ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ ТА МЕРЕЖ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ТА ВИЯВЛЕННЯ АНОМАЛІЙ У МЕРЕЖЕВОМУ ТРАФІКУ»

на здобуття освітнього ступеня магістра
зі спеціальності 172 Електронні комунікації та радіотехніка
(код, найменування спеціальності)
освітньо-професійної програми Телекомунікаційні системи та мережі
(назва)

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

_____ Андрій ЛИСАКІВСЬКИЙ
(підпис) *(Ім'я ПРІЗВИЩЕ)*

Виконав: здобувач вищої освіти гр. ТСДМ-61
Андрій ЛИСАКІВСЬКИЙ
Ім'я, ПРІЗВИЩЕ

Керівник: Катерина ТРЕНЬОВА
PhD (доктор філософії) *Ім'я, ПРІЗВИЩЕ*

Рецензент: _____
Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Телекомунікацій

Кафедра ___Телекомунікаційних систем та мереж_____
Ступінь вищої освіти __магістр_____
Спеціальність ___172 Електронні комунікації та радіотехніка _____
Освітньо-професійна програма _____Телекомунікаційні системи та мережі_____

ЗАТВЕРДЖУЮ
Завідувач кафедри __ТСМ_____
_____ Віктор ЗАЇКА
«____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Лисаківському Андрію Олеговичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Розробка системи моніторингу та виявлення аномалій у мережевому трафіку
керівник кваліфікаційної роботи Катерина ТРЕНЬОВА, PhD (доктор філософії)
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» 10. 2025 р. № 467

2. Строк подання кваліфікаційної роботи «___» _____ 2025 р.

3. Вихідні дані до кваліфікаційної роботи:
1. Аномалії.
2. Системи моніторингу.
3. Мережевий трафік.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Мережеві вузли та аномалії.
2. Аномалії в часових рядах.
3. Аномалії в мікросервісній архітектурі.
4. Аномалії в моніторингу безпеки мережі.

5. Перелік ілюстративного матеріалу: ___презентація на____слайдах_____

6. Дата видачі завдання «___» _____ 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	20.10.2025	Викон.
2	Розробка першого розділу	27.10.2025	Викон.
3	Розробка другого розділу	07.11.2025	Викон.
4	Розробка третього розділу	17.11.2025	Викон.
5	Розробка четвертого розділу	27.11.2025	Викон.
6	Висновки, вступ, реферат	07.12.2025	Викон.
7	Розробка презентації	17.12.2025	Викон.

Здобувач вищої освіти _____ Андрій ЛИСАКІВСЬКИЙ
(підпис) *(Ім'я ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Катерина ТРЕНЬОВА
(підпис) *(Ім'я ПРІЗВИЩЕ)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут телекомунікацій
ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач Лисаківський А.О. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 172 Електронні комунікації та радіотехніка
(код, найменування спеціальності)
освітньо-професійної програми Телекомунікаційні системи та мережі
(назва)
на тему: «Розробка системи моніторингу та виявлення аномалій у мережевому трафіку»

Кваліфікаційна робота і рецензія додаються.

Директор ННІТ _____ Владислав КРАВЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач Лисаківський Андрій Олегович показав гарну теоретичну та інженерну підготовку, уміння володіти новими комп'ютерними технологіями, користуватися навчальною, довідковою і науково-технічною літературою. Працюючи над завданнями, які були поставлені керівником, проявив ініціативність, сумлінність та хист до наукової роботи.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Лисаківського Андрія Олеговича на оцінку «відмінно» та присвоїти йому кваліфікацію магістр з електронних комунікацій та радіотехніки.

Керівник кваліфікаційної роботи _____ Катерина ТРЕНЬОВА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

«____» _____ 20___ року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Лисаківський А.О. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри телекомунікаційних систем та мереж _____ Віктор ЗАЇКА
(назва) *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

«____» _____ 20___ року

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу
на здобуття освітнього ступеня магістра

здобувача вищої освіти Лисаківського Андрія Олеговича
(прізвище, ім'я, по батькові)
на тему «Розробка системи моніторингу та виявлення аномалій у мережевому трафіку»

Актуальність.
Стрімке ускладнення телекомунікаційних мереж, зростання обсягу трафіку та використання мікросервісних, хмарних і промислових інфраструктур потребують більш інтелектуальних методів моніторингу. Традиційні IDS/IPS та статичні пороги не здатні вчасно розпізнавати складні або приховані аномалії. Flow-телеметрія (NetFlow, IPFIX, sFlow) у поєднанні з методами машинного навчання та кластеризацією характеристичних векторів дозволяє пасивно формувати поведінкові профілі вузлів і виявляти відхилення. Розробка таких рішень є актуальною для підвищення кіберстійкості, зниження MTTD та захисту критичної інфраструктури.
Позитивні сторони.
1. У роботі запропоновано математичну модель характеристичних векторів мережевих вузлів, придатну для класифікації та виявлення аномалій.
2. У роботі реалізовано застосування методів машинного навчання та AutoML (TPOT, stacking), що підвищує точність та адаптивність системи.
Недоліки.
1. У роботі немає інтеграції з існуючими системами моніторингу (Grafana, Zabbix, ELK, Suricata), що знижує прикладну цінність.
2. В роботі не розглянуто аспекти оптимізації ресурсів при реальному впровадженні (CPU/RAM/затримки).

Відзначені зауваження не впливають на загальну позитивну оцінку магістерської кваліфікаційної роботи

Висновок: *кваліфікаційна робота на здобуття ступеня магістра заслуговує оцінку "відмінно", а здобувач Лисаківський Андрій Олегович заслуговує присвоєння кваліфікації: магістр з електронних комунікацій та радіотехніки.*

Рецензент
науковий ступінь, вчене звання _____ (підпис) _____ *(Ім'я ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 82 стор., 32 рис., 4 табл., 22 джерела.

Мета роботи – розробити інтелектуальний підхід до моніторингу та виявлення аномалій у мережевому трафіку на основі flow-телеметрії та машинного навчання, що забезпечує своєчасне виявлення відхилень і підвищення кіберстійкості телекомунікаційних систем..

Об’єкт дослідження – мережевий трафік телекомунікаційних систем корпоративного, хмарного та промислового типу у форматах flow-телеметрії (NetFlow, IPFIX, sFlow).

Предмет дослідження – методи формування характеристичних векторів мережевих вузлів та застосування алгоритмів машинного навчання (класифікації, кластеризації, реконструкції) для виявлення аномалій у поведінці мережі.

Короткий зміст роботи: у роботі розроблено підхід до побудови системи моніторингу та виявлення аномалій у мережевому трафіку на основі flow-телеметрії та методів машинного навчання. Проаналізовано переваги представлення трафіку у вигляді потоків NetFlow/IPFIX замість сирих пакетів, що забезпечує до 80–90% охоплення мережевої активності. Сформовано модель характеристичних векторів вузлів, придатну для класифікації та кластеризації аномалій. Доведено вплив змін профілю одного вузла на поведінку інших елементів мережі. Особливу увагу приділено виявленню аномалій у часових рядах, зокрема reconstruction-based моделям і TadGAN, що здатні фіксувати як точкові, так і групові відхилення.

КЛЮЧОВІ СЛОВА: FLOW-ТЕЛЕМЕТРІЯ, NETFLOW, IPFIX, SFLOW, МАШИННЕ НАВЧАННЯ, КЛАСТЕРИЗАЦІЯ, КЛАСИФІКАЦІЯ, ВИЯВЛЕННЯ АНОМАЛІЙ, ЧАСОВІ РЯДИ, RECONSTRUCTION-BASED МЕТОДИ, TADGAN, ХАРАКТЕРИСТИЧНІ ВЕКТОРИ ВУЗЛІВ, МЕРЕЖЕВИЙ ТРАФІК, КІБЕРБЕЗПЕКА, МОНІТОРИНГ МЕРЕЖІ

ABSTRACT

Text part of the master's qualification work: 82 pages, 32 figures, 4 table, 22 sources.

The purpose of the work is to develop an intelligent approach to monitoring and detecting anomalies in network traffic based on flow telemetry and machine learning, which ensures timely detection of deviations and increases the cyber resilience of telecommunication systems.

Object of the study is network traffic of telecommunication systems of corporate, cloud and industrial type in flow telemetry formats (NetFlow, IPFIX, sFlow).

Subject of the study is methods for forming characteristic vectors of network nodes and applying machine learning algorithms (classification, clustering, reconstruction) to detect anomalies in network behavior.

Summary of the work: the work develops an approach to building a system for monitoring and detecting anomalies in network traffic based on flow telemetry and machine learning methods. The advantages of representing traffic in the form of NetFlow/IPFIX flows instead of raw packets are analyzed, which provides up to 80–90% coverage of network activity. A model of node characteristic vectors suitable for classification and clustering of anomalies has been formed. The influence of changes in the profile of one node on the behavior of other network elements has been proven. Special attention is paid to the detection of anomalies in time series, in particular reconstruction-based models and TadGAN, which are capable of recording both point and group deviations.

KEYWORDS: FLOW-TELEMETERY, NETFLOW, IPFIX, SFLOW, MACHINE LEARNING, CLUSTERIZATION, CLASSIFICATION, ANOMALIES DETECTION, TIME SERIES, RECONSTRUCTION-BASED METHODS, TADGAN, NODE CHARACTERISTIC VECTORS, NETWORK TRAFFIC, CYBERSECURITY, NETWORK MONITORING

ЗМІСТ

ВСТУП 9
РОЗДІЛ 1 МЕРЕЖЕВІ ВУЗЛИ ТА АНОМАЛІЇ 11
1.1 Характеристики мережевих вузлів 11
1.2 Виявлення аномалій в мержевому трафіку 18
1.3 Методи реалізації 22
РОЗДІЛ 2 АНОМАЛІЇ В ЧАСОВИХ РЯДАХ 29
2.1 Використання методів аналітики 29
2.2 Моніторинг аномалій в технологічних системах 32
2.3 Нейронні мережі для виявлення аномалій 42
РОЗДІЛ 3 АНОМАЛІЇ В МІКРОСЕРВІСНІЙ АРХІТЕКТУРІ 49
3.1 Підхід in-house 50
3.2 Prometheus Anomaly Detector (PAD) 53
3.3 Виявлення аномалій як частина системи АРМ 55
3.4 Множинний доступ із бітовим розділенням 58
РОЗДІЛ 4 АНОМАЛІЇ В МОНІТОРИНГУ БЕЗПЕКИ МЕРЕЖІ 61
4.1 Основні джерела даних для моніторингу 61
4.2 Flow-протоколи та їхні відмінності 68
ВИСНОВКИ 81
ПЕРЕЛІК ПОСИЛАНЬ 83
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ 85

ВСТУП

Сучасні телекомунікаційні мережі стрімко ускладнюються: у корпоративних, промислових, державних та хмарних інфраструктурах постійно зростає кількість пристроїв, сервісів і мікросервісних компонентів. Разом з цим зростає і обсяг мережевого трафіку, який необхідно контролювати, аналізувати та оперативно оцінювати для забезпечення надійної роботи систем та їх захищеності. У таких умовах традиційні засоби моніторингу — зокрема інструменти, що працюють лише з сирим трафіком або ручними пороговими правилами, — виявляються недостатніми. Вони не здатні своєчасно реагувати на складні, нетипові або приховані відхилення у поведінці мережевих вузлів, особливо в умовах високої динамічності та сезонності навантажень.

Переходи на мікросервісні архітектури, хмарні інфраструктури, контейнеризацію та автоматизацію DevOps значно ускладнили завдання своєчасного розпізнавання нестандартних ситуацій. Навіть незначне збільшення латентності, аномальні HTTP-потоки, зміщення частоти запитів або нетипові SSH-з’єднання можуть сигналізувати про майбутні відмови, компрометації або внутрішні порушення. У таких умовах ручне налаштування порогів перестає бути ефективним, а стандартні підходи до аналізу логів чи пакетів вже не дають повної картини.

Ефективним підходом стає робота з мережевою телеметрією, зокрема з потоками типу NetFlow, IPFIX або sFlow. На відміну від пакетного аналізу, потокова аналітика дозволяє отримувати узагальнену, але надзвичайно інформативну модель поведінки мережі, що охоплює до 90% усіх можливих проявів аномалій. Використання flow-телеметрії дає змогу бачити взаємодії між вузлами, вимірювати інтенсивність використання протоколів, визначати структуру навантаження, оцінювати кореляції та зміни у часових рядах трафіку. Саме цей тип даних сьогодні стає основою інтелектуальних систем мережевого моніторингу.

Не менш важливим елементом сучасних підходів є методи машинного навчання, зокрема класифікація, кластеризація, автоматизоване моделювання та

реконструкція сигналів. Рішення на основі ієрархічної кластеризації, стекінга моделей, TPOT-оптимізації або глибинних нейронних мереж дозволяють будувати точні профілі роботи вузлів, виділяти характерні ознаки трафіку та виявляти відхилення з високою точністю. Особливо ефективними є reconstruction-based методи, такі як TadGAN, що дають змогу навчати модель «нормальній» поведінці та фіксувати будь-які відхилення, які вона не здатна відтворити.

У рамках дослідження розглянуто весь цикл обробки трафіку: від отримання потоків за допомогою DPI-модулів і побудови характеристичних векторів вузлів — до класифікації, побудови кластерів, оцінювання критичності аномалій та автоматизованого тестування моделей. Продемонстровано, як зміна структури мережі, додавання нових пристроїв або нетипових протоколів впливає на профілі вузлів, та як алгоритми машинного навчання здатні це адаптивно враховувати. Показано практичні приклади формування векторів ознак, роботи моделі на реальних та синтетичних наборах даних, а також досліджено різні види аномалій: точкові, групові та контекстні.

Актуальність обраної теми обумовлена необхідністю створення систем, здатних автоматично та з високою точністю виявляти аномальні стани, знижувати ризики внутрішніх загроз, скорочувати MTTD та забезпечувати стабільність мережевих сервісів. Розробка інтелектуальної системи моніторингу та виявлення аномалій у мережевому трафіку є важливим кроком у напрямку побудови сучасних, стійких та кібербезпечних інфраструктур.

1 МЕРЕЖЕВІ ВУЗЛИ ТА АНОМАЛІЇ

1.1 Характеристики мережевих вузлів

У сучасних промислових підприємствах, енергетичних комплексах, транспортних системах та великих інфраструктурних об’єктах усе ширше застосовуються мережі передавання даних, які забезпечують координацію технологічних процесів і обмін важливою службовою інформацією. Фактично будь-яке сучасне виробництво — від хімічних заводів до систем залізничного руху — залежить від стабільної роботи мережевого середовища. У таких системах передані дані часто мають критичний характер: від них напряму залежить безпека людей, цілісність обладнання, безперервність робочих процесів і навіть підтримання сталого функціонування технологічних ланцюгів.

Саме тому питання захисту мереж та розуміння їхньої структури набувають першорядного значення. Для того щоб застосовувати ефективні механізми мережевої безпеки — від систем виявлення вторгнень і фільтрації трафіку до механізмів контролю доступу та сегментації — потрібно володіти повною інформацією про всі мережеві вузли: їх розташування, функції, типи, способи взаємодії та характер поведінки у мережі. Проте отримати ці дані традиційними методами в промислових системах часто складно або навіть небезпечно.

Стандартні засоби виявлення пристроїв, такі як Nmap та інші інструменти активного сканування, були створені для гнучких IT-мереж, але не для критичних виробничих середовищ. Активне зондування, яке у звичайній мережі не становить проблеми, у промисловій може викликати небажані наслідки. Багато ПЛК, контролерів чи спеціалізованих пристроїв не розраховані на додаткове навантаження, і навіть кілька зайвих пакетів можуть призвести до їх зависання або аварійного перезапуску. Таке втручання здатне зупинити виробничу лінію або порушити логіку роботи системи керування.

Інша проблема полягає в тому, що традиційні сканери працюють на основі визначених наперед сигнатур та правил. Якщо пристрій має нетипові

характеристики або не входить у відомий клас, він залишається невпізнаним або позначається як «невідомий». Для промислових мереж, де використовуються унікальні або застарілі пристрої, це цілком звична ситуація. У результаті отримана карта мережі виявляється неповною або неточною.

Запропонований підхід вирішує ці недоліки: класифікація типів вузлів здійснюється пасивно, виключно на основі аналізу реального мережевого трафіку, без жодного втручання у роботу обладнання. Метод опирається на машинне навчання, яке здатне самостійно адаптуватися до особливостей конкретної мережі та виявляти закономірності, недоступні класичним методам аналізу.

Крім того, цей підхід має додаткову значущу перевагу — можливість виявлення аномалій. Оскільки кожен мережевий вузол має характерний набір протоколів, частоту взаємодій та власний поведінковий «підпис», зміна цього профілю може сигналізувати про потенційні загрози: стороннє втручання, несанкціоновані команди, зараження шкідливим ПЗ або навіть помилки у конфігурації мережі. Таким чином, описуваний метод може виконувати і функцію автоматизованої інвентаризації, і функцію інтелектуальної системи раннього виявлення інцидентів.

У ситуаціях, коли активні інструменти неприйнятні або заблоковані політикою безпеки, ручне описання та класифікація всіх пристроїв займає значний час і ресурси. Використання автоматизованого методу на основі профайлінгу мережевого трафіку дозволяє мінімізувати людський фактор, підвищити точність аналізу та зберегти стабільність і безпеку системи.

Ідея методу базується на концепції пасивного аналізу, подібного до того, що застосовується в системах виявлення вторгнень (IDS). Суть полягає у тому, щоб спостерігати за трафіком у мережі, не змінюючи його та не створюючи додаткового навантаження. Кожен пакет фіксується, після чого увесь трафік групується у потоки — унікальні з’єднання між вузлами, які можна охарактеризувати набором задіяних протоколів, IP-адресами, портами та іншими атрибутами.

Для кожного мережевого вузла формується узагальнений профіль — вектор ознак, який відображає частоту використання певних протоколів, типову

інтенсивність взаємодій, особливості комунікацій з іншими пристроями тощо. Такий вектор фактично є цифровим описом поведінки пристрою у мережевому середовищі, подібним до відбитку, який складно підробити або випадково змінити.

Після цього використовується набір заздалегідь розмічених вузлів, які слугують тренувальною вибіркою. Застосовуючи засоби AutoML, можна автоматично підібрати найкращу архітектуру моделі машинного навчання, яка найбільш ефективно виконує класифікацію. Модель навчається розпізнавати патерни поведінки, притаманні різним класам пристроїв: серверам, робочим станціям, контролерам, маршрутизаторам, сенсорам тощо.

Після тренування модель може приймати на вхід вектори ознак невідомих вузлів та видавати прогноз щодо їхнього найбільш імовірного типу. Цей процес повністю автоматизований і може виконуватися регулярно, у режимі реального часу або періодичного аналізу.

У завданні виявлення аномалій підхід спрощується ще більше. Оскільки мета не визначити конкретний тип пристрою, а виявити відхилення від норми, застосовується алгоритм навчання без учителя. Спочатку за векторами ознак виконується кластеризація — автоматичне групування вузлів за схожими поведінковими характеристиками. Оптимальна кількість кластерів визначається за метричними критеріями, після чого кожен вузол отримує належність до певної групи.

Далі ці дані використовуються для навчання нейронної мережі, яка дозволяє швидко перевіряти, чи належить оновлений профіль вузла до того самого кластеру. Якщо класи співпадають — його поведінка лишається типовою. Якщо ні — це ознака потенційної підозрілої активності.

Щоб оцінити критичність виявленого відхилення, можна застосувати ієрархічну кластеризацію, яка показує «відстань» між вузлами на дереві подібностей. Чим далі розташовані два стани пристрою, тим суттєвіше відхилення. Таким чином легко розрізнити, наприклад, появу нового клієнта у файлового сервера — явище цілком нормальне — та появу незвичного керуючого трафіку до комутатора, який ніколи не приймав подібних команд.

Математична формальна модель

Математику часто називають універсальною мовою науки, адже саме вона дозволяє строго й однозначно описувати складні процеси. Спробуємо подати ідею визначення мережевого класу вузла за його трафіком у математичній формі та одночасно уточнити окремі технічні деталі.

На вхід моделі надходить мережевий трафік захищуваної системи. Під потоком будемо розуміти множину мережевих пакетів, які:

- використовують однаковий набір протоколів на всіх задіяних мережевих рівнях,
- мають однакову адресу джерела,
- та однакову адресу призначення.

Кожен потік FlowFlowFlow описується вектором характеристик:

$$Flow = \{p_1, p_2, \dots, p_{numProto}\} \tag{1.1}$$

де довжина вектора numProtonumProtonumProto визначається кількістю протоколів, які присутні у стеку конкретного пакета або групи пакетів.

Усі потоки системи, позначимо їх множиною FlowsFlowsFlows, де кількість потоків lenFlowslenFlowslenFlows, можуть бути представлені як набір таких векторів:

$$Flows = \{Flow_1, Flow_2, \dots, Flow_{lenFlows}\} \tag{1.2}$$

Ця формальна модель слугує основою для подальшого аналізу трафіку, формування ознак та класифікації мережевих вузлів.

Таким чином, весь набір потоків Flows довжиною lenFlows описується набором характеристик

$$Flows = \{Flow_i\}_{i=0}^{lenFlows} \tag{1.3}$$

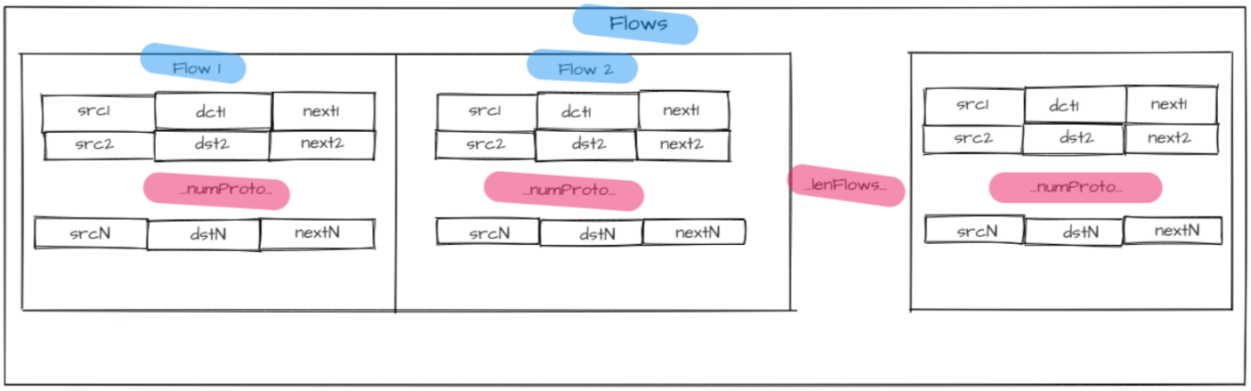


Рис. 1.1 Група потоків

Як ідентифікатор для кожного мережевого вузла *host* з множини класифікованих вузлів *Hosts* довжиною *lenHosts* використовується набір адрес джерела на всіх рівнях мережевої моделі, кількість яких дорівнює *numProto*. Значення *numProto* відповідає числу вкладених протоколів у структурі мережі:

$$Hosts = \{host\}_{i=0}^{\text{numHosts}} = \left\{\{host.src\}_{j=0}^{\text{numProto}}\right\}_{i=0}^{\text{numHosts}} \quad (1.4)$$

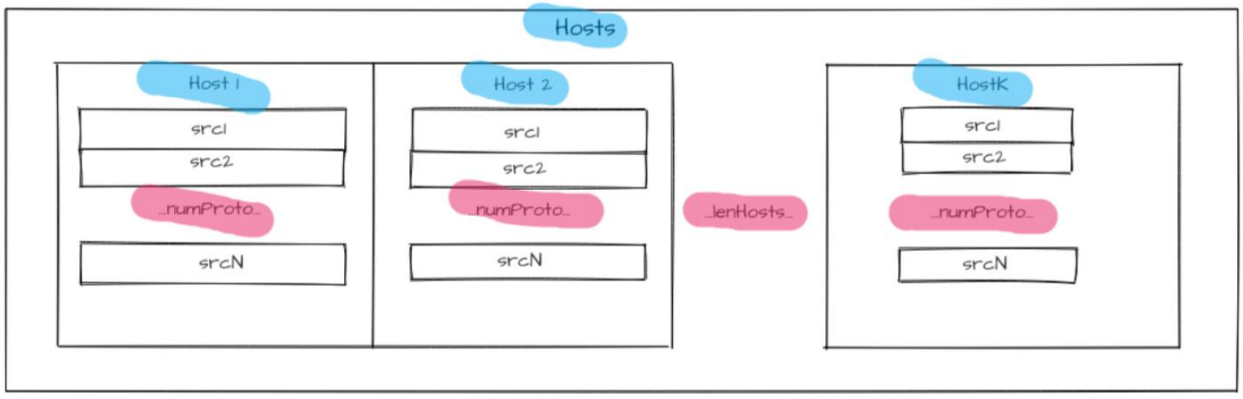


Рис. 1.2 Ідентифікатори вузлів

Основними характеристиками вузла є кількість потоків *host.num_flows*, що належать до нього, а також протоколи, які використовуються в цих потоках. Для кожного вузла формується відповідний вектор параметрів:

$$host.proto = \{frequency\}_{j=0}^{\text{host.numKnownProto}} \quad (1.5)$$

Довжина цього вектора *host.numKnownProto* відповідає загальній кількості відомих протоколів усіх рівнів у всьому масиві потоків. Кожен його елемент *frequency* має значення від 0 до 1 і відображає частку потоків, що належать даному вузлу та містять відповідний протокол.

Вузол має набір ярликів:

$$host.labels = \{label\}_{k=0}^{\text{numTypes}} \quad (1.6)$$

Кожен елемент цього набору визначає належність вузла до певного класу. Групи класів формуються окремо за двома ознаками: за роллю вузла (наприклад, ПЛК, АРМ, АСО тощо) та за операційною системою (Windows XP, Windows 7, Linux Ubuntu, Linux CentOS тощо).

Вузол також описується часткою потоків, що з'єднують його з іншими вузлами певних класів — цей показник позначається як *host.connectedWith*. Він дає змогу оцінити, наскільки інтенсивно та з якими саме типами вузлів взаємодіє даний елемент мережі. Така інформація є важливою для розуміння структури мережевих зв'язків і поведінкових особливостей кожного вузла.

Для того щоб представити вузол у форматі, придатному для подальшого аналізу методами машинного навчання, усі ключові параметри поєднуються у єдиний характеристичний вектор *host.vector*. До цього вектора можуть додаватися додаткові відомості про вузол — *host.some_inf*, наприклад дані про його роль, особливості взаємодій, тип операційної системи чи інші специфічні властивості.

Об'єднуючи цю інформацію з частотою використання різних протоколів, отримуємо профіль трафіку вузла, який формально описується як:

host.vector = {host.proto, host.some_inf}

Такий характеристичний вектор відображає як структурні, так і поведінкові особливості вузла та може бути використаний у задачах класифікації, кластеризації чи виявлення аномалій у мережевому трафіку.

Вихідні дані:

$$data = \{host.vector\}_{i=0}^{\text{lenHosts}} \quad (1.6)$$

Для моделі дані подаються як набір характеристичних векторів окремих вузлів.

Для навчання моделі потрібні заздалегідь відомі відомості про класи вузлів, тому кожен із них має бути позначений відповідним класом.

$$host.labels = \{label\}_{k=0}^{\text{numTypes}} \quad (1.7)$$

Тобто дамп трафіку повинен складатися з набору потоків для всіх вхідних даних (адрес джерела для кожного вузла).

У яких наперед відома відповідь – тип вузла label.

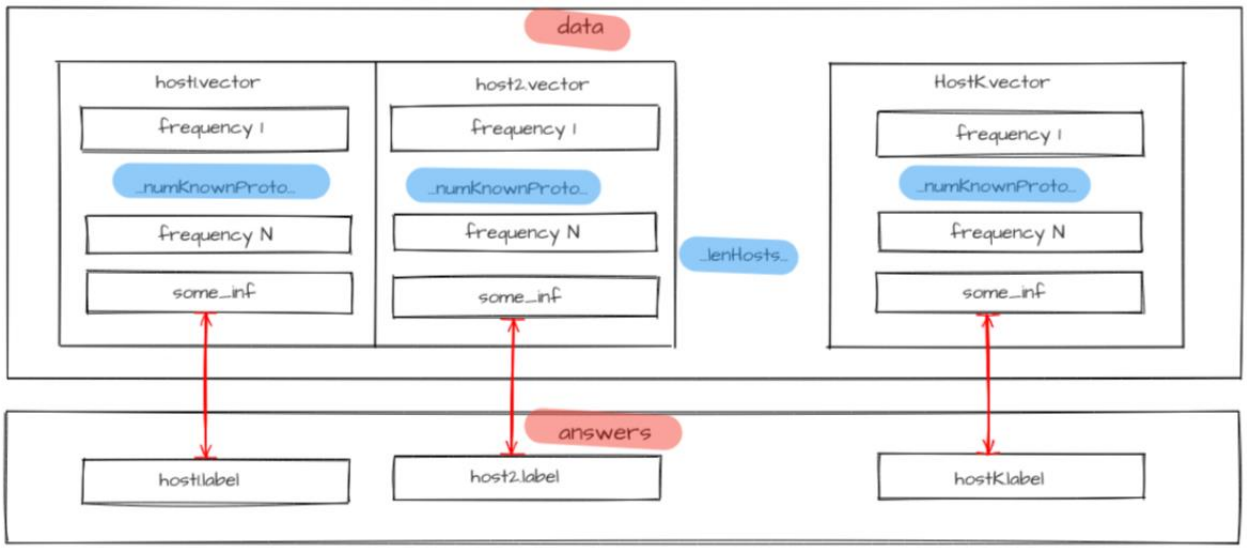


Рис. 1.3 Дані для навчання моделі

Наявний мережевий трафік, у якому вузли вже мають визначені класи, поділяється на дві частини: тренувальну та тестову вибірки. На першому етапі модель навчається на тренувальних даних, де для кожного вузла відомі його реальні мітки класів — *host.label*. Це дозволяє моделі сформувати залежності між характеристичними векторами вузлів та їхніми класами.

Після завершення навчання модель переходить до етапу розпізнавання вузлів із тестової вибірки. Тут для кожного вузла модель генерує передбачену мітку класу — *host.labelPredicted*. Далі відбувається порівняння цих прогнозованих значень із заздалегідь відомими «правильними» мітками *host.label*. На основі цього порівняння, із використанням відповідних метрик оцінювання (таких як точність, повнота, F1-міра тощо), визначається якість роботи моделі.

Якщо отримані результати відповідають встановленим вимогам, модель вважається достатньо якісною і може бути збережена для подальшого застосування. Якщо ж якість виявляється недостатньою, структуру моделі або її гіперпараметри коригують, після чого цикл навчання → тестування → налаштування повторюється доти, доки не буде досягнуто бажаного рівня точності передбачення.

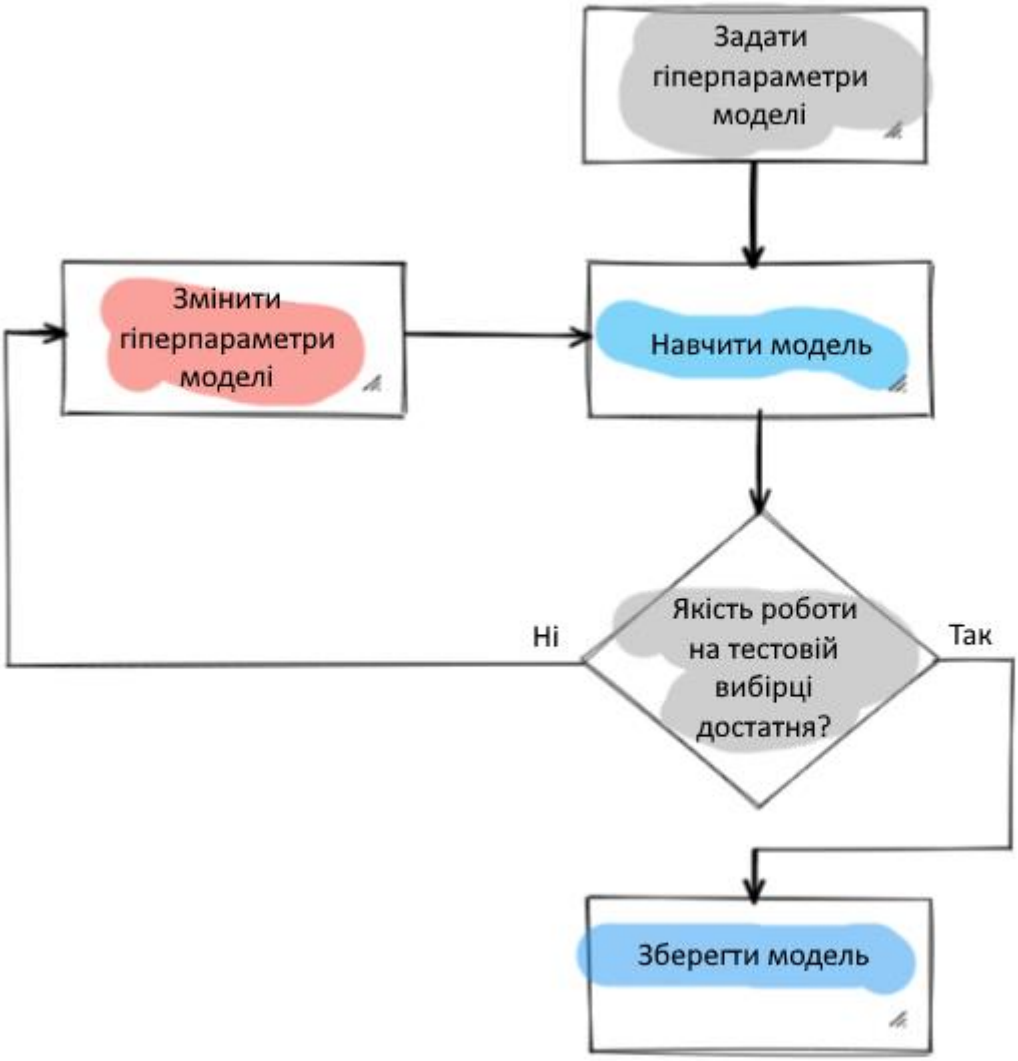


Рис. 1.4 Процес створення моделі

1.2 Виявлення аномалій в мержевому трафіку

Тепер розглянемо другу важливу задачу — виявлення аномалій у мережевому трафіку. Для цього зі сформованого масиву потоків *Flows* виділяються характеристичні вектори *host.vector*, що відображають ключові параметри кожного вузла. Після цього отримані вектори подаються на вхід моделі кластеризації, яка автоматично групує вузли за схожістю їх поведінкових та структурних ознак. У результаті кожному вузлу присвоюється певний клас *host.label*, що відповідає номеру кластера, до якого він був віднесений.

Таким чином, усе багатовимірне простір, розмірність якого дорівнює довжині характеристичного вектора *host.vector*, поділяється на низку областей, число яких збігається з кількістю отриманих кластерів. Кожна така область відповідає певному типу поведінки вузлів у мережі: від нормальної активності до потенційно аномальної.

Для кластеризації застосовується модель ієрархічного кластеризатора, яка вибудовує деревоподібну структуру взаємозв'язків між кластерами та дозволяє оцінити близькість між групами на різних рівнях деталізації. Результат цього ієрархічного розбиття зберігається та використовується на наступних етапах аналізу, наприклад для пошуку відхилень, повторного класифікування нових вузлів або побудови додаткових моделей виявлення аномалій.

Такий підхід дає змогу ефективно виявляти нетипову поведінку вузлів у мережі, навіть коли попередня розмітка відсутня або неповна, а також забезпечує основу для глибшого аналізу структури та динаміки мережевого трафіку.

Коли з'являється нова група потоків *Flows*, характеристичні вектори вузлів *host.vector* оновлюються та передаються на вхід моделі. Результати роботи моделі зберігаються у властивість *host.new_label* відповідного вузла і порівнюються з відомими для нього класами *host.label*. Якщо виявляється невідповідність

$$(host_i.\ label \neq host_i.\ new_label) \tag{1.8}$$

то така зміна вважається аномалією. Потоки *Flows*, що спричинили це відхилення (тобто ті, які змінили характеристичний вектор вузла *host.vector*), визначаються як ініціатори аномалії. Вони передаються оператору для подальшого аналізу як потенційно небезпечні. Одночасно зазначається і сам вузол *host*, чию поведінку система розпізнала як нетипову.

Для оцінювання рівня критичності аномалії використовується попередньо сформована ієрархічна кластерна модель. Ступінь критичності визначається на основі того, наскільки «далеко» вузол перемістився у кластерній структурі. Зокрема, критичність переходу вузла з одного кластера до іншого обчислюється як кофенетичне відстань — довжина шляху між двома вузлами у дереві ієрархії: між

поточним аномальним вузлом та його новим найближчим кластерним сусідом. Чим більшою є ця відстань, тим серйознішим вважається відхилення у поведінці вузла.

Такий метод дозволяє не лише виявляти факт аномалії, а й оцінювати її значущість, що підвищує точність та інформативність аналізу мережевого трафіку.

Якщо ж розбіжностей між відомими класами вузлів та новими, отриманими в результаті роботи моделі, не виявлено, тобто поведінка всіх вузлів узгоджується з раніше сформованою кластерною структурою,

$$(host_i.\ label = host_i.\ new_label) \tag{1.9}$$

то модель проходить повторне навчання, використовуючи оновлені характеристичні вектори вузлів *host.vector* як вхідні дані. На цьому етапі фактично виконується перерозмітка багатовимірного простору, оскільки характеристики вузлів могли змінитися під впливом нових потоків. Модель таким чином адаптується до оновленого стану мережі та підтримує актуальність кластерної структури.

Після повторного навчання система переходить у режим очікування наступної партії потоків *Flows*, готова знову аналізувати зміни в поведінці мережевих вузлів. Одночасно знову будується оновлене ієрархічне розбиття, яке відображає свіжу структуру взаємозв’язків між кластерами.

У такій ситуації вважається, що аномальної поведінки цього разу не зафіксовано, оскільки всі вузли залишилися у межах своїх типових груп, а їх нові характеристики не спричинили переходу в інші області кластерного простору. Такий підхід дозволяє моделі постійно адаптуватися до нормальної еволюції трафіку, не сигналізуючи про «хибні» аномалії.

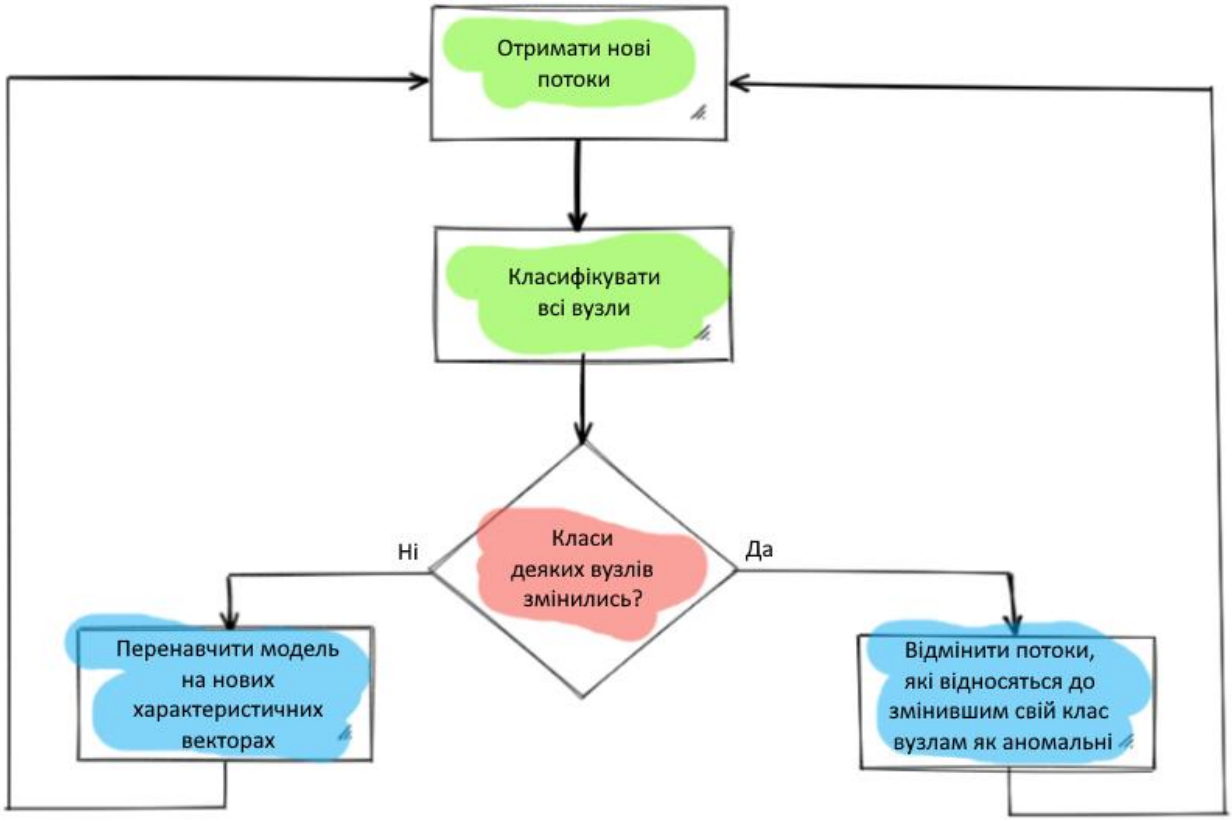


Рис. 1.5 Виявлення аномальних потоків

Описана формальна модель дає змогу ефективно вирішувати дві ключові задачі: визначення класифікаційних ознак мережевих вузлів та виявлення аномалій у їхній поведінці. Завдяки структурованому представленню характеристик вузлів у вигляді векторів ознак, модель дозволяє точно віднести кожен вузол до відповідного класу, враховуючи як його мережеві параметри, так і особливості взаємодій з іншими елементами системи.

Крім того, можливість автоматичного аналізу змін у цих характеристичних векторах забезпечує виявлення нетипових або підозрілих відхилень у поведінці вузлів. Використання кластеризації та ієрархічних структур дозволяє отримати більш глибоке уявлення про внутрішню організацію мережі, виділити природні групи елементів та оцінити ступінь критичності аномалій.

Таким чином, модель служить універсальним інструментом для аналізу мережевого трафіку, формування профілів вузлів, підтримки задач безпеки та забезпечення своєчасного реагування на потенційні загрози в інфраструктурі.

1.3 Методи реалізації

Для отримання потоків із сирого мережевого трафіку було використано модуль DPI. За його допомогою здійснюється глибинний аналіз пакетів, на основі якого формується масив потоків *Flows*. Надалі ці потоки обробляються спеціально розробленим Python-скриптом, який генерує характеристичні вектори вузлів — структуроване числове представлення їхніх властивостей, необхідне для подальшого машинного аналізу.

Для вибору оптимальної архітектури моделей застосовано один із сучасних підходів автоматизованого машинного навчання — TPOT, реалізований у відповідному модулі Python. Цей інструмент проводить еволюційний пошук найкращих комбінацій алгоритмів і їхніх гіперпараметрів, що дає змогу автоматично знаходити високоефективні моделі без ручного підбору.

У результаті роботи TPOT було сформовано ансамбль, що складається з двох блоків, об’єднаних методом *стекінгу* (stacking). Перший блок також є ансамблем моделей, створених за принципом стекінгу, але базується виключно на алгоритмах к-найближчих сусідів (k-NN). У такій архітектурі результати (передбачення) моделей нижчого рівня використовуються як додаткові вхідні ознаки для моделей вищого рівня. Це дозволяє суттєво підвищити якість класифікації, оскільки старші моделі отримують доступ до проміжних рішень і можуть коригувати або узагальнювати їх.

Такий підхід забезпечує гнучкість, підвищену точність та здатність ансамблю адаптуватися до складної багатовимірної структури мережевого трафіку.

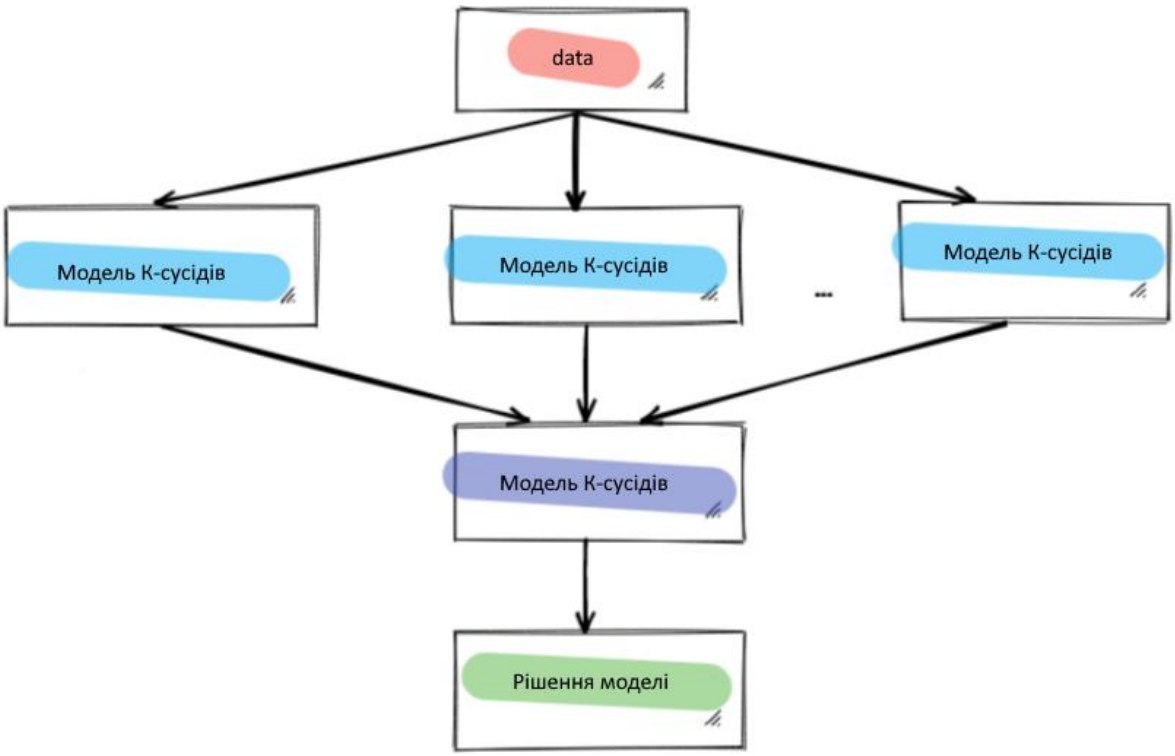


Рис. 1.6 Перший блок ансамблю підсумкової моделі

Другий блок – модель наївного байєсівського класифікатора з мінімальним згладжуванням.



Рис. 1.7 Підсумкова модель

Наразі наявного обсягу даних недостатньо для того, щоб повною мірою оцінити точність і стабільність роботи розроблених модулів. Очікується, що після переходу продукту до етапу дослідної експлуатації з’явиться більший масив

реальних даних, що дозволить отримати статистично значущі показники та вже впевнено оперувати конкретними цифрами.

Попри це, попереднє тестування на доступних даних (звичайно ж, на відмінних від тих, що використовувалися в навчальній вибірці) показало досить обнадійливі результати: значення F1-міри склало близько 80% (див. Таблицю 1.1). Це свідчить про те, що модель уже на ранньому етапі демонструє прийнятну якість класифікації.

Крім того, було запропоноване цікаве практичне рішення для випадків, коли модель не має достатньої впевненості у виборі класу. У таких ситуаціях система може пропонувати користувачеві два найбільш ймовірні класи замість одного. Застосування такого підходу значно підвищує якість передбачень: F1-міра у цьому випадку зростає до 95–98%, що є суттєво кращим показником. Важливо й те, що частка таких «невизначених» ситуацій у загальному обсязі вибірки є порівняно малою, тож використання цього механізму виглядає практичним і доцільним.

Таблиця 1.1

Точність роботи класифікатора

Тип класу	**Точність**	**Повнота**	**F1-міра**
Ролі вузлів (АРМ, ПЛК, Сервер, …)	0.81	0.80	0.81
Операційна система (WinXP, Win8, LinuxCentOS, …)	0.86	0.80	0.78

Для демонстрації можливостей підходу розглянемо спрощений, майже ідеалізований приклад, який, хоча й є певною мірою виродженим, зате легко сприймається та дозволяє наочно пояснити принципи роботи моделі.

Уявімо, що в мережі підприємства одночасно функціонують лише чотири вузли: ПЛК, SCADA-сервер та два АРМ. Взаємодія між ними організована таким чином:

- ПЛК обмінюється даними зі SCADA за протоколом Modbus, а на нижчих рівнях стеку використовуються стандартні протоколи — Ethernet, IPv4 та TCP.
- SCADA комунікує з АРМ №1 через HTTP, при цьому набір нижньорівневих протоколів такий самий: Ethernet → IPv4 → TCP.

- Крім того, SCADA додатково спілкується з АРМ №2 за допомогою SSH, що створює ще один набір потоків.

Таким чином, ми маємо мінімальну, але показову мережеву структуру з чітко визначеними напрямами обміну даними та різними застосованими протоколами. Схематична карта потоків, яка ілюструє ці взаємозв'язки, наведена на рисунку нижче.

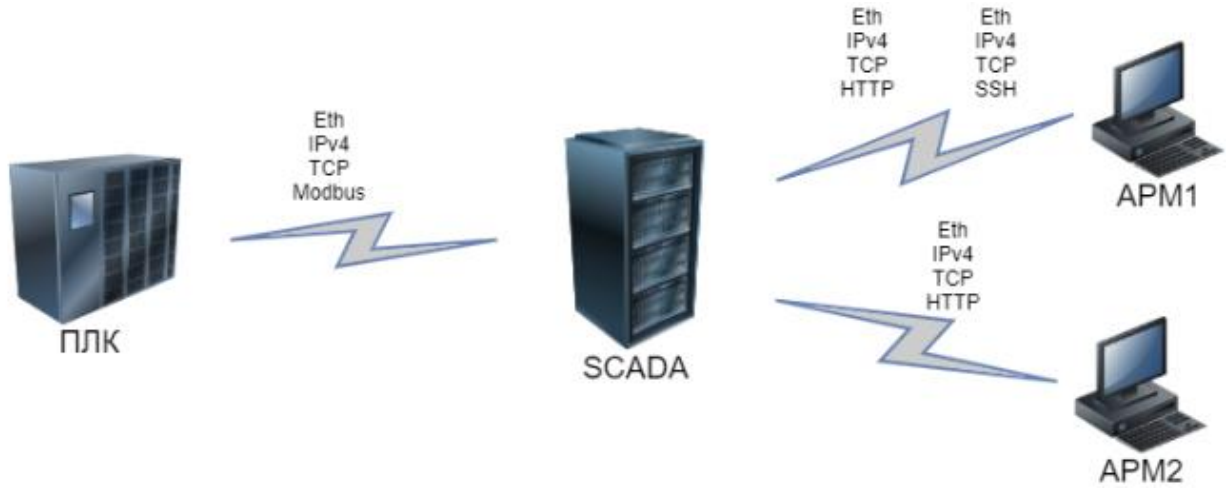


Рис. 1.8 Мапа потоків

Складемо таблицю частот використання протоколів кожному рівні OSI кожним з вузлів.

Таблиця 1.2

Частоти використання протоколів вузлами

Признак \ Вузол	**ПЛК**	**SCADA**	**АРМ №1**	**АРМ №2**
L2 – Ethernet	1	1	1	1
L3 – IPv4	1	1	1	1
L4 – TCP	1	1	1	1
L5 – Modbus	1	0.25	0	0
L5 – HTTP	0	0.5	0.5	1
L5 – SSH	0	0.25	0.5	0

За наведеною таблицею нескладно сформувати характеристичні вектори для кожного вузла:

ПЛК = [1, 1, 1, 1, 0, 0]

SCADA = [1, 1, 1, 0.25, 0.5, 0.25]

АРМ №1 = [1, 1, 1, 0, 0.5, 0.5]

АРМ №2 = [1, 1, 1, 0, 1, 0]

Добре помітно, що перші елементи всіх векторів збігаються для кожного типу вузла. Це означає, що вони не матимуть впливу під час навчання моделі, оскільки не несуть інформації, яка допомагає розрізняти класи. Справді важливими для класифікації будуть лише останні компоненти векторів, адже саме вони відображають специфічні протокольні особливості окремих вузлів.

Тепер перевіримо, як саме модель поводитиметься під час обробки нових, раніше невідомих даних. Припустімо, що в мережі з’явився ще один додатковий АРМ. Однак він відрізняється від двох початкових робочих станцій: цей новий вузол використовується виключно для керування SCADA-сервером і виконує спеціалізовані функції.

Для спрощення вважатимемо, що новий АРМ генерує лише один тип трафіку — SSH-з’єднання зі SCADA. Тобто він не працює з HTTP, Modbus чи іншими протоколами, а здійснює лише захищений віддалений доступ.

За таких умов схема взаємодії в мережі змінюється: з’являється новий потік між SCADA та цим АРМ, який представлений виключно SSH-трафіком. На основі цього оновленого набору потоків можна проаналізувати, чи правильно модель віднесе новий вузол до відповідного класу.

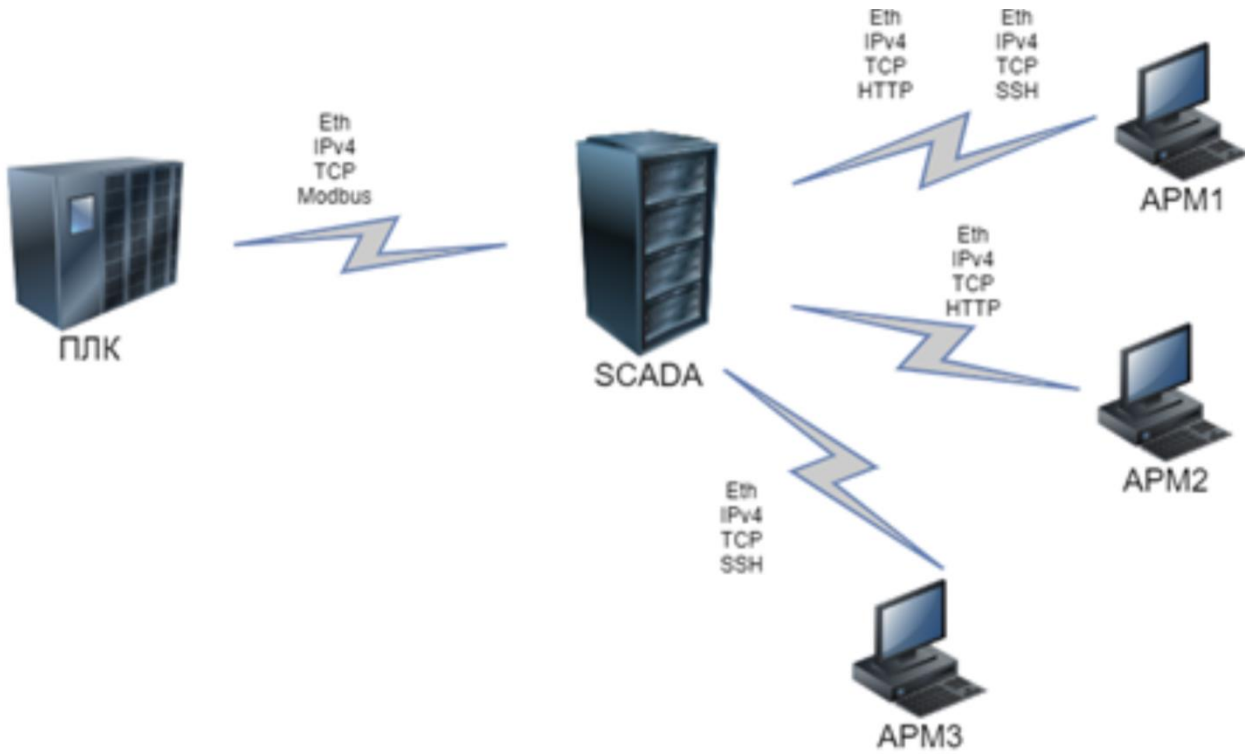


Рис. 1.9 Оновлена мапа потоків

Поява нового вузла призводить не лише до додавання ще одного вектора ознак у загальний набір даних. Одночасно змінюються й характеристичні вектори тих вузлів, з якими цей новий елемент мережі взаємодіє.

У нашому прикладі таким вузлом є SCADA. Оскільки новий АРМ встановлює з нею з’єднання виключно через протокол SSH, частота використання SSH у її характеристичному векторі зростає. Водночас частка інших протоколів у загальній структурі трафіку SCADA пропорційно зменшується, оскільки їхній внесок стає менш значним порівняно з новими потоками.

Таким чином, додавання навіть одного нового пристрою може суттєво вплинути на поведінковий профіль інших вузлів, що важливо враховувати під час аналізу мережевого середовища.

Таблиця 1.3

Частоти використання протоколів вузлами після появи нового вузла

Признак \ Вузол	**ПЛК**	**SCADA**	**АРМ №1**	**АРМ №2**	**АРМ №3**
L2 – Ethernet	1	1	1	1	1
L3 – IPv4	1	1	1	1	1
L4 – TCP	1	1	1	1	1
L5 – Modbus	1	0.2	0	0	0
L5 – HTTP	0	0.4	0.5	1	0
L5 – SSH	0	0.4	0.5	0	1

З наведеного прикладу добре видно, що модель змогла коректно визначити клас нового вузла. Незважаючи на те, що приклад є спрощеним і містить мінімальну кількість елементів, він достатньо точно ілюструє принципи роботи системи та логіку формування рішень.

У реальних умовах сервіс аналізує значно більшу кількість вузлів, а характеристичні вектори кожного з них містять набагато більше ознак — як на рівні протоколів, так і на рівні поведінкових та структурних параметрів. Це робить модель більш стійкою та дозволяє ідентифікувати складніші закономірності у мережевому трафіку.

Попри це, наведений приклад добре демонструє базову концепцію: модель здатна адаптуватися до появи нових пристроїв, переобчислювати їхні профілі та правильно визначати їхню належність до відповідного класу навіть у динамічному середовищі. Це підтверджує, що обраний підхід є ефективним і перспективним для практичного застосування.

2 АНОМАЛІЇ В ЧАСОВИХ РЯДАХ

2.1 Використання методів аналітики

У нашому житті з кожним роком з’являється дедалі більше найрізноманітніших пристроїв, систем, комплексів та платформ — технологічних, інформаційних і кіберфізичних. Ми користуємося ними щодня й майже не замислюємося про їхній внутрішній устрій. Нас мало хвилює, як саме кавоварка готує каву, за яким принципом робот-пилосос обирає маршрут прибирання, яким чином система біометричної ідентифікації розпізнає людину на відео, або як портал державних послуг опрацьовує нашу електронну заявку на довідку про несудимість.

Для пересічного користувача всі ці системи постають як «чорні скриньки»: ми очікуємо передбачуваного результату, не замислюючись про те, у якому стані сама система, чи не з’являються в ній ознаки збою, чи працює вона стабільно. Для споживача такий підхід природний. Але зовсім інакше на це дивляться компанії-розробники та фахівці, які відповідають за експлуатацію технологічних рішень. Саме для них і описано один з оригінальних методів виявлення аномалій у часових рядах — тобто відхилень у поведінці системи, що свідчать про можливу несправність або про ризик її появи найближчим часом.

Ефективна експлуатація складних технологічних систем неможлива без якісного моніторингу та застосування різноманітних аналітичних підходів, які дозволяють контролювати стан системи, керувати параметрами її роботи та своєчасно реагувати на зміни. Моніторинг зазвичай забезпечується типово застосовуваними інструментами — надійними системами збору, зберігання та візуалізації даних. Однак створення дієвих аналітичних модулів вимагає глибших досліджень, експериментів і ґрунтовного розуміння предметної області.

У загальному випадку виділяють чотири основні типи аналітики даних:

Описова аналітика показує накопичені дані у наочному вигляді, часто з попередньою обробкою для кращої інтерпретації. Хоча цей тип аналізу вважається найпростішим, він є фундаментом для використання більш складних методів.

Діагностична аналітика дозволяє зрозуміти причини подій, що вже відбулися. У процесі аналізу виявляються тренди, аномалії, характерні ознаки процесу, шукаються причинно-наслідкові зв’язки та кореляції.

Прогнозна аналітика дає змогу передбачати ймовірні результати на основі виявлених закономірностей, статистичних підходів і моделей, побудованих за історичними даними.

Приписова (предписуюча) аналітика формує оптимальні рішення для виробничих або технологічних завдань, використовуючи прогнози та моделювання. Це може бути оптимізація режимів роботи обладнання, налаштування бізнес-процесів або перелік заходів для запобігання аварійним ситуаціям.

Такий системний підхід дозволяє не лише оцінити поточний стан технологічних процесів, а й прогнозувати ризики, приймати обґрунтовані рішення та забезпечувати стабільну роботу складних інженерних систем.

Для реалізації прогнозної та приписової аналітики найчастіше застосовують методи моделювання, серед яких важливе місце займають алгоритми машинного навчання. Саме вони дають змогу виявляти приховані закономірності, будувати прогнози та пропонувати оптимальні рішення на основі великих обсягів даних. Однак ефективність таких моделей значною мірою залежить від того, наскільки якісно організовано процес збирання, очищення, обробки та попереднього аналізу даних. Якщо дані неповні, зашумлені або некоректні, навіть найсучасніша модель не зможе забезпечити точних результатів.

Кожен із видів аналітики має власний рівень складності — як щодо побудови моделей, так і стосовно необхідного залучення фахівця-аналітика. Описова аналітика потребує мінімальної участі людини й здебільшого базується на візуалізації даних. Діагностична аналітика вимагає більш глибокого розуміння предметної області та здатності інтерпретувати виявлені аномалії або закономірності. Прогнозна аналітика ґрунтується на математичних моделях і машинному навчанні, що значно підвищує її складність. Найвищий рівень — приписова аналітика, де моделі не лише прогнозують можливі сценарії, а й

пропонують конкретні рішення, що потребує як складних алгоритмів, так і експертного контролю за їх коректністю.

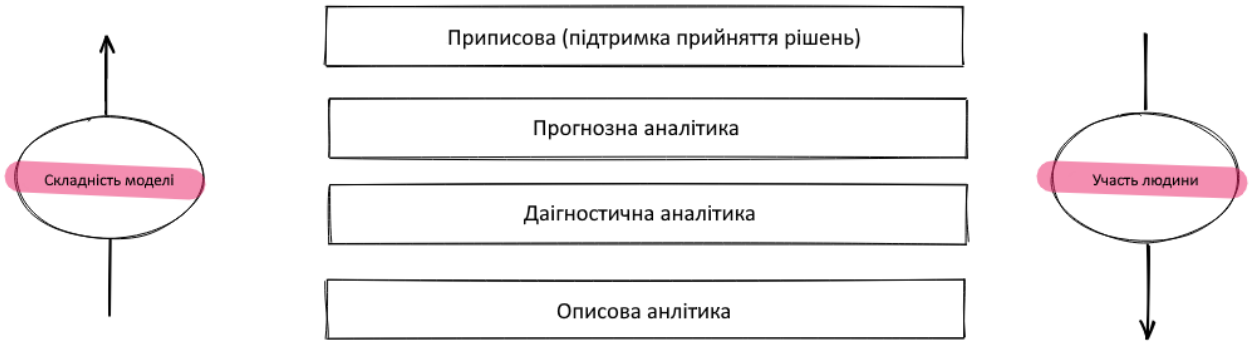


Рис. 2.1 Види аналітик

Сфера застосування аналітичних інструментів надзвичайно широка: це і інформаційна безпека, і банківський сектор, і державне управління, і медицина, а також безліч інших напрямів, де необхідно приймати обґрунтовані рішення на основі даних. Аналітика допомагає виявляти загрози, прогнозувати ризики, оптимізувати робочі процеси, діагностувати відхилення та формувати рекомендації для підвищення ефективності роботи систем.

Досить часто один і той самий метод аналізу даних може успішно застосовуватися у зовсім різних предметних областях. Наприклад, алгоритми кластеризації або методи прогнозування однаково добре працюють як у фінансових операціях, так і у медичних обстеженнях чи в аналізі мережевої активності. Саме тому розробники аналітичних платформ прагнуть створювати універсальні модулі, які містять широкий набір алгоритмів і підходів.

Такі модулі здатні адаптуватися до різних задач: від пошуку аномалій у часових рядах і класифікації об’єктів до оптимізації процесів чи автоматичного прийняття рішень. Гнучкість і універсальність інструментів дозволяють з мінімальними змінами переносити їх між різними сферами, що робить аналітику особливо цінною у сучасних високотехнологічних системах.

2.2 Моніторинг аномалій в технологічних системах

Для більшості сучасних технологічних систем результати моніторингу можуть бути подані у вигляді часових рядів. Це один із найпоширеніших способів представлення даних, коли стан системи фіксується у вигляді послідовності вимірювань, прив'язаних до конкретних моментів часу. Такий підхід дозволяє простежувати динаміку процесів, виявляти тенденції та вчасно помічати відхилення.

Часовий ряд має кілька ключових властивостей:

- **Прив'язка кожного вимірювання до часу його появи.** Кожен елемент ряду — це пара «значення + час», що дозволяє аналізувати зміни параметра у часовому контексті.
- **Рівномірність інтервалів між вимірюваннями.** Дані зазвичай фіксуються через однакові проміжки часу, що забезпечує коректність математичної обробки, зокрема для фільтрації, згладжування чи прогнозування.
- **Можливість відновлення поведінки процесу на основі історичних даних.** Завдяки закономірностям, що містяться у попередніх вимірюваннях, часовий ряд дозволяє передбачати стан системи у поточний момент або навіть прогнозувати її майбутню поведінку. Це робить часові ряди надзвичайно корисними для моделювання та прогнозування.

Загалом часові ряди є універсальним інструментом аналізу, який дає змогу оцінювати стабільність роботи технологічних систем, виявляти аномалії та прогнозувати розвиток подій.

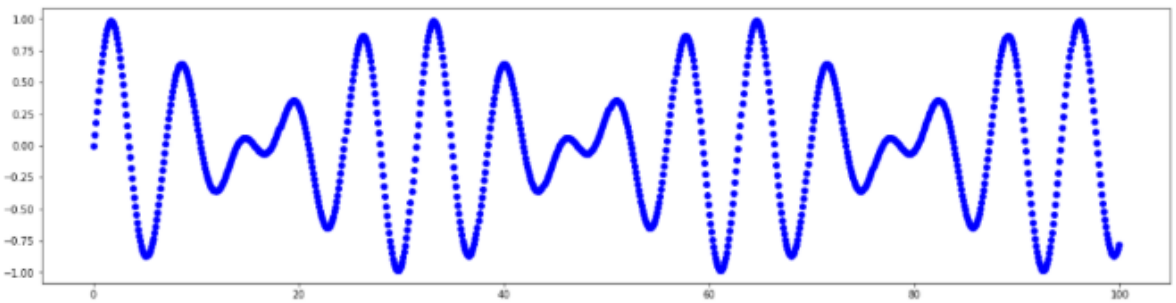


Рис. 2.2 Часовий ряд

Часові ряди здатні описувати не лише процеси, що вимірюються чисельно й мають чітко визначені фізичні параметри. Завдяки розвитку сучасних методів аналізу даних, зокрема глибинного навчання, часові ряди можна використовувати для опрацювання дуже різних типів інформації — навіть такої, що на перший погляд не є кількісною.

Сучасні архітектури нейронних мереж, такі як рекурентні моделі, трансформери чи згорткові мережі, дозволяють опрацьовувати часові послідовності у задачах обробки природної мови (NLP), комп'ютерного зору (CV), багатовимірних сенсорних даних тощо. Наприклад:

- повідомлення в чаті можна перетворити на числові вектори — ембеддинги, які з'являються у певній часовій послідовності;
- відео — це по суті набір числових матриць (кадрів), що змінюються з часом, а отже також утворюють часовий ряд.

Таким чином, часові ряди стають універсальним інструментом, здатним описувати роботу складних систем та процесів різної природи. Вони широко застосовуються у типових аналітичних завданнях, таких як:

- моделювання та відтворення поведінки системи,
- прогнозування майбутніх станів,
- виділення ознак і побудова характеристичних профілів,
- класифікація та кластеризація,
- пошук повторюваних патернів,
- виявлення аномалій.

Приклади практичного використання часових рядів надзвичайно різноманітні: електрокардіограма, коливання курсу акцій або валют, метеорологічні прогнози, зміна обсягу мережевого трафіку, параметри роботи двигуна, технічні показники промислового обладнання тощо.

Завдяки своїй універсальності часові ряди стали одним із ключових інструментів у сучасній аналітиці даних та машинному навчанні.

Часові ряди мають низку типових характеристик, які дозволяють досить точно описати їхню поведінку та структуру. До основних належать:

- **Період** — це відрізок часу постійної довжини, після завершення якого значення ряду повертаються до близьких або подібних. Наприклад, добові зміни температури або трафік у мережі, що повторюється щодня.
- **Сезонність** — прояв періодичності, коли певні закономірності повторюються з однаковим інтервалом (сезон = період). Це може бути тижнева активність користувачів, квартальні економічні показники тощо.
- **Цикл** — зміни значень ряду, що зумовлені глобальними чи довготривалими факторами. На відміну від сезонності, цикл не має фіксованої тривалості. Наприклад, економічні підйоми та спади, багаторічні кліматичні коливання.
- **Тренд** — стійка довгострокова тенденція до зростання або зниження значень ряду. Це може бути, наприклад, поступове збільшення навантаження на сервери або довготривале зростання температури.

Часові ряди можуть містити аномалії — тобто нетипові відхилення у поведінці процесу. Аномалія означає, що в певний момент значення або структура ряду змінилася таким чином, який не відповідає очікуваним закономірностям. Для їх виявлення застосовуються алгоритми машинного пошуку аномалій, що працюють з даними про поведінку системи (датасетами).

У залежності від предметної області та природи даних у часовому ряді можуть зустрічатися різні види аномалій:

- **Точкові аномалії** — це поодинокі значення, що різко відхиляються від загального тренду чи характерного діапазону. Наприклад, раптовий стрибок навантаження на сервер.
- **Групові аномалії** — випадки, коли окремі значення не видаються підозрілими, але їхня сукупність утворює аномальний фрагмент. Наприклад, нехарактерна комбінація параметрів двигуна протягом кількох секунд.
- **Аномалії контексту** — ситуації, коли значення є нормальним самі по собі, але нетиповим у певному контексті. Наприклад, низька температура взимку є нормальною, але така сама температура влітку — вже аномалія.

Точкові аномалії є найпростішим типом для розпізнавання, оскільки вони проявляються у вигляді окремих значень, що різко вибиваються із загальної

картини поведінки процесу. Це такі ситуації, коли в одному конкретному моменті часу параметр приймає значення, яке значно відрізняється від усіх інших спостережень.

Наприклад, можна зафіксувати раптовий різкий стрибок температури, напруги, навантаження на сервер або інший вимірюваний параметр — тоді як усі інші точки ряду мають відносно стабільні значення. Подібне точкове відхилення часто свідчить про короткочасний збій, помилку вимірювання або потенційну проблему в роботі обладнання.

Саме через свою локальність такі аномалії виявляються найпростішими з погляду алгоритмів — вони не потребують аналізу довготривалих патернів, а визначаються за принципом різкого контрасту з сусідніми значеннями.

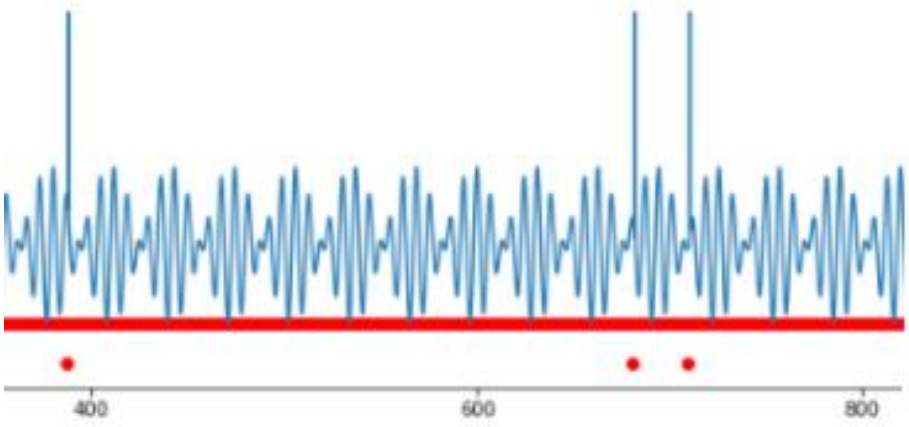


Рис. 2.3 Декілька точкових аномалій

Такі значення часто називають «викидами», оскільки вони різко вибиваються із загальної картини та суттєво впливають на статистичні характеристики процесу. Викиди можуть сильно спотворювати середні значення, дисперсію та інші метрики ряду. Їх виявлення зазвичай не становить труднощів — достатньо встановити порогові значення або діапазони допустимих величин, за межами яких точка автоматично вважається аномальною.

Набагато складніше розпізнати аномалію у випадках, коли кожне окреме значення виглядає цілком нормальним, але їх сукупність формує «дивну» або нетипову поведінку. Такі ситуації є більш тонкими та часто не помітні при простому перегляді графіка.

До подібних групових аномалій можна віднести, наприклад:

- зміну форми сигналу без різких стрибків;
- зміну статистичних характеристик — середнього, моди, медіани, дисперсії;
- появу або зникнення кореляції між двома параметрами;
- невеликі, але підозрілі коливання амплітуди;
- короткочасні відхилення, що самі по собі не здаються аномальними, але порушують загальну структуру процесу.

У таких випадках стандартні статистичні методи вже не здатні виявити проблему, оскільки вони оцінюють лише окремі точки, а не поведінку фрагмента ряду в цілому. Тому основне завдання полягає в тому, щоб навчитися визначати аномальні патерни, які проявляються тільки при розгляді групи значень разом. Саме такі аномалії найчастіше є ознакою складних відхилень у роботі системи й потребують застосування більш просунутих алгоритмів аналізу та машинного навчання.

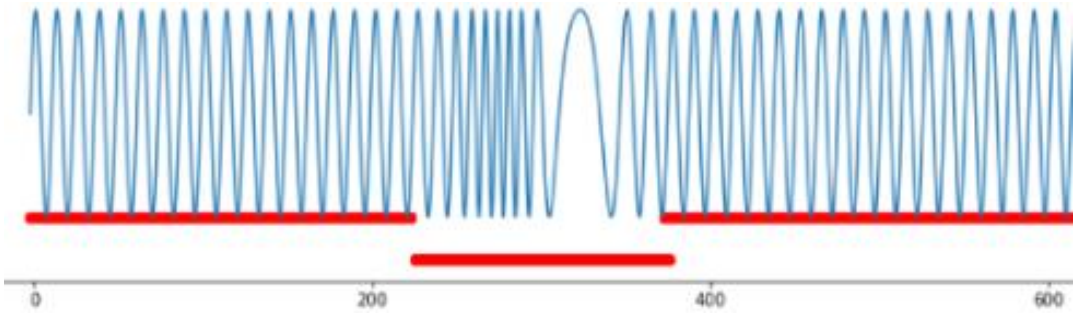


Рис. 2.4 Групова аномалія, зміна частоти

Пошук аномалій відіграє надзвичайно важливу роль у роботі з даними та технічними системами. В одних випадках необхідно видаляти аномальні значення, щоб очистити дані й отримати більш достовірну картину процесів. В інших ситуаціях, навпаки, аномалії потребують особливої уваги, оскільки вони можуть бути ранніми ознаками наближення аварійного або нестабільного режиму роботи обладнання.

Виявлення аномалій у часових рядах є складним завданням. Це зумовлено кількома факторами: нечітким визначенням того, що саме вважати аномалією; відсутністю готової розмітки даних; неочевидними зв’язками між параметрами. Навіть сучасні SOTA-алгоритми пошуку аномалій у часових рядах демонструють

досить високий рівень False Positive, тобто помилкових тривог, коли нормальні значення помилково позначаються як аномальні.

Людина здатна виявити лише обмежену кількість аномалій — переважно точкових — і то за умови якісної візуалізації часових рядів. Групові аномалії помітити значно важче, особливо коли йдеться про великі масиви даних або про одночасний аналіз параметрів різних пристроїв. Додаткову складність становлять аномалії у часі, коли сигнал за своїми значеннями є абсолютно нормальним, але з'являється у «неправильний» момент — наприклад, пікове навантаження в період, коли його не повинно бути.

Через це під час обробки часових рядів доцільно використовувати автоматизовані методи пошуку аномалій, здатні аналізувати великі обсяги даних та виявляти приховані відхилення.

Однією з найбільших проблем у роботі з реальними даними є те, що вони переважно не розмічені. У таких випадках заздалегідь невідомо, які стани вважати нормальними, а які — аномальними. Немає чітких правил або готових шаблонів для пошуку відхилень. Саме тому в подібних задачах широко використовуються методи навчання без учителя (unsupervised learning). Такі моделі самостійно виявляють закономірності, формують структуру даних, визначають типові групи поведінки та виділяють все те, що не вписується у норму.

Таким чином, автоматизований пошук аномалій є необхідним інструментом для аналізу часових рядів і забезпечує раннє виявлення проблем у складних техніч иних системах.

Методи, які застосовуються для пошуку аномалій у часових рядах, зазвичай поділяють на кілька основних груп. Кожна з них має власні сильні сторони та обмеження, тому підходить для різних типів аномалій:

1. Proximity-based методи (методи близькості)

Ці підходи визначають аномалії на основі порівняння значень параметрів або коротких фрагментів часових рядів між собою. Ідея полягає в тому, що «нормальні» точки утворюють щільні кластери, тоді як аномалії помітно віддаляються від групи.

Такі методи добре справляються із виявленням точкових аномалій та викидів, тобто окремих різких відхилень. Однак вони майже не реагують на зміни форми сигналу або складні групові аномалії.

Найпростіший приклад proximity-based методу — перевірка на перевищення порогового значення, коли сигнал виходить за межі встановлених норм.

2. Prediction-based методи (методи прогнозування)

Цей підхід передбачає побудову моделі, що здатна прогнозувати значення часової послідовності. Далі модельні (прогнозні) значення порівнюють із фактичними.

- Якщо прогноз збігається з реальними даними, вважається, що поведінка є нормальною.
- Якщо ж прогноз і реальний сигнал суттєво розходяться, цей фрагмент визначається як аномальний.

Prediction-based методи особливо ефективні для рядів із чітко вираженою сезонністю, періодичністю або циклічністю, наприклад: добові зміни, регулярні піки навантаження, робочі цикли обладнання.

3. Reconstruction-based методи (методи реконструкції)

Ці моделі намагаються «відтворити» (реконструювати) фрагмент вихідних даних, використовуючи інформацію з інших ділянок ряду. Якщо модель добре реконструює сигнал — він вважається нормальним. Якщо ж реконструкція значно відрізняється від реальних значень, фрагмент позначається як аномальний.

Перевага reconstruction-based підходів у тому, що вони здатні виявляти як точкові аномалії, так і групові, включно з:

- зміною форми сигналу,
- нетиповими шаблонами,
- поступовим дрейфом,
- аномальними комбінаціями параметрів.

Прикладами таких методів є автоенкодери, PCA-реконструкція, денойзинг-моделі та ін.

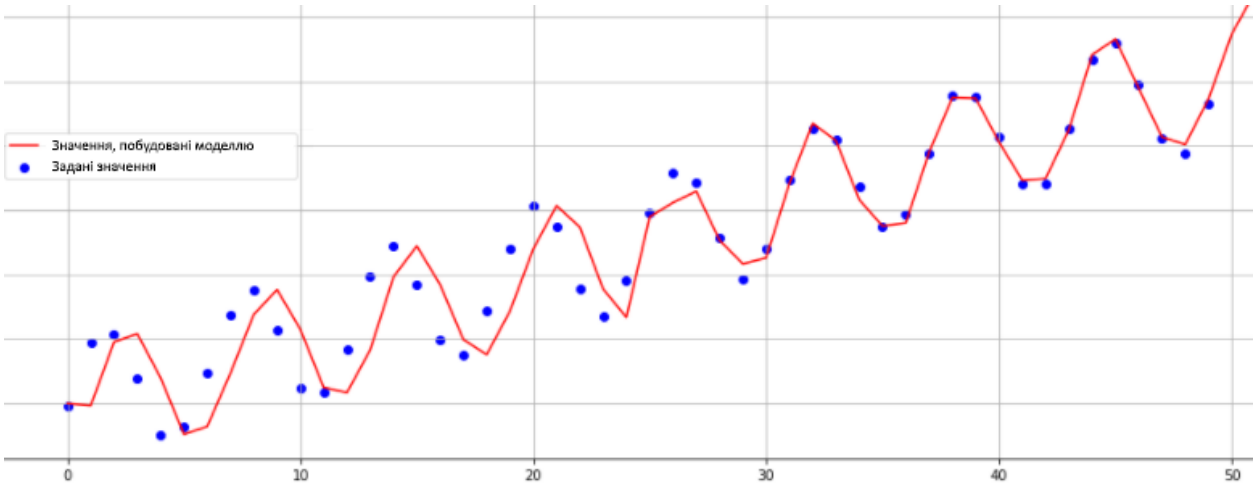


Рис. 2.5 Моделювання часового ряду

Найпоширенішими підходами до моделювання часових рядів сьогодні є SARIMA-моделі та рекурентні нейронні мережі (RNN), які добре працюють із послідовними даними та здатні враховувати залежності у часі.

У reconstruction-based моделях застосовується більш оригінальний і водночас потужний підхід. Спочатку модель навчають кодувати та декодувати сигнали з наявного набору даних. Під час такого навчання модель отримує фрагмент часового ряду на вході, а на виході має відтворити його з максимальною точністю.

Особливість у тому, що на проміжному етапі дані перетворюються на представлення значно меншої розмірності — компактний вектор ознак. Тобто мережа повинна навчитися стискати інформацію, зберігаючи при цьому найважливіші закономірності сигналу. Саме здатність моделі відновити сигнал із цього стислого подання дозволяє виявляти аномалії: якщо відновлення «провалюється», значить у фрагменті є нетипова поведінка.

Наприклад, у задачах комп’ютерного зору такий принцип використовується для роботи з зображеннями: картинку розміром 32×32 пікселі можна перетворити в компактний вектор із 32 чисел. Попри настільки сильне стискання, модель зазвичай зберігає основні риси об’єкта і може відтворити зображення з достатньою точністю. Якщо ж у даних з’являється аномальна структура, модель не зможе її коректно «закодувати» і «розкодувати», а значить — аномалія буде виявлена.

Такий підхід працює не лише для картинок, а й для часових рядів, дозволяючи фіксувати відхилення форми сигналу, раптові стрибки, зміну циклічності, поєднання нетипових патернів і багато інших видів аномалій.

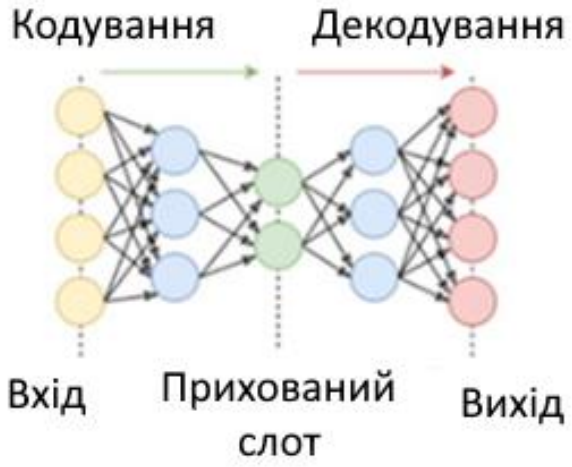


Рис. 2.6 Схема роботи автокодувальника

Після завершення навчання моделі їй на вхід подають фрагменти сигналів, які представляють собою відрізки досліджуваного часового ряду. Якщо процес «кодування → декодування» проходить успішно та відновлений сигнал достатньо близький до оригіналу, то таке поведінкове відхилення вважається нормальним.

Однак у випадках, коли модель не здатна якісно відтворити поданий відрізок, це свідчить про аномальну поведінку системи, і відповідний фрагмент позначається як підозрілий. Такий підхід добре працює для виявлення не лише окремих «викидів», а й складних групових аномалій — наприклад, змін у формі сигналу або незвичних поєднань параметрів.

Одним з найсучасніших і найефективніших методів reconstruction-based аналізу, який продемонстрував високі результати у задачах пошуку аномалій, є TadGAN. Цей підхід був розроблений дослідниками MIT наприкінці 2020 року та поєднує в собі архітектурні принципи автокодировувачів і генеративно-змагальних нейронних мереж (GAN).

TadGAN працює таким чином, що одночасно навчається реконструювати сигнал та генерувати реалістичні його фрагменти. Завдяки цьому метод вміє

«розуміти», яким має бути природний вигляд часового ряду, і дуже точно помічає будь-які відхилення від норми. У порівнянні з класичними автокодировувачами, TadGAN точніше відтворює складні процеси, зменшує кількість хибнопозитивних спрацьовувань та краще підлаштовується під специфіку реальних технічних сигналів.

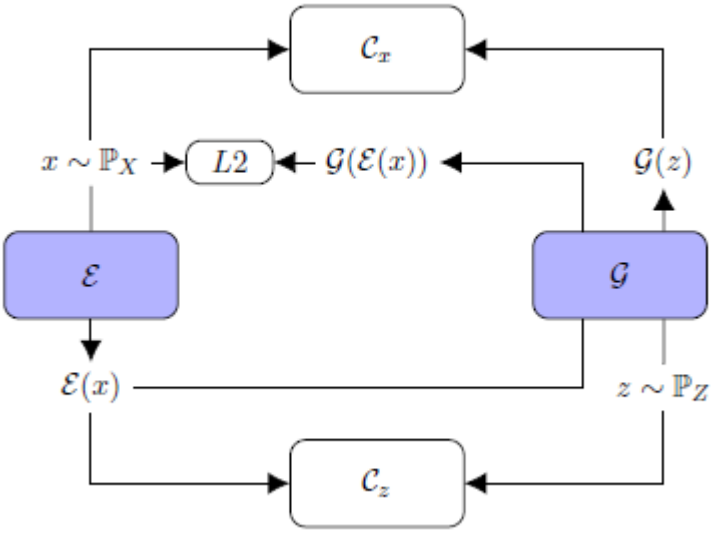


Рис. 2.7 Архітектура TadGAN

Модельний компонент Ɛ виконує роль кодера, який перетворює фрагменти часового ряду *x* у вектори прихованого простору *z*. У свою чергу G діє як декодер, що з цього прихованого подання *z* відновлює відповідні відрізки часового ряду.

Блок Cx виступає критиком, який оцінює якість відновлення сигналу — тобто порівнює вихід *G(Ɛ(x))* з оригінальним фрагментом *x*. Блок Cz аналізує, наскільки приховане подання *z = Ɛ(x)* схоже на випадковий шум, що є характерною вимогою для генеративних моделей.

Крім того, у методі застосовується контроль схожості між початковим та відновленим зразками, який обчислюється за допомогою L2-міри відповідно до принципів *cycle consistency loss*. Такий підхід забезпечує загальну близькість синтезованих фрагментів до оригінальних даних, що є важливою умовою для моделей типу GAN [6].

Підсумкова цільова функція є комбінацією всіх перелічених складників:

— оцінок критика Cx,

— оцінок критика Cz,

— а також міри схожості між вихідним сигналом і відновленим.

Таким чином досягається комплексна оцінка якості роботи моделі та її здатності коректно відтворювати «нормальні» сигнали.

2.3 Нейронні мережі для виявлення аномалій

Для створення та навчання нейронних мереж сьогодні доступна велика кількість стандартних бібліотек і фреймворків, що надають високорівневі інструменти для побудови складних архітектур. Завдяки таким інструментам розробник може сконцентруватися на логіці моделі, не витрачаючи час на реалізацію низькорівневих обчислень.

Приклад реалізації архітектури, подібної до TadGAN, разом із процесом навчання ваг, можна знайти у відкритому репозиторії [7]. Під час навчання цієї моделі оптимізувалося п’ять ключових метрик, кожна з яких відповідає за певний аспект роботи мережі:

- aeLoss — середньоквадратичне відхилення між початковим і відновленим часовим рядом, тобто різниця між вихідним фрагментом x та реконструйованим сигналом $G(\mathcal{E}(x))$. Фактично ця метрика показує, наскільки добре модель може відтворити реальний сигнал.
- cxLoss — бінарна крос-ентропія критика Cx, який визначає, наскільки reconstructed-сигнал відрізняється від справжнього фрагмента ряду. Чим точніше критик розрізняє реальний та штучно отриманий відрізок, тим краще працює ця частина мережі.
- cx_g_Loss — бінарна крос-ентропія, що характеризує помилку генератора $G(\mathcal{E}(x))$ у спробах «обманути» критика Cx. Якщо ця метрика велика, це означає, що генератор ще недостатньо добре навчився створювати правдоподібні реконструкції.
- czLoss — бінарна крос-ентропія критика Cz, який порівнює вектори прихованого простору $z = \mathcal{E}(x)$ із випадковим шумом. Це забезпечує те, що

приховані вектори не зберігають чітких шаблонів з історичних даних, а мають риси хаотичності. Таким чином модель не «запам’ятовує» конкретні патерни, а навчається узагальнювати.

- cz_g_Loss — бінарна крос-ентропія, що описує помилку генератора $\mathcal{E}(x)$ при спробі формувати приховані вектори, максимально схожі на випадкові, аби «заплутати» критика *Cz*. Якщо це значення зменшується, модель краще узагальнює структуру даних.

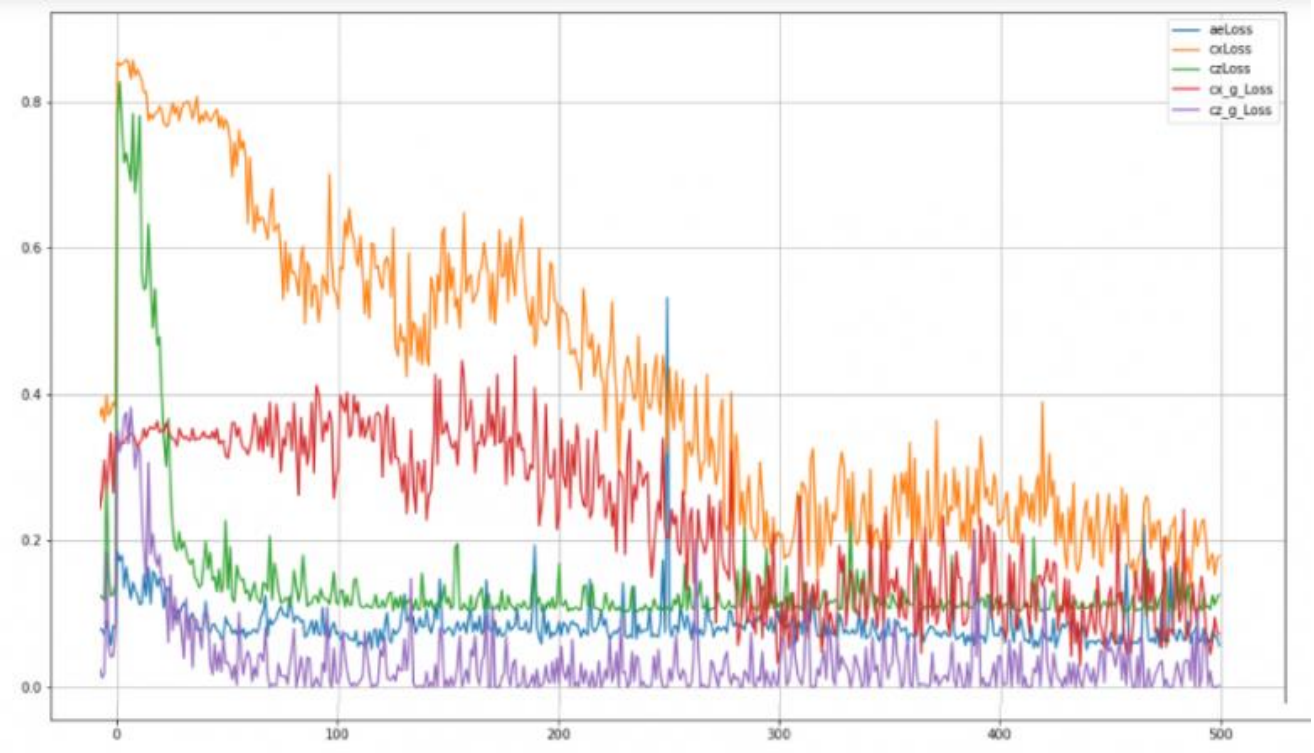


Рис. 2.8 Графік навчання моделі TadGAN на 500 епох

Після завершення навчання моделі переходять до етапу реконструкції фрагментів досліджуваного часового ряду. На цьому етапі модель отримує послідовні відрізки сигналу й намагається відтворити їх на основі свого внутрішнього уявлення. Далі виконується порівняння між оригінальним фрагментом та його реконструкцією. Саме якість цього відновлення й слугує основою для визначення наявності або відсутності аномалій.

Порівняння може виконуватися кількома способами, серед яких найпоширеніші:

- Поточкове порівняння.

- У цьому випадку значення сигналу порівнюються «точка до точки». Якщо різниця перевищує певний поріг, відповідний фрагмент вважається аномальним. Метод простий, але чутливий до шуму.
- Порівняння площ під кривими у визначеній області навколо кожного сэмплу. Довжина цієї області задається як гіперпараметр моделі. Такий підхід дозволяє враховувати локальну форму сигналу, що підвищує точність порівняння та робить аналіз менш чутливим до випадкових коливань.
- Dynamic Time Warping (DTW). Це популярний алгоритм, який дозволяє порівнювати часові ряди різної швидкості або ряди, в яких певні фрагменти «розтягнуті» чи «стиснуті у часі». DTW знаходить оптимальне вирівнювання між двома послідовностями, що робить його особливо ефективним для аналізу складних та нерівномірних сигналів.

Після виконання реконструкції та порівняння визначається, які ділянки ряду можна вважати нормальними, а які — аномальними. Для кількісної оцінки якості роботи моделі застосовують F1-метрику у постановці задачі бінарної класифікації, де:

- Positive (нулева гіпотеза) означає: *аномалія присутня*;
- Negative (альтернативна гіпотеза) — *аномалії немає*.

F1-міра дозволяє збалансовано оцінити якість виявлення аномалій з урахуванням як точності, так і повноти, що особливо важливо в умовах, де критичними є як пропущені аномалії, так і хибні спрацювання.

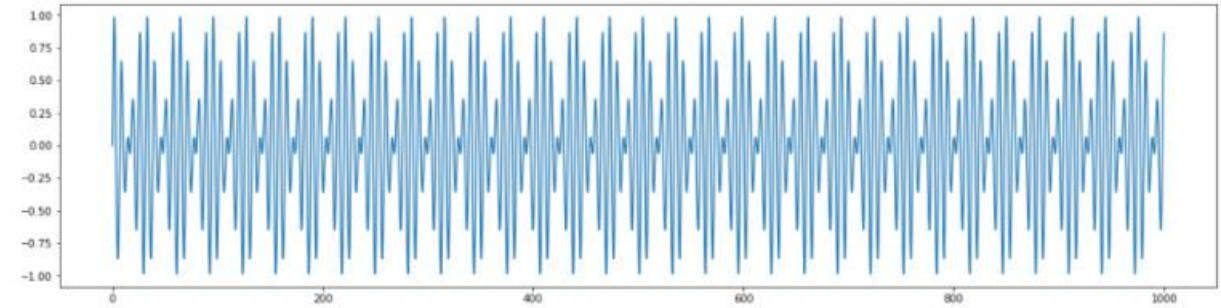
Для демонстрації роботи методу використаємо синтетичний (штучно згенерований) часовий ряд, у якому відсутні будь-які аномалії. Цей ряд сформовано як суму двох синусоїд, а його значення змінюються в діапазоні від –1 до 1.

$$y(x) = \frac{1}{2}\sin(x) + \frac{1}{2}\sin(0.8x) \tag{2.1}$$

Графік даного ряду:

```
xs = np.linspace(0, 1000, 10000)
ys1 = 0.5*np.sin(xs) + 0.5*np.sin(0.8*xs)
label1 = np.zeros(10000)
```

```
plt.figure(figsize=(20,5))
plt.plot(xs, ys1)
plt.show()
```



```
plt.figure(figsize=(20,5))
plt.plot(xs[:1000], ys1[:1000])
plt.show()
```

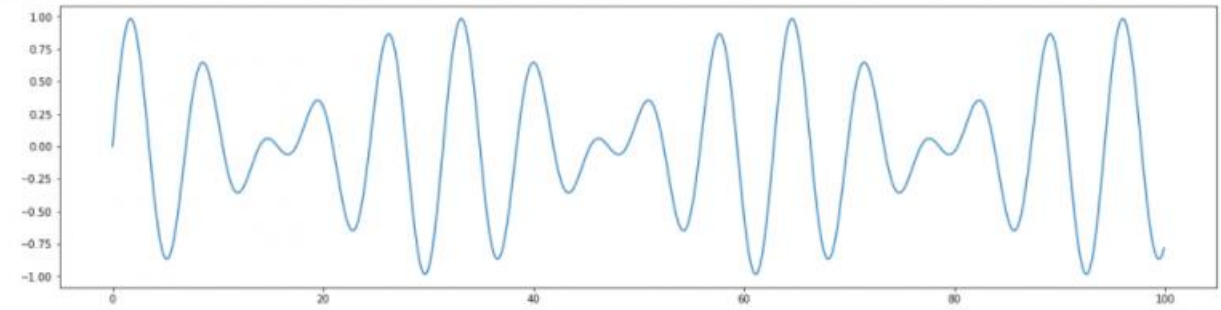


Рис. 2.9 Графік синтетичного ряду

Відновлений ряд з використанням TadGAN для різної кількості епох (4 та 80) виглядає так:

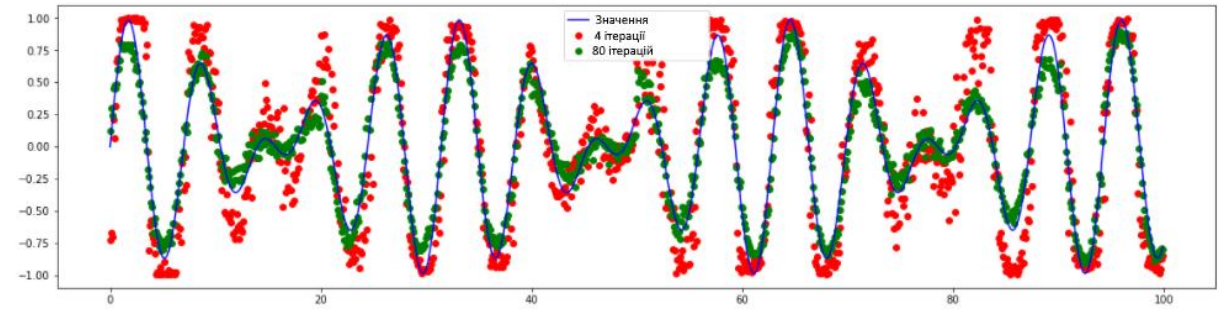


Рис. 2.10 Моделювання ряду за допомогою TadGAN для різної кількості епох (4 та 80)

Добре видно, що модель досить точно навчилася відтворювати основні закономірності у даних. Тепер спробуємо ускладнити ситуацію: додамо до часового ряду кілька різних аномалій і перевіримо, чи зможе модель TadGAN коректно їх виявити. Для початку внесемо у дані кілька точкових аномалій.

```
xs = np.linspace(0, 1000, 10000)
ys2 = 0.5*np.sin(xs) + 0.5*np.sin(0.8*xs)
idxs = np.random.choice(len(ys2), size=5)
max_ = np.max(ys2)
ys2[idxs] = max_*np.random.randint(5)
label2 = np.zeros(10000)
label2[idxs] = 1
```

```
plt.figure(figsize=(20,5))
plt.plot(xs, ys2)
plt.scatter(xs, label2*(-2)-1, c='red')
plt.show()
```

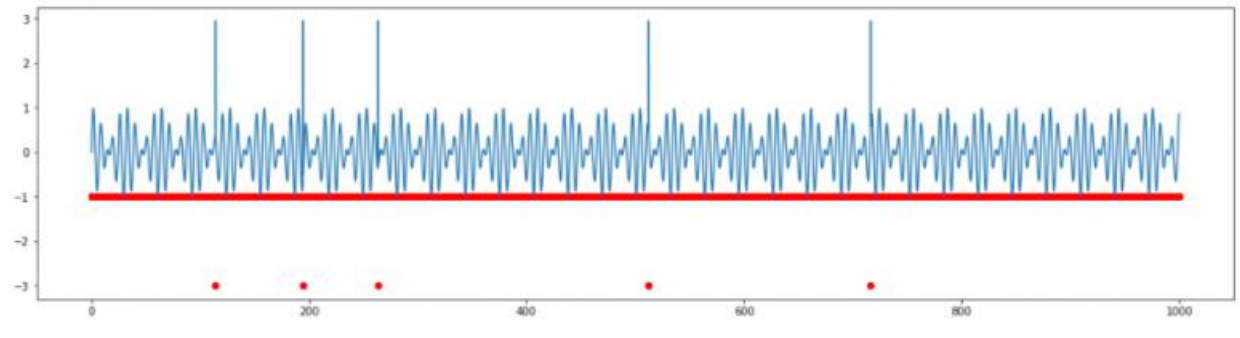


Рис. 2.11 Виявленя точкових аномалій за допомогою TanGAN

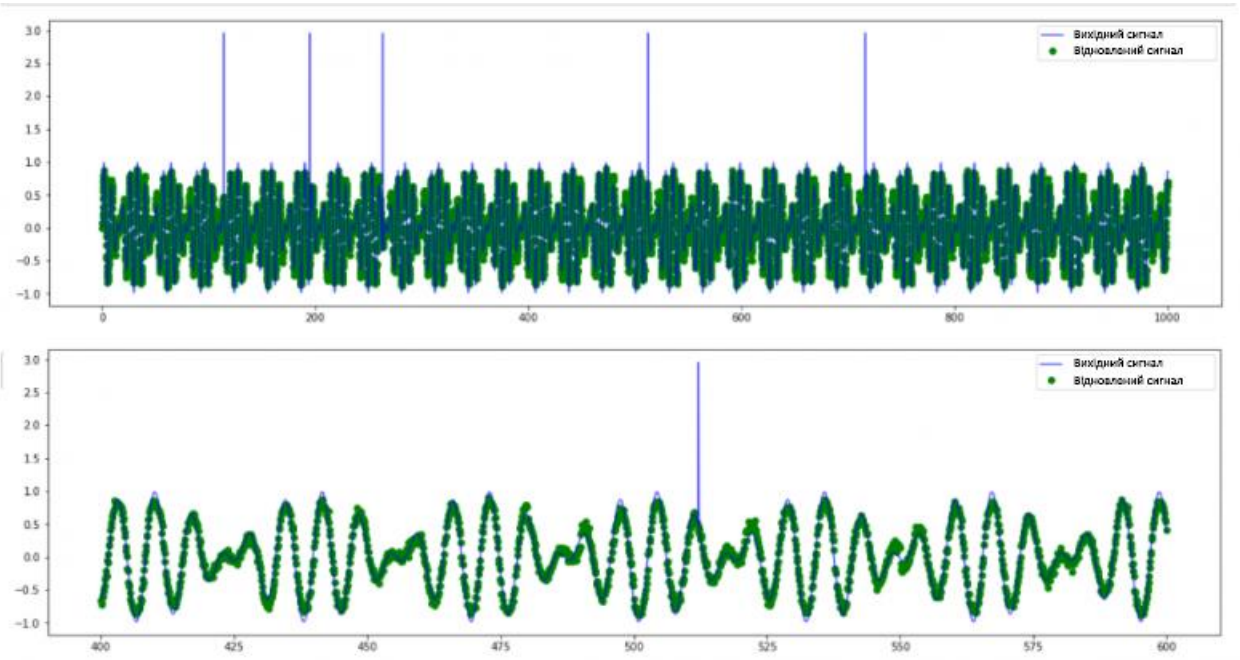


Рис. 2.12 Виявленя точкових аномалій за допомогою TanGAN

З графіка, який демонструє порівняння вихідного та відновленого сигналів, добре помітно, що модель не здатна точно реконструювати різкі «пікові» відхилення в даних. Такі точки істотно вибиваються із загальної структури ряду, і TadGAN їх не відтворює — саме це і може слугувати надійною ознакою точкових аномалій. Однак варто зазначити, що для таких простих випадків використання складної моделі на кшталт TadGAN не завжди є виправданим: подібні аномалії можна виявити й елементарними методами, зокрема пороговою фільтрацією.

Перейдемо до більш складного прикладу аномалії. Розглянемо сигнал, у якому присутні «нормальні» за амплітудою значення, але спостерігається аномальна зміна частоти. Такий тип відхилення не проявляється у вигляді перевищення порогів, адже амплітуда залишається в межах допустимого діапазону. Відхилення видно лише у груповій поведінці кількох сусідніх точок, де порушується типова закономірність зміни сигналу.

У цій ситуації модель TadGAN також демонструє здатність виявляти аномалію: вона не може коректно відтворити такий фрагмент через порушення звичної структури сигналу. На відповідному графіку це проявляється як значне розходження між відновленим і вихідним рядом. Саме така невідповідність може бути використана як ознака групової аномалії, що є набагато складнішим типом відхилення, ніж точкові аномалії.

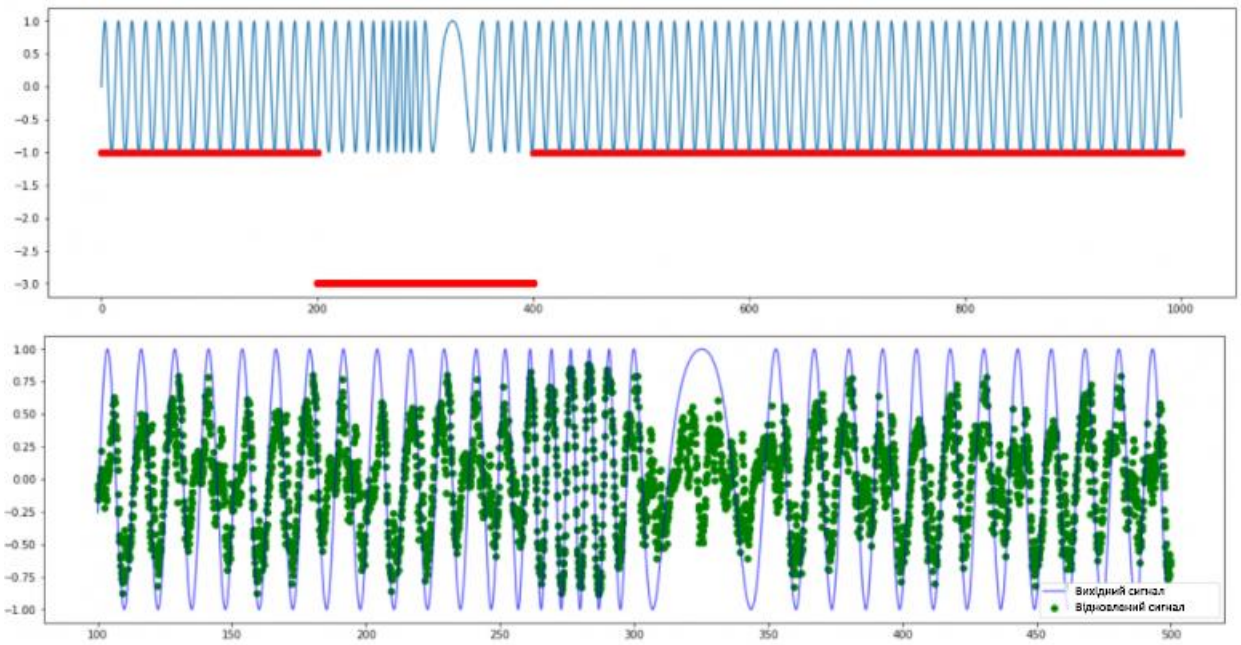


Рис. 2.13 Результат роботи TadGAN на датасеті з аномальною зміною частоти

На наведених прикладах продемонстровано базові можливості методу та його поведінку на даних із різними типами аномалій. Читач за бажанням може самостійно сформувати власні набори даних та перевірити, як модель реагуватиме на ті чи інші відхилення, експериментуючи з формою сигналу, шумами, періодичністю чи амплітудою.

Більш складні приклади наборів даних, а також повне описання експериментів авторів, наведені у статті розробників методики TadGAN. У ній також міститься посилання на бібліотеку Orion — інструмент, створений у MIT, що використовує методи машинного навчання для виявлення рідкісних аномалій у часових рядах на основі навчання без учителя (unsupervised learning). Це робить його зручним засобом для аналізу реальних, погано структурованих або нерозмічених даних.

Підсумовуючи, варто зазначити, що сьогодні існує велика кількість методів, які використовують реконструкцію сигналу (reconstruction-based підходи) для виявлення аномалій. На платформі arXiv можна знайти десятки наукових статей, де описано модифікації алгоритмів автокодировщиків, генеративно-змагальних мереж та їхніх гібридів. Кожен метод має свої сильні сторони й обмеження, тому для кожної конкретної задачі важливо обирати модель, виходячи з вимог, типу даних і знання предметної області.

Грамотно підібраний алгоритм у поєднанні з якісною підготовкою даних дозволить максимально ефективно виявляти відхилення в роботі складних систем та забезпечувати їхню стабільність.

3 АНОМАЛІЇ В МІКРОСЕРВІСНІЙ АРХІТЕКТУРІ

Тема виявлення аномалій сьогодні є надзвичайно актуальною. З переходом компаній на мікросервісну архітектуру для команд SRE та DevOps значно зростає пріоритет задач, пов’язаних із перетворенням великої кількості алертів на осмислені сигнали, зменшенням MTTD (mean time to detect) та спрощенням налаштування правил оповіщення у розподілених системах.

Чому виявлення аномалій набуває критичного значення?

Перехід до мікросервісної моделі означає, що кількість окремих сервісів збільшується в рази, а кількість контрольованих метрик — у десятки чи навіть сотні разів. Відповідно, складність моніторингу зростає експоненційно. Під «контролем» мається на увазі спроможність вчасно отримати осмислений сигнал, коли продуктивність інфраструктури або сервісів починає погіршуватися.

У більшості систем оповіщення базуються на тригерах — необхідно вручну визначити порогові значення, при перевищенні яких надсилається алерт. Але як встановити пороги для сотень сервісів і тисяч різнорідних метрик? І чи потрібно взагалі робити це вручну?

Приклади ситуацій, де аномалії відіграють ключову роль

- Різке збільшення latency лише на одному сервісі серед ста.
- Зростання кількості помилок на одному з тисяч endpoint’ів.
- Зменшення кількості запитів у період, коли зазвичай спостерігається пік активності.

Такі зміни можуть загубитися серед величезного потоку даних. До того ж реалістично важко налаштувати пороги для кожної метрики — про щось можна забути, щось пропустити, а інколи зміни настільки слабкі, що їх не помітить людина.

Навіщо потрібне виявлення аномалій у моніторингу?

Методи автоматичного пошуку аномалій дозволяють:

- виявляти проблеми ще до того, як вони призведуть до відмови сервісу;

- автоматизувати моніторинг, усуваючи необхідність ручного налаштування порогів;
- знаходити «слабкі сигнали», які вказують на ранні ознаки деградації системи;
- працювати з дуже великими наборами метрик, де класичний підхід «поріг → алерт» більше не працює.

З чого почати впровадження пошуку аномалій?

Сьогодні існують три основні підходи до реалізації систем виявлення аномалій:

1. Зробити самому (in-house)

   Створення власної системи з використанням ML/AI, статистики, нейронних мереж та кастомних моделей.
2. Використовувати функціонал APM-систем

   Сучасні Application Performance Monitoring інструменти (New Relic, Datadog, Dynatrace тощо) вже мають готові модулі для автоматичного аналізу аномалій у метриках і логах.
3. Anomaly Detection as a Service

   Хмарні рішення або спеціалізовані сервіси, які підключаються до метрик та самостійно виконують аналіз, рекомендації та генерацію сигнальних сповіщень.

3.1 Підхід in-house

Готових реалізацій алгоритмів для виявлення аномалій сьогодні існує дуже багато — практично для будь-якої мови програмування. Це можуть бути як популярні інструменти на Python, так і спеціалізовані засоби на R чи інших платформах.

Ба більше, навіть у стандартній системі моніторингу Prometheus вже містяться вбудовані можливості для роботи зі статистикою, які дають змогу знаходити аномальні значення у часових рядах метрик. Для кожної метрики, що аналізується, зазвичай потрібно створити кілька recording rules — окремих правил,

які фіксують розраховані статистичні величини. Проте цей процес є доволі клопітким і потребує часу, особливо коли метрик багато.

Якщо ж прийняти припущення, що аналізовані дані мають нормальний розподіл, то можна використати простий, але ефективний статистичний підхід: вважати аномальними всі значення, що виходять за межі трьох стандартних відхилень від середнього (так зване *правило трьох сигм*). У цьому випадку реалізація виявлення аномалій зводиться до розрахунку та збереження трьох ключових величин:

- середнього значення метрики,
- стандартного відхилення,
- z-оцінки (z-score), яка показує, наскільки далеко спостережуване значення відхиляється від середнього у одиницях стандартних відхилень.

Наприклад, якщо розглядати метрику http_requests_total, то першим кроком буде її агрегація — оскільки в сирому вигляді вона може бути надто деталізованою або нерівномірною для подальшого аналізу. Після агрегування можна переходити до розрахунків статистичних показників і побудови правил для автоматичного виявлення відхилень.

Такий підхід підходить для базових сценаріїв і може слугувати відправною точкою для впровадження автоматизованого пошуку аномалій у системах моніторингу. Якщо згодом виникне потреба — його завжди можна розширити складнішими моделями машинного навчання або готовими рішеннями з APM-платформ.

```
# агрегація за п’ять хвилин
- record: job:http_requests:rate5m
  expr: sum by (app) (rate(http_requests_total[5m]))
```

Далі обчислюємо три необхідні набори даних:

```
# average — середнє значення
- record: job:http_requests:rate5m:avg_over_time_1w
  expr: avg_over_time(job:http_requests:rate5m[1w])
```

```
# stddev — стандартне відхилення
- record: job:http_requests:rate5m:stddev_over_time_1w
  expr: stddev_over_time(job:http_requests:rate5m[1w])
# z-оцінка
(job:http_requests:rate5m - job:http_requests:rate5m:avg_over_time_1w)
  / job:http_requests:rate5m:stddev_over_time_1w
```

Більшість даних, що описують продуктивність і навантаження застосунків (наприклад, швидкість обробки запитів чи latency), мають виразну **сезонність**. Це означає, що навантаження на систему протягом доби, тижня або року розподіляється нерівномірно, але такі коливання повторюються з певною регулярністю. Саме ця повторюваність і є сезонністю.

Через сезонний характер даних порівнювати поточне значення метрики лише із загальним середнім — некоректно. Значення, яке є нормальним для пікового часу доби, виглядатиме як аномальне, якщо порівнювати його із середнім за день або тиждень.

Щоб вирішити цю проблему, необхідно враховувати сезонність при визначенні аномалій. Для цього використовують підхід, за якого:

1. На основі історичних даних будується прогноз майбутніх значень метрики.
2. Поточне значення порівнюється не з усередненим показником, а саме з очікуваним (прогнозованим) значенням.
3. Відхилення між фактичним та прогнозованим значенням оцінюється за допомогою статистичних методів — наприклад, за z-оцінкою, яка показує, наскільки сильно метрика відхилилася від прогнозу в одиницях стандартного відхилення.

Для побудови прогнозу можна застосовувати:

- класичні статистичні методи (наприклад, ARIMA або її сезонне розширення SARIMA);
- методи згладжування;
- ансамблі кількох моделей, коли декілька прогнозів об’єднуються в один узагальнений ряд;

- прості моделі, що враховують повторюваність поведінки за попередні періоди (доба, тиждень, місяць).

Після того як прогноз отримано, постає наступне завдання — визначити межі “допустимого” відхилення від прогнозу. Саме тут знову корисною стає z-оцінка: вона дозволяє оцінити, наскільки значення вибивається з очікуваного сезонного патерну.

Таким чином, для систем із сезонними навантаженнями найефективнішим є підхід, коли:

- поточне значення порівнюється з прогнозом,
- а не з фіксованим порогом чи середнім;
- аномалія визначається за відхиленням від прогнозованого значення з урахуванням статистичної значущості.

Цей підхід забезпечує значно точніше виявлення реальних проблем у продуктивності системи та мінімізує кількість хибних спрацьовувань у моніторингу.

3.2 Prometheus Anomaly Detector (PAD)

Проєкт Prometheus Anomaly Detector (PAD), який розвивається під егідою Red Hat, значною мірою автоматизує усі дії, описані у попередньому розділі. Фактично PAD бере на себе рутинну частину роботи, пов’язану з налаштуванням правил, прогнозуванням та визначенням меж нормальної поведінки метрик.

У PAD користувач може обрати вже існуючі метрики з Prometheus, які потребують аналізу. Після цього система автоматично:

- створює необхідні *recording rules*;
- будує прогноз поведінки метрики;
- обчислює верхні та нижні порогові значення;
- пропускає всі дані через механізм прогнозування Prophet, який враховує сезонність, тренди, цикли та інші властивості часових рядів.

Таким чином, PAD значно спрощує впровадження аномалійного моніторингу: замість ручного налаштування десятків формул і правил, система генерує весь необхідний набір обчислень автоматично. Це особливо корисно для великих мікросервісних систем, де кількість метрик і сервісів може вимірюватися тисячами, а отже автоматизація стає критично важливою.

PAD дозволяє суттєво знизити навантаження на інженерів SRE/DevOps і підвищити швидкість виявлення проблем у складних розподілених середовищах.

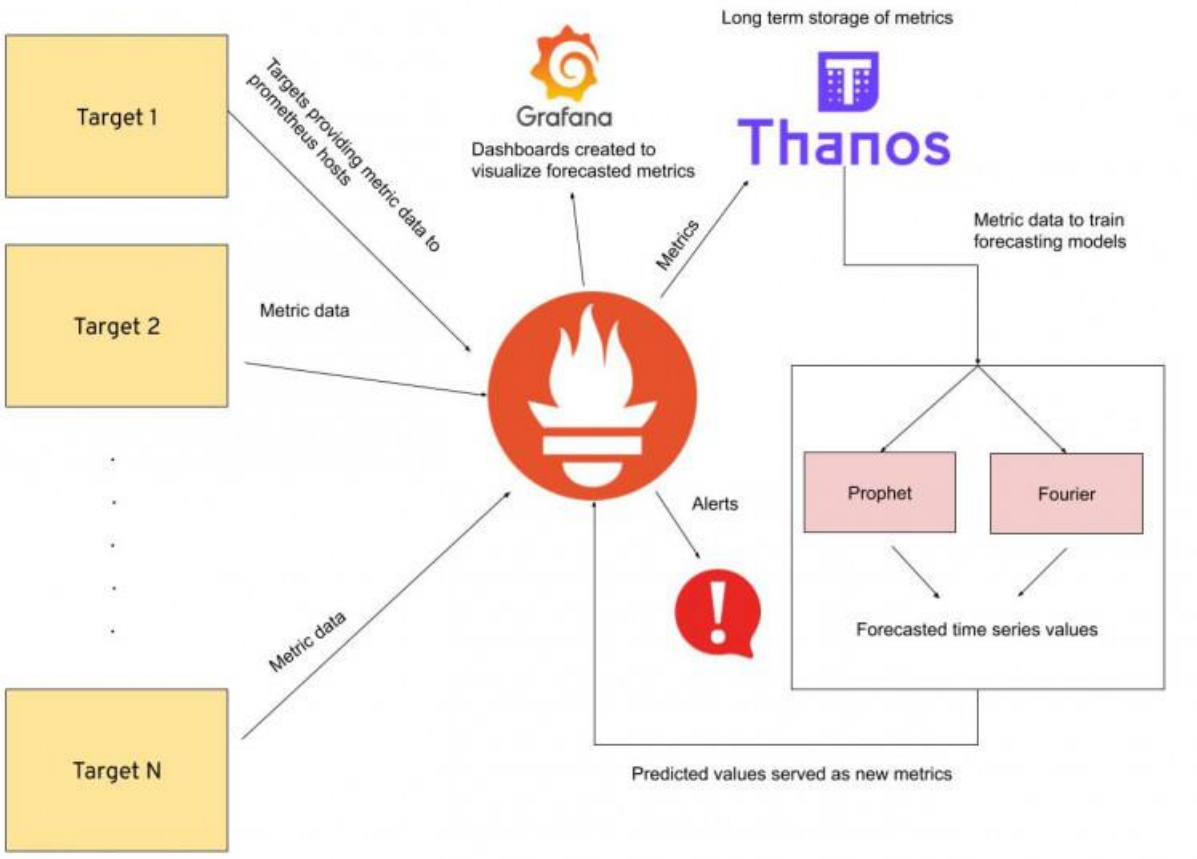


Рис. 3.1 Prometheus Anomaly Detector (PAD)

PAD створить дашборди в Grafana та алерти для оповіщення у разі виявленої аномалії.

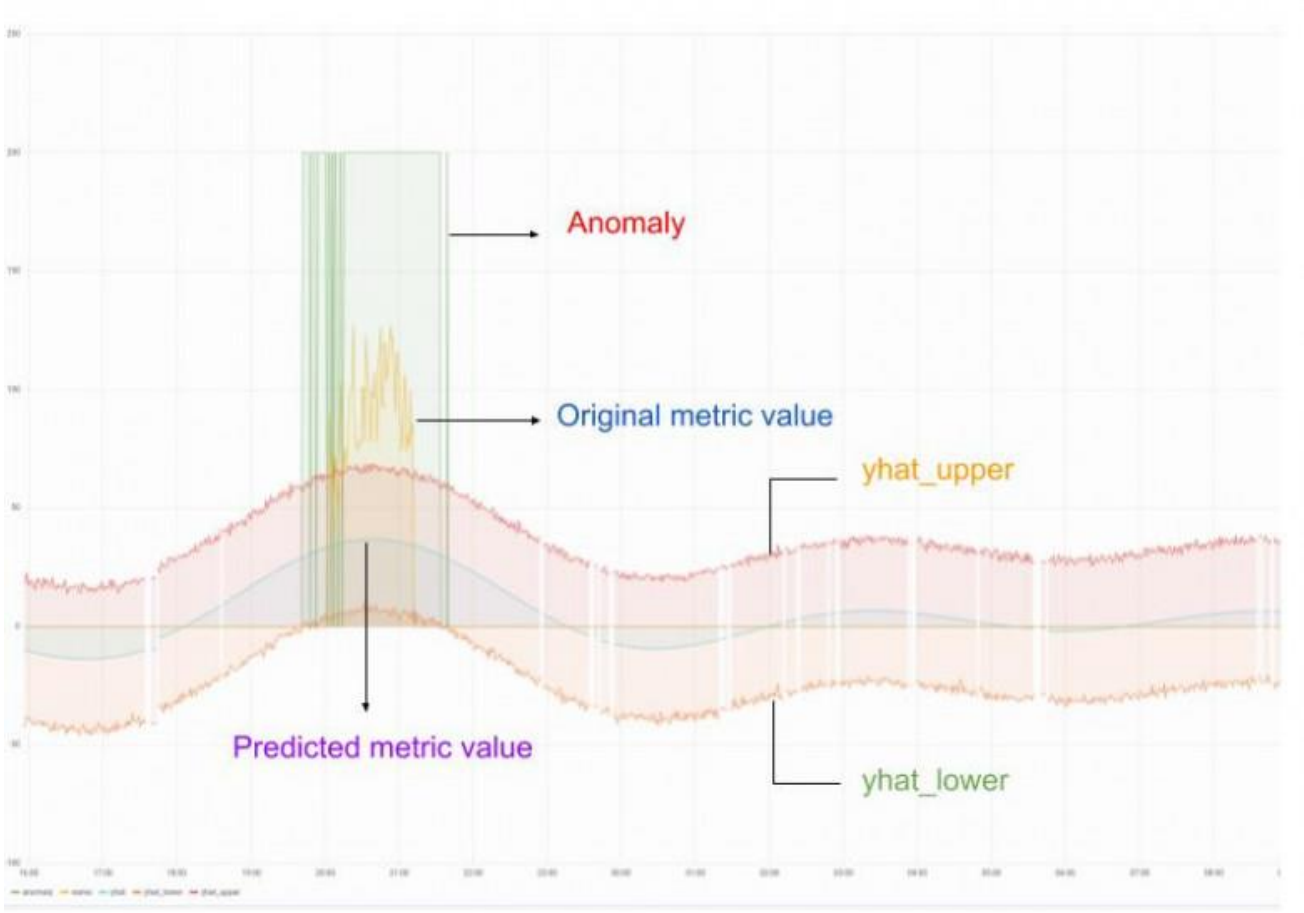


Рис. 3.2 Дашборди та алерти створені PAD

3.3 Виявлення аномалій як частина системи APM

Системи моніторингу продуктивності (Application Performance Monitoring, APM) дедалі частіше інтегрують у себе компоненти класу AIOps — інтелектуальні інструменти, що використовують машинне навчання для аналізу поведінки системи та автоматичного виявлення аномалій. Це природний розвиток APM-платформ, адже вони вже володіють величезними масивами телеметричних даних: метриками, логами, трасуваннями, подіями. Логічно, що наступним кроком стає здатність не просто збирати інформацію, а й робити з неї висновки.

Розгляньмо, як саме сучасні APM-системи реалізують функції автоматизованого виявлення аномалій, і наскільки ці механізми зручні з точки зору налаштування та подальшої експлуатації. Зокрема, нас цікавить:

- чи є підтримка «розумного» аналізу поведінки сервісів у режимі реального часу;
- наскільки автоматичною є адаптація до нових метрик і нових сервісів у мікросервісній архітектурі;
- чи може система сама визначати аномалію без ручного встановлення порогів;
- як відбувається робота з сезонністю та довгостроковими трендами;
- чи надаються рекомендації або пріоритизація алертів через машинне навчання.

Сучасні платформи APM прагнуть зменшити навантаження на інженерів SRE/DevOps, автоматизувавши те, що раніше вимагало великих зусиль: налаштування порогів, створення складних правил, ручне відслідковування аномалій у десятках тисяч метрик. Саме тому функціонал AIOps — один із найдинамічніших напрямів розвитку інструментів моніторингу.

Таким чином, зростання складності інфраструктури та збільшення кількості метрик зробили автоматизоване виявлення аномалій невід’ємною складовою сучасних APM-рішень, а ступінь автоматизації цього процесу — ключовим фактором ефективності системи моніторингу.

Dynatrace

У Dynatrace механізми виявлення аномалій уже вбудовані «з коробки» — тобто після встановлення система одразу пропонує набір готових правил та відповідний алертинг. Це дозволяє дуже швидко розпочати моніторинг без необхідності ручного налаштування кожного окремого параметра.

Dynatrace автоматично формує сигнали про виявлені відхилення, наприклад про зниження трафіку, що видно на наведеному прикладі системного оповіщення.

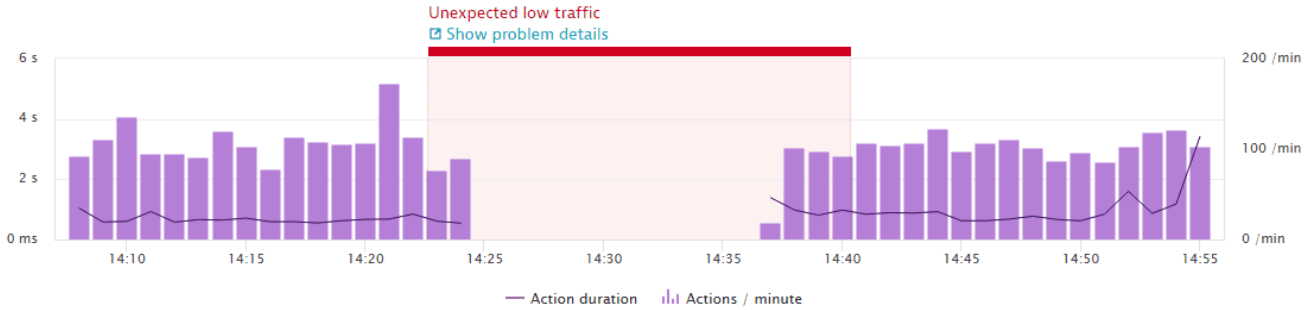


Рис. 3.3 Системне оповіщення Dynatrace

Серед доступних правил виявлення аномалій передбачені такі можливості:

- виявлення деградації ключових показників продуктивності (KPI) — сигналізує про проблеми в роботі сервісу, коли певні метрики погіршуються порівняно зі звичайними значеннями;
- виявлення падіння трафіку — корисно для ситуацій, коли сервіс перестає отримувати очікувану кількість запитів;
- контроль збільшення часу виконання транзакцій — дає змогу оперативно помітити подовження часу обробки запитів.

Основні параметри, варіанти налаштування та конфігурація правил аномалій відображені на скріншоті, де можна обрати:

- пороги чутливості,
- типи метрик, що аналізуються,
- умови запуску алертів,
- сценарії реагування.

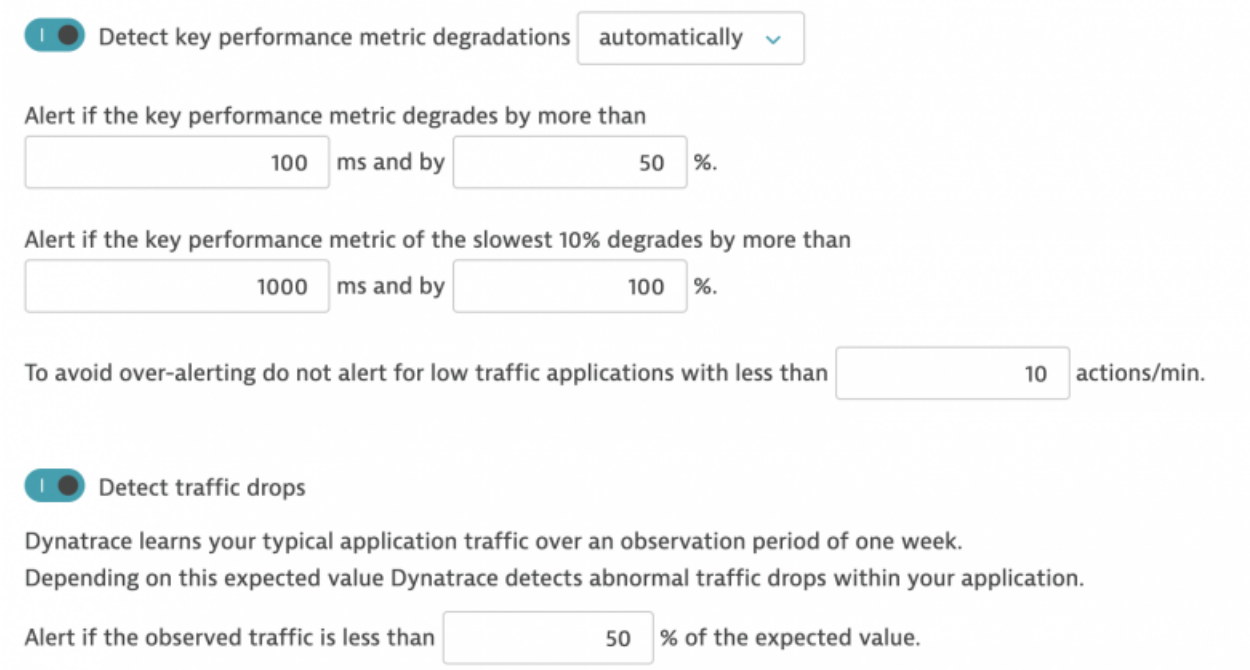


Рис. 3.4 Основні параметри, варіанти налаштування та конфігурація

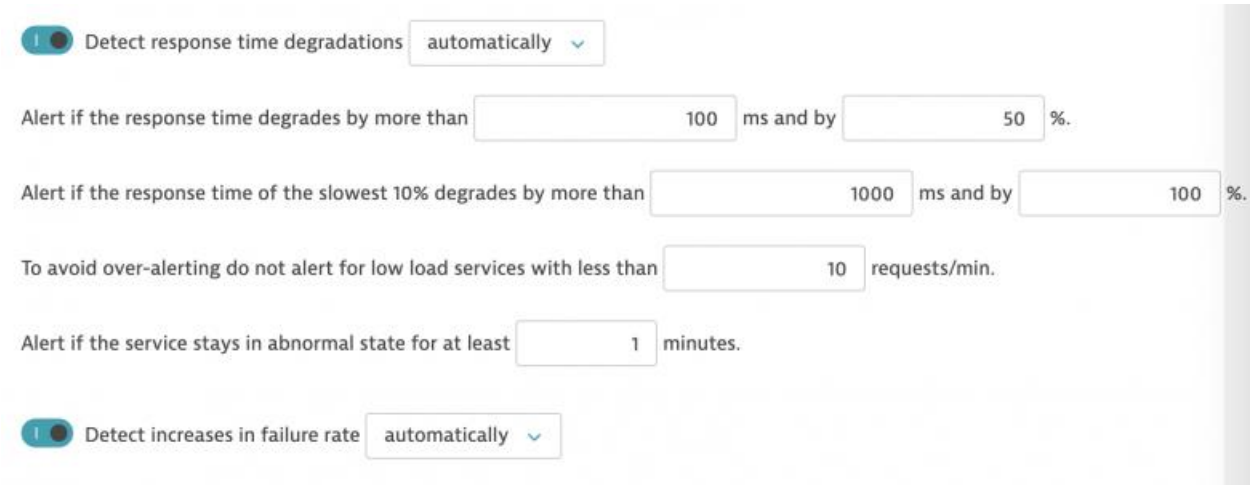


Рис. 3.5 Основні параметри, варіанти налаштування та конфігурація

Завдяки такому рівню автоматизації Dynatrace значно спрощує підтримку великих розподілених систем і дозволяє швидко виявляти проблеми навіть у середовищах із тисячами метрик та сотнями сервісів.

Azure Metric Advisor

Одним із сервісів екосистеми Microsoft є Azure Metric Advisor — потужний інструмент, призначений для виявлення аномалій у великих потоках метрик і даних. Сервіс підтримує широкий спектр джерел даних та може працювати з різними форматами і типами інформації.

З огляду на офіційну документацію, основний акцент платформи зроблено на застосуваннях у сфері e-commerce — там, де важливо відстежувати зміни в поведінці користувачів, показниках продажів, навантаженні на сервіси, конверсіях та інших бізнес-метриках, чутливих до аномалій.

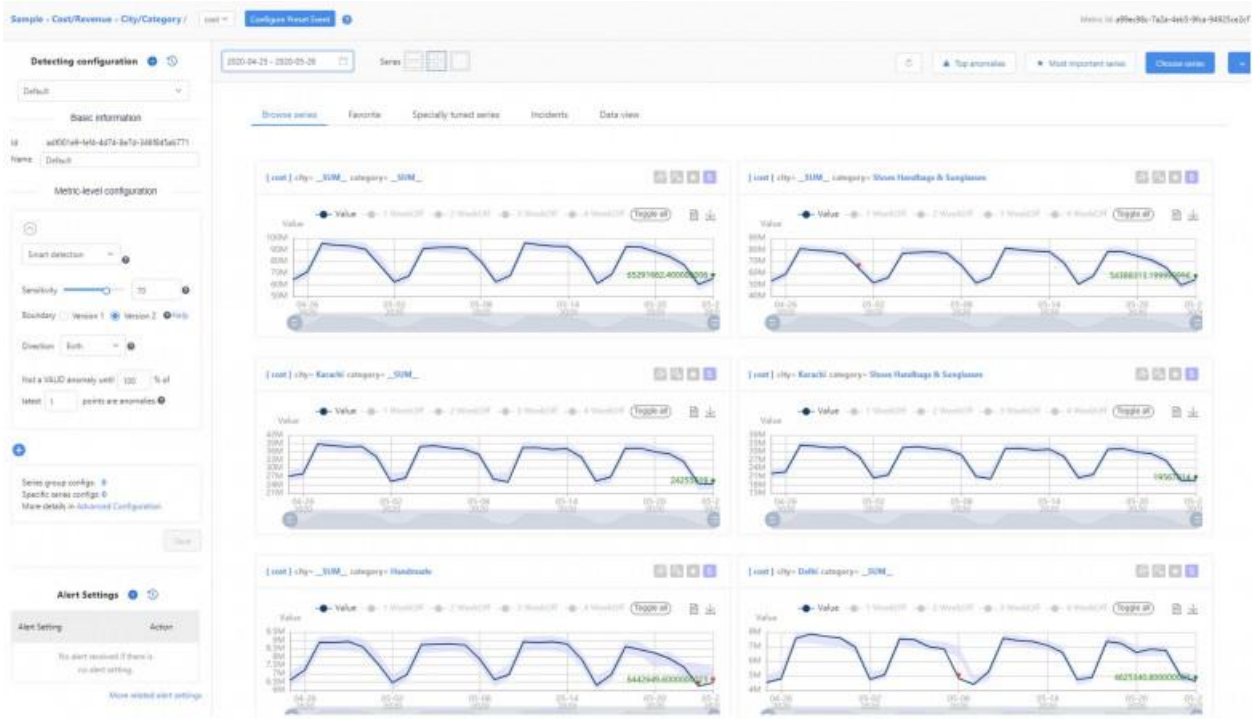


Рис. 3.6 Функціонал Azure Metric Advisor

Azure Metric Advisor підтримує інтеграцію з багатьма популярними сховищами та джерелами даних, серед яких:

- SQL Server,
- ElasticSearch,
- InfluxDB,
- MongoDB,
- MySQL,
- PostgreSQL,
- а також низка інших сервісів і баз даних.

При цьому варто зазначити, що підтримка Prometheus як джерела метрик відсутня. Це може стати обмеженням для команд, що використовують Prometheus як основний інструмент моніторингу у середовищах Kubernetes або мікросервісних архітектурах.

Загалом Azure Metric Advisor — гнучке рішення для автоматичного пошуку аномалій у бізнес-метриках і телеметрії, однак у технічних сценаріях інфраструктурного моніторингу він може вимагати додаткових шарів інтеграції.

4 АНОМАЛІЇ В МОНІТОРИНГУ БЕЗПЕКИ МЕРЕЖІ

4.1 Основні джерела даних для моніторингу

Коли мова заходить про моніторинг безпеки внутрішньої корпоративної чи відомчої мережі, у багатьох одразу виникає асоціація з контролем витоку даних та використанням DLP-рішень. Якщо ж уточнити питання й запитати, як саме виявляються атаки всередині мережі, найчастіше у відповідь можна почути лише згадку про системи виявлення вторгнень (IDS). І хоча ще 10–20 років тому саме IDS були практично єдиним інструментом для таких задач, сьогодні вони вже не покривають усіх загроз і вважаються застарілими як єдиний метод моніторингу.

У сучасних умовах набагато ефективнішим, а в деяких випадках навіть єдино можливим способом відстеження підозрілої активності та виявлення аномалій у мережевому трафіку є використання flow-протоколів. Спочатку ці протоколи створювалися суто для діагностики мережевих проблем (troubleshooting), але з часом перетворилися на потужний інструмент безпеки. Саме аналіз flow-даних дозволяє бачити нетипову поведінку систем, виявляти аномалії у взаємодії вузлів, фіксувати ознаки атак ще на ранніх стадіях та робити це з мінімальними витратами ресурсів у порівнянні з повним перехопленням трафіку.

Питання «Навіщо потрібен моніторинг безпеки внутрішньої інфраструктури?» зазвичай вже не потребує додаткових пояснень. В умовах сучасних загроз без цього просто неможливо забезпечити стійкість корпоративних ТК-систем. Щоб ще раз переконатися в цьому, достатньо переглянути коротке відео про те, якими різноманітними способами зловмисники можуть проникнути в мережу, захищену навіть сучасним міжмережевим екраном. Отже, вважаємо очевидним: внутрішній моніторинг необхідний, і залишилося тільки зрозуміти, як його правильно побудувати та як ефективно виявляти аномалії в поведінці вузлів мережі.

Для комплексного контролю інфраструктури на мережевому рівні можна виділити три ключові джерела даних:

1. Сирий мережевий трафік, який перехоплюється і передається на аналіз спеціалізованим системам.
2. Події та журнали з мережевого обладнання, через яке проходить трафік (маршрутизатори, комутатори, firewall).
3. Flow-дані, тобто агрегована інформація про з’єднання, отримана за допомогою одного з flow-протоколів.

Саме третій варіант — flow-дані — сьогодні вважається одним із найоптимальніших для виявлення аномалій, підозрілих шаблонів та ознак атак. Flow-моніторинг дозволяє відстежувати поведінку мережі у динаміці, знаходити нетипові відхилення (аномалії) у обсягах трафіку, кількості з’єднань, напрямках комунікації між вузлами та інших характеристиках. Потужні алгоритми аналізу — зокрема машинне навчання — значно підсилюють цей підхід, дозволяючи автоматично розпізнавати аномальні патерни навіть у великих і складних мережах.

Захоплення «сирого» мережевого трафіку — це найбільш поширений і традиційний підхід у сфері кібербезпеки. Він з’явився першим і багато років залишався основним інструментом для аналізу мережевих атак. Перші комерційні системи виявлення вторгнень, такі як NetRanger від Wheel Group (компанію згодом у 1998 році придбала Cisco), працювали саме так: вони перехоплювали окремі пакети або цілі сесії та шукали в них сигнатури — характерні шаблони, що вказують на атаку (те, що в термінології ФСТЕК називають «вирішальними правилами»).

Зрозуміло, аналізувати необроблений трафік можна не лише за допомогою IDS. Існують популярні IT-інструменти — Wireshark, tcpdump, аналітичний функціонал NBAR2 у Cisco IOS тощо. Проте їм, як правило, не вистачає тієї великої бази знань та глибини аналізу, якими володіють повноцінні системи безпеки, тому їх використання обмежене технічними задачами, а не виявленням складних атак.

IDS довгі роки були основою мережевої безпеки. Цей метод дійсно добре працює на периметрі — чи то межа корпоративної мережі, чи дата-центр, чи окремий сегмент. Але сучасні мережі — високошвидкісні, комутовані,

віртуалізовані та програмно-орієнтовані — створюють серйозні виклики для класичних сенсорів IDS.

Якщо мережа побудована на великій кількості комутаторів, то інфраструктура сенсорів стає просто гігантською. Для відстеження атак на конкретний сервер або робочу станцію доведеться встановити сенсор на кожен порт або канал, через який проходить трафік.

Будь-який виробник без вагань продасть сотні чи тисячі сенсорів, але такий підхід:

- вкрай дорогий,
- складний у впровадженні,
- вимагає значного обслуговування,
- не масштабується для великих мереж.

Навіть у Cisco, яка сама розробляє рішення NGIPS (Next-Generation Intrusion Prevention System), ми не змогли побудувати подібну інфраструктуру в повному обсязі — хоча, здавалося б, питання ціни для власних розробників не повинно бути критичним.

Технічні складнощі використання IDS-сенсорів

Виникає низка додаткових проблем:

- Як підключати сенсор?

  У розрив каналу (inline)? Це ризиковано: якщо сенсор виходить з ладу, падає і весь сервіс.
- Потрібен bypass-модуль?

  Це різко збільшує вартість рішення.
- Використовувати TAP-розгалужувачі?

  Це також ускладнює інфраструктуру і потребує додаткових витрат.

Усе це робить підхід «ставимо сенсори скрізь» малопридатним для сучасних великих компаній, незалежно від їх масштабу.

Як результат, хоч захоплення сирого трафіку й залишається корисним і важливим інструментом, воно вже давно перестало бути універсальним рішенням. Сучасні підходи до виявлення аномалій, поведінкового аналізу, аналізу flow-

трафіку та машинного навчання набагато ефективніше вирішують задачі безпеки у розподілених та високонавантажених мережах.

Розгляньмо тепер другий підхід — аналіз подій, що надходять від мережевих пристроїв. Він також може застосовуватися для виявлення атак, проте не може слугувати основним механізмом захисту, адже дозволяє фіксувати лише обмежений набір вторжень. До того ж цей метод має яскраво виражений реактивний характер: спочатку атака має статися, потім її має зареєструвати мережевий пристрій, і лише після цього він повідомить про інцидент засобами журналювання.

Таких засобів декілька — syslog, RMON та SNMP.
Два останні протоколи (RMON та SNMP) у контексті інформаційної безпеки доцільно використовувати переважно для виявлення атак типу DoS проти самих мережевих пристроїв, оскільки вони дозволяють контролювати завантаження процесора, стан інтерфейсів тощо. Наприклад, різкий стрибок завантаження CPU або відсотка використання порту може сигналізувати про спробу перевантаження пристрою.

Це один із найдешевших, адже доступний практично «з коробки» у будь-якому обладнанні, але водночас один із найменш ефективних методів моніторингу безпеки внутрішньої мережі. Значна частина атак просто не буде помічена, оскільки ці протоколи не дають повної картини того, що відбувається в трафіку.

Втім, відкидати його повністю не варто: аналіз syslog може допомогти вчасно помітити зміни конфігурації пристрою, ознаки його компрометації або інші критичні події. Але коли йдеться про виявлення атак у масштабах усієї мережі, цей варіант явно недостатній.

Третій підхід — це аналіз інформації про трафік, яку генерує пристрій, що підтримує один із flow-протоколів. Незалежно від того, який саме протокол застосовується (NetFlow, sFlow, IPFIX тощо), інфраструктура обробки потоків складається з трьох ключових компонентів:

1. Генерація або експорт потоків (flow export)

Це функція маршрутизатора, комутатора або іншого мережевого пристрою, який пропускає через себе трафік і виокремлює з нього ключові параметри. Потім ці параметри надсилаються на модуль збору.

У Cisco, наприклад, протокол NetFlow підтримують різні категорії обладнання: маршрутизатори, комутатори (включно з промисловими та віртуальними), безпроводові контролери, міжмережеві екрани й навіть сервери.

2. Збір потоків (flow collection)

Оскільки в сучасних мережах зазвичай працюють десятки або навіть сотні мережевих пристроїв, виникає потреба у спеціальному рішенні для їх консолідації. Цю функцію виконують колектори flow — системи, які отримують потоки, агрегують їх, нормалізують і передають далі для аналізу.

3. Аналіз потоків (flow analysis)

Саме на цьому етапі відбувається основна «інтелектуальна» робота. Аналізатор застосовує різні алгоритми до зібраних потоків, на основі яких робить висновки.

- Для ІТ-функцій аналіз може виявити перевантажені ділянки мережі, незбалансоване використання ресурсів або потребу в оптимізації.
- Для ІБ-функцій аналіз потоків дозволяє виявляти витоки даних, підозрілу активність, поширення шкідливого ПЗ, DoS/DDoS-атаки, аномальне мережеве сканування тощо.

Не варто вважати, що така трирівнева архітектура є надмірно складною чи важкою для впровадження. Насправді більшість інших рішень — за винятком хіба що найпростіших систем мережевого моніторингу, які працюють виключно через SNMP чи RMON, — функціонують за тією ж логікою.

У будь-якій системі моніторингу присутні три ключові компоненти:

1. **Джерело даних**, яке формує інформацію для аналізу

   Це може бути мережевий пристрій (маршрутизатор, комутатор, міжмережевий екран), або окремий сенсор, що збирає та структурує дані про трафік.

2. **Підсистема збору та консолідації подій**

Вона приймає інформацію від різних пристроїв, агрегує дані, перетворює у єдиний формат та передає їх на вищий рівень для подальшого аналізу. У невеликих мережах ця функція може бути поєднана з аналітичним модулем, але у більших інфраструктурах колектори зазвичай виділяють в окремі вузли для підвищення продуктивності й масштабованості.

3. **Аналітичний модуль або система керування моніторингом**

   Саме тут виконуються інтелектуальні операції: аналіз потоків, пошук аномалій, визначення потенційних атак, сповіщення адміністраторів. У великих розгортаннях такі системи працюють автономно, приймають рішення на основі машинного навчання й дозволяють автоматично реагувати на загрози.

У невеликих мережах ці елементи інколи об’єднуються в одному пристрої чи сервері. Проте в реальних корпоративних або відомчих інфраструктурах, де важливими є надійність, безперервність роботи та масштабованість, їх логічно розділяти між двома або навіть кількома окремими вузлами. Це дозволяє легко розширювати систему, обробляти великий обсяг потоків і забезпечувати стійкість до відмов.

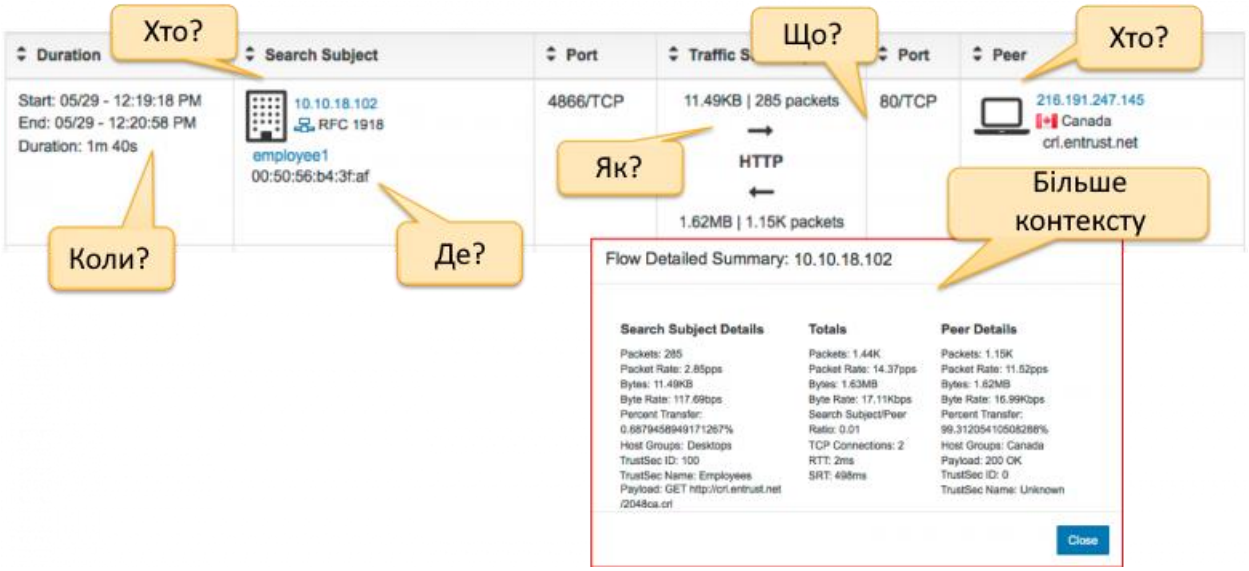


Рис. 4.1 Приклад flow-запису з розширеним контекстом у процесі аналізу мережевих потоків

На відміну від аналізу пакетів, який ґрунтується на детальному вивченні заголовків і вмісту кожного окремого пакета та сформованих із них сесій, аналіз

потоків працює з метаданими мережевого трафіку. Саме на питання *коли? скільки? звідки? куди? яким способом?* відповідають flow-протоколи, збираючи узагальнену телеметрію без необхідності аналізувати кожен байт трафіку.

Спершу такі протоколи застосовувалися переважно для статистичного аналізу мережі, пошуку перевантажень та усунення IT-проблем. З розвитком аналітичних можливостей з'явилася можливість використовувати ці самі метадані і для цілей кібербезпеки, зокрема для пошуку аномалій та ознак мережевих атак.

Важливо розуміти: *аналіз потоків не замінює повного захоплення пакетів.* Обидва методи мають свою чітку сферу застосування. Однак у контексті моніторингу внутрішньої інфраструктури, саме аналіз потоків часто є найефективнішим рішенням. Адже яким би не був шлях атаки, вона мусить пройти через мережеві пристрої — маршрутизатори, комутатори чи міжмережеві екрани. Усі ці пристрої здатні експортувати flow-дані. На відміну від класичного сенсора IDS, який зловмисник може обійти, мережеве обладнання обійти неможливо — атака обов'язково залишить слід у телеметрії.

З іншого боку, якщо вам потрібні докази для служби безпеки, аудиту чи правоохоронних органів, одного лише аналізу потоків недостатньо. Flow не містить вмісту пакетів і не є повним аналогом трафіку. Тут свою роль відіграє packet capture (PCAP) — саме пакети можуть бути використані як доказова база. Проте зберігати *увесь* трафік надзвичайно дорого.

Розглянемо приклад. Нехай мережа працює на швидкості 250 Мбіт/с. Тоді для повного запису трафіку потрібно:

- 31 МБ — за одну секунду,
- 1,8 ГБ — за хвилину,
- 108 ГБ — за годину,
- 2,6 ТБ — за добу.

Для мережі у 10 Гбіт/с добовий обсяг складе 108 ТБ. Якщо регулятор вимагає зберігати дані рік — такі обсяги стають практично неможливими. Натомість аналіз потоків дозволяє писати лише потрібний трафік, тобто лише те, що має ознаки аномалії чи атаки. У цьому випадку розмір даних зменшується на

порядок: співвідношення обсягу потоку до повного дампу становить приблизно 1:500. Наприклад, замість сотень терабайтів вам потрібно всього кілька гігабайтів.

4.2 Flow-протоколи та їхні відмінності

На відміну від систем packet capture, де підходи у різних вендорів майже однакові, у світі flow-протоколів існує низка важливих відмінностей, особливо актуальних у сфері кібербезпеки.

Найпоширенішим є NetFlow, створений Cisco. На сьогодні найбільш функціональною є версія NetFlow v9, на основі якої розроблений промисловий стандарт IPFIX (NetFlow v10). Більшість мережевих виробників підтримують саме ці два протоколи.

Існують і інші варіанти: sFlow, jFlow, cFlow, rFlow, NetStream тощо. Серед них найпопулярнішим є sFlow, який часто підтримують вітчизняні виробники завдяки простішій реалізації.

Ключові відмінності між NetFlow/IPFIX та sFlow:

1. Гнучкість полів

   NetFlow/IPFIX дозволяють змінювати набір полів експорту, тоді як sFlow працює з фіксованою структурою запису.

2. Тип телеметрії
   - *NetFlow/IPFIX — несемпльована телеметрія.* Фіксується кожен потік, що дає повну картину мережевої активності.
   - *sFlow — семпльована телеметрія.* Пристрій періодично відбирає лише окремі пакети, що дає статистичну, але не повну картину.

Це ключова різниця для задач кібербезпеки: семпліровані дані можуть пропустити атакуючу активність, тоді як NetFlow/IPFIX з високою деталізацією фіксують усі потоки.

Таблиця 4.1

Різниця між *NetFlow* та *sFlow*

Функція	**NetFlow**	**sFlow**
Захват пакетів	Не захоплює пакети взагалі	Копіює всі пакети та семплює 1 із N для відправлення на колектор
Підтримка протоколів	Рівень 2, IP та IPv6	Не залежить від мережевого рівня
Конфігуровані поля	Гнучкі: NetFlow має настраювані користувачем поля (шаблони)	Фіксовані поля протоколу
Записи потоків	Підтримка запису потоків IPv4, IPv6 для всього трафіку	Запис потоку не створюється — копіюються перші N байтів пакета
Апаратне прискорення	Так: записи потоків створюються «в залізі» без впливу на data plane	Апаратне прискорення можливе, але зазвичай пакети захоплюються ПЗ
Індустріальний стандарт	IPFIX	sFlow v5
Тимчасові мітки потоків (початок і кінець)	Так	Так
Packet rates (кількість пакетів у потоці)	Так	Ні
Підрахунок байтів (кількість байтів у потоці)	Так	Так (частково)

Наступне важливе питання, яке постає перед фахівцями з мережевої безпеки, звучить так: де саме слід впроваджувати підтримку flow-протоколів для ефективного виявлення аномалій та атак? Проте таке формулювання не зовсім

коректне. На більшості сучасного мережевого обладнання підтримка flow-телеметрії вже присутня «з коробки». Тому питання варто ставити по-іншому: де саме найефективніше збирати телеметрію з точки зору безпеки?

Найкраща відповідь — на рівні доступу, тобто там, де підключені кінцеві користувачі та сервери. Саме на цьому рівні:

- можна побачити 100% трафіку, який проходить між вузлами;
- є максимально детальна інформація про хости (MAC-адреси, VLAN, порти, ідентифікатори інтерфейсів);
- можна відстежувати P2P-трафік між пристроями, що критично для виявлення сканування мережі, бічних переміщень зловмисника та розповсюдження шкідливого ПЗ.

На рівні ядра частина трафіку може взагалі не потрапляти у зону видимості, а на периметрі ви спостерігатимете добре якщо 20–30% всього внутрішнього руху. А якщо у мережі з’являться сторонні пристрої, через які можлива прихована комунікація, периметрові засоби взагалі нічого не побачать. Саме тому оптимальний варіант — збір телеметрії на рівні доступу.

Варто додати, що навіть у середовищах з віртуалізацією чи контейнерами (KVM, VMware, OpenStack, Kubernetes) сучасні віртуальні комутатори часто підтримують flow-телеметрію, тому відстеження трафіку можливе й у цих середовищах.

Що робити, якщо обладнання не підтримує flow?

Бувають ситуації, коли:

- обладнання (фізичне або віртуальне) не підтримує flow-протоколи;
- їх увімкнення заборонено політиками (наприклад, у промислових сегментах для зниження ризиків);
- увімкнення flow призводить до надмірного навантаження на CPU, що характерно для застарілих пристроїв.

У таких випадках застосовують спеціалізовані віртуальні або фізичні flow-сенсори (flow sensor). По суті, це мережеві TAP-пристрої або «розгалужувачі», які

отримують копію трафіку, аналізують його й експортують flow-телеметрію до колектора.

Проте такий варіант має ті самі недоліки, які були згадані у випадку аналізу сирого трафіку:

- складність у розгортанні,
- додаткові витрати на обладнання,
- ризик появи «вузьких місць»,
- потреба у виділених каналах або SPAN-портів.

Тому, обираючи підхід до впровадження аналізу потоків, слід чітко розуміти не лише переваги, але й обмеження цієї технології, аби побудувати ефективну й масштабовану систему моніторингу безпеки.

Ще один важливий аспект, який варто враховувати під час роботи зі засобами аналізу потоків, — це специфіка вимірювання їх навантаження. Якщо для класичних систем генерації подій безпеки використовується метрика EPS (events per second — кількість подій за секунду), то для систем, що працюють із телеметрією, вона вже не підходить. У цьому випадку застосовується інша метрика — FPS (flows per second — кількість потоків за секунду).

Як і у випадку з EPS, заздалегідь точно визначити рівень FPS неможливо, адже він залежить від характеру роботи кожного конкретного пристрою та від навантаження, яке на нього припадає. Проте завжди можна розрахувати орієнтовні значення: у відкритих джерелах існують спеціальні таблиці з типовими параметрами FPS для різних класів корпоративних пристроїв і мережевих сегментів. Це дає змогу заздалегідь оцінити, які ліцензії знадобляться для інструментів аналізу та якою має бути архітектура системи моніторингу у конкретному випадку.

Важливо розуміти, що, так само як IDS-сенсор має обмежену пропускну здатність, виражену у максимальній кількості трафіку, який він здатен обробити, — так само і колектор потоків має власні обмеження. Саме тому у великих, географічно розподілених корпоративних мережах, як правило, використовується декілька колекторів. Наприклад, у внутрішній мережі Cisco розгорнуто 21 колектор,

що обслуговують інфраструктуру, розподілену на п’яти континентах та налічує близько пів мільйона активних пристроїв.

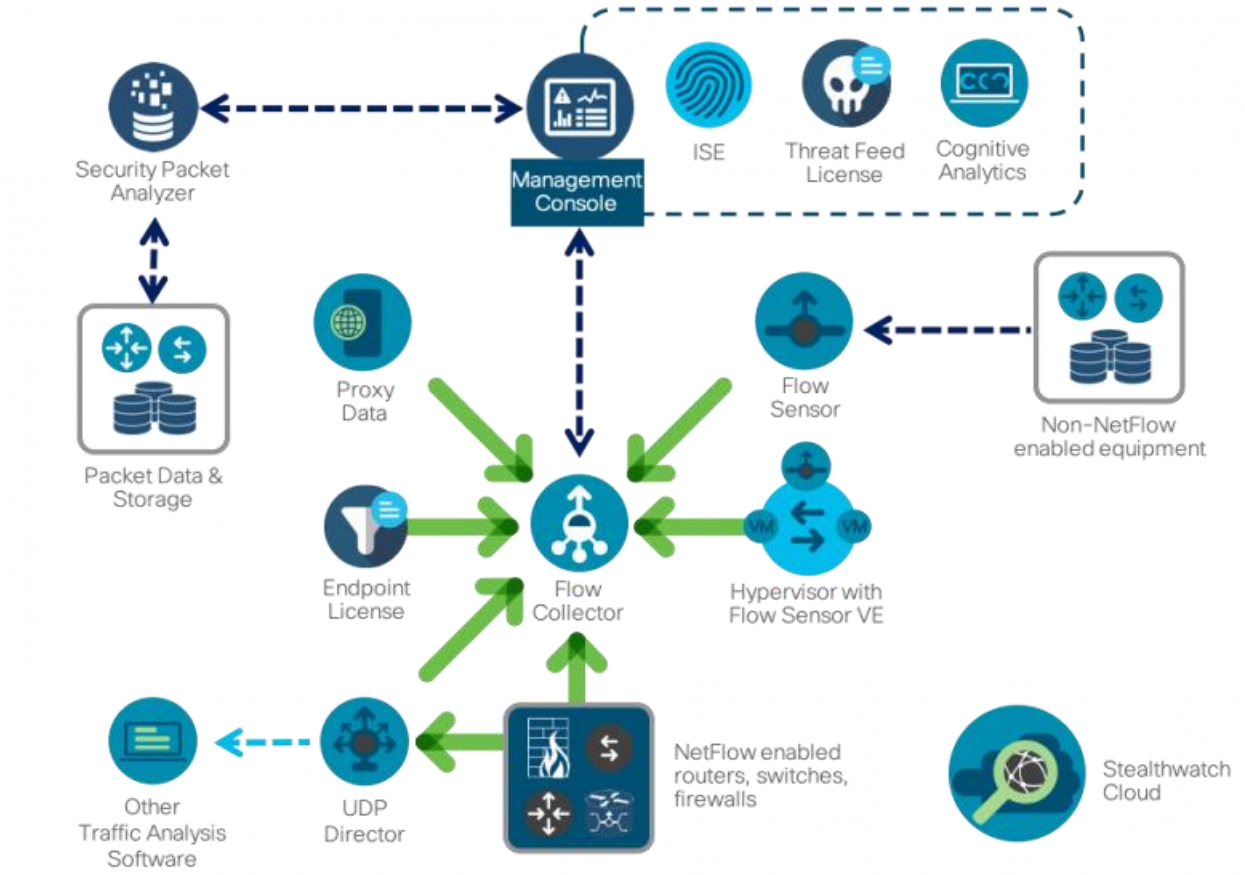


Рис. 4.2 Ситема моніторингу Netflow на базі обладнання Cisco

Для аналізу NetFlow у Cisco використовується власне рішення Cisco Stealthwatch, спеціально створене для вирішення задач кібербезпеки. Платформа оснащена численними вбудованими механізмами виявлення аномалій, підозрілої активності та явних загроз. Вона здатна розпізнавати широкий спектр інцидентів — від криптомайнінгу та витоків даних до поширення шкідливого програмного забезпечення чи шахрайських дій у мережі.

Stealthwatch реалізовано за класичною трирівневою архітектурою:

1. **Генератор потоків** — мережеве обладнання або інші пристрої, що експортують дані у форматі flow.
2. **Колектор потоків** — модуль, який приймає, нормалізує та агрегує інформацію з усіх джерел.

3. **Аналізатор потоків** — інтелектуальний модуль, що застосовує аналітичні алгоритми до зібраних потоків, виявляє підозрілі поведінкові моделі, формує попередження та аналітичні звіти.

Однак у Stealthwatch ця архітектура істотно розширена низкою додаткових можливостей, що суттєво підвищують ефективність виявлення атак.

- Інтеграція із засобами захоплення трафіку

  Наприклад, із Cisco Security Packet Analyzer, що дозволяє на вимогу зберігати окремі мережеві сесії для поглибленого аналізу інцидентів.

- Підтримка спеціального протоколу nvzFlow

  Це унікальний механізм, який дає змогу передавати телеметрію про активність додатків безпосередньо з кінцевих вузлів — серверів, робочих станцій та інших пристроїв. Такий рівень деталізації дозволяє Stealthwatch бачити більше загроз, ніж традиційні мережеві аналізатори потоків, які працюють виключно з інформацією від мережевого обладнання.

Таким чином, завдяки поєднанню аналізу потоків, можливості запису трафіку «на вимогу» та додаткових телеметричних джерел, Stealthwatch перетворюється на комплексну платформу поведінкового аналізу, здатну виявляти складні та малопомітні атаки в мережі.

Цілком очевидно, що ринок систем аналізу NetFlow з позиції кібербезпеки далеко не обмежується єдиним продуктом Cisco. Доступні як комерційні рішення, так і безкоштовні чи умовно безкоштовні інструменти. Було б дивно у блозі Cisco почати детально описувати продукти конкурентів, але варто звернути увагу на два популярні засоби аналізу телеметрії, які часто плутають через схожість назв, хоча їхня природа суттєво відрізняється — SiLK та ELK.

SiLK — гнучкий, але складний інструмент командного рядка

SiLK (System for Internet-Level Knowledge) — це набір утиліт, створений CERT/CC для глибокого аналізу мережевої телеметрії. Він підтримує NetFlow (версії 5 і 9), IPFIX та sFlow і дозволяє за допомогою численних інструментів (rwfilter, rwcount, rwflowpack тощо) виконувати фільтрацію, агрегування та пошук підозрілої активності в потоці даних.

Проте існує кілька важливих нюансів:

- **SiLK — це CLI-інструмент**, і робота з ним потребує нескінченних команд. Наприклад, пошук ICMP-пакетів розміром понад 200 байт може виглядати так:

  ```
  rwfilter --flowtypes=all/all --proto=1 --bytes-per-packet=200- --pass=stdout \
  | rwcut --fields=sIP,dIP,iType,iCode --num-recs=15
  ```

- Існує GUI-надбудова **iSiLK**, але вона значно не спрощує аналіз — лише допомагає візуалізувати дані.
- **SiLK не містить готових аналітичних моделей** — весь процес пошуку аномалій, побудови гіпотез, написання скриптів та логіки лежить на аналітикові.

  Це не є недоліком — радше особливість інструменту, який орієнтований на досвідчених спеціалістів.

Як виглядає робочий цикл аналітика з SiLK?

1. **Формулювання гіпотези.**

   Необхідно чітко розуміти, який тип аномалій чи атак ми шукаємо, та знати, за якими атрибутами можливо їх виявити.

2. **Побудова моделі.**

   Аналітик самостійно програмує логіку аналізу за допомогою Python, Shell чи інших засобів.

3. **Тестування.**

   Готова гіпотеза перевіряється за допомогою інструментів SiLK (rw*, set*, bag*).

4. **Робота з реальними даними.**

   Аналітик аналізує результати:

   - Чи знайдено очікувані патерни?
   - Чи відповідає знайдене початковій гіпотезі?
   - Як зменшити кількість хибних спрацювань?
   - Як покращити точність?

5. **Оптимізація та покращення.**

Гіпотеза уточнюється, код оптимізується, створюються шаблони для прискорення роботи.

Порівняння з Cisco Stealthwatch

У Cisco Stealthwatch цей цикл максимально автоматизований. Аналітик отримує готові механізми:

- автоматичні алгоритми виявлення аномалій;
- внутрішні бази знань;
- кореляцію з зовнішніми списками шкідливих IP (в SiLK це потрібно робити вручну);
- сценарії (workflow) розслідування інцидентів.

Наприклад, той самий пошук взаємодій з небезпечними IP:

- в SiLK треба писати окремий скрипт;
- у Stealthwatch попередження з’являється автоматично.

Таким чином, SiLK — це потужний, але “ручний” інструмент для експертів, тоді як Stealthwatch забезпечує високий рівень автоматизації, швидку реакцію на інциденти та масштабованість для мереж будь-якого розміру.

Якщо підніматися вище по вартості програмних засобів для аналізу потокової телеметрії, то після повністю безкоштовного SiLK наступним рівнем є умовно безкоштовний стек ELK, який складається з трьох основних компонентів:

- **Elasticsearch** — система для індексації, пошуку та аналітичної обробки великих обсягів даних;
- **Logstash** — інструмент для збирання, перетворення та перенаправлення даних;
- **Kibana** — платформа для візуалізації, створення графіків, панелей моніторингу та аналітичних дашбордів.

На відміну від SiLK, де практично всі операції потрібно реалізовувати вручну — писати фільтри, сценарії та аналітичні пайплайни, — екосистема ELK уже має великий набір готових модулів, плагінів та розширень. Частина з них є безкоштовними, інші доступні за передплатою, але в будь-якому випадку вони

суттєво спрощують роботу з даними та пришвидшують розробку складних аналітичних рішень.

Наприклад, модуль GeoIP у Logstash автоматично визначає географічне розташування IP-адрес, що надходять у потоці телеметрії, і додає цю інформацію до подальшого аналізу. Це дає можливість відразу бачити трафік, що походить із підозрілих регіонів або країн. У спеціалізованих рішеннях на кшталт Cisco Stealthwatch така функція вже інтегрована «з коробки», але ELK дозволяє досягти подібного результату за рахунок додаткової конфігурації.

ELK надає набагато більшу гнучкість, ніж класичні засоби обробки flow-телеметрії. Він дозволяє:

- створювати власні аналітичні сценарії без глибокого програмування;
- індексувати мільйони записів потоків і проводити пошук за будь-якими параметрами;
- будувати деталізовані дашборди для оперативного моніторингу;
- застосовувати машинне навчання (у більш розширеній комерційній версії X-Pack) для виявлення аномалій.

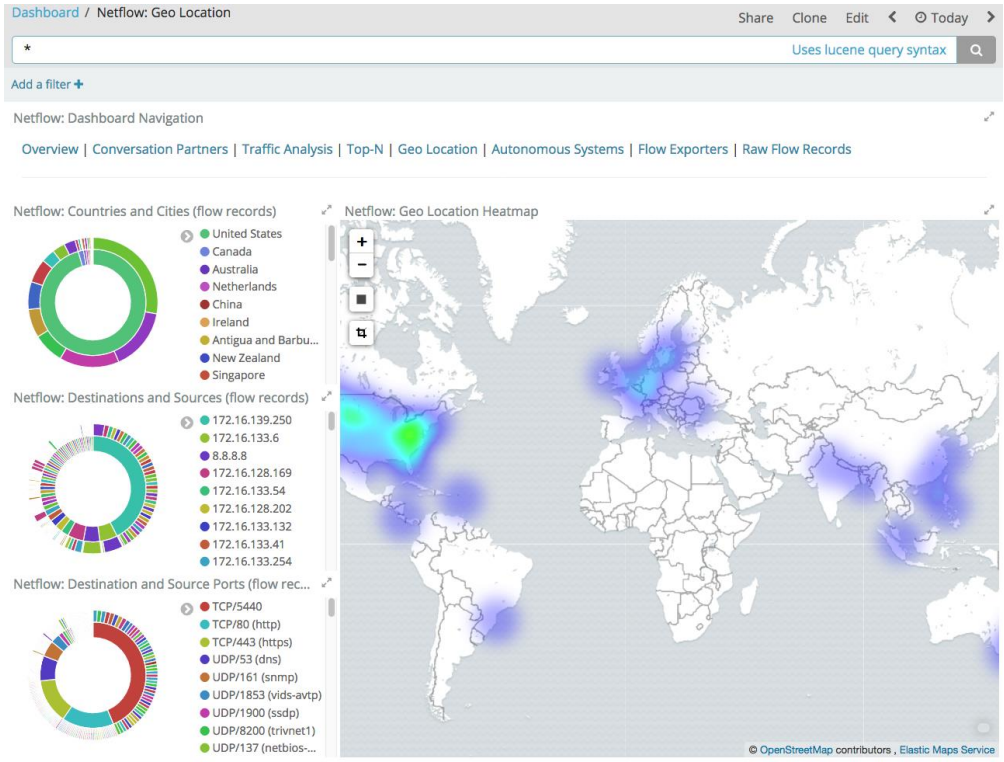


Рис. 4.3 Фільтр GeoIP в Logstash дозволяє прив'язати спостерігаються IP-адреси до їх географічного розташування

Таким чином, ELK виступає як універсальна платформа, що може бути адаптована під дуже різні сценарії — від базового збору та візуалізації telco-телеметрії до складного поведінкового аналізу у корпоративних мережах. Він займає проміжне місце між повністю ручними інструментами рівня SiLK та потужними комерційними продуктами з вбудованою аналітикою, надаючи оптимальний баланс можливостей та вартості.

Підсумовуючи викладений матеріал, варто виділити основні рекомендації, яких доцільно дотримуватися при побудові якісного моніторингу інформаційної безпеки внутрішньої інфраструктури:

1. Не обмежуйтесь лише периметром мережі.

Сучасні корпоративні інфраструктури потребують багаторівневого підходу. Використовуйте мережеве обладнання не тільки як транспортний механізм для передачі даних, а й як інструмент кіберзахисту. Сучасні комутатори та маршрутизатори містять вбудовані засоби телеметрії, які здатні виявляти значну частину небезпечної активності.

2. Вивчіть функціонал вашого мережевого обладнання та задійте наявні інструменти безпеки.

Багато систем уже мають вбудовані механізми моніторингу, налаштування яких може значно підвищити рівень захищеності мережі без додаткових витрат.

3. Для внутрішнього моніторингу обирайте аналіз мережевої телеметрії.

Аналіз потоків (flow) дозволяє виявити 80–90% інцидентів інформаційної безпеки, включно з аномальною поведінкою вузлів, зловмисною активністю та ознаками компрометації. Flow-фреймворк охоплює повний спектр внутрішнього трафіку та працює навіть там, де неможливо підключити сенсори IDS.

4. Для аналізу потоків використовуйте Netflow v9 або IPFIX.

Ці протоколи забезпечують найширше охоплення телеметрії, включно з IPv4, IPv6, MPLS тощо. Вони містять детальні метадані, необхідні для точного аналізу поведінки в мережі.

5. Перевагу варто надавати несемпльованим flow-протоколам.

Несемпльований Netflow/IPFIX дає повну картину трафіку, що критично для точного виявлення кіберзагроз. Семпльований протокол містить прогалини в даних і може пропустити частину атаки.

6. Перевірте продуктивність мережевого обладнання.

Активувавши експорт потоків, не всі пристрої здатні витримати додаткове навантаження. Якщо продуктивність падає, доцільно розглянути використання:

- віртуальних сенсорів,
- Netflow Generation Appliance,
- або оновлення обладнання.

7. Впроваджуйте збір телеметрії на рівні доступу.

Саме тут видно 100% трафіку, включно з East-West взаємодіями між хостами, що дозволяє виявляти:

- сканування,
- внутрішнє поширення шкідливого ПЗ,
- P2P-комунікації,
- незвичну активність у VLAN-сегментах.

Це найважливіший рівень для моніторингу безпеки.

8. Якщо ви змушені використовувати російське мережеве обладнання (у контексті вимог або обмежень), — обирайте лише ті моделі, які підтримують:

- flow-протоколи,
- або SPAN/RSPAN-порти для дублювання трафіку.

9. Комбінуйте різні системи безпеки.

Найкращий результат дає поєднання:

- IDS/IPS-систем на периметрі,
- аналізу потоків у внутрішній мережі,
- моніторингу віртуальної інфраструктури та хмарних платформ.

Така гібридна архітектура забезпечує найповніше охоплення загроз та значно знижує час виявлення (MTTD).

Також, підсумовуючи викладене, варто звернути увагу на один важливий висновок. На наведеній ілюстрації (рисунок нижче) добре видно, що якщо раніше

служба інформаційної безпеки Cisco будувала свою систему моніторингу майже повністю на основі засобів виявлення вторгнень та сигнатурних механізмів, то сьогодні їхня частка становить лише близько 20% усіх зафіксованих інцидентів.

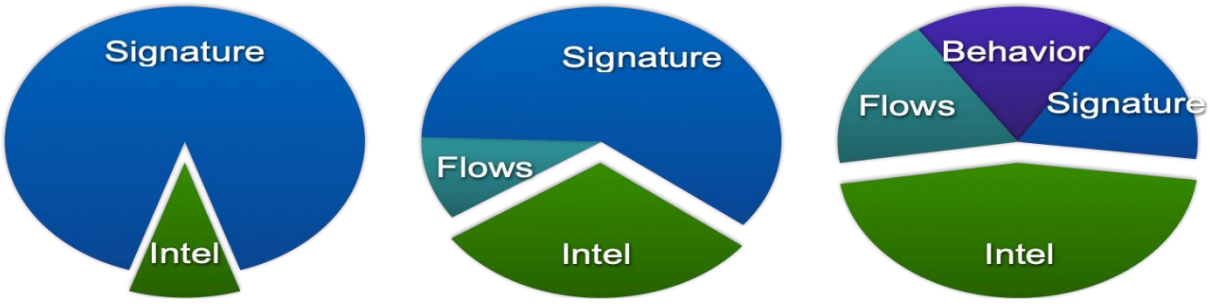


Рис. 4.4 Система моніторингу Cisco

Ще приблизно 20% — це результат роботи систем аналізу мережевих потоків. Така статистика демонструє, що аналіз потоків — це не модна тенденція і не експериментальний інструмент, а повноцінний та ефективний механізм забезпечення безпеки для сучасних підприємств. Він дозволяє виявляти ті події, які класичні IDS можуть пропустити, особливо у складних, розподілених і динамічних інфраструктурах.

Більш того, більшість підприємств уже мають необхідну основу для впровадження цього підходу — мережеву інфраструктуру, у яку й так інвестуються значні кошти. Тому логічно підсилити захист цієї інфраструктури й одночасно підвищити її цінність, поклавши на неї також функції моніторингу поведінки користувачів, трафіку та телеметрії.

Хоча в даній роботі не було розглянуто деталі реагування на виявлені аномалії чи загрози, важливо пам’ятати: моніторинг не повинен завершуватися лише фіксацією інциденту. За виявленням підозрілої активності обов’язково мають слідувати дії реагування — бажано не лише ручні, але й автоматизовані. Адже саме швидкість та точність реакції визначають здатність підприємства мінімізувати збитки, зменшити час простою та запобігти подальшому поширенню атаки.

Тема автоматизованої відповіді на загрози (SOAR-підходи, автоматичне блокування, інтеграція з мережевою інфраструктурою) заслуговує окремого глибокого розгляду та може стати основою для подальших робіт.

ВИСНОВКИ

У ході дослідження було розроблено та проаналізовано підходи до побудови системи моніторингу та виявлення аномалій у мережевому трафіку, що можуть використовуватися для підвищення рівня інформаційної безпеки корпоративних мереж. Значний обсяг теоретичного матеріалу, а також практичні приклади дозволили сформувати цілісне бачення процесів обробки мережевих потоків, побудови характеристичних векторів вузлів та застосування методів машинного навчання для аналізу поведінки мережевих елементів.

Перш за все було розглянуто концепцію представлення трафіку у вигляді потоків (flows), що містять ключові метадані про взаємодію між вузлами. На відміну від класичного аналізу сирих пакетів, підхід, заснований на потоках, виявився більш придатним для моніторингу внутрішньої інфраструктури, оскільки дає змогу охопити більшу частину мережі, включно з сегментами, де використання IDS є обмеженим або економічно недоцільним. Завдяки цьому поточні системи телеметрії можуть виявляти до 80–90% аномальних активностей, що підтверджує ефективність методу.

У роботі було детально розглянуто принципи формування характеристичних векторів вузлів, що описують частоти використання протоколів, роль вузла, операційну систему, зв’язки з іншими класами пристроїв та інші поведінкові параметри. Саме така багатовимірна модель дозволяє застосовувати як методи класифікації, так і методи кластеризації для виявлення аномалій. Наведені приклади показали, що зміна трафіку одного вузла впливає на профілі інших елементів, а отже, система має враховувати динамічні зміни у мережі.

Окрему увагу було приділено сучасним підходам до виявлення аномалій у часових рядах, зокрема методам reconstruction-based, які демонструють відмінні результати для складних, нелінійних сигналів. Одним із найефективніших підходів є модель TadGAN, що поєднує властивості автокодировувачів та генеративно-змагальних мереж. Під час аналізу показано, що reconstructive-моделі здатні якісно

виявляти як точкові, так і групові аномалії, включно зі зміною форми сигналу чи частоти.

Також розглянуто значення сезонності та впливу зовнішніх факторів на варіації параметрів мережевого трафіку. Важливу роль відіграє правильна обробка часових рядів, зокрема врахування періодичності, циклічності та трендів. Як демонструють сучасні системи на кшталт Prometheus Anomaly Detector або APM-рішень (Dynatrace, Azure Metric Advisor), автоматизоване прогнозування та порівняння з очікуваними значеннями значно полегшує процес моніторингу в складних мікросервісних інфраструктурах.

Проведений аналіз підтвердив, що комплексне поєднання потокового моніторингу, аналітики часових рядів та методів машинного навчання дає можливість своєчасно виявляти аномалії у мережевому середовищі, мінімізуючи ризики інцидентів безпеки. Застосування таких підходів забезпечує високу масштабованість, зручність інтеграції та здатність адаптуватися до змін у мережевій структурі.

Отже, розроблена концептуальна модель моніторингу та виявлення аномалій у мережевому трафіку є актуальною, ефективною та може бути основою для побудови повноцінної системи ІБ, орієнтованої на сучасні виклики. У перспективі її можна доповнювати автоматизацією реагування на інциденти, інтеграцією з SIEM/SOAR-платформами та адаптивними механізмами прогнозування, що забезпечить ще вищий рівень захисту інфраструктури.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ahmed M., Mahmood A., Hu J. A survey of network anomaly detection techniques // Journal of Network and Computer Applications. – 2016.
2. Zhou P., Wang L., Chen X., et al. A survey of streaming data anomaly detection in network traffic // PeerJ Computer Science. – 2025.
3. Geiger A., Liu S., et al. TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks // Proc. IEEE Big Data. – 2020.
4. Silva I.O., Souza R., Martins D. Time Series Anomaly Detection Using TadGAN with Meta-Level Representations // Lecture Notes in Computer Science. – 2024.
5. Satheesh N., Rathnamma B. Flow-based anomaly intrusion detection using machine learning model with SDN for OpenFlow networks // Computer Communications. – 2020.
6. Li Z., Liu Y., Zhang Q., et al. FlowGANAnomaly: Flow-Based Anomaly Network Intrusion Detection Using GAN // Chinese Journal of Electronics. – 2022/2024.
7. Mutambik I., Alharbi F., Almutairi S., et al. An Efficient Flow-Based Anomaly Detection System for IoT Network Intrusion Detection (IoT-FIDS) // Sensors. – 2024.
8. Sattar S., Gupta R., Singh P., et al. Anomaly detection in encrypted network traffic using self-supervised learning // Scientific Reports. – 2025.
9. Gajin S., Savić S., Dimitrijević S. Network traffic anomaly detection and analysis based on the entropy of flow data // CEUR Workshop Proc. – 2023.
10. Agrawal S., Agrawal J. Survey on anomaly detection using data mining techniques // Procedia Computer Science. – 2015.
11. Гайдур Г.І., Романенко М.О. Виявлення аномалій трафіку в інформаційних системах // Телекомунікаційні та інформаційні технології. – 2021.
12. Петляк Н., Білецький К., Заставна Ю. Підхід до виявлення аномального мережевого трафіку з використанням алгоритмів LOF та HBOS // Measuring and Computing Devices… – 2024.
13. Савченко Т. Аналіз ефективності виявлення аномалій мережевого трафіку з використанням цифрового двійника // Cybersecurity. – 2025.

14. Лагун І.І., Лагун О.І. Розуміння природи аномалій трафіку у мережі // Вісник НУ «Львівська політехніка». – 2011.
15. Морозов Б. Дослідження методів аналізу мережевого трафіку // Матеріали конференції. – 2016.
16. Хархонов А.В. Методи аналізу мережевого трафіку для виявлення вторгнень : магістерська дисертація. – КПІ, 2020.
17. Келеберда А.М. Використання малохвильового перетворення для виявлення аномалій мережевого трафіку : магістерська робота. – КПІ, 2021.
18. Мальченко Г. Розробка системи моніторингу мережевого трафіку : бакалаврська робота. – ДУІКТ, 2023/2024.
19. Яворський Г., Флуд Л. Система моніторингу мережевого трафіку у Kubernetes-кластері // Вісник ХНУ. – 2025.
20. Шовкошитний І., Гаврилюк О., та ін. Підхід до захисту ІТС із застосуванням адресно-часової селекції для виявлення аномалій // Сучасні інформаційні технології… – 2023.
21. Бондаренко К.О. Математичні моделі та методи виявлення аномалій та вторгнень : монографія. – НТУ «ХПІ». – 2024.
22. Львівська політехніка.Інтелектуальна система моніторингу та аналізу трафіку для виявлення атак у сучасних мережах // ICtEE Journal. – 2022.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ

КВАЛІФІКАЦІЙНА РОБОТА

РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ТА ВИЯВЛЕННЯ АНОМАЛІЙ У МЕРЕЖЕВОМУ ТРАФІКУ

ВИКОНАВ:
СТУДЕНТ ГРУПИ ТСДМ-61
АНДРІЙ ЛИСАКІВСЬКИЙ

МЕТА РОБОТИ ТА АКТУАЛЬНІСТЬ

- Мета роботи – розробити інтелектуальний підхід до моніторингу та виявлення аномалій у мережевому трафіку на основі flow-телеметрії та машинного навчання, що забезпечує своєчасне виявлення відхилень і підвищення кіберстійкості телекомунікаційних систем..
- Об’єкт дослідження – мережевий трафік телекомунікаційних систем корпоративного, хмарного та промислового типу у форматах flow-телеметрії (NetFlow, IPFIX, sFlow).
- Предмет дослідження – методи формування характеристичних векторів мережевих вузлів та застосування алгоритмів машинного навчання (класифікації, кластеризації, реконструкції) для виявлення аномалій у поведінці мережі.
- **Актуальність.**
- Стрімке ускладнення телекомунікаційних мереж, зростання обсягу трафіку та використання мікросервісних, хмарних і промислових інфраструктур потребують більш інтелектуальних методів моніторингу. Традиційні IDS/IPS та статичні пороги не здатні вчасно розпізнавати складні або приховані аномалії. Flow-телеметрія (NetFlow, IPFIX, sFlow) у поєднанні з методами машинного навчання та кластеризацією характеристичних векторів дозволяє пасивно формувати поведінкові профілі вузлів і виявляти відхилення. Розробка таких рішень є актуальною для підвищення кіберстійкості, зниження MTTD та захисту критичної інфраструктури.

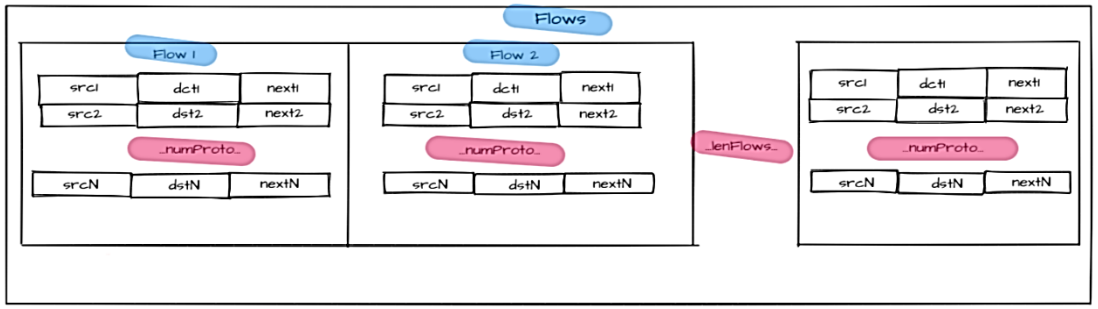
Характеристики мережевих вузлів

У системах активне сканування (Nmap тощо) є небезпечним — може викликати збої роботи ПЛК чи контролерів. Тому застосовується **пасивний підхід**: спостереження за трафіком без втручання.

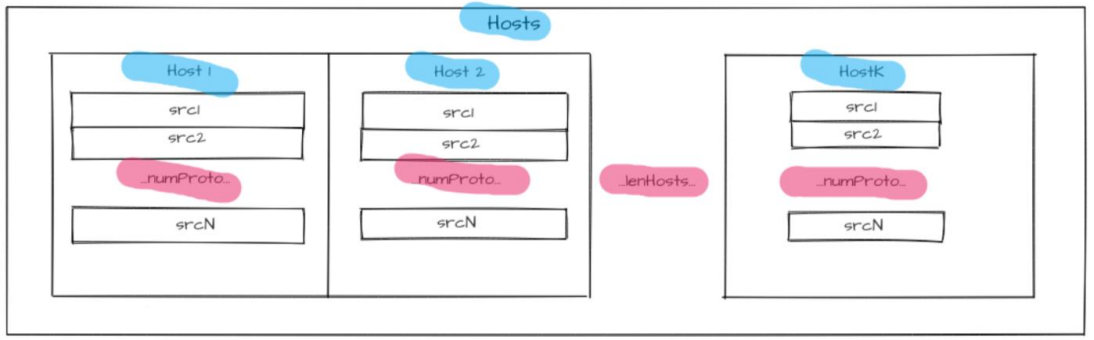
Увесь трафік групується в **потоки Flows**, а для кожного вузла формується **характеристичний вектор host.vector**, що описує частоти протоколів, типи взаємодій та особливості поведінки. Це його **цифровий профіль**, який складно підробити.

Далі:

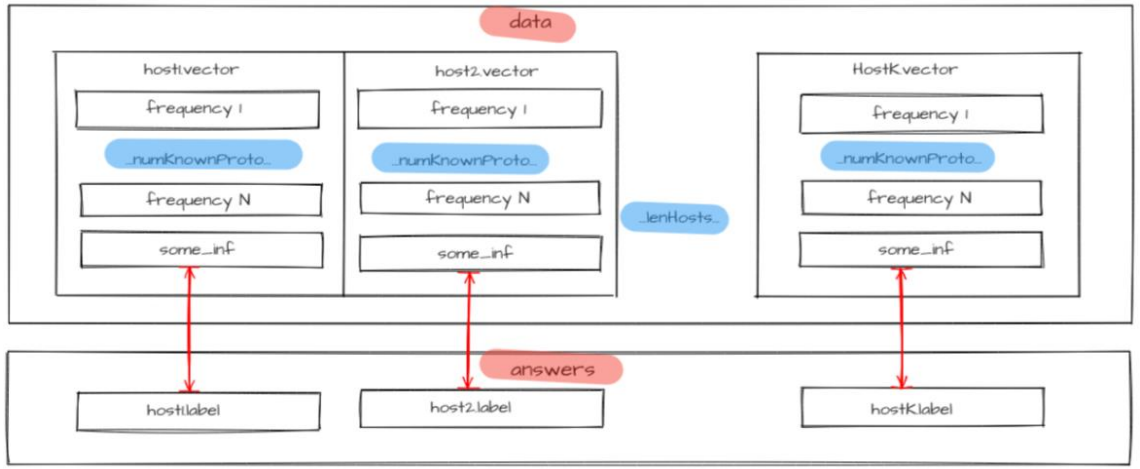
- за відомими класами формують **тренувальну вибірку**;
- **AutoML** підбирає оптимальну модель;
- після навчання система розпізнає типи вузлів та формує **host.labelPredicted**;
- якість оцінюється за **точністю, повнотою, F1-мірою**.



Група потоків



Ідентифікатори вузлів



Дані для навчання моделі

Виявлення аномалій в мережевому трафіку

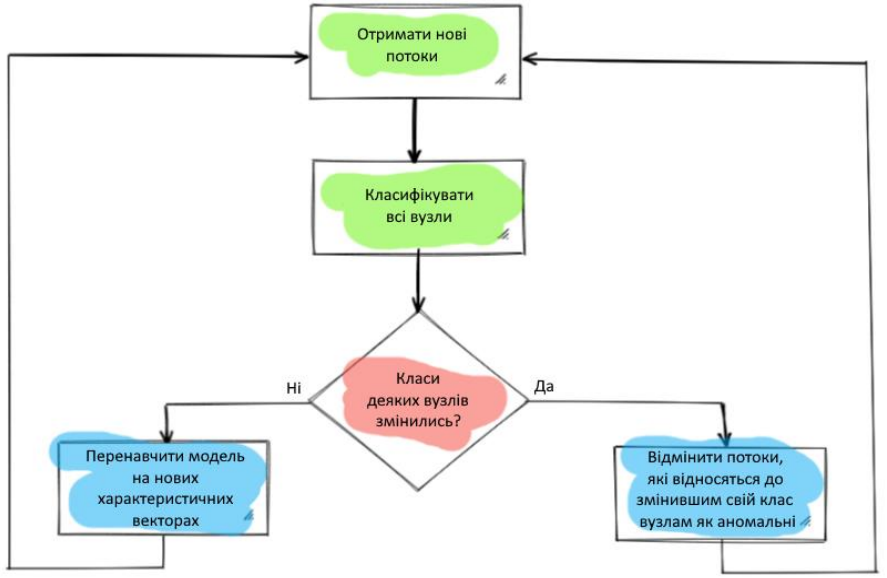
Для пошуку аномалій із потоків Flows формуються характеристичні вектори host.vector, які подаються в модель кластеризації. Система автоматично групує вузли та присвоює їм клас host.label. Якщо після появи нових потоків модель кластеризує вузол по-іншому(host_label ≠ host_new_label) — фіксується аномалія.

Потоки, що змінили поведінку вузла, позначаються як ініціатори аномалії, а рівень критичності визначається за відстанню між кластерами в ієрархічній структурі.

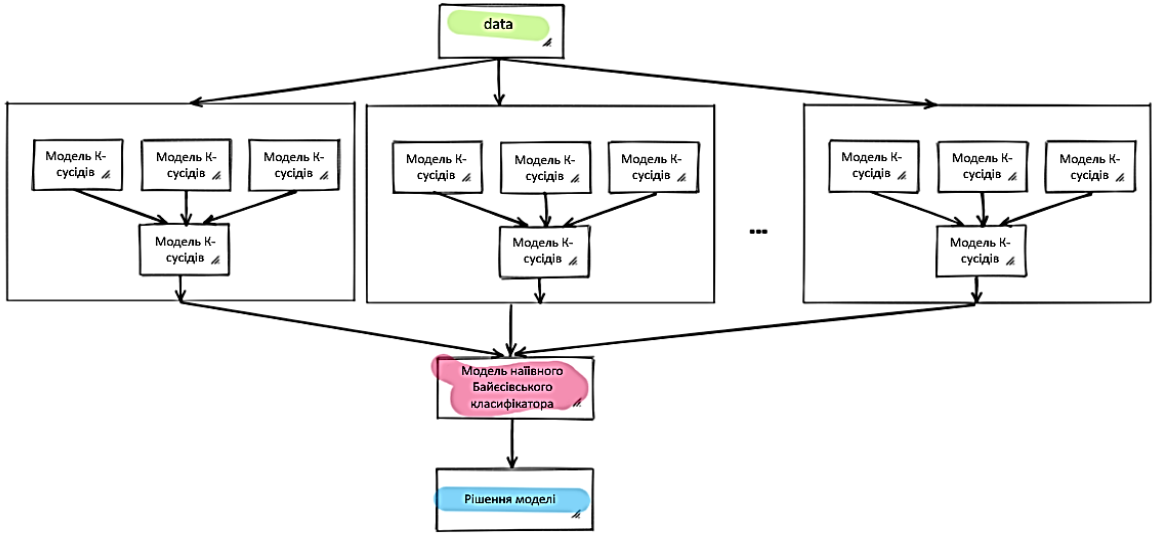
Якщо поведінка не змінюється(host_label = host_new_label) — модель проходить оновлене навчання, адаптуючись до нового стану мережі без сигналів тривоги.

Такий підхід дозволяє:

- автоматично виявляти аномалії без попередньої розмітки;
- оцінювати їх критичність;
- підтримувати актуальність профілів вузлів у динамічному середовищі.



Виявлення аномальних потоків



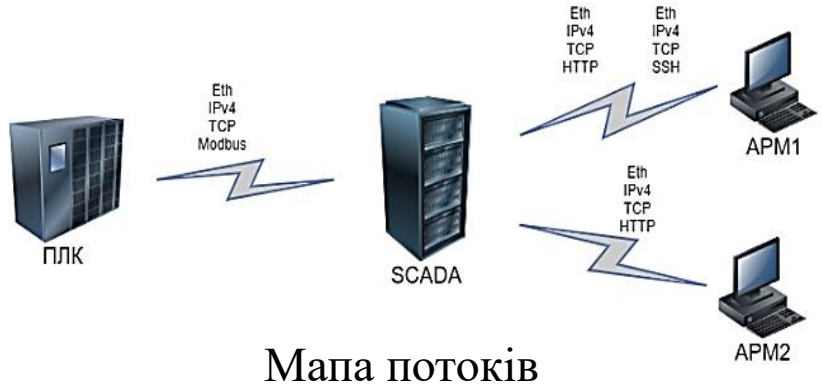
Підсумкова модель

Методи реалізації

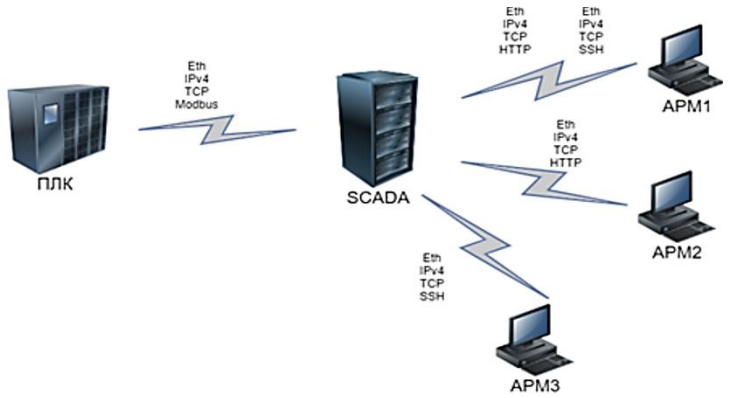
Сирий трафік оброблявся модулем **DPI**, після чого Python-скрипт формував **характеристичні вектори вузлів**. Для вибору моделі застосовано автоML-інструмент **TPOT**, який автоматично підібрав оптимальну архітектуру: **ансамблевий стекінг k-NN + наївний байєсівський класифікатор**.

Модель показала **F1 ≈80%**, а при пропозиції **двох можливих класів — 95–98%**, що свідчить про практичну ефективність.

У демонстраційному прикладі мережі з PLC, SCADA та АРМ протоколи Modbus, HTTP та SSH використовувалися для формування ознак. Модель змогла правильно класифікувати навіть новий вузол і адаптуватися до змін трафіку.



Мапа потоків



Оновлена мапа потоків

Точність роботи класифікатора

Тип класу	Точність	Повнота	F1-міра
Ролі вузлів (АРМ, ПЛК, Сервер, …)	0.81	0.80	0.81
Операційна система (WinXP, Win8, LinuxCentOS, …)	0.86	0.80	0.78

Частоти використання протоколів вузлами

Признак \ Вузол	ПЛК	SCADA	АРМ №1	АРМ №2
L2 – Ethernet	1	1	1	1
L3 – IPv4	1	1	1	1
L4 – TCP	1	1	1	1
L5 – Modbus	1	0.25	0	0
L5 – HTTP	0	0.5	0.5	1
L5 – SSH	0	0.25	0.5	0

Частоти використання протоколів вузлами після появи нового вузла

Признак \ Вузол	ПЛК	SCADA	АРМ №1	АРМ №2	АРМ №3
L2 – Ethernet	1	1	1	1	1
L3 – IPv4	1	1	1	1	1
L4 – TCP	1	1	1	1	1
L5 – Modbus	1	0.2	0	0	0
L5 – HTTP	0	0.4	0.5	1	0
L5 – SSH	0	0.4	0.5	0	1

Моніторинг аномалій в технологічних системах

Часові ряди та виявлення аномалій

Більшість технічних процесів можна описати **часовими рядами** — послідовністю значень у часі. Це дає змогу аналізувати тенденції, моделювати поведінку системи та виявляти відхилення. Часові ряди містять тренд, цикл, сезонність і можуть мати аномалії.

Типи аномалій:

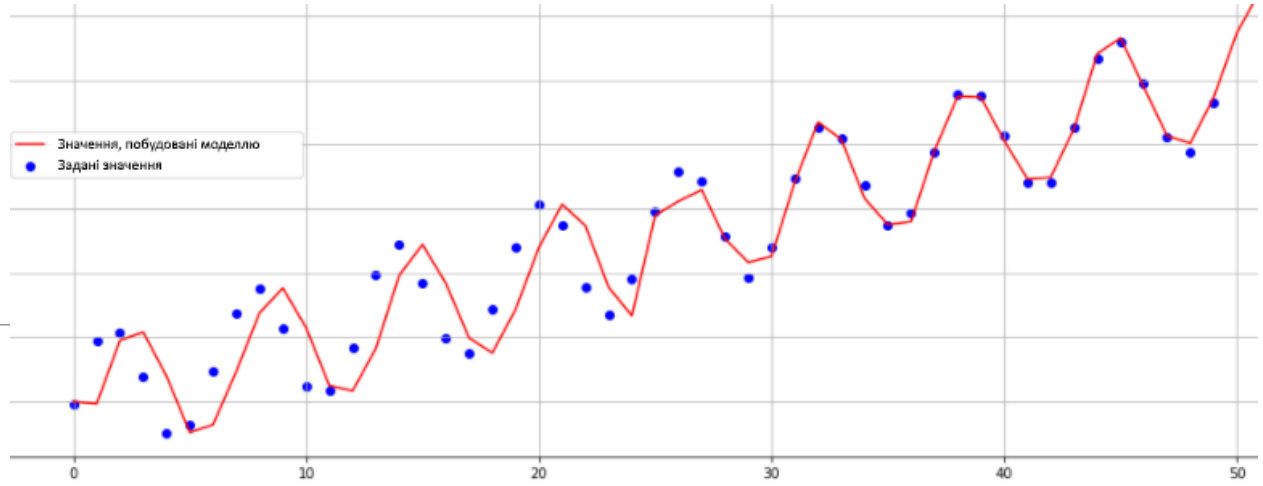
- **точкові** — різкий стрибок значення;
- **групові** — нетипова форма сигналу;
- **контекстні** — значення «не в свій час».

Статистичні методи виявляють лише прості випадки. Для складних даних застосовують **машинне навчання**, особливо **unsupervised learning**, бо реальні дані часто не мають розмітки.

Ефективні підходи:

- **Proximity-based** — порогові перевірки, кластеризація;
- **Prediction-based** — порівняння прогнозу з реальністю;
- **Reconstruction-based** — відтворення сигналу та пошук відхилень (основа TadGAN).

Остання група — найефективніша. Вона лежить в основі автокодерів, PCA, GAN та архітектури TadGAN, що одночасно реконструє і генерує сигнали, чутливо реагуючи на відхилення.



Моделювання часового ряду

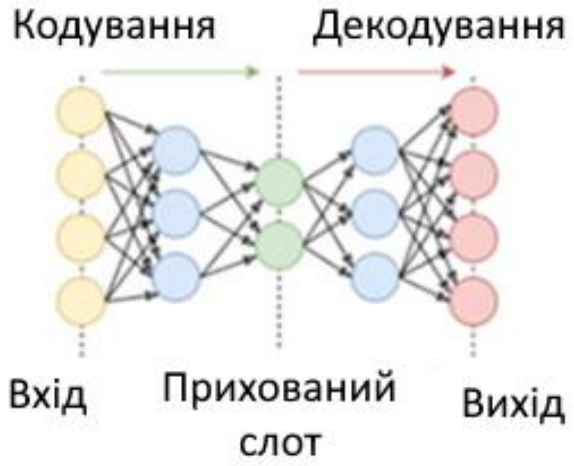
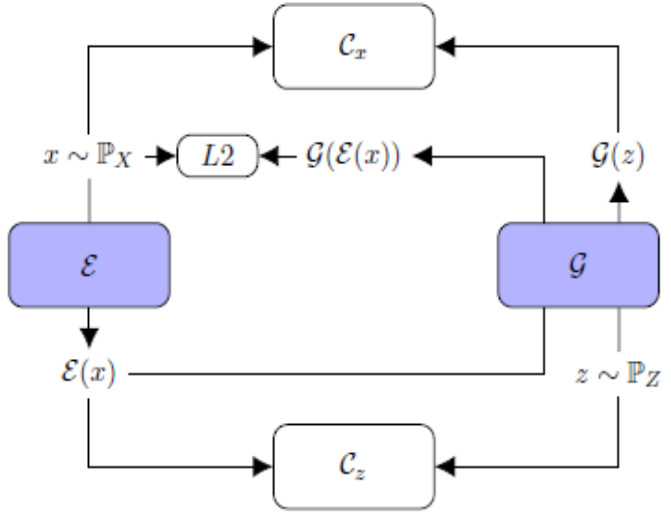


Схема роботи автокодувальника



Архітектура TadGAN

Нейронні мережі для виявлення аномалій

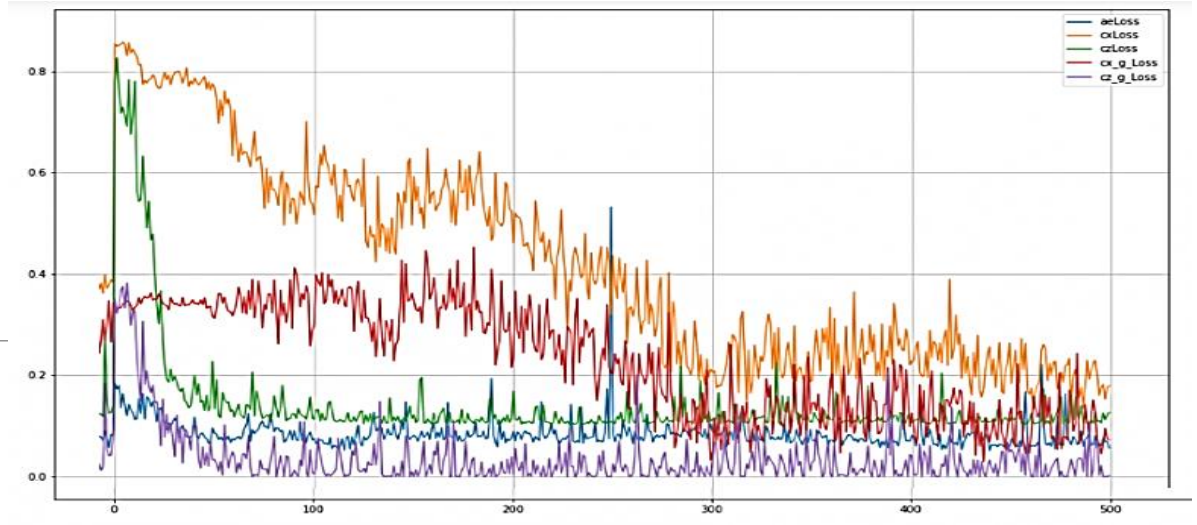
Сучасні фреймворки дають змогу швидко створювати та навчати нейромережі без реалізації низькорівневих обчислень. Архітектура TadGAN доступна у відкритих репозиторіях і навчається за п’ятьма основними метриками:

- **aeLoss** — точність реконструкції сигналу;
- **cxLoss / cx_g_Loss** — якість розрізнення реальних та згенерованих даних;
- **czLoss / cz_g_Loss** — узагальнення прихованого простору без «запам’ятовування» історії.

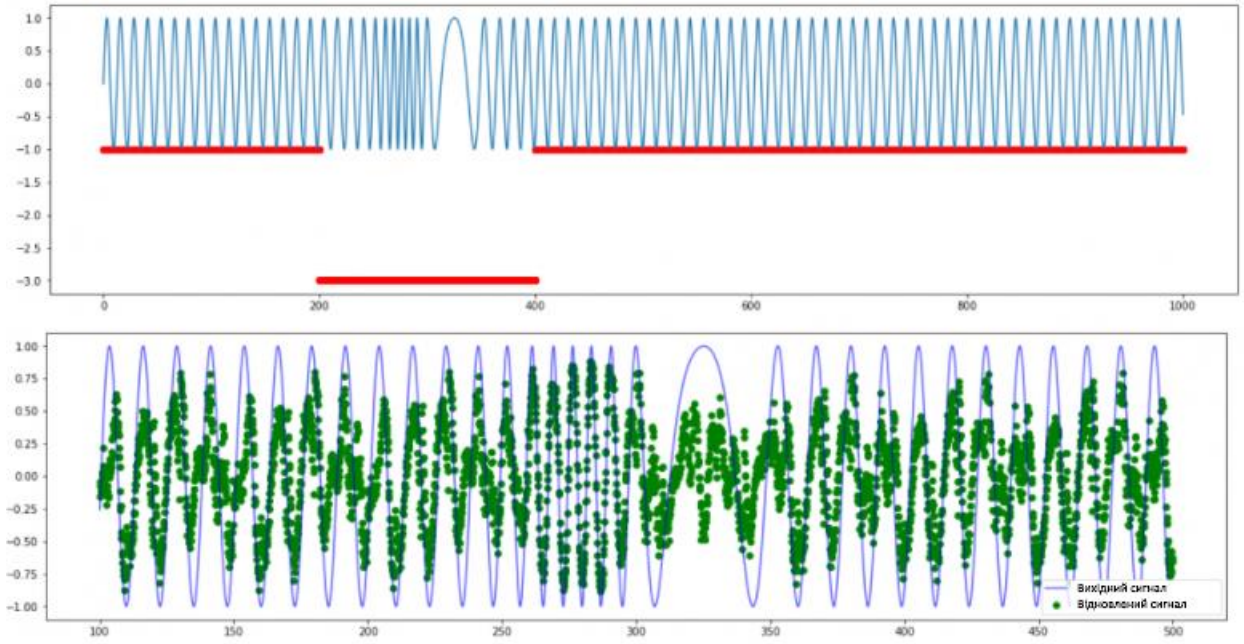
Після навчання модель реконструює часовий ряд і порівнює його з початковим сигналом для пошуку аномалій. Використовуються такі методи:

- точкове порівняння;
- оцінка площ під кривими;
- **Dynamic Time Warping (DTW)** — для сигналів з різною швидкістю або деформованою структурою.

TadGAN здатен виявляти як точкові, так і групові аномалії. У простих випадках можна застосовувати порогову фільтрацію, але для складних змін частоти або структури сигналу генеративні моделі дають значно кращі результати.



Графік навчання моделі TadGAN на 500 епох

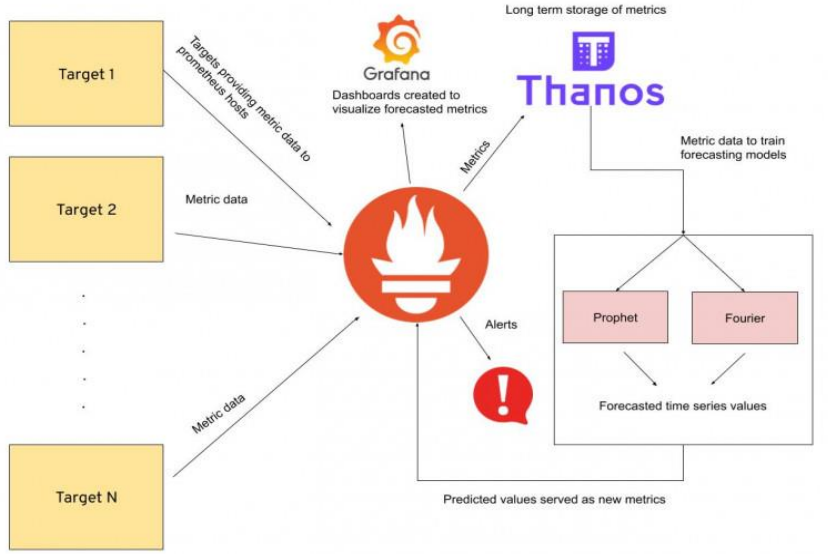


Результат роботи TadGAN на датасеті з аномальною зміною частоти

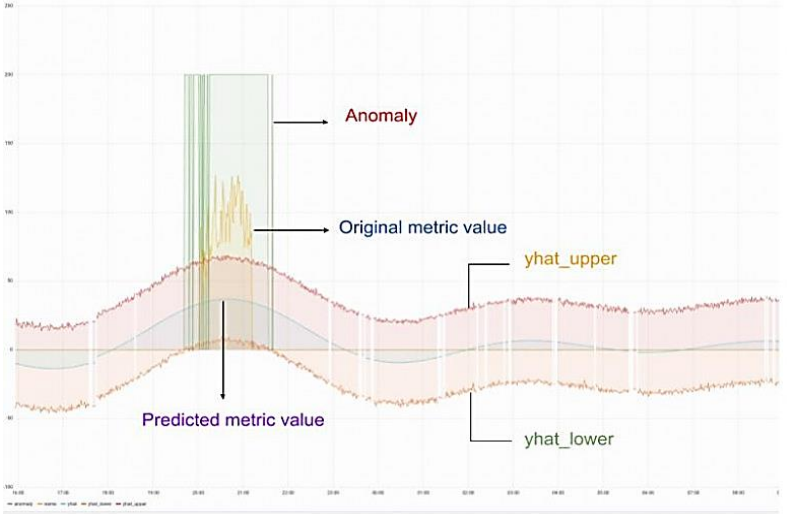
Prometheus Anomaly Detector (PAD)

Проєкт **Prometheus Anomaly Detector (PAD)** від Red Hat автоматизує виявлення аномалій у метриках Prometheus. Система самостійно обирає метрики, створює recording rules, будує прогноз поведінки та визначає нормальні порогові значення. Для прогнозування використовується алгоритм **Prophet**, який враховує сезонність, цикли й тренди часових рядів.

PAD суттєво зменшує навантаження на SRE/DevOps-команди та дозволяє швидко запускати аномалійний моніторинг у масштабних мікросервісних системах. Вона також автоматично створює дашборди в Grafana та алерти для сповіщення про відхилення.



Prometheus Anomaly Detector (PAD)



Дашборди та алерти створені PAD

Виявлення аномалій як частина системи APM

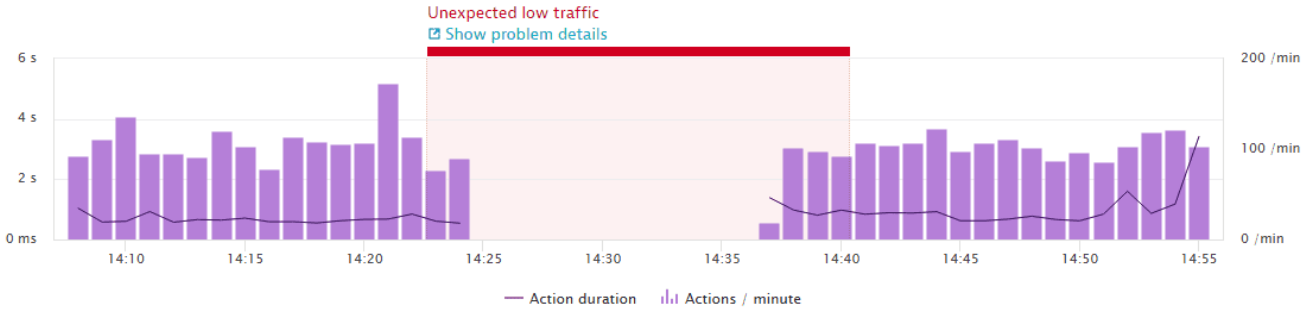
Сучасні системи APM дедалі частіше використовують AIOps — модулі з машинним навчанням, що автоматично аналізують телеметрію (метрики, логи, події, трасування) та визначають аномалії без ручного налаштування порогів. Це знімає навантаження з SRE/DevOps-команд і забезпечує проактивний моніторинг великих інфраструктур.

Основні задачі AIOps:

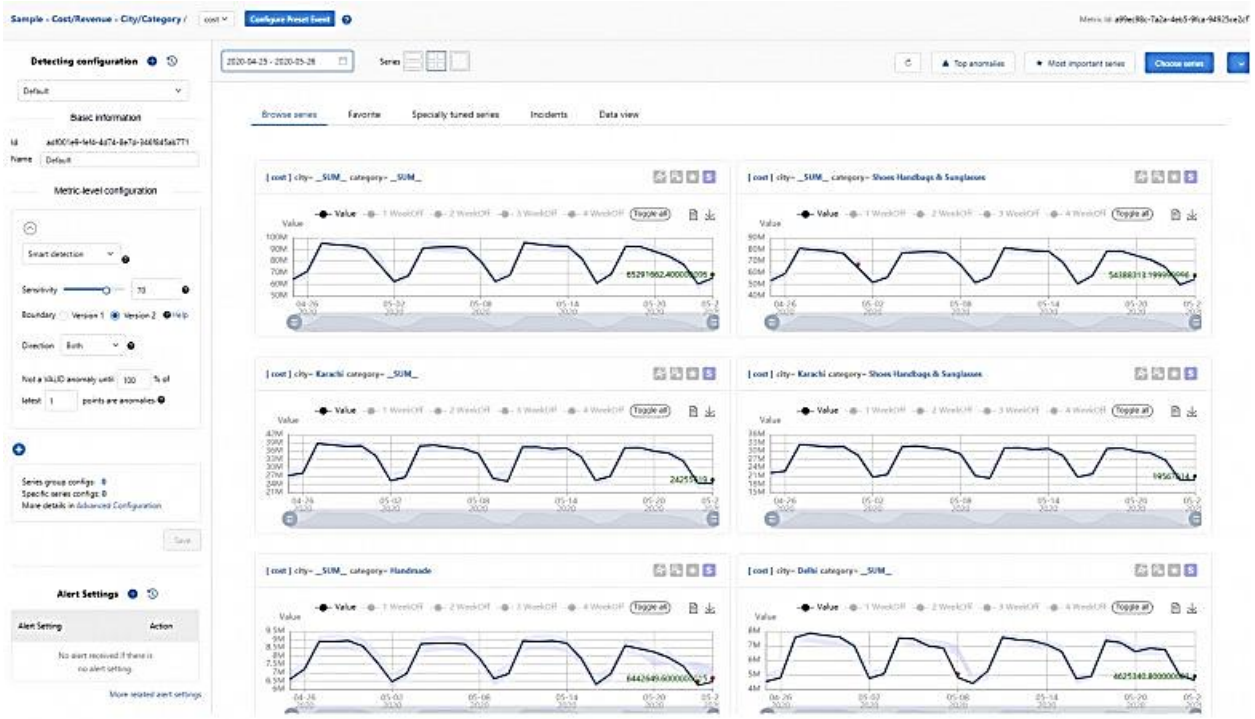
- аналіз поведінки сервісів у реальному часі;
- автоматична адаптація до нових метрик і сервісів;
- робота із сезонністю та трендами;
- пріоритизація алертів та рекомендації.

Dynatrace має вбудоване виявлення аномалій «з коробки» — система автоматично генерує алерти щодо змін трафіку, KPI чи часу обробки запитів.

Azure Metric Advisor орієнтований на бізнес-метрики та e-commerce, підтримує різні БД, але не інтегрується з Prometheus, що обмежує використання у Kubernetes-середовищах.



Системне оповіщення Dynatrace



Функціонал Azure Metric Advisor

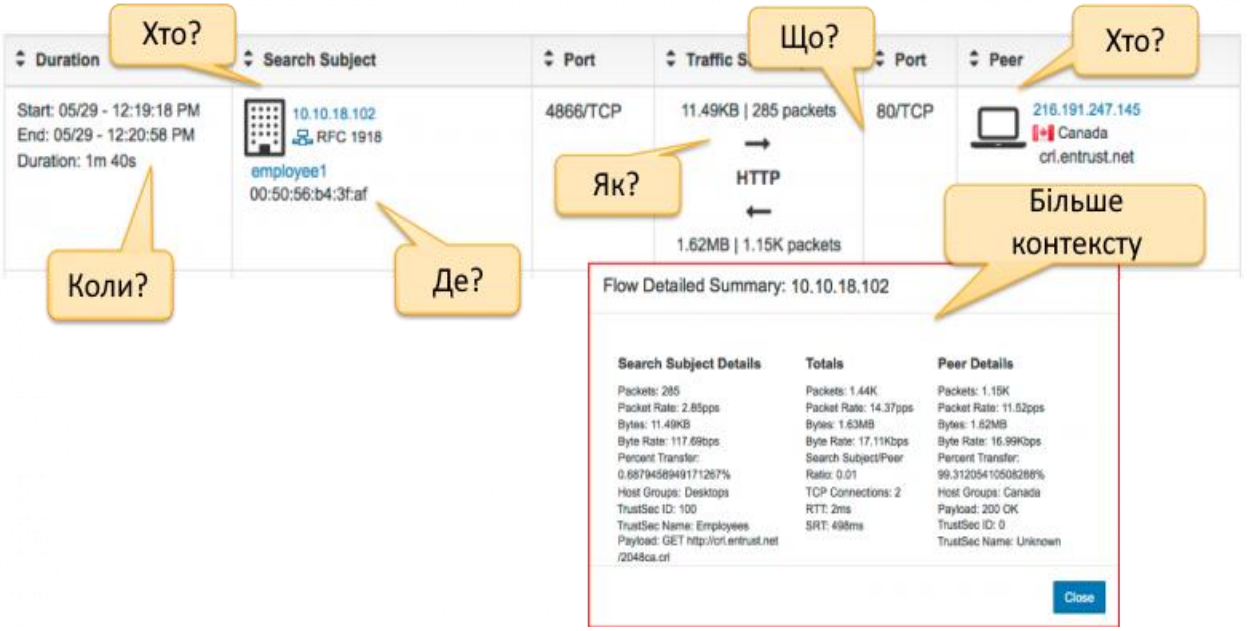
Основні джерела даних для моніторингу

Раніше безпека внутрішньої мережі асоціювалася переважно з DLP-системами та IDS. Проте IDS сьогодні не покривають усіх загроз і не масштабуються в складних мережах. Найефективнішим інструментом стає **аналіз flow-даних**, який спочатку створювався для діагностики, а сьогодні використовується для виявлення аномалій, ранніх ознак атак та поведінкових відхилень між вузлами мережі.

Flow-телеметрія має кілька переваг:

- не потребує повного перехоплення трафіку;
- дає динамічну картину взаємодій у мережі;
- охоплює 80–90% типових атак;
- відповідає на запитання: **коли, звідки, куди, яким способом і наскільки інтенсивно** відбувається зв'язок.

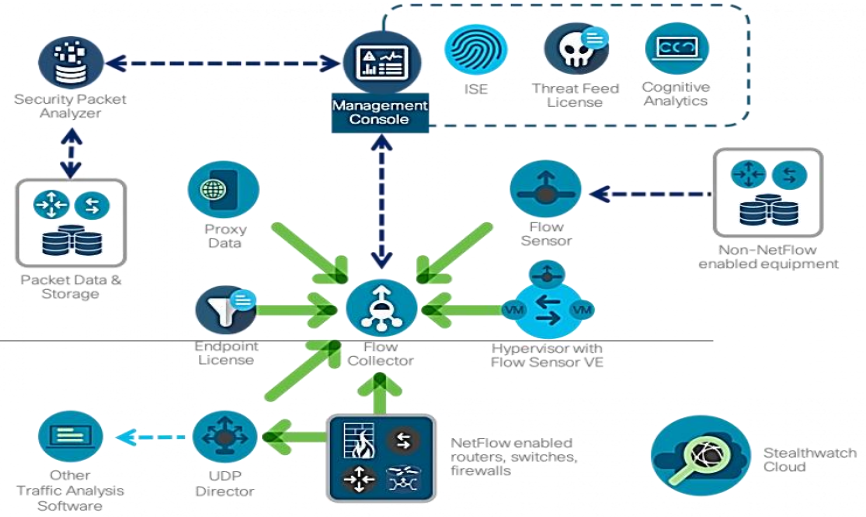
Повне захоплення трафіку (packet capture, PCAP) забезпечує глибокий аналіз, але є надто ресурсомістким. Наприклад, при швидкості 250 Мбіт/с потрібно **≈2,6 ТБ на добу**. Натомість потоки вимагають у **500 разів менше місця**, що робить їх оптимальним рішенням для постійного моніторингу.



Приклад flow-запису з розширеним контекстом у процесі аналізу мережевих потоків

Flow-протоколи та їхні відмінності

Ситема моніторингу Netflow на базі обладнання Cisco



Flow-протоколи мають критичні відмінності для кібербезпеки. NetFlow/IPFIX забезпечують **несемпльовану**, гнучку телеметрію з повним охопленням трафіку, що дозволяє виявляти аномалії та атаки. sFlow використовує **семплювання**, через що може пропустити критичні події. Найефективніше збирати телеметрію **на рівні доступу**, де видно реальну взаємодію вузлів. Якщо обладнання не підтримує flow, застосовують TAP-сенсори, але це потребує додаткових ресурсів. Інструменти аналізу різняться: **SiLK** — ручний CLI, **ELK** — гнучка платформа, **Cisco Stealthwatch** — повністю автоматизована система поведінкового аналізу.

Різниця між *NetFlow* та *sFlow*

Функція	NetFlow	sFlow
Захват пакетів	Не захоплює пакети взагалі	Копіює всі пакети та семплює 1 із N для відправлення на колектор
Підтримка протоколів	Рівень 2, IP та IPv6	Не залежить від мережевого рівня
Конфігуровані поля	Гнучкі: NetFlow має настраювані користувачем поля (шаблони)	Фіксовані поля протоколу
Записи потоків	Підтримка запису потоків IPv4, IPv6 для всього трафіку	Запис потоку не створюється — копіюються перші N байтів пакета
Апаратне прискорення	Так: записи потоків створюються «в залізі» без впливу на data plane	Апаратне прискорення можливе, але зазвичай пакети захоплюються ПЗ
Індустріальний стандарт	IPFIX	sFlow v5
Тимчасові мітки потоків (початок і кінець)	Так	Так
Packet rates (кількість пакетів у потоці)	Так	Ні
Підрахунок байтів (кількість байтів у потоці)	Так	Так (частково)

ВИСНОВКИ

- У роботі розроблено підхід до побудови системи моніторингу та виявлення аномалій у мережевому трафіку на основі flow-телеметрії та методів машинного навчання.
- Проаналізовано переваги представлення трафіку у вигляді потоків NetFlow/IPFIX замість сирих пакетів, що забезпечує до 80–90% охоплення мережевої активності.
- Сформовано модель характеристичних векторів вузлів, придатну для класифікації та кластеризації аномалій. Доведено вплив змін профілю одного вузла на поведінку інших елементів мережі.
- Особливу увагу приділено виявленню аномалій у часових рядах, зокрема reconstruction-based моделям і TadGAN, що здатні фіксувати як точкові, так і групові відхилення.
- Запропонована концепція показала високу ефективність і може слугувати основою для повноцінної системи кіберзахисту корпоративних мереж.

Дякую за увагу!

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Андрій ЛИСАКІВСЬКИЙ

Назва: РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ТА ВИЯВЛЕННЯ АНОМАЛІЙ У МЕРЕЖЕВОМУ ТРАФІКУ

Координатор: Катерина ТРЕНЬОВА

Підрозділ: ННІТ

Коефіцієнт подібності 1:0.7

Коефіцієнт подібності 2:0.4

Тривога: 7

Після аналізу Звіту подібності констатую наступне:

☐ виявлені в роботі запозичення є сумлінними і не мають ознак плагіату. Тому робота визнається самостійною і допускається до захисту;

☐ виявлені в роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на доопрацювання;

☐ виявлені в роботі запозичення є недобросовісними і мають ознаки плагіату або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. У зв'язку з чим, робота не допускається до захисту.

…… ……

Дата *Підпис Наукового керівника*