МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н. Каразіна
Факультет математики і інформатики
Кафедра теоретичної та прикладної інформатики

Кваліфікаційна робота

магістр

на тему «Пошук аномалій в часових рядах»

Виконав: студент 2 курсу, групи МФ-62
спеціальність 122 «Комп'ютерні науки»
освітньо-професійна програма
«Інформатика»

Зайцев Ю.А.
(прізвище та ініціали)
Керівник Власенко Д.І.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)

Харків – 2024 рік

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CPD – Change Point Detection

IoT – Internet of Things

ARIMA – Autoregressive Integrated Moving Average

STL – Seasonal-Trend decomposition using LOESS

RNN – Recurrent Neural Network

CNN – Convolutional Neural Network

LSTM – Long Short-Term Memory

CUSUM – CUmulative SUM

HMM – Hidden Markov Model

PCA – Principal Component Analysis

PDF – Probability Density Function

KDE – Kernel Density Estimation

SVM – Support Vector Machine

ВСТУП

Актуальність теми дослідження. У сучасному світі, орієнтованому на дані, часові ряди стали ключовим елементом багатьох галузей — від фінансів і охорони здоров'я до виробництва й кібербезпеки. Цей тип даних, що складається з вимірювань, зібраних через послідовні проміжки часу, є критично важливим для моніторингу систем, прогнозування трендів і прийняття стратегічних рішень. З розвитком технологій обсяг часових рядів зріс експоненційно завдяки поширенню сенсорів, підключених пристроїв і цифрових систем. Цей стрімкий ріст створив величезні можливості для отримання нових інсайтів, але також поставив перед дослідниками серйозні виклики у виявленні незвичних чи критичних подій, прихованих у даних.

Аномалії, які часто називають викидами або нерегулярними патернами, являють собою відхилення від очікуваної поведінки системи. Ці відхилення — не просто статистичні аномалії: вони часто несуть у собі глибокий зміст, таким чином, наприклад, аномалія у фінансових транзакціях може свідчити про шахрайські дії, тоді як незвичайні патерни у показниках здоров'я пацієнта можуть вказувати на початок медичного стану. У виробництві виявлення тонких відхилень у даних про продуктивність обладнання може запобігти катастрофічним збоям, заощаджуючи час і ресурси. Отже, виявлення цих аномалій є критично важливим для забезпечення безпеки, ефективності та надійності в різних галузях [1]. У фінансових ринках роль виявлення аномалій особливо помітна, бо незвичайні рухи цін, обсяги торгів або патерни транзакцій можуть надати значущу інформацію про маніпуляції на ринку або економічні тренди. Так само в охороні здоров'я моніторинг життєво важливих показників, таких як частота серцевих скорочень або артеріальний тиск,

залежить від здатності швидко виявляти відхилення від нормальних значень. Такі аномалії можуть забезпечити раннє попередження для медичного втручання, потенційно рятуючи життя. Зростання Інтернету речей (IoT) ще більше підсилило важливість виявлення аномалій, оскільки сенсори, вбудовані в промислове обладнання, генерують потоки даних, які потребують постійної оцінки, адже непомічена аномалія у цьому контексті може призвести до незапланованих простоїв, дорогих ремонтів або небезпечних для людей ситуацій.

Виявлення аномалій в часових рядах є складним завданням через те, що реальні часові ряди часто демонструють складну та динамічну поведінку. Патерни можуть змінюватися з часом через сезонність, змінні тренди або зовнішні порушення, що ускладнює розмежування природної мінливості та дійсно аномальних подій, до того ж аномалії, за своєю природою, є рідкісними явищами. Всі ці властивості ускладнюють розробку моделей, які можуть точно їх виявляти, не породжуючи надмірної кількості хибних результатів. Додатковим викликом є величезний обсяг даних, який генерується сучасними системами, де аномалії часто потрібно виявляти в реальному часі для забезпечення оперативної реакції [2].

Актуальність цієї галузі зростає завдяки поєднанню кількох технологічних досягнень як цього століття, так і минулих: машинне навчання, глибинне навчання та аналітика великих даних революціонізували підходи до виявлення аномалій, де раніше панували традиційні статистичні методи, які, хоча й ефективні у простіших сценаріях, часто не справляються зі складністю сучасних систем. Нові методи використовують потужність нейронних мереж, ансамблевих моделей і гібридних підходів для виявлення тонких, контекстно-залежних аномалій, які раніше були недоступними, водночас зростаюча взаємопов’язаність систем підвищила ставки, бо одна непримітна аномалія

може спричинити каскадні наслідки у мережах, що призведе до масштабних збоїв.

Важливість виявлення аномалій у часових рядах виходить за межі технічних викликів: воно задовольняє фундаментальні суспільні потреби, такі як захист фінансових систем, охорона критичної інфраструктури та підвищення якості життя. У міру того, як дані продовжують зростати за обсягом і складністю, здатність виявляти та реагувати на аномалії відіграватиме ключову роль у формуванні майбутнього багатьох галузей.

Метою дослідження було розкрити сутність поняття «аномалія в часовому ряді» та зробити огляд методів пошуку аномалій в часових рядах.

Об’єктом дослідження є процес пошуку аномалій в часових рядах.

Предметом дослідження є сучасні методи пошуку аномалій в часових рядах.

Для досягнення мети необхідно розв’язати такі **задачі дослідження**:

1. Вивчити та проаналізувати наукову літературу, розкрити зміст понять «часовий ряд», «аномалії в часових рядах» та «модель трафіку в комп’ютерних мережах».
2. Дослідити сучасну методологію пошуку аномалій в часових рядах.
3. Порівняти на практиці сучасні методи пошуку аномалій в часових рядах на різних наборах даних.

Робота складається зі вступу, двох розділів, загальних висновків, списку використаної літератури. Зміст роботи висвітлено на 57 сторінках основного тексту і містить 22 рисунки та 2 таблиці.

РОЗДІЛ 1. НАУКОВО-ТЕОРЕТИЧНІ ЗАСАДИ ПОШУКУ АНОМАЛІЙ В ЧАСОВИХ РЯДАХ

1.1. Визначення поняття «часовий ряд» та його характеристики.

Часовий ряд — це послідовність точок даних, зібраних або записаних через послідовні проміжки часу (рис. 1). Ці інтервали можуть бути регулярними (щоденні ціни на акції, щомісячні показники продажів) або нерегулярними (журнали подій у кібербезпеці, викликані певними подіями). Основною особливістю часових рядів, яка відрізняє їх від інших типів даних, є їхня притаманна тимчасова структура, де порядок точок даних має значення, оскільки вони відображають процес, що змінюється з часом [3].

Концепція часових рядів має корені у вивченні періодичних явищ в астрономії та метеорології у XVII–XVIII століттях. Перші дослідники прагнули зрозуміти повторювані патерни таких явищ, як рух планет і сезонні зміни погоди. У XIX столітті поява статистичних методів сприяла поширенню аналізу часових рядів у наукових і промислових додатках. Наприклад, сер Френсіс Гальтон і Карл Пірсон розробили методи вивчення кореляцій у даних, що змінюються з часом, а на початку XX століття такі дослідники, як Джордж Юдні Юл і Норберт Вінер, формалізували техніки моделювання часових рядів, впровадивши ключові концепції, такі як автокореляція та спектральний аналіз.

Часовий ряд — це послідовність спостережень X_t, де кожне спостереження індексується часовою міткою t. Формально часовий ряд можна визначити як:

$$\{X_t\}_{t\in T}$$

де:

- X_t - значення спостереження у момент часу t_r

- T - множина часових індексів, зазвичай $T = \{t_1, t_2, \dots, t_n\}$, де $t_i < t_{i+1}$,
- X_t може бути скаляром (одновимірний часовий ряд) або векторною величиною (багатовимірний часовий ряд).

У разі регулярних часових рядів індекси T рівновіддалені (наприклад, щоденні, щомісячні чи щорічні інтервали). Для нерегулярних часових рядів інтервали між індексами змінюються, часто зумовлені подієвими явищами.

Часовий ряд часто моделюється як сума кількох компонентів:

$$X_t = T_t + S_t + R_t$$

де:

- T_t - трендова компонента, що відображає довгострокову зміну в ряді,
- S_t - сезонна компонента, яка враховує періодичні коливання,
- R_t - залишкова (шумова) компонента, що відповідає випадковим змінам або неврахованим факторам.

Альтернативно часовий ряд можна розглядати як реалізацію випадкового процесу. Випадковий процес $\{X_t\}$ — це колекція випадкових змінних, індексованих часом t. Такий ймовірнісний підхід є основою для багатьох статистичних і машинних методів аналізу часових рядів.

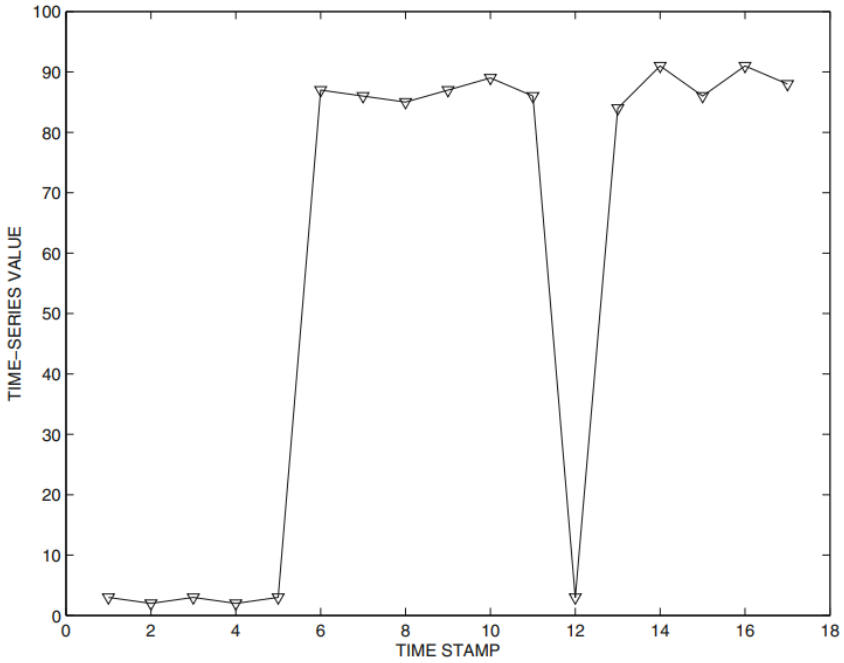

Рисунок 1. Приклад зображення часового ряду *[4]*.

Аналіз часових рядів базується на розумінні його основних властивостей, таких як стаціонарність, тренд і сезонність. Ці концепції є ключовими для ефективного аналізу, моделювання та інтерпретації, оскільки вони розкривають основну структуру й поведінку даних, що залежать від часу [3] [5].

Стаціонарність означає властивість часового ряду, коли його статистичні характеристики залишаються незмінними в часі. Це ключове припущення для багатьох статистичних моделей, оскільки стаціонарні часові ряди легше аналізувати й прогнозувати.

Часовий ряд $\{X_t\}$ є:

1. Суворо стаціонарним, якщо спільний розподіл ймовірностей (X_t, X_{t+h}) не змінюється з часом. Формально, для будь-якого моменту часу t та зсуву h :

$$P(X_t, X_{t+h}) = P(X_{t+k}, X_{t+h+k}), \forall k.$$

2. Слабко стаціонарним (стаціонарним другого порядку), якщо виконуються наступні умови:
   - Постійне математичне сподівання:
$$E[X_t] = \mu, \forall t.$$
   - Постійна дисперсія:
$$\mathrm{Var}(X_t) = \sigma^2, \forall t$$
   - Автоковаріація залежить лише від зсуву:
$$\mathrm{Cov}(X_t, X_{t+h}) = \gamma(h), \forall t$$

Тренд у часових рядах відображає довгострокові зміни в даних, які можуть включати поступове зростання, зниження або стабільність у часі. На відміну від короткострокових коливань, які часто є результатом випадкового шуму або сезонних ефектів, тренд демонструє основну динаміку системи, що формується під впливом зовнішніх або системних факторів. Розуміння трендів є ключовим для багатьох сфер, таких як економіка, кліматологія та охорона здоров'я, оскільки вони надають уявлення про загальні закономірності та допомагають відрізнити суттєві зміни від тимчасових варіацій.

Тренди часто виникають через структурні зміни у системі, такі як економічне зростання, збільшення населення або технологічний прогрес. Наприклад, глобальне підвищення середніх температур є трендом, який здебільшого пояснюється кліматичними змінами, а стабільне зростання

фондового ринку протягом десятиліть відображає економічний розвиток. Виявлення та моделювання таких трендів може надати критично важливу інформацію про довгострокову динаміку, що сприяє кращому прогнозуванню та ухваленню рішень [6].

Тренд може мати різні форми. Найпростішим є лінійний тренд, де дані демонструють стабільну швидкість зростання або зниження в часі. Наприклад, ціни на житло в міських районах із постійним попитом можуть демонструвати лінійний висхідний тренд. Лінійний тренд часто моделюється як пряма лінія:

$$T_t = \beta_0 + \beta_1 t,$$

де T_t представляє значення тренду в момент часу, $t_1\beta_0$ - це перетин, а β_1 - нахил, що вказує на швидкість змін. Ця проста модель дозволяє легко ідентифікувати й інтерпретувати лінійні тренди.

Однак багато реальних явищ не підпорядковуються прямій лінії. Нелінійні тренди охоплюють більш складні закономірності змін, такі як прискорення чи уповільнення з часом. Наприклад впровадження технологій часто слідує експоненціальній кривій зростання на початкових етапах, коли продукт або послуга набувають популярності. Це можна змоделювати як:

$$T_t = e^{\beta_0 + \beta_1 t}.$$

З іншого боку, деякі тренди демонструють зменшення темпів змін із часом, що часто спостерігається у видобутку природних ресурсів або насиченні ринку продуктів. Такі тренди можуть слідувати логарифмічній формі:

$$T_t = \beta_0 + \beta_1 \log(t).$$

Окрім лінійних і нелінійних трендів, деякі часові ряди демонструють сегментовані тренди, де напрямок або швидкість змін різко змінюються у певні моменти. Наприклад, введення нових урядових політик, технологічні прориви або економічні кризи можуть створювати розриви у тренді. Такі сегментовані

тренди часто моделюються як частково-лінійні тренди, де різні сегменти ряду мають власні нахили й перетини.

Виявлення трендів у даних часових рядів є ключовим етапом аналізу. Зазвичай використовують графічне зображення даних у часі, що дозволяє візуально ідентифікувати напрямок руху. Для точнішого визначення тренду застосовуються статистичні методи, такі як регресійний аналіз, для оцінки нахилу та перетину тренду. Також широко використовуються методи згладжування, такі як ковзні середні чи експоненційне згладжування, щоб виділити тренди шляхом фільтрації короткострокових коливань. Наприклад, 12-місячне ковзне середнє може згладжувати щомісячні дані продажів, демонструючи основну тенденцію росту на тлі сезонних варіацій.

Незважаючи на їх важливість, тренди можуть приховувати інші компоненти часових рядів, такі як сезонність чи шум. Тому тренди часто видаляються під час попередньої обробки для досягнення стаціонарності, що є ключовою вимогою багатьох аналітичних моделей. Процес видалення тренду називається детрендуванням. Один із найпростіших методів детрендування — це різницювання, яке передбачає віднімання послідовних спостережень:

$$X_t' = X_t - X_{t-1}.$$

Різницювання ефективно усуває лінійні тренди, залишаючи стаціонарні залишки для подальшого аналізу.

Для складніших трендів можна застосовувати поліноміальне згладжування або логарифмічні перетворення. Наприклад, для моделювання й видалення криволінійного тренду може бути використаний поліном другого ступеня, залишаючи після цього дані без тренду. Іншим підходом є методи декомпозиції, які розділяють часовий ряд на складові: тренд, сезонність і залишкову компоненту. Методи, такі як декомпозиція за допомогою STL або

класична декомпозиція, надають структурований спосіб виділення й аналізу трендів (рис. 2).

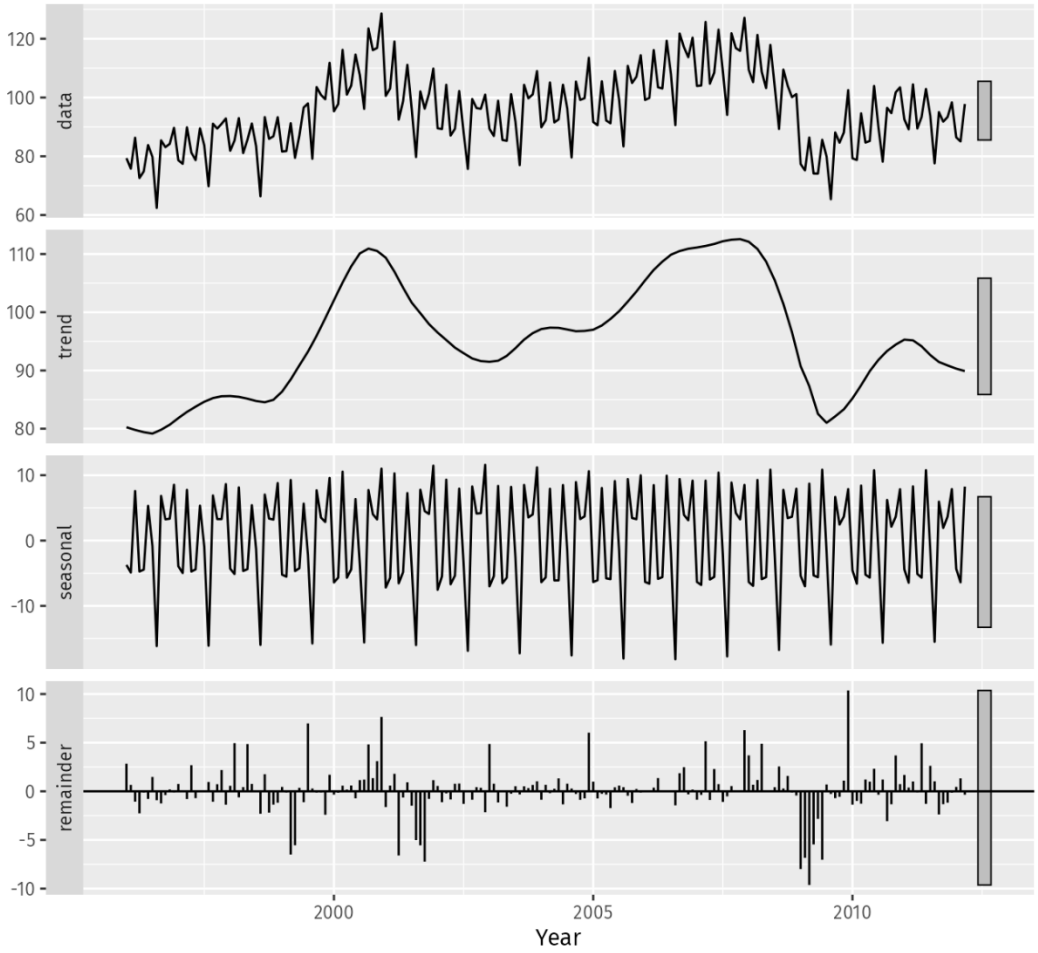

Рисунок 2. Приклад декомпозиції часового ряду методом STL [7].

Тренди мають значний вплив у реальних додатках: фінансах визначення довгострокових трендів у цінах акцій чи економічних індикаторах допомагає інвесторам приймати обґрунтовані рішення, а у кліматології аналіз трендів температури та рівня моря є критично важливим для оцінки наслідків змін клімату. Так само в охороні здоров’я відстеження трендів у показниках

пацієнтів, таких як артеріальний тиск або вага, може забезпечити раннє попередження про можливі проблеми зі здоров'ям [2].

Однак інтерпретація трендів може бути не завжди очевидною, бо у деяких випадках те, що виглядає як тренд, може бути результатом зовнішніх факторів, таких як помилки вимірювань або зміни в методах збору даних. Наприклад, раптове збільшення трафіку на веб-сайті може здаватися трендом, але насправді це може бути результатом одноразової події, наприклад, маркетингової кампанії, тому відокремлення справжніх трендів від хибних ефектів потрібен ретельний аналіз і часто знання предметної області.

Підсумовуючи, тренди є фундаментальною властивістю часових рядів, що відображає довгострокові зміни, зумовлені системними факторами. Їх виявлення, моделювання та видалення є важливими етапами аналізу часових рядів, дозволяючи виділити інші компоненти й покращити точність прогнозів. Розуміння трендів дає змогу отримувати важливі інсайти, приймати рішення та краще передбачати майбутню поведінку систем.

Сезонність — це фундаментальна характеристика багатьох часових рядів, що відображає повторювані патерни або цикли через регулярні інтервали. Така періодична поведінка виникає через природні, соціальні або економічні явища. Наприклад, роздрібні продажі часто досягають піку в святковий сезон, споживання електроенергії має добові та річні цикли, а температурні показники демонструють щорічні коливання. Сезонність відрізняється від трендів і залишкових варіацій тим, що є передбачуваною і прив'язаною до фіксованого періоду (рис. 3).

Сезонність відіграє важливу роль у прогнозуванні та виявленні аномалій. Ідентифікація та розуміння сезонних компонентів дозволяють аналітикам коригувати прогнози, виявляти відхилення та відокремлювати сезонні патерни від інших базових поведінкових характеристик у даних [5] [6].

Сезонна поведінка часто виражається математично як періодична функція. Для часового ряду X_t сезонну компоненту S_t можна моделювати за допомогою тригонометричних функцій, які ефективно описують циклічну поведінку:

$$S_t = A\sin\left(\frac{2\pi t}{P}\right) + B\cos\left(\frac{2\pi t}{P}\right),$$

де:

- A і B - параметри амплітуди, що визначають величину коливань,
- t - часовий індекс,
- P - період сезонності, що представляє кількість кроків часу до повторення циклу.

Ця формула ґрунтується на аналізі Фур'є, який розкладає складні періодичні патерни на суму синусоїдальних і косинусоїдальних функцій. Підбираючи ці компоненти, можна апроксимувати сезонність у часовому ряді.

Альтернативно сезонність можна моделювати за допомогою фіктивних змінних або базисних функцій у рамках регресійного аналізу, особливо при роботі з нерегулярними або несинусоїдальними циклами.

Взаємодія сезонності із загальним рівнем часового ряду визначає два основні типи:

1. Адитивна сезонність

Припускає, що сезонний ефект S_t є постійним у часі й незалежним від тренду. У цьому випадку часовий ряд можна записати як

$$X_t = T_t + S_t + R_t$$

де T_t - трендова компонента, а R_t - залишковий шум.

2. Мультиплікативна сезонність

Передбачає, що сезонний ефект масштабується разом із рівнем тренду. Часовий ряд у цьому випадку моделюється як:

$$X_t = T_t \cdot S_t \cdot R_t$$

Ця форма поширена в економічних даних, де вищі обсяги продажів підсилюють сезонні ефекти. Наприклад, у роздрібній торгівлі відсоткове збільшення продажів у святковий сезон може залишатися постійним, але абсолютне зростання буде більшим при високих базових продажах [7].

Одним із ключових завдань аналізу часових рядів є виділення сезонної компоненти з решти даних. Найпоширенішим підходом є декомпозиція, яка розділяє ряд на складові: тренд (T_t), сезонність (S_t) і залишки (R_t). Це можна математично виразити так:

1. Адитивна декомпозиція:

$$X_t = T_t + S_t + R_t$$

2. Мультиплікативна декомпозиція:

$$X_t = T_t \cdot S_t \cdot R_t$$

Методи декомпозиції, такі як STL, використовуються для виділення цих компонентів, після чого сезонність можна аналізувати окремо, що дозволяє виявляти патерни та коригувати прогнози [1] [5].

Сезонність також можна вивчати у частотній області за допомогою спектрального аналізу. Періодичний характер сезонності відповідає пікам у спектрі частот часового ряду. Наприклад, перетворення Фур'є розкладає ряд на його частотні компоненти, дозволяючи визначати домінуючі цикли.

Частота (f) сезонної компоненти пов'язана з її періодом (P) якс

$$f = \frac{1}{P}$$

Наприклад, річна сезонність у місячних даних $(P = 12)$ відповідає частоті $f = \frac{1}{12}$.

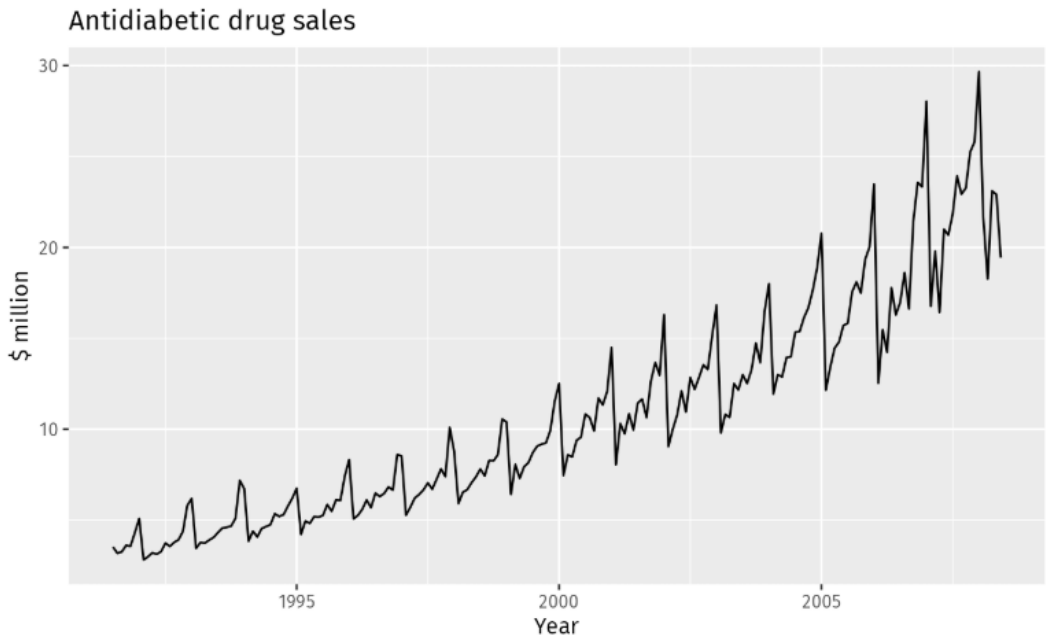

Рисунок 3. Графік продажу лік проти діабету, на якому чітко простежується і лінійний тренд, і сезонність (остання пов’язана з діями влади, що роблять накопичення ліків більш вигідним для споживачів в кінці кожного календарного року) [7].

1.2. Класифікація аномалій в часових рядах.

Аномалії в часових рядах - це точки даних, патерни або послідовності, які суттєво відхиляються від очікуваної поведінки базової системи. Їх часто називають викидами, відхиленнями або новизною, і їхнє виявлення є критично важливим для ідентифікації рідкісних або несподіваних подій. На відміну від аномалій у статичних наборах даних, аномалії в часових рядах тісно пов’язані з тимчасовими залежностями та структурами, що робить їх виявлення як складним, так і контекстно залежним [8].

Формально, аномалія в часовому ряді - це точка даних X_t або послідовність точок даних $\{X_t, X_{t+1}, \dots, X_{t+k}\}$, яка не відповідає очікуваному розподілу або часовому патерну ряду. Це відхилення може проявлятися в різних формах, залежно від контексту та специфічних особливостей часового

ряду. Поняття «очікуваної поведінки» випливає з основних статистичних або структурних властивостей даних, таких як тренд, сезонність та автокореляція. Важливість аномалій часто полягає в їхніх наслідках. Наприклад, у фінансових системах аномалії можуть вказувати на шахрайські транзакції. У сфері охорони здоров'я аномалія в життєвих показниках пацієнта може сигналізувати про критичну медичну подію. У промислових умовах відхилення в показниках сенсорів можуть попереджати про наближення відмови обладнання. Своєчасне виявлення таких аномалій є важливим для запобігання більшим збоям, зменшення ризиків та забезпечення надійності системи [4] [1].

Аномалії в часових рядах можна широко класифікувати за їхньою природою, контекстом та виникненням. Три основні типи аномалій — це точкові аномалії, контекстуальні аномалії та структурні аномалії. Кожен тип вимагає особливого підходу до виявлення та інтерпретації.

Точкові аномалії — це окремі точки даних, які суттєво відхиляються від очікуваного діапазону або розподілу значень. Ці аномалії часто називають «глобальними» аномаліями, оскільки їх можна ідентифікувати без урахування тимчасового контексту. Наприклад, раптовий сплеск температури, зафіксований сенсором, або незвично велика сума транзакції у фінансових даних можуть бути віднесені до точкових аномалій.

Зі статистичної точки зору, точкові аномалії виявляються шляхом аналізу відхилень від середнього або очікуваного значення. Техніки, такі як Z-оцінки, інтерквартильні діапазони або ймовірнісні методи, як-от моделі нормального розподілу (гаусівського), широко використовуються. Однак у часових рядах при виявленні точкових аномалій також необхідно враховувати тимчасові залежності, що робить прості статистичні методи менш ефективними у складних випадках.

Контекстуальні аномалії виникають, коли точка даних є аномальною лише у своєму тимчасовому або середовищному контексті. На відміну від точкових аномалій, контекстуальні аномалії вимагають оцінки навколишніх даних для визначення відхилення. Наприклад, температура 30°C може бути нормальною влітку, але аномальною взимку. Подібним чином, високе навантаження на сервер може бути очікуваним під час робочих годин, але аномальним у непіковий час.

Виявлення контекстуальних аномалій спирається на методи, які враховують часову структуру, такі як авторегресійні моделі, рекурентні нейронні мережі або гібридні методи, так як ці методи беруть за основу очікувану поведінку системи з часом і виявляють відхилення відносно навколишнього контексту.

Структурні аномалії відносяться до відхилень у загальному патерні, структурі або розподілі часового ряду. Вони можуть включати різкі зміни в тренді, зрушення в сезонності або раптові порушення автокореляції. Прикладами можуть бути раптове падіння трафіку на веб-сайті після порушення безпеки, зміна волатильності фондового ринку після значущої події або зрушення в показниках сенсорів, викликане переналаштуванням системи.

Структурні аномалії часто складніше виявити, оскільки вони включають зміни, що охоплюють кілька точок даних. Методи, такі як виявлення точок змін (change-point detection), сегментація часових рядів та кластеризація, зазвичай застосовуються для ідентифікації цих порушень. Структурні аномалії особливо важливі в довгостроковому моніторингу, оскільки вони можуть сигналізувати про фундаментальні зрушення в базовому процесі.

1.3. Підходи та методи пошуку аномалій в часових рядах.

Основою виявлення аномалій є порівняння спостережуваних даних із очікуваною базовою поведінкою. Очікувана поведінка може бути виведена з статистичних принципів, часової динаміки або навчання моделей. Статистичні підходи акцентують увагу на властивостях розподілу, де аномалії — це точки даних, що суттєво відхиляються від очікуваного розподілу. Натомість підходи, засновані на машинному навчанні та глибокому навчанні, використовують здатність алгоритмів вивчати складні патерни та часові залежності в даних.

Внутрішні властивості часових рядів суттєво впливають на підходи до виявлення аномалій. У стаціонарних часових рядах основна увага приділяється виявленню відхилень у значеннях на основі стабільних статистичних характеристик, таких як середнє і дисперсія. Для нестаціонарних часових рядів підходи повинні адаптуватися до трендів, сезонності та змінюваних патернів. Це часто вимагає попередньої обробки, наприклад, різницювання, усунення тренду чи масштабування, щоб стандартизувати дані перед застосуванням алгоритму виявлення [4] [5].

Підходи до виявлення аномалій часто залежать від конкретної галузі, в якій генеруються дані часового ряду. Наприклад, у фінансах аномалії зазвичай вказують на шахрайські транзакції або ринкові порушення. У цій галузі підходи значною мірою покладаються на історичні патерни та статистичні методи, які дозволяють виявляти відхилення у грошових значеннях або частотах транзакцій. У сфері охорони здоров’я аномалії можуть свідчити про раптові зміни у життєвих показниках пацієнтів, що потребує аналізу фізіологічних ритмів і адаптації до індивідуальних базових рівнів.

Індустріальні застосування, такі як прогнозуюче обслуговування, часто базуються на даних сенсорів, де аномалії можуть проявлятися як раптові сплески або поступові зрушення у роботі обладнання. У таких випадках

підходи використовують часові патерни та експлуатаційні пороги для розрізнення нормального зносу і потенційних збоїв. Кібербезпека створює ще одну унікальну задачу, оскільки аномалії у даних мережевого трафіку можуть свідчити про спроби вторгнення, а це вимагає підходів, які інтегрують часову та просторову динаміку для виявлення прихованих загроз та загроз, що еволюціонують [9] [10].

Ще одним підходом може бути створення моделі та виявлення відхилення як аномалії. Такі підходи ефективні, коли базовий процес добре зрозумілий і може бути точно змодельований. Наприклад, моделі ARIMA часто використовуються для врахування лінійних часових залежностей. У противагу, більш гнучкі підходи, засновані на даних, зокрема ті, що базуються на машинному навчанні, дозволяють виявляти складні, нелінійні залежності без необхідності явного моделювання.

Підходи, засновані на даних, особливо цінні в сценаріях з багатими наборами даних і складними патернами. Наприклад, моделі глибокого навчання, такі як рекурентні нейронні мережі (RNN) або довго-короткострокова пам’ять (LSTM), ефективно навчаються часовим залежностям, тоді як згорткові нейронні мережі (CNN) дозволяють враховувати просторові особливості. Ці підходи особливо підходять для багатовимірних часових рядів, де аномалії можуть включати тонкі зміни одразу в декількох змінних.

Ще одним важливим аспектом вибору підходу є те, чи виявлення аномалій здійснюється у реальному часі, чи ретроспективно. Підходи реального часу обробляють дані по мірі їх надходження, що є критичним для таких застосувань, як моніторинг промислових процесів або виявлення шахрайства, де потрібна негайна реакція. Ретроспективні підходи аналізують

історичні дані для виявлення аномалій постфактум, що корисно для судового аналізу або стратегічного планування.

Підходи реального часу часто використовують технології потокової обробки даних і легковагові моделі, здатні швидко приймати рішення. Ретроспективні підходи, у свою чергу, можуть використовувати обчислювально інтенсивніші методи для досягнення більшої точності, оскільки вони не обмежені часом обробки [10].

Динамічний характер багатьох систем вимагає підходів до виявлення аномалій, які можуть адаптуватися з часом. Адаптивні підходи постійно оновлюють свої моделі, щоб відображати змінювані патерни в даних, що робить їх придатними для нестаціонарних часових рядів. Такі підходи часто використовують методи онлайн-навчання або поступові оновлення моделей.

Гібридні підходи поєднують кілька методик для використання їхніх сильних сторін. Наприклад, статистичні моделі можуть використовуватися для попередньої обробки даних і виявлення початкових відхилень, а алгоритми машинного навчання — для уточнення процесу виявлення. Аналогічно, комбінація керованого навчання з некерованими методами може компенсувати відсутність позначених даних шляхом створення псевдоміток для навчання.

Один із найпростіших статистичних методів для виявлення аномалій - це метод порогових значень. Цей підхід передбачає встановлення попередньо визначених верхніх і нижніх меж, за межами яких точки даних вважаються аномаліями. Математично, для часового ряду X_t, аномалія визначається, якщо:

$$X_t < L \text{ або } X_t > U$$

де L і U - нижній та верхній пороги відповідно. Цей метод інтуїтивно зрозумілий і обчислювально ефективний, але обмежений своєю статичною природою. Він не адаптується до змін у часовому ряді, таких як тренди або сезонність, що робить його непридатним для складних або

змінюваних наборів даних. Проте цей підхід часто використовується як базовий у системах, де очікувані діапазони добре визначені, наприклад, у моніторингу фізичних параметрів промислового обладнання.

Метод Z-оцінок ідентифікує аномалії на основі того, наскільки точка даних відхиляється від середнього значення ряду у стандартних відхиленнях. Для часового ряду із середнім значенням μ і стандартним відхиленням σ, Z-оцінка для точки X_t обчислюється як:

$$Z_t = \frac{X_t - \mu}{\sigma}$$

Точки даних, для яких $|Z_t|$ перевищує попередньо визначений поріг, позначаються як аномалії (рис. 4). Цей метод припускає, що часовий ряд має нормальний (гаусівський) розподіл, що робить його придатним для багатьох природних наборів даних [11].

Однак у випадках, коли часовий ряд є нестаціонарним або ненормальним, метод Z-оцінок може призводити до хибних спрацьовувань або пропусків. У таких ситуаціях необхідні попередні етапи обробки, такі як різницювання або перетворення (наприклад, логарифмічне), для стабілізації статистичних характеристик ряду.

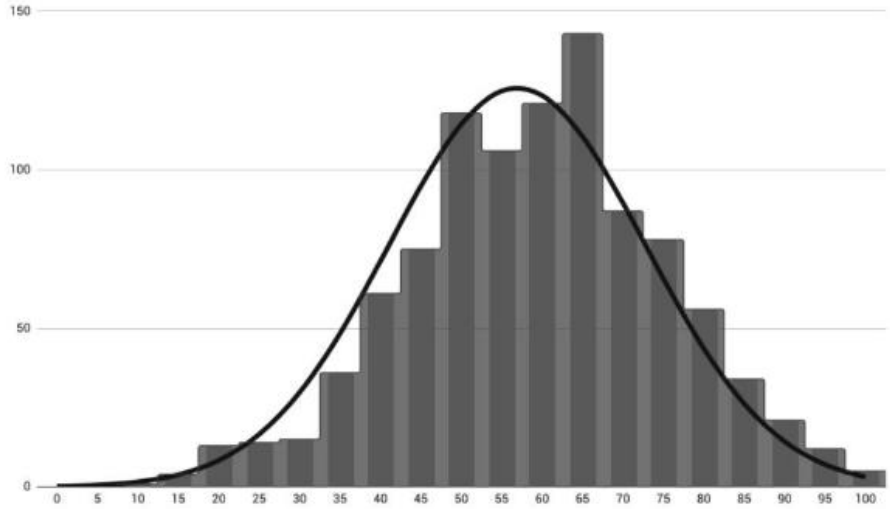

Рисунок 4. Приклад роботи методу Z-оцінка [11].

Статистичне тестування гіпотез включає формулювання нульової гіпотези H_0, яка представляє нормальну поведінку часового ряду, і альтернативної гіпотези H_1, що вказує на наявність аномалій. Аномалії виявляються, коли спостережувані дані суттєво відхиляються від очікуваних значень за H_0.

Прикладом є тест (або критерій) Граббса, який ідентифікує викиди шляхом перевірки, чи є максимальне абсолютне відхилення від середнього значення суттєво більшим за очікуване за нормального розподілу:

$$G = \frac{\max|X_t - \mu|}{\sigma}$$

Якщо G перевищує критичне значення, відповідна точка позначається як аномалія. Хоча цей метод ефективний для одномірних часових рядів, він стає обчислювально дорогим у багатовимірних наборах даних.

Авторегресійні моделі, такі як ARIMA, широко використовуються для виявлення аномалій у часових рядах. Ці моделі враховують часові залежності в даних, представляючи кожну точку як лінійну комбінацію попередніх

значень і стохастичного похибкового члена. Загальна форма автокореляційної моделі:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t$$

де $\phi_1, \phi_2, \ldots, \phi_p$ - коефіцієнти моделі, p - порядок лагу, а ϵ_t - білий шум. У ARIMA ряд спочатку трансформується для досягнення стаціонарності за допомогою різницювання (компонента «I»), після чого моделюється за допомогою авторегресійних і ковзних середніх компонент. Аномалії виявляються шляхом порівняння спостережуваних значень із прогнозами моделі. Великі похибки прогнозу, визначені як:

$$e_t = X_t - \hat{X}_t$$

Методи виявлення точок змін зосереджені на ідентифікації структурних аномалій, таких як різкі зміни середнього, дисперсії або інших статистичних характеристик часового ряду. Ці зміни часто вказують на значущі події або зрушення у режимах роботи системи.

Метод кумулятивної суми (CUSUM) є класичним підходом до виявлення точок змін. Він відстежує кумулятивну суму відхилень від середнього:

$$S_t = \sum_{i=1}^{t} (X_i - \mu)$$

де μ - очікуване середнє значення. Раптові зміни у S_t вказують на потенційні точки змін.

Баєсові методи виявлення точок змін оцінюють апостеріорну ймовірність точок змін за наявними даними. Цей підхід особливо корисний для ідентифікації кількох точок змін у серії.

Методи на основі густини виявляють аномалії, оцінюючи функцію густини ймовірності (PDF) часового ряду і позначаючи точки в областях із

низькою густиною як аномалії. Ядрова оцінка густини розподілу (KDE) є поширеним підходом для цього. Для точки X_t KDE оцінює густину як:

$$f(X_t) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_t - X_i}{h}\right)$$

де K - ядрова функція, а h - ширина вікна. Точки з низьким значенням $f(X_t)$ вважаються аномаліями.

Ймовірнісні методи розширюють цей підхід, моделюючи часовий ряд як стохастичний процес, наприклад, за допомогою прихованої моделі Маркова (HMM). Аномалії ідентифікуються як послідовності з низькою ймовірністю за навченою моделлю. HMM особливо ефективні для часових рядів із дискретними станами, таких як сигнали мовлення або журнали систем.

Статистичні методи є обчислювально ефективними і добре працюють із одновимірними, добре структурованими часовими рядами. Проте вони мають труднощі з обробкою складних, нелінійних або багатовимірних наборів даних. Ці методи часто припускають стаціонарність і нормальний розподіл, що може не відповідати реальним даним. Незважаючи на ці обмеження, статистичні методи залишаються основою для виявлення аномалій, слугуючи базовою лінією та доповнюючи більш складні техніки.

Методи машинного навчання набули значного поширення у виявленні аномалій у часових рядах завдяки їх здатності моделювати складні залежності, обробляти багатовимірні дані та адаптуватися до змінюваних характеристик даних. На відміну від традиційних статистичних методів, підходи машинного навчання базуються на даних і здатні виявляти складні патерни, які важко описати за допомогою явних правил або попередньо визначених порогів.

Керовані методи покладаються на позначені набори даних, де аномалії явно позначені під час навчання. Модель навчається розрізняти нормальну та

аномальну поведінку, мінімізуючи помилки прогнозування або максимізуючи точність класифікації [9] [12].

Зазвичай процес застосування керованого навчання для виявлення аномалій у часових рядах включає такі етапи: вилучення ознак, навчання моделі та оцінка. Ознаки часто включають часові властивості, такі як відкладені значення, ковзні середні або перетворення Фур'є, для опису динаміки ряду.

Популярні алгоритми керованого навчання включають:

- Опорновекторні машини (SVM): Ці моделі знаходять граничну поверхню, яка розділяє нормальні та аномальні точки даних у просторі ознак. Для нелінійних залежностей використовуються ядрові функції [4].
- Дерева рішень і випадкові ліси: Ці моделі створюють ієрархічні правила для класифікації точок даних. Вони інтуїтивно зрозумілі та ефективні для наборів даних із чіткими відмінностями між нормальними та аномальними патернами.

Хоча керовані методи можуть досягати високої точності за наявності достатньої кількості позначених даних, отримання таких наборів часто є складним у реальних сценаріях. Аномалії є рідкісними та різноманітними, що ускладнює покриття всіх можливих випадків під час навчання [9].

Некеровані методи не потребують позначених даних, що робить їх добре придатними для виявлення аномалій у часових рядах, де аномалії часто невідомі або трапляються рідко. Ці методи припускають, що аномалії рідкісні та відрізняються від більшості даних.

Алгоритми кластеризації групують схожі точки даних у кластери, а аномалії визначаються як точки, які не належать жодному кластеру або знаходяться в малонаселених кластерах. Приклади:

- Кластерізація методом k-середніх (англ. k-Means clustering): Розбиває дані на k кластерів, мінімізуючи внутрішньокластерні відстані. Точки з великими відстанями до центрів кластерів позначаються як аномалії.
- DBSCAN: Алгоритм, що групує точки даних на основі щільності та визначає аномалії як точки в малощільних областях (рис. 5).

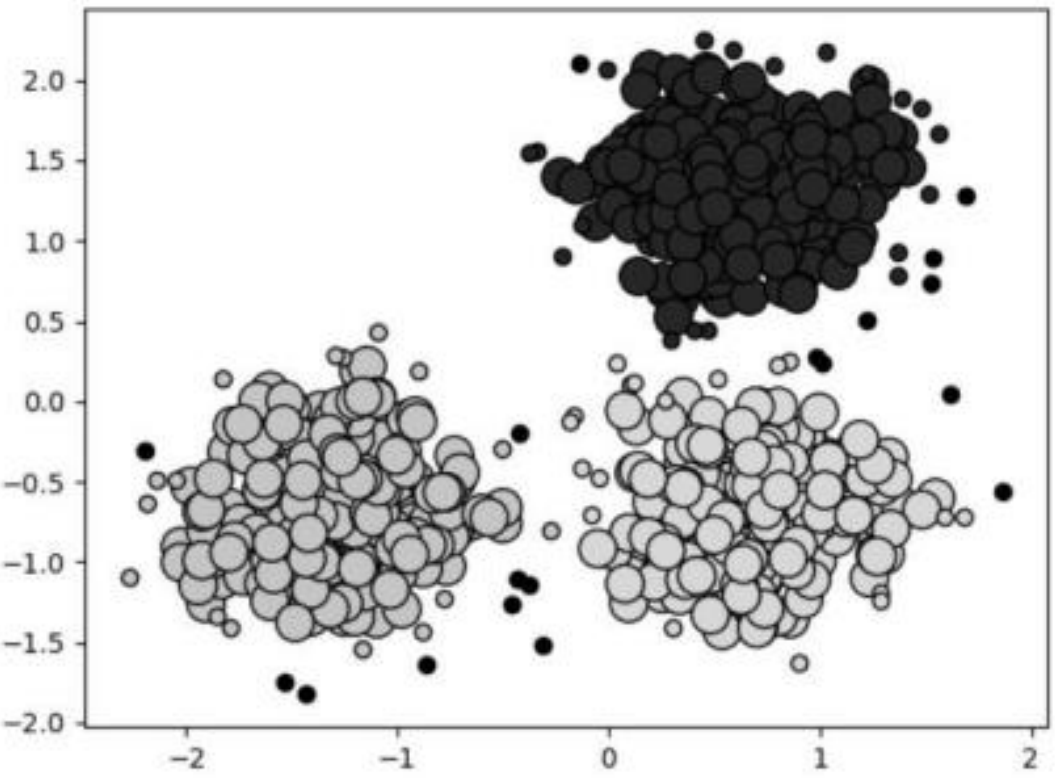

Рисунок 5. Приклад роботи методу DBSCAN [11].

Методи, такі як метод головних компонентів (PCA), перетворюють дані у простір меншої розмірності, зберігаючи якомога більше варіації. Аномалії визначаються як точки з високою похибкою відновлення або ті, що лежать далеко від основного кластеру даних у зменшеному просторі [7].

Ізоляційний ліс (англ. Isolation Forest) — це некерований алгоритм виявлення аномалій, який ґрунтується на ідеї ізоляції викидів. На відміну від багатьох інших методів, Isolated Forest не намагається створити модель розподілу даних. Натомість він використовує властивість аномалій, яка полягає в тому, що вони легше ізолюються через свою рідкість та віддаленість від основної маси даних.

Ізоляція здійснюється за допомогою побудови випадкових дерев, які розділяють дані. Для кожної точки даних алгоритм генерує шлях через дерево, починаючи від кореня і закінчуючи ізоляцією. Довжина цього шляху є мірою «ізольованості» точки. Аномалії, як правило, мають коротші шляхи через те, що вони легко ізолюються, тоді як нормальні точки вимагають більшої кількості розділень.

Кожне дерево в Isolation Forest будується на підмножині даних шляхом ітеративного розбиття. Для кожного вузла:

1. Вибирається атрибут X_i випадковим чином.
2. Обирається порогове значення p з діапазону значень атрибуту X_i.

Процес триває, поки всі точки даних не будуть ізольовані або поки не буде досягнуто максимальну глибину дерева.

Для точки x, ізольованість вимірюється як середня довжина шляху $h(x)$ через ліс із t дерев:

$$s(x,n) = 2^{-\frac{h(x)}{(n)}}$$

де:

- n - кількість точок у вибірці,
- $c(n)$ - середня довжина шляху в повному бінарному дереві, що визначається як:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}$$

- $H(i)$ - гармонійне число:

$$H(i) = \sum_{k=1}^{i} \frac{1}{k}$$

Чим нижче значення $s(x, n)$, тим більша ймовірність, що точка x є аномалією.

Метод коефіцієнту локального відхилення (LOF) є популярним підходом для виявлення аномалій, особливо в складних наборах даних зі змінною локальною щільністю. Головна ідея методу полягає в тому, щоб оцінити, наскільки незвичною є точка даних порівняно з її найближчими сусідами. На відміну від глобальних методів, які шукають викиди на рівні всього набору даних, LOF фокусується на локальному контексті, що дозволяє ефективно працювати зі структурами, де щільність даних варіюється в різних областях.

Метод базується на кількох ключових поняттях. Спершу обчислюється відстань до *k*-го найближчого сусіда кожної точки, що визначає масштаб локальної області, у якій аналізується її щільність. Потім вводиться концепція досяжності, яка враховує не лише відстань до сусідів, але й мінімальну відстань, необхідну для «досягнення» цих точок. Це дозволяє враховувати випадки, коли точка лежить у малонаселеній області, але її сусіди є щільно згрупованими.

На основі цих понять для кожної точки розраховується її локальна щільність. Ця щільність, по суті, є зворотною величиною середньої досяжності до всіх її сусідів. Після цього LOF визначається як співвідношення локальної щільності точки до середньої щільності її сусідів. Якщо локальна щільність точки значно нижча за щільність її сусідів, LOF набуває високого значення, що

вказує на можливу аномалію. Нормальні точки, навпаки, мають LOF, близький до одиниці.

Метод LOF є особливо ефективним у випадках, коли дані мають неоднорідну структуру, і аномалії важко виявити глобальними методами. Наприклад, у наборах даних зі змінною щільністю LOF дозволяє виявляти точки, які виглядають нормально у глобальному масштабі, але є аномальними в локальному контексті.

Напівкероване навчання комбінує переваги керованих і некерованих методів. Ці техніки навчаються здебільшого на нормальних даних, моделюючи очікувану поведінку. Відхилення від цієї поведінки позначаються як аномалії.

Моделі одного класу, такі як One-Class SVM, призначені для розмежування однокласових (нормальних) даних від аномалій. Ці моделі визначають межу навколо нормальних точок, а аномалії ідентифікуються як точки, що виходять за межі цієї границі (рис. 6).

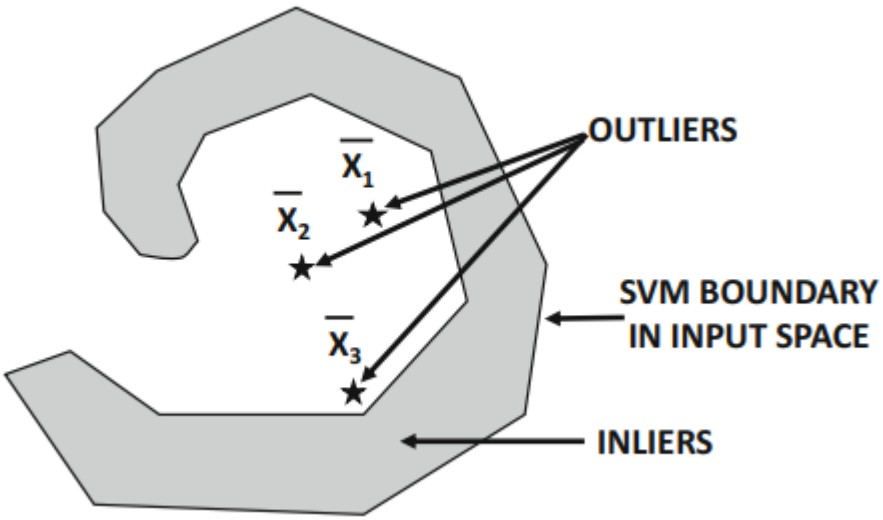

Рисунок 6. Гіпотетична ілюстрація використання опорновекторної машини (SVM) для задання границь [4].

Також для пошуку аномалій можуть використовуватися автоенкодери (рис. 7). Це нейронні мережі, які навчаються відновлювати свої вхідні дані. Вони складаються з енкодера, який стискає дані до латентного представлення, та декодера, який відновлює вхідні дані з цього представлення. Похибка відновлення використовується для виявлення аномалій:

$$E_t = \left\| X_t - \hat{X}_t \right\|$$

де X_t - вихідні дані, а $\hat{X}_t$ - відновлені дані.

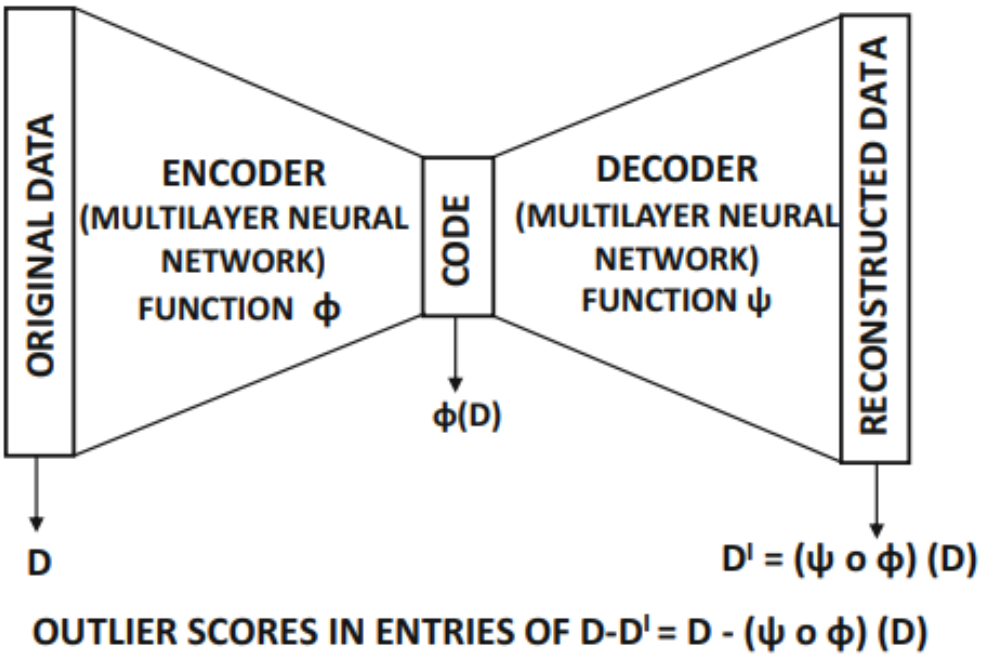

Рисунок 7. Принцип роботи автоенкодерів [4].

Часові ряди за своєю природою містять тимчасові залежності, які необхідно враховувати під час виявлення аномалій.

RNN, а також їх варіант LSTM, розроблені спеціально для послідовних даних. Ці моделі зберігають пам’ять про попередні стани, дозволяючи їм

навчатися довгостроковим залежностям. Для виявлення аномалій RNN можуть навчатися прогнозувати майбутні значення часового ряду, а аномалії визначаються як точки з високою похибкою прогнозу:

$$e_t = \left| X_t - \hat{X}_t \right|$$

де $\hat{X}_t$ - прогнозоване значення.

Методи машинного навчання забезпечують гнучкість, масштабованість і здатність обробляти складні, багатовимірні дані. Вони особливо ефективні у випадках, коли залежності є нелінійними або розподіл даних невідомий. Проте ці методи вимагають значних обчислювальних ресурсів, особливо підходи глибокого навчання.

Незважаючи на ці виклики, методи машинного навчання є потужним інструментом для виявлення аномалій, що дозволяють долати обмеження традиційних підходів і отримувати нові інсайти зі складних наборів даних.

Поєднання методів машинного навчання зі статистичними підходами підвищує їх ефективність. Наприклад, статистичні моделі, такі як ARIMA, можуть попередньо обробляти дані, усуваючи тренди та сезонність, що дозволяє моделям машинного навчання зосереджуватися на залишкових аномаліях.

1.4. Інструменти та програмні засоби для пошуку аномалій в часових рядах.

Розвиток аналітики, заснованої на даних, призвів до появи різноманітних інструментів і програмного забезпечення для виявлення аномалій у часових рядах. Ці інструменти варіюються від спеціалізованих бібліотек для аналізу часових рядів до універсальних платформ, які можуть обробляти різноманітні типи даних. Найбільш помітними є інструменти, що

пропонують розширені можливості моделювання, зручні інтерфейси та масштабованість для роботи з великими наборами даних. У цьому розділі розглядаються ключові інструменти з акцентом на їхню роль у виявленні аномалій.

Facebook Prophet — це універсальна бібліотека з відкритим кодом, розроблена для прогнозування даних часових рядів із вбудованими можливостями виявлення аномалій. Prophet особливо підходить для рядів із сильними сезонними патернами та пропусками даних. Завдяки простоті використання та потужним технікам моделювання цей інструмент здобув популярність серед практиків і дослідників [13].

Модель Prophet представляє часові ряди як адитивну комбінацію компонентів, що описуються рівнянням:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t,$$

де:

- $g(t)$: трендова компонента, що відображає довгострокові зміни;
- $s(t)$: сезонна компонента, яка враховує періодичні варіації;
- $h(t)$: ефекти свят або спеціальних подій;
- ϵ_t : шум або залишкові помилки.

Prophet виявляє аномалії, порівнюючи фактичні значення з прогнозованими. Значні залишкові відхилення свідчать про потенційні аномалії. Завдяки автоматичному врахуванню сезонності та можливості додавання подій цей інструмент ефективно застосовується у таких сферах, як роздрібні продажі, споживання енергії та аналіз веб-трафіку.

Метод Гольта-Вінтерса, хоча й не є окремою бібліотекою, часто реалізується у таких інструментах, як бібліотека statsmodels у Python або пакет forecast у R. Цей метод розширює просте експоненційне згладжування, додаючи компоненти для тренду та сезонності, що робить

його придатним для виявлення аномалій у рядах із передбачуваними патернами.

Метод Гольта-Вінтерса розкладає часовий ряд на три компоненти: рівень (L_t), тренд (T_t) і сезонність (S_t) за допомогою таких рівнянь:

$$L_t = \alpha\left(X_t - S_{t-p}\right) + (1-\alpha)(L_{t-1} + T_{t-1}),$$
$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$$
$$S_t = \gamma(X_t - L_t) + (1-\gamma)S_{t-p}$$

де:

- α, β, γ : параметри згладжування;
- p : сезонний період.

Аномалії визначаються як значні відхилення від прогнозованих значень, особливо якщо похибки перевищують попередньо визначені пороги.

PyCaret — це бібліотека машинного навчання з відкритим кодом, яка спрощує процес створення та впровадження моделей, включаючи ті, що використовуються для виявлення аномалій. Її модуль для часових рядів підтримує виявлення аномалій через інтеграцію різних алгоритмів, таких як ізоляційні ліси, автоенкодери та методи кластеризації.

Однією з сильних сторін PyCaret є автоматизація. Користувачі можуть виконувати попередню обробку даних, навчати декілька моделей і оцінювати їх продуктивність з мінімальним обсягом коду. Для аналізу часових рядів PyCaret пропонує додавання відкладених ознак, ковзних середніх і сезонних коригувань, що дозволяє більш точно ідентифікувати аномалії.

Kats — це ще один потужний інструмент для аналізу часових рядів і виявлення аномалій, розроблений Facebook. На відміну від Prophet, орієнтованого на прогнозування, Kats пропонує комплексну платформу для

завдань із часовими рядами, включаючи моделювання, вилучення ознак і виявлення аномалій. Kats включає сучасні алгоритми, такі як Seasonal Hybrid Extreme Studentized Deviate (S-H-ESD) та Bayesian Change Point Detection (BCPD).

Метод S-H-ESD ідентифікує аномалії шляхом ітеративного вилучення викидів на основі надійних статистичних оцінок. Інтеграція BCPD дозволяє Kats виявляти структурні аномалії, такі як раптові зміни трендів або сезонних патернів. Завдяки модульному дизайну та сумісності з Python Kats є універсальним інструментом для дослідників і практиків.

Azure Anomaly Detector — це хмарний сервіс від Microsoft, який використовує передові алгоритми машинного навчання для виявлення аномалій у даних часових рядів. Сервіс підтримує як одновимірні, так і багатовимірні ряди, що робить його придатним для таких застосувань, як моніторинг IoT-сенсорів або виявлення шахрайства у фінансових системах.

Цей сервіс автоматично виявляє тренди та сезонність, застосовуючи методи, такі як STL, для попередньої обробки даних. Azure Anomaly Detector дозволяє виявляти аномалії як у режимі реального часу, так і у пакетному режимі. Його інтеграція з екосистемою Microsoft робить його особливо корисним для корпоративних додатків.

Серед інших інструментів для пошуку аномалій в часових рядах також є:

- TensorFlow та PyTorch: Ці фреймворки глибокого навчання дозволяють створювати кастомні моделі для аналізу часових рядів, такі як LSTM, VAE та GAN.
- Elastic Stack (раніше ELK Stack): Використовується для аналізу логів і метрик, включаючи функції виявлення аномалій за допомогою моделей машинного навчання.

- Anomalize (R): Пакет для виявлення аномалій у часових рядах, який безшовно інтегрується з екосистемою tidyverse.

Ці інструменти пропонують різні рівні автоматизації, гнучкості та масштабованості, задовольняючи потреби широкого кола користувачів.

Висновки до розділу 1

Аномалії в часових рядах є відхиленнями від очікуваної поведінки і мають критичне значення у таких сферах, як фінанси, охорона здоров'я та промисловий моніторинг. Виявлення цих аномалій потребує глибокого розуміння характеристик часових рядів, включаючи їхню часову структуру, тренди, сезонність і шум. Ці властивості визначають вибір методів виявлення та формують етапи попередньої обробки, необхідні для ефективного аналізу.

Класифікація аномалій на точкові, контекстуальні та структурні підкреслює різноманітність викликів у виявленні аномалій. Кожен тип вимагає індивідуального підходу, що акцентує увагу на важливості контексту та часових залежностей у розрізненні аномалій від природних варіацій.

Методи виявлення аномалій охоплюють статистичні, машинного навчання та глибинного навчання. Статистичні методи пропонують базові інструменти для ідентифікації відхилень на основі припущень щодо розподілу або часових патернів. Підходи машинного навчання розширюють ці можливості, використовуючи дані для виявлення складних і багатовимірних залежностей. Методи глибинного навчання, особливо ефективні для масштабних і багатовимірних часових рядів, дозволяють ідентифікувати тонкі та контекстно залежні аномалії.

Сучасні інструменти, такі як Facebook Prophet, Kats та Azure Anomaly Detector, забезпечують доступну реалізацію цих концепцій, інтегруючи передові алгоритми з практичними робочими процесами. Ці інструменти

демонструють зростаючу синергію між теоретичними досягненнями та реальними застосуваннями.

Таким чином, виявлення аномалій у часових рядах є динамічною галуззю, яка поєднує фундаментальні принципи та передові техніки. Комбінуючи глибоке розуміння властивостей часових рядів із інноваційними методологіями, можливо ефективно виявляти критичні аномалії, сприяючи покращенню прийняття рішень і надійності систем.

РОЗДІЛ 2. ПРАКТИЧНЕ ПОРІВНЯННЯ ЕФЕКТИВНОСТІ МЕТОДІВ ПОШУКУ АНОМАЛІЙ

2.1. Набори даних, інструменти та методи пошуку аномалій.

Метою практичної частини цієї роботи є порівняння ефективності роботи різних методів пошуку аномалій в часових рядах на різних даних. Для цього важливо вирішити наступні задачі:

1. Обрати набори даних та визначити критерії ефективності.

Вибір наборів даних є критичним етапом у проведенні практичної роботи з виявлення аномалій у часових рядах, адже саме від цього залежить релевантність отриманих результатів. Набори даних мають відповідати цілям дослідження та забезпечувати умови для об’єктивної оцінки ефективності методів.

Одним із ключових аспектів вибору є наявність чітко задокументованих аномалій у даних. Використання таких наборів дозволяє порівнювати результати роботи методів із відомими еталонними значеннями, що значно підвищує точність оцінки. Наприклад, якщо набір даних містить аномалії, які були ідентифіковані та підтверджені експертами, це надає можливість об’єктивно виміряти такі показники, як чутливість і точність методів.

Різноманіття наборів даних є ще одним важливим фактором. Часові ряди з різними властивостями — такими як сезонність, наявність трендів, короткі або довгі часові періоди, багатовимірність тощо — дозволяють тестувати методи у широкому спектрі умов. Це дає змогу зрозуміти, які методи краще працюють у різних сценаріях, наприклад, в умовах шуму, при різному співвідношенні аномалій до загальної кількості точок або за наявності складних залежностей між змінними.

Ще один аспект, який варто враховувати, — це реалістичність даних. Дані, що імітують реальні сценарії використання, допомагають оцінити методи у практичних умовах. Наприклад, у сфері охорони здоров’я часові ряди можуть відображати фізіологічні параметри пацієнтів, а в промисловості — показники роботи обладнання. Тестування на таких наборах дозволяє оцінити не лише математичну ефективність методів, але й їхню здатність адаптуватися до вимог конкретних галузей.

Важливо також врахувати обсяг і розмірність наборів даних. Великі набори даних допомагають перевірити масштабованість методів, тобто їхню здатність ефективно працювати в умовах великих обсягів інформації. Багатовимірні часові ряди, у свою чергу, дозволяють оцінити, як методи обробляють складні взаємозв’язки між змінними.

Першим набором даних було обрано часовий ряд з показниками попиту на таксі в Нью-Йорк Сіті з датасету NAB (Numenta Anomaly Benchmark) [14]. Набір даних демонструє чіткі та повторювані патерни, які характеризуються денними піками та спадами, що відповідають передбачуваним циклам активності, таким як міська динаміка або попит на транспорт. Ці регулярні патерни забезпечують добре визначену базову лінію, що дозволяє моделям ефективно ідентифікувати основну поведінку та виявляти відхилення як аномалії (рис. 8). А так як авторами набору даних чітко зазначені аномальні точки, можна чітко виявити хибно-позитивні результати. Надалі істині аномалії на графіках позначатимуться червоними точками.

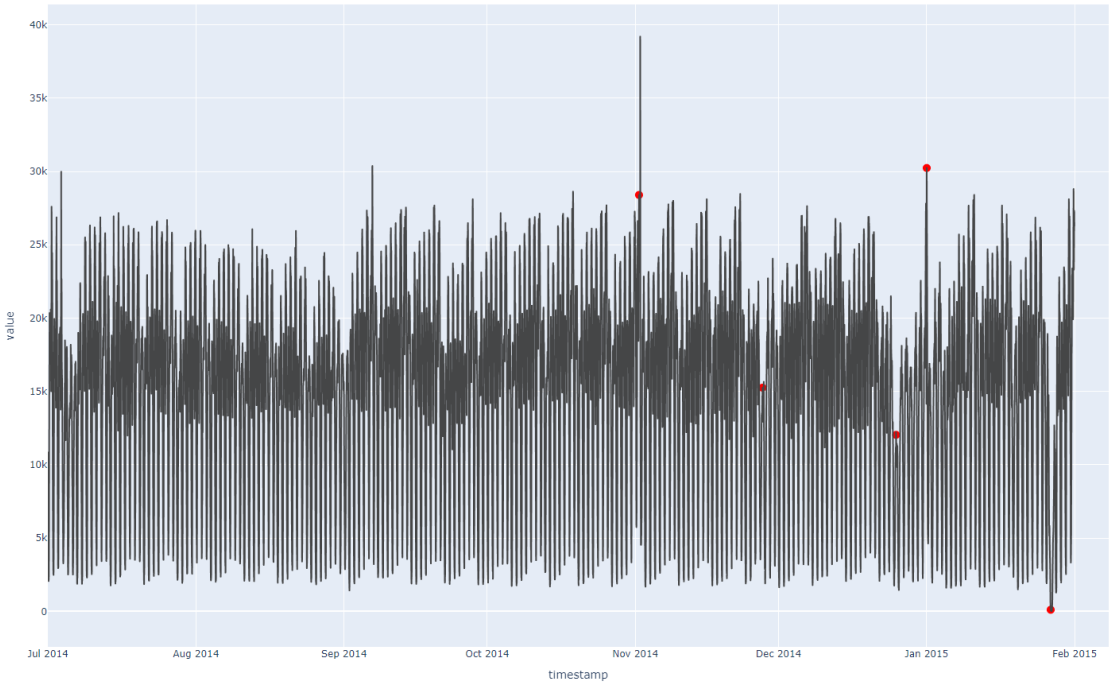

Рисунок 8. Попит таксі Нью-Йорк Сіті.

Другим часовим рядом для тестування різних методів пошуку аномалій було взято ряд з показниками температур промислової машини з датасету NAB [14]. Він був обраний через його IoT-походження, достатній обсяг даних (близько двадцяти двох тисячі точок) та наявність чітко визначених аномалій, що зробить оцінку результатів більш точною. В ньому значення являють собою показники температури кожні п’ять хвилин (рис. 9).

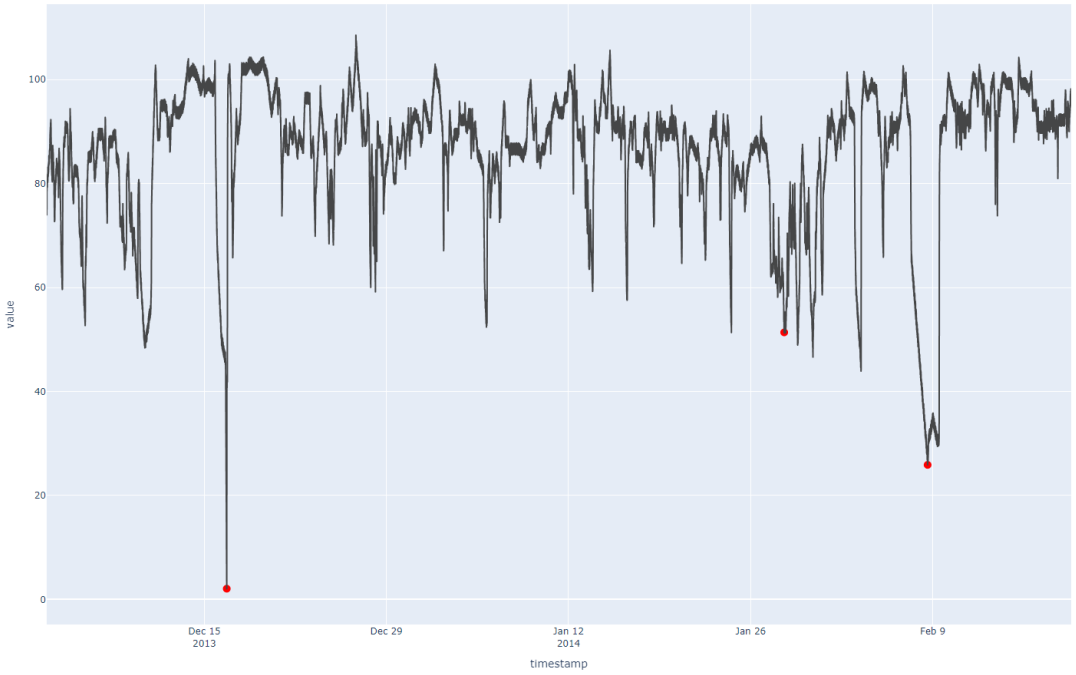

Рисунок 9. Температура промислової машини.

Згідно з документацією, перша аномалія була спричинена плановим відключенням машини. Друга – критичною помилкою, що призвела до третьої аномалії. Таким чином, більш ефективним в даній ситуації буде вважатися метод, що знайде усі три аномальні проміжки з мінімумом хибно-позитивних результатів за менший проміжок часу. Варто зазначити, що цей часовий ряд, хоча і є одновимірним, може бути складним для пошуку аномалій через високу волатильність та різну природу та характеристики аномалій, бо якщо порівняти першу і третю аномалію з другою, то друга є неявною навіть візуально для людини, таким чином є вірогідність, що методи пошуку можуть або взагалі не знаходити другий кластер, або знаходити його, але в той самий час хибно спрацьовувати в інших місцях, що залежить від чутливості використаних методів.

В якості третього набору даних було обрано багатовимірний часовий ряд з датасету Controlled Anomalies Time Series (CATS) [15]. Його перевагами є

великий обсяг даних (5 мільйонів рядків), достатня кількість характеристик та наявність позначених автором аномалій. Таким чином з характеристик було виділено 3 стовпчики, що, згідно з метаданими датасету, є основними джерелами аномалій, для їх подальшого аналізу та тестування методів пошуку аномалій (рис. 10).

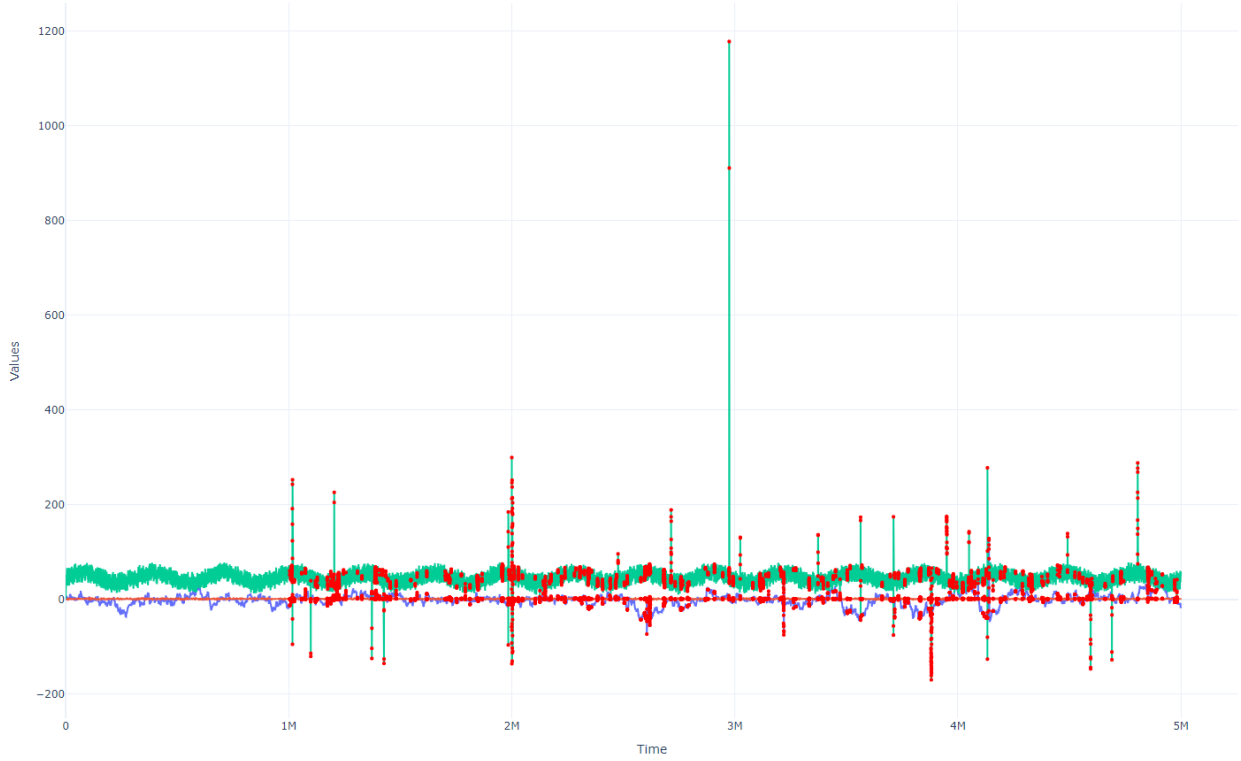

Рисунок 10. Показники сенсорів.

Було вирішено використати F1-міру для оцінки результатів методів, і її значення наведені у відповідних розділах із результатами. Проте слід зазначити, що ця метрика не є достатньо змістовною та релевантною в контексті задачі, поставленої в цій роботі. Метою дослідження є не лише кількісне оцінювання методів, а глибший аналіз їхньої поведінки та застосовності до різних типів аномалій і наборів даних. Хоча наявність істинних міток дозволяє розраховувати такі метрики, як влучність, повнота чи

F1-міра, аномалії часто мають суб’єктивну природу, а їхня оцінка залежить від доменного контексту, який числові показники не можуть повністю відобразити. Обрати методи та інструменти.

Для пошуку аномалій в часових рядах було вирішено використати такі методи:

1. Z-значення. Цей метод забезпечує простий статистичний підхід до виявлення аномалій та є ефективним для ідентифікації простих викидів в одновимірних часових рядах, де аномалії значно відхиляються від середнього значення. Він є обчислювально легким і простим у реалізації, що робить його чудовим базовим методом.
2. Виявлення точок змін (CPD): CPD зосереджується на ідентифікації структурних змін у часовому ряді, таких як зрушення трендів, середніх значень або дисперсії. Це особливо корисно для виявлення аномалій, які представляють значні переходи у даних і можуть залишитися непоміченими стандартними методами пошуку викидів.
3. Isolation Forest — це надійний некерований алгоритм виявлення аномалій, який ізолює викиди у багатовимірних даних. Його здатність моделювати багатовимірні взаємозв’язки та виявляти широкий спектр аномалій робить його придатним для складних часових рядів із залежностями між змінними.
4. LOF — це метод, заснований на щільності, який ідеально підходить для виявлення аномалій у наборах даних зі змінною щільністю. На відміну від Isolation Forest, він виявляє аномалії, які не є глобально-виразними, але є локально незвичайними, що важливо для динамічних часових рядів.
5. Prophet, хоча являє собою інструмент, а не метод, був обраний з саме ціллю підійти до пошуку аномалій більш комплексно. Цей інструмент

використовує прогнозування та добре справляється з моделюванням часових рядів із сильними сезонними або трендовими компонентами. Хоча він сам по собі не є методом виявлення аномалій, Prophet допомагає виявляти відхилення від очікуваних патернів через аналіз залишків. Це робить його цінним для розуміння базової структури часового ряду та надання контексту для виявлених аномалій.

Більшість перелічених методів була застосована за допомогою бібліотеки з відкритим кодом PyCaret для мови програмування Python. Хоча PyCaret не дозволяє здійснювати дуже тонке налаштування параметрів методів, її головною перевагою є можливість швидко змінювати підходи та проводити експерименти з різними алгоритмами. Ця бібліотека забезпечує автоматизацію основних етапів обробки даних, таких як попередня підготовка, навчання моделей та оцінювання результатів, що робить її зручною для роботи з кількома методами одночасно. Завдяки цьому PyCaret дозволив ефективно порівнювати методи виявлення аномалій і швидко адаптувати їх до різних наборів даних. Також була використана імплементація ізоляційного лісу з бібліотеки Scikit-learn, що широко використовується в пов’язаних з машинним навчанням роботах.

2.2. Результати використання різних методів на обраних даних.

У цьому підрозділі представлено результати практичного застосування різних методів виявлення аномалій на часових рядах. Для кожного методу наведено графіки часових рядів із позначеними реальними аномаліями та тими, які були виявлені відповідним підходом. Детально аналізується, як методи справлялися з поставленими задачами на різних наборах даних, включаючи їх сильні сторони, обмеження та специфіку поведінки в кожному випадку. Це

дозволяє оцінити практичну ефективність кожного методу та окреслити їхні особливості у контексті реальних задач. Таким чином:

- Червоними точками позначено істині аномалії, зазначені в документації к наборам даних.
- Синім кольором позначаються точки, що були позначені методами як аномальні (їх можна трактувати як хибно-позитивні результаті).
- Фіолетовими точками позначені ті істині аномалії, що були також знайдені використаними методами (позитивні результати).
- Зеленим пунктиром у випадках використання Facebook Prophet позначено прогноз, на основі якого потім були виявлені аномалії.

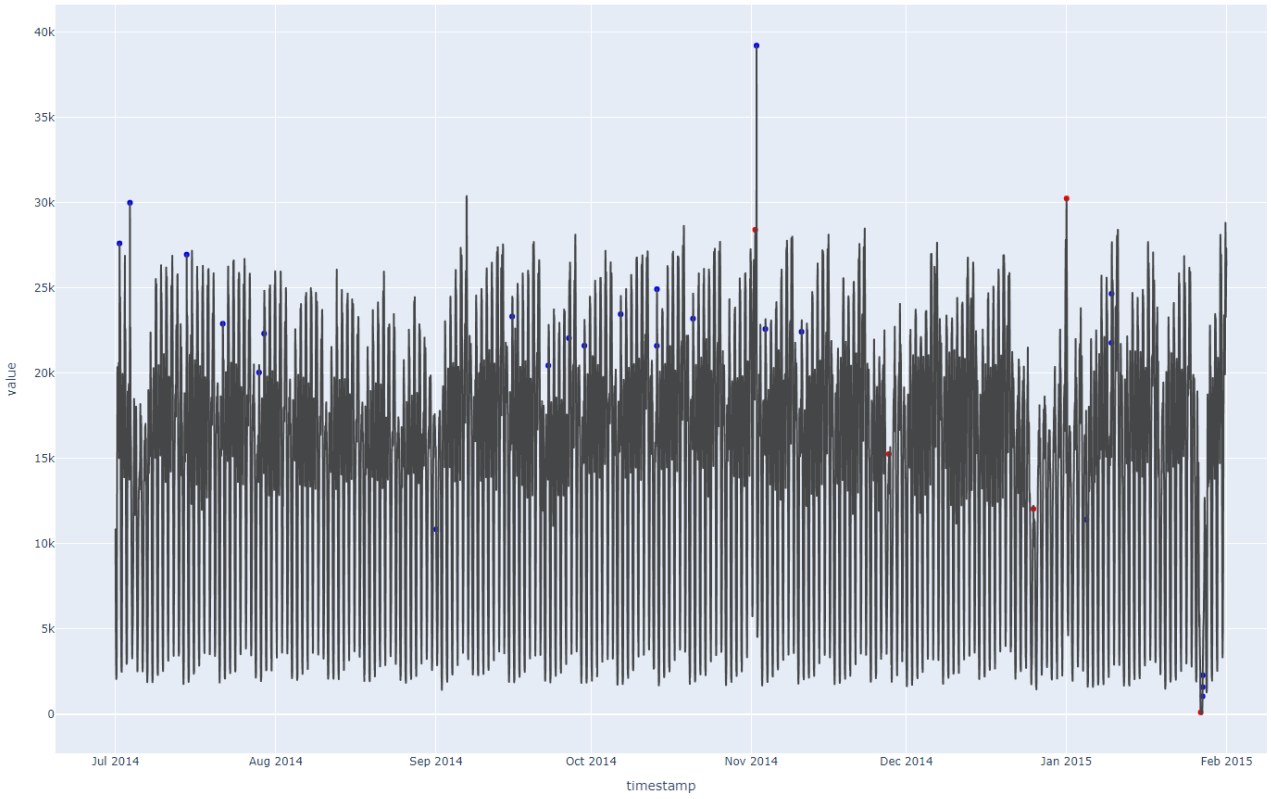

Рисунок 11. Метод Z-значень. Попит таксі Нью-Йорк Сіті.

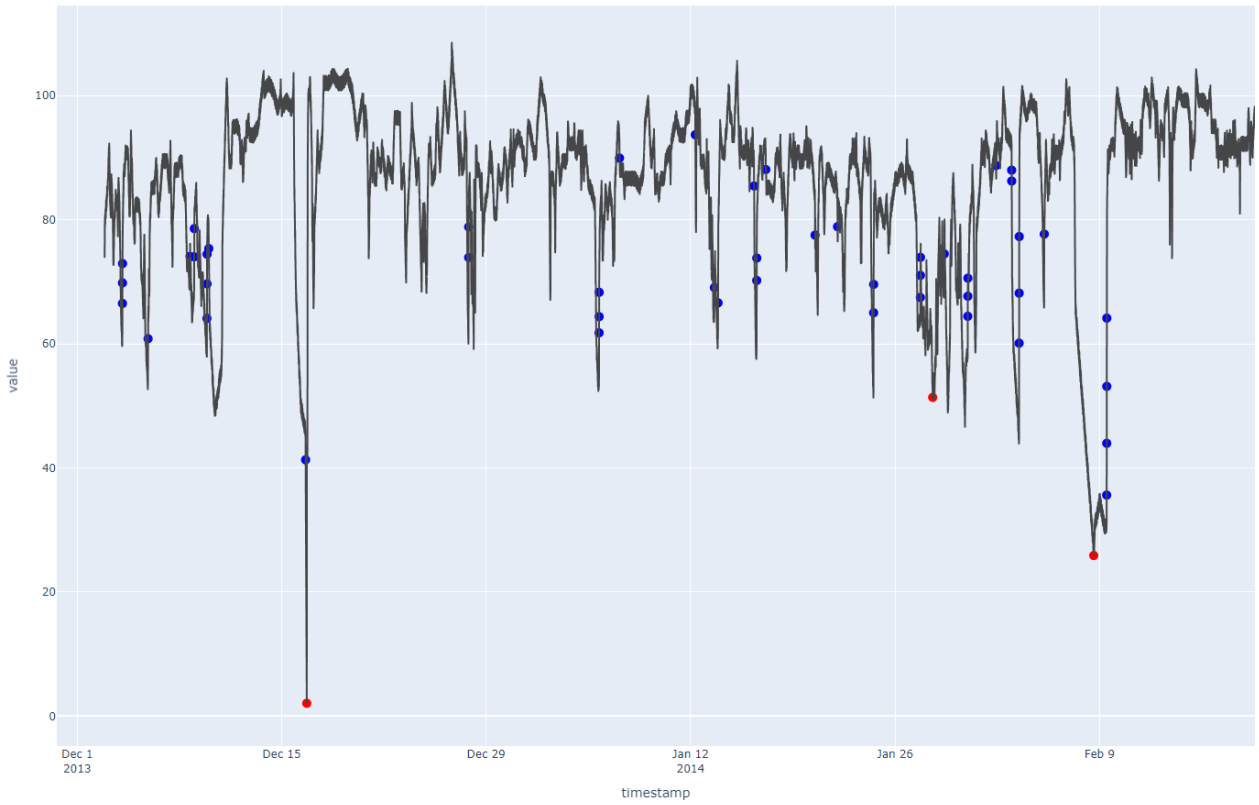

Рисунок 12. Метод Z-значень. Температура промислової машини.

Незважаючи на свою простоту та обчислювальну ефективність, метод Z-значень показав значні обмеження (рис. 11-12). Основна проблема полягала в тому, що метод не зміг виявити (або виявив пізно) швидкі й різкі зміни значень, які явно виглядали аномальними, а також помилково позначав точки, які не були аномальними судячи з контексту або візуального погляду. Такі результати пов’язані з особливостями самого методу та набору даних, зокрема, використання ковзної статистики згладжує різкі зміни, а припущення про нормальний розподіл у межах ковзного вікна може бути непридатним для цього конкретного випадку. Крім того, вибір параметрів, таких як розмір вікна чи поріг, істотно впливає на результати, тому цей метод потребує ретельного налаштування під конкретний набір даних.

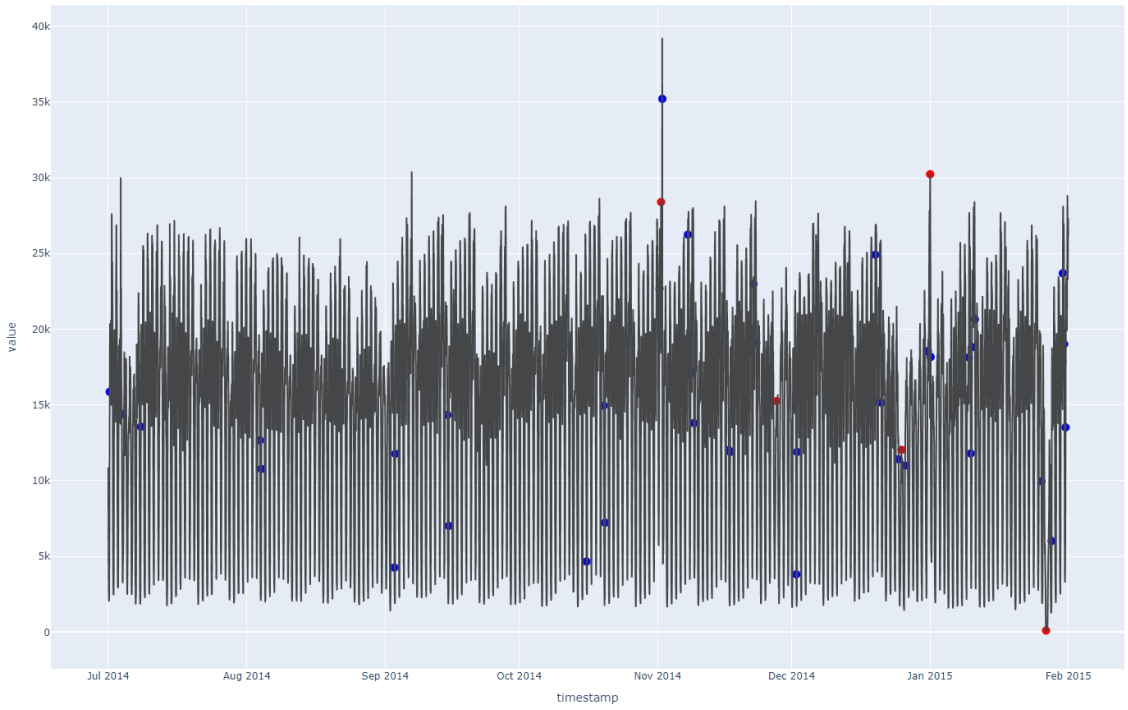

Рисунок 13. Метод виявлення точок змін (CPD). Попит таксі Нью-Йорк Сіті.

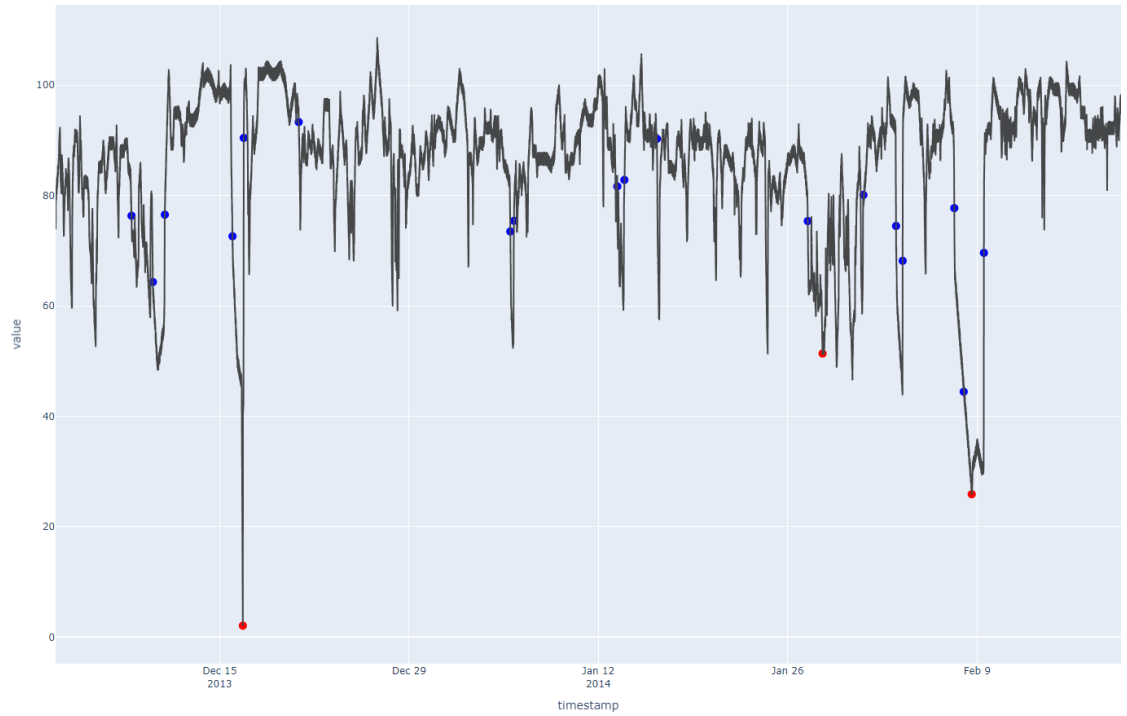

Рисунок 14. Метод виявлення точок змін (CPD). Температура промислової машини.

Метод виявлення точок змін (CPD) успішно ідентифікував початок усіх аномалій і цей результат підтверджує його здатність точно визначати моменти структурних змін, що є ключовою характеристикою CPD, проте в цих випадках метод часто позначає стабільні ділянки часового ряду як точки змін, інтерпретуючи флуктуації або шум як значущі зрушення. Ці хибні спрацьовування ускладнюють аналіз, особливо коли дані є шумними або містять багато природних коливань. Хоча ці помилки і можна корегувати зміною чутливості методу, зниження даного параметру може призвести до неспрацьовування методу там, де це необхідно (рис. 13-14).

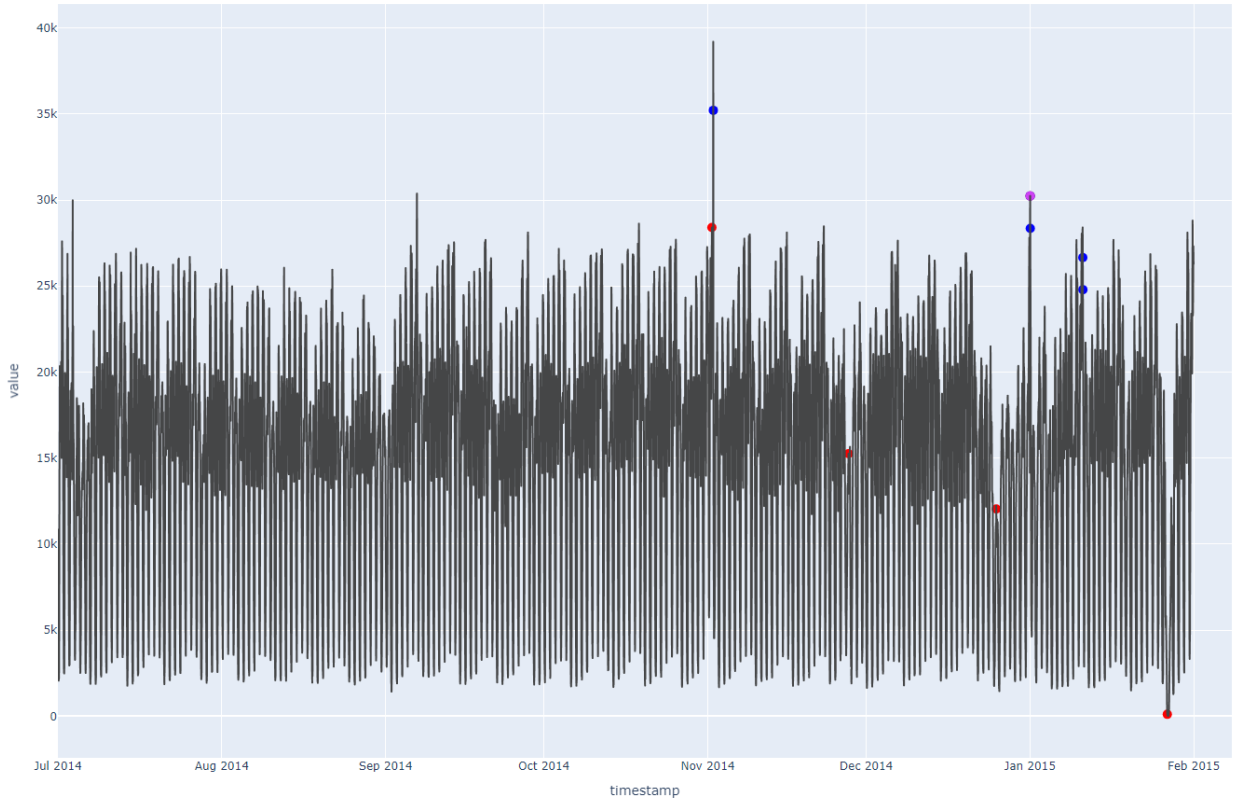

Рисунок 15. Ізоляційний ліс (Isolated Forest). Попит таксі Нью-Йорк Сіті.

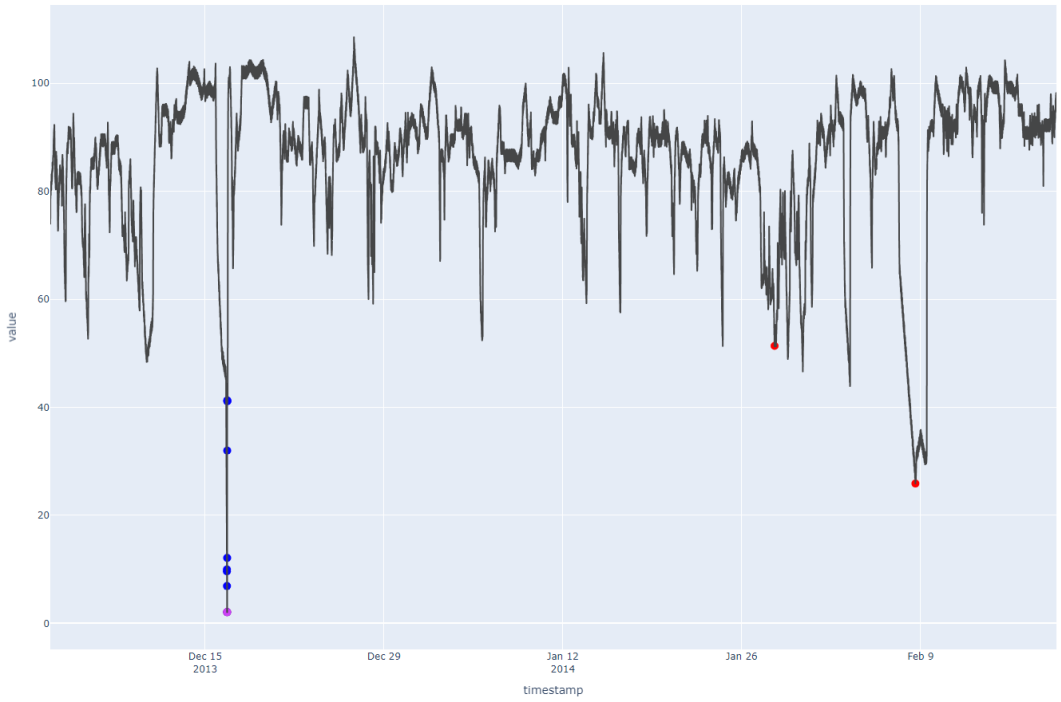

Рисунок 16. Ізоляційний ліс (Isolated Forest). Температура промислової машини.

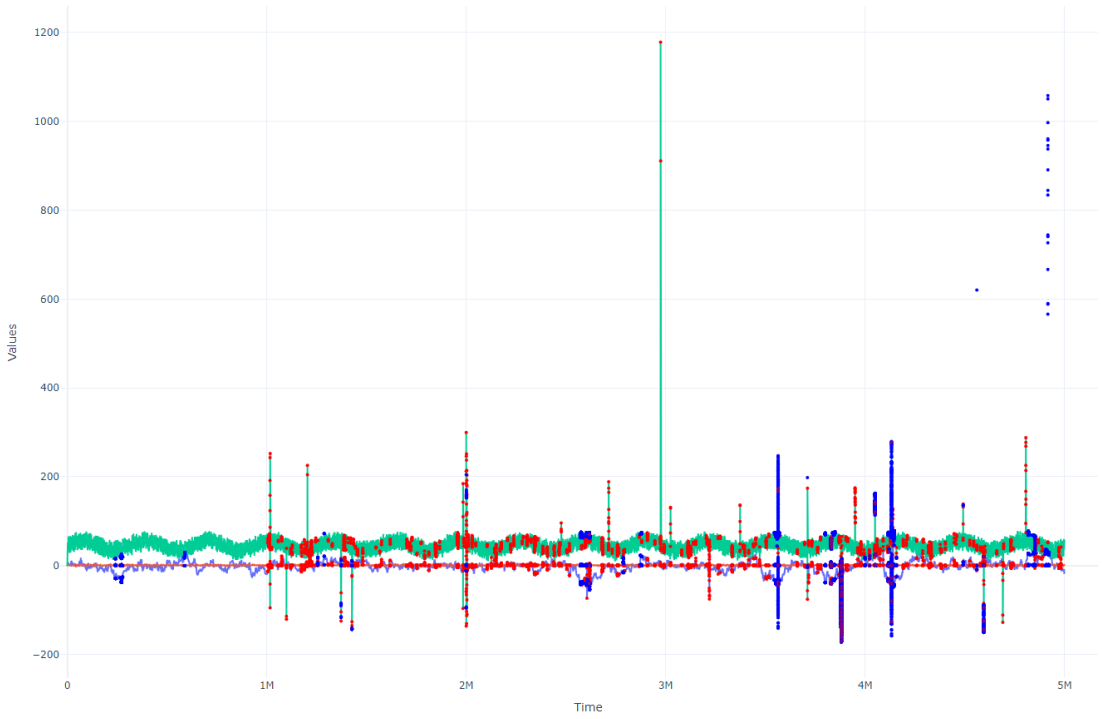

Рисунок 17. Ізоляційний ліс (Isolated Forest). Показники сенсорів.

Однією з очевидних переваг використання ізоляційного лісу є його здатність ідентифікувати області з високою варіативністю даних, таким чином багато з виявлених аномалій збігаються з різкими падіннями та переходами у наборі даних, які інтуїтивно сприймаються як ознаки ненормальної поведінки. У цьому сенсі Isolation Forest ефективно виявляє основні відхилення, підтверджуючи свою придатність для наборів даних із чіткими зрушеннями або значними викидами. Однак надійність моделі знижується, коли мова йде про аномалії, які є більш тонкими або контекстно залежними. Хоча між виявленими аномаліями (сині точки) та істинними аномаліями (червоні точки) є певні збіги, спостерігаються явні пропуски, де модель повністю ігнорує справжні аномалії. Ще однією проблемою є наявність хибно-позитивних результатів, що свідчить про те, що Isolation Forest може мати труднощі з розпізнаванням аномалій, які вимагають розуміння часових патернів або глибших взаємозв’язків між змінними. Це особливо помітно в стабільних регіонах часового ряду, де модель демонструє надмірну чутливість до дрібних відхилень. Така поведінка може вносити шум у результати та знижувати практичну корисність моделі в певних сценаріях (рис. 15-17).

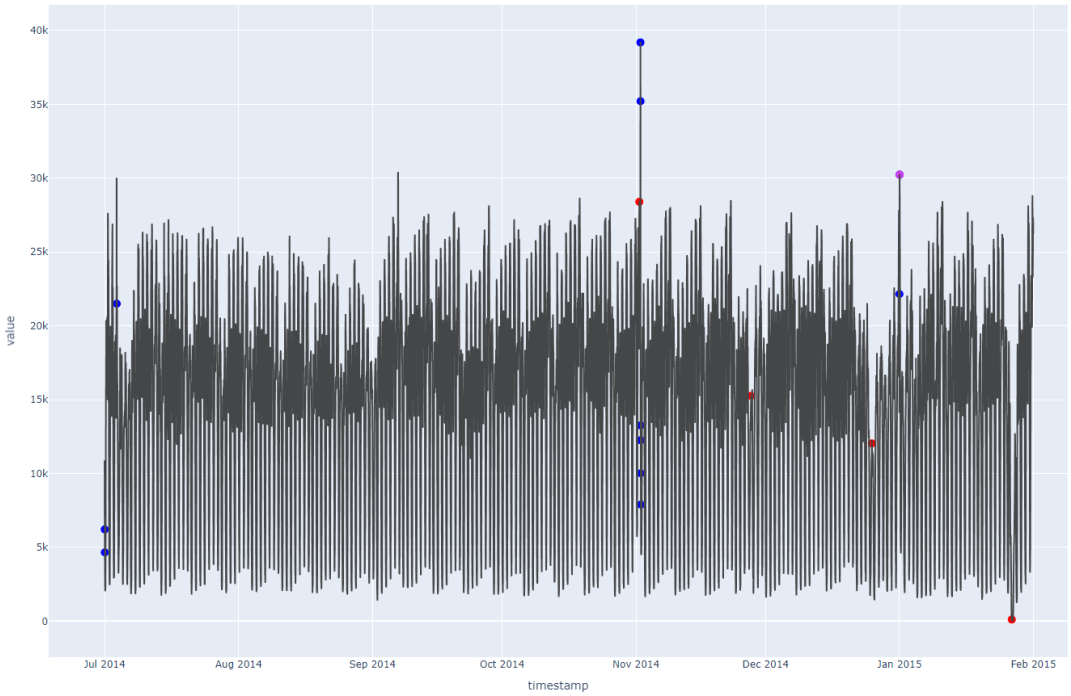

Рисунок 18. Коефіцієнт локального відхилення (LOF). Попит таксі Нью-Йорк Сіті.

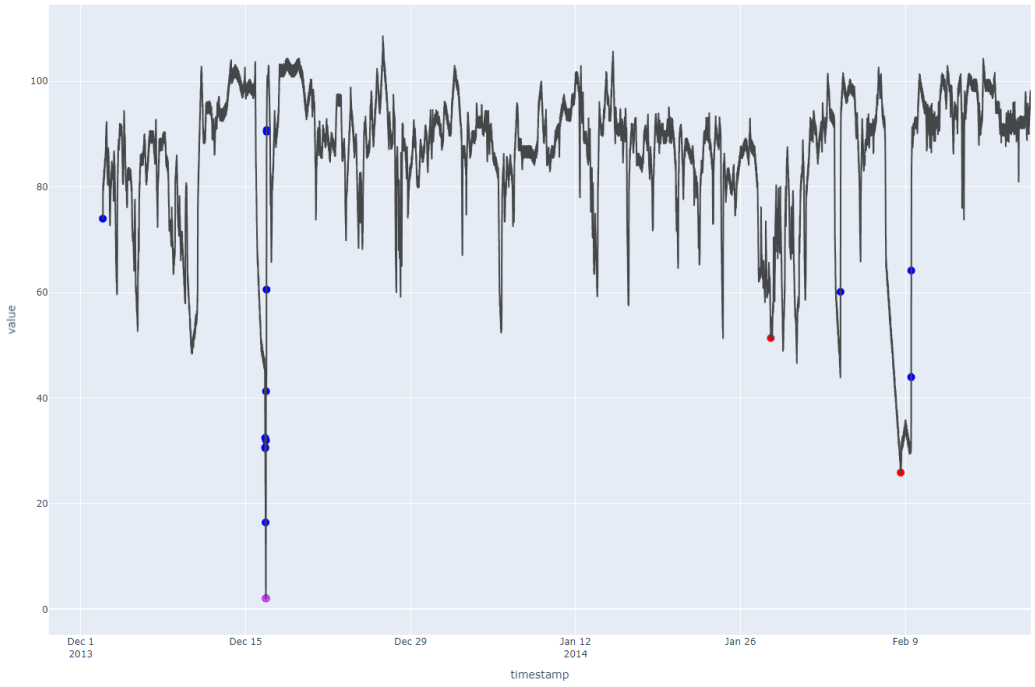

Рисунок 19. Коефіцієнт локального відхилення (LOF). Температура промислової машини.

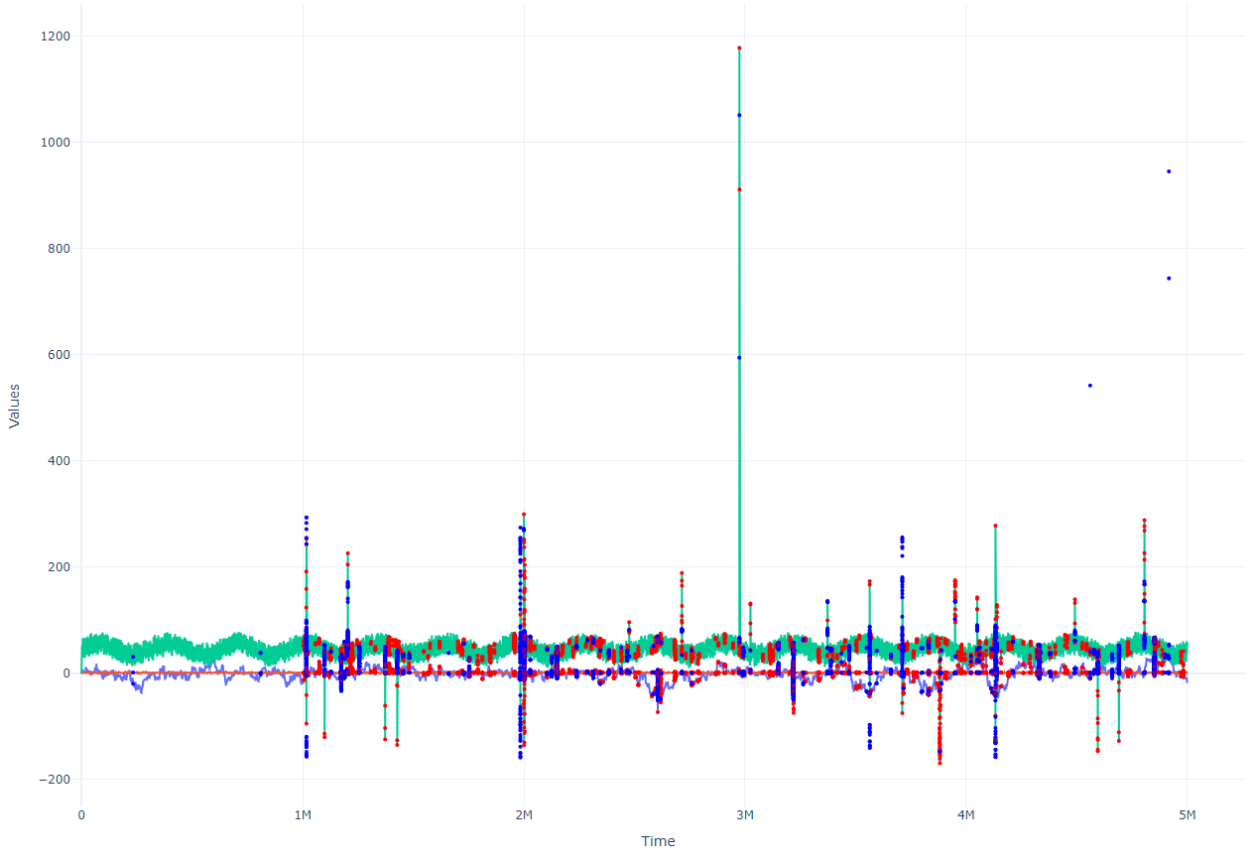

Рисунок 20. Коефіцієнт локального відхилення (LOF). Показники сенсорів.

Хоча LOF вдалося ідентифікувати деякі важливі аномальні області, його продуктивність була непослідовною. Метод зосереджується на аналізі щільності сусідніх точок, що обмежує його здатність працювати з даними, які мають значну варіативність та різкі переходи. Деякі критичні аномалії, позначені в наборі даних, залишилися непоміченими. Це, та дуже велика кількість хибно-позитивних результатів, вказує на слабкість LOF у роботі з глобальними патернами чи різкими змінами, що виходять за межі локального контексту (рис. 18-20).

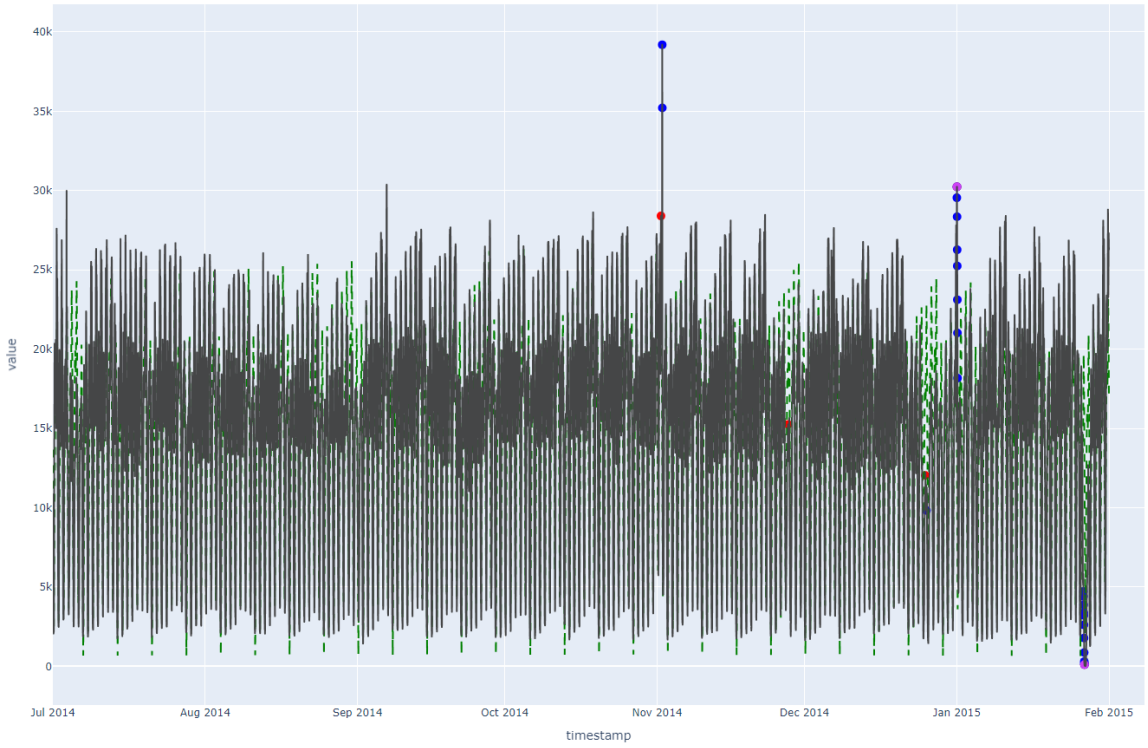

Рисунок 21. Facebook Prophet. Попит таксі Нью-Йорк Сіті.

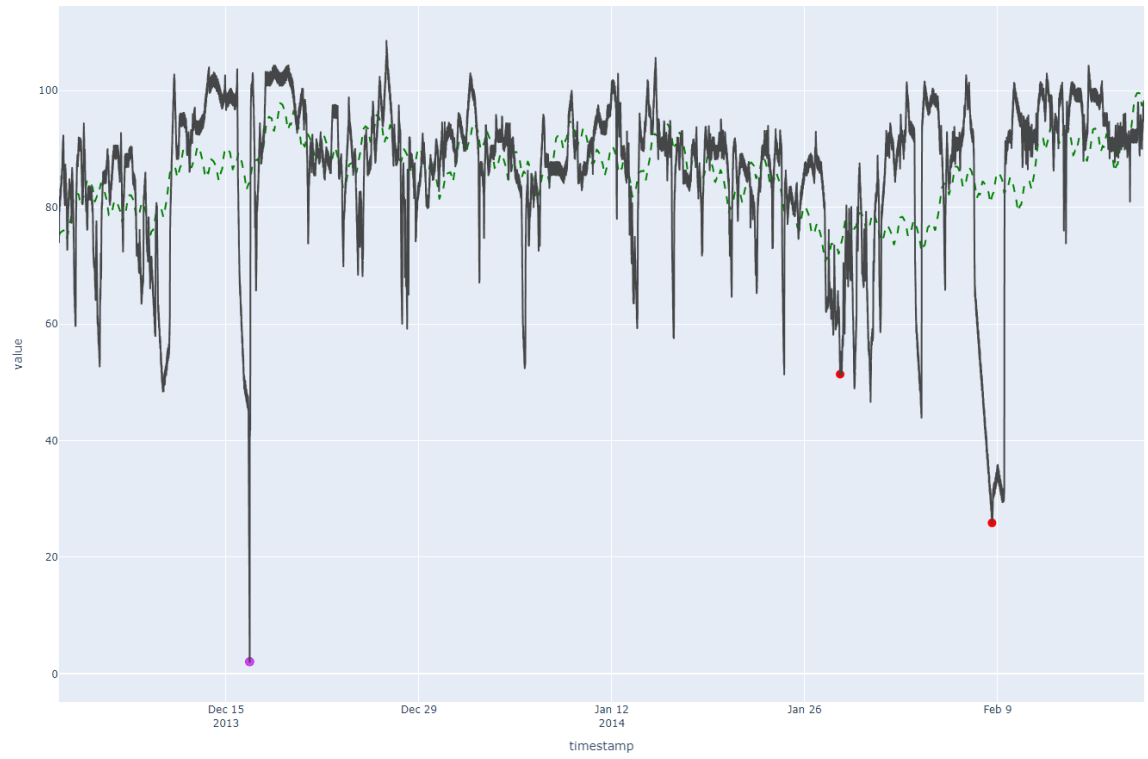

Рисунок 22. Facebook Prophet. Температура промислової машини.

Прогнозований Prophet тренд більш-менш ефективно відображає загальну поведінку набору даних, особливо в періоди з мінімальною варіативністю. Це дозволяє моделі розрізняти нормальні коливання і відхилення, які виходять за межі загального тренду. Prophet успішно ідентифікував кілька істинних аномалій, про що свідчать фіолетові маркери, де виявлені та реальні аномалії збігаються. Ці результати демонструють, що модель добре працює з великими відхиленнями або аномаліями, які тісно пов'язані із загальними трендами, однак обмеження Prophet стають очевидними в більш складних регіонах даних. Наприклад, друга аномалія в другому часовому ряді не була повністю виявлена, що свідчить про те, що прогноз моделі надмірно згладжує цю область. Подібним чином, у періоди високої варіативності модель виявляє труднощі у розпізнаванні дрібніших відхилень як аномалій, що призводить до пропусків. Така поведінка підкреслює залежність Prophet від припущень про тренд і сезонність, які можуть бути недостатньо ефективними для наборів даних без чітких патернів або з контекстно-залежними аномаліями. Незважаючи на свої переваги у роботі з великими відхиленнями та динамічними порогами, Prophet виявляється менш придатним для наборів даних, у яких домінують шум або тонкі зрушення в аномаліях.

Таблиця 1 ілюструє швидкість роботи різних методів виявлення аномалій на трьох наборах даних. Варто зазначити, що ці вимірювання не є абсолютними і мають значення лише відносно один одного. Для даних про попит на таксі в Нью-Йорку метод Z-значень показав найвищу швидкість (0.0010 секунди), що свідчить про його легкість у реалізації та обчислювальну ефективність. Change Point Detection (CPD) та Facebook Prophet були значно

повільнішими (1.4934 і 2.0680 секунди відповідно), що можна пояснити їхньою складнішою структурою розрахунків.

На наборі даних про температуру промислової машини швидкість роботи Z-значень залишалася високою (0.0010 секунди), тоді як CPD зайняв значно більше часу (25.9536 секунди) через необхідність аналізу структурних змін. Isolation Forest та LOF мали швидкість у межах 0.3107 та 0.2731 секунди, що демонструє їхню оптимізовану продуктивність навіть для складніших наборів даних.

Для показників сенсорів методи Isolation Forest і LOF потребували найбільше часу (26.6460 і 68.1021 секунди відповідно), що пояснюється складністю обчислень у цьому багатовимірному наборі. Інші методи для цього датасету не використовувались, так як вони здебільшого релевантні тільки для звичайних одновимірних часових рядів.

Метод / Датасет	Z-значення	CPD	Ізоляційний ліс	LOF	Facebook Prophet
Попит таксі Нью-Йорк Сіті	0.0010	1.4934	0.2204	0.1203	2.0680
Температура промислової машини	0.0010	25.9536	0.3107	0.2731	5.4857
Показники сенсорів	-	-	26.6460	68.1021	-

Таблиця 1. Час виконання пошуку аномалій у датасетах у секундах.

Показники середньої ефективності використаних методів наведено в таблиці 2. Результати, представлені в таблиці, демонструють влучність, повноту та F-міру для застосованих методів виявлення аномалій. Низькі значення цих метрик є очікуваними та цілком виправданими в контексті даного дослідження, так як основною метою цієї роботи було не максимізувати числові показники методів через тонке налаштування, а порівняти їх ефективність у "чистому" використанні, оцінюючи здатність методів адаптуватися до різних наборів даних без складного попереднього налаштування.

	Z-Values	CPD	Isolation Forest	LOF	Prophet
Влучність	0	0	0.3270	0.0914	0.4009
Повнота	0	0	0.1895	0.1137	0.2523
F-міра	0	0	0.1260	0.0496	0.3240

Таблиця 2. Середня ефективність методів пошуку аномалій.

Для методів, таких як Z-значення та CPD, низькі результати F-міри (0 для обох) вказують на їхню обмеженість у цьому контексті, що можна пояснити їхньою залежністю від статичних параметрів, таких як фіксовані пороги, що робить їх малоефективними у складних або шумних наборах даних.

Методи машинного навчання, такі як Isolation Forest та LOF, показали трохи кращі результати, зокрема, Isolation Forest досяг повноти 0.1895 і F-міри 0.1260. Ці результати свідчать про здатність цих методів виявляти деякі типи аномалій, але також підкреслюють їхню чутливість до параметрів, таких як частка вибраних аномалій чи розмір локального сусідства. Низька влучність,

наприклад, 0.3270 для Isolation Forest, демонструє, що метод часто позначає нормальні точки як аномальні, що може бути зумовлено відсутністю попередньої адаптації параметрів до даного набору.

Prophet, який добре працює з трендами та сезонністю, показав найкращий баланс серед F-міри (0.3240), хоча й ці значення залишаються низькими. Це підтверджує його обмеження у виявленні тонких аномалій або в роботі з даними, які не мають чітких трендових структур. Загалом, простежується тенденція покращення результатів пошуку аномалій з ускладненням використаного методу: прості статистичні методи вимагають сильної обробки даних та комбінування з іншими методами, в той час як комплексні інструменти як Prophet демонструють кращі результати навіть без попередньої підготовки.

Висновки до розділу 2

У практичній частині цієї роботи було протестовано кілька методів виявлення аномалій на різних наборах даних, що дозволило продемонструвати їх сильні сторони та обмеження в різних сценаріях. Стало очевидним, що жоден метод не може надійно виявляти всі типи аномалій. Успіх виявлення аномалій значною мірою залежить від якості підготовки даних, включаючи інженерію ознак і попередню обробку, а також від ретельного поєднання методів. Зрештою, результати підкреслюють, що вибір відповідного методу та поєднання взаємодоповнюючих технік є ключовими для надійного виявлення аномалій, при цьому підготовка даних відіграє вирішальну роль у підвищенні продуктивності.

ВИСНОВКИ

Це дослідження охопило як теоретичні, так і практичні аспекти виявлення аномалій у часових рядах, надаючи огляд сучасних методів і підходів до цієї задачі.

У теоретичній частині було закладено фундаментальне розуміння часових рядів, їх ключових характеристик, таких як тренди та сезонність, а також класифікації аномалій на три основні типи: точкові, контекстуальні та структурні. Точкові аномалії охоплюють окремі відхилення, контекстуальні вимагають урахування локальних умов, тоді як структурні вказують на значущі зміни у структурі даних, наприклад, зсув тренду чи зміну дисперсії. Було розглянуто широкий спектр методів для виявлення аномалій, від традиційних статистичних підходів, таких як Z-значення, до сучасних алгоритмів машинного навчання і методів глибокого навчання. Також увагу було приділено сучасним інструментам та платформам для аналізу часових рядів та пошуку аномалій в них. Це створило теоретичну основу для оцінки різних методів і їх застосування у реальних сценаріях.

У практичній частині дослідження було проведено експерименти із застосуванням кількох методів виявлення аномалій на різних наборах даних. Результати підкреслили, що жоден метод не є універсальним і не може гарантувати ідеальне виявлення аномалій у всіх сценаріях.

Ключовим висновком є важливість підготовки даних перед застосуванням будь-якого методу. Інженерія ознак, нормалізація та врахування доменної експертизи виявилися критичними для досягнення успіху.

Дослідження також наголошує на важливості ітеративного процесу: тестування, оцінювання та адаптація методів залежно від специфіки набору даних є основою успішного виявлення аномалій. Цей підхід дозволяє гнучко

реагувати на різноманітність реальних задач, де кожен набір даних може вимагати унікального підходу.

Дослідження продемонструвало, що виявлення аномалій у часових рядах є багатогранним завданням, яке вимагає як теоретичних знань, так і практичних навичок. Воно підкреслило необхідність адаптивного, інтегративного підходу, який враховує різноманітність характеру даних, типів аномалій і поставлених цілей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1-58..

[2] Cook, D. J., & Das, S. K. (2007). How smart are our environments? An updated look at the state of the art. Pervasive and Mobile Computing, 3(2), 53-73..

[3] Brockwell, P. J., & Davis, R. A. (2016). Introduction to Time Series and Forecasting (3rd ed.). Springer..

[4] Aggarwal, C. C. (2017). Outlier Analysis (2nd ed.). Springer..

[5] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control (5th ed.). Wiley..

[6] Chatfield, C. (2003). The Analysis of Time Series: An Introduction (6th ed.). CRC Press..

[7] Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice (2nd ed.). OTexts..

[8] Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. Neurocomputing, 262, 134-147..

[9] Pimentel, M. A. F., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. Signal Processing, 99, 215-249..

[10] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19-31..

[11] О.Ю. Лихач,М.Л. Угрюмов , Д.О. Шевченко, С.І. Шматков (2022), Методи виявлення викидів в пробних вибірках при управлінні процесами в системах за станом, Вісник Харківського національного університету імені В. Н. Каразіна, випуск 53.

[12] Marsland, S. (2014). Machine Learning: An Algorithmic Perspective (2nd ed.). CRC Press..

[13] Taylor, S. J., & Letham, B. (2018). Forecasting at scale. The American Statistician, 72(1), 37-45..

[14] Numenta Anomaly Benchmark (NAB). URL: https://github.com/numenta/NAB (дата звернення: 25/10/2024).

[15] Controlled Anomalies Time Series (CATS) Dataset. URL: https://www.kaggle.com/datasets/patrickfleith/controlled-anomalies-time-series-dataset/ (дата звернення: 2/11/2024).

ДОДАТКИ

Додаток 1

Приклад коду пошуку аномалій в наборі даних різними методами

Файл anomalydetection-taxi.py

```
import pandas as pd
import plotly.express as px
from prophet import Prophet
from pycaret.anomaly import *
import ruptures as rpt
import time
import os

os.environ["CUDA_VISIBLE_DEVICES"] = "-1"

data = pd.read_csv('nyc_taxi.csv')
print(data.head())

data['timestamp'] = pd.to_datetime(data['timestamp'])

anomaly_points = [
    "2014-11-01 19:00:00.000000",
    "2014-11-27 15:30:00.000000",
    "2014-12-25 15:00:00.000000",
    "2015-01-01 01:00:00.000000",
    "2015-01-27 00:00:00.000000"
```

```
]

anomaly_points = data[data['timestamp'].isin(anomaly_points)]

data['hour'] = data['timestamp'].dt.hour
data['day_of_week'] = data['timestamp'].dt.dayofweek
data['is_weekend'] = (data['day_of_week'] >= 5).astype(int)
data['month'] = data['timestamp'].dt.month
data['year'] = data['timestamp'].dt.year

data['rolling_mean'] = data['value'].rolling(window=5).mean()
data['rolling_std'] = data['value'].rolling(window=5).std()
data['lag_1'] = data['value'].shift(1)

print(data.head())

######################

fig = px.line(data, x="timestamp", y="value", title='Попит таксі Нью-Йорк Сіті')

fig.data[0].line.color = '#454647'

fig.add_scatter(
    x=anomaly_points['timestamp'],
    y=anomaly_points['value'],
    mode='markers',
    marker=dict(size=10, color='red'),
```

```
    name='Real Anomalies'
)

fig.show()

######################

columns_to_use = ['value', 'hour', 'day_of_week', 'is_weekend', 'month', 'year',
'rolling_mean', 'rolling_std', 'lag_1']
pycaret_data = data[columns_to_use]

anomaly_setup = setup(pycaret_data, session_id=123)



def calculate_metrics(real_anomalies, detected_anomalies):
    real_anomalies_set = set(real_anomalies['timestamp'])
    detected_anomalies_set = set(detected_anomalies['timestamp'])

    tp = len(real_anomalies_set & detected_anomalies_set)
    fp = len(detected_anomalies_set - real_anomalies_set)
    fn = len(real_anomalies_set - detected_anomalies_set)

    precision = tp / (tp + fp) if (tp + fp) > 0 else 0
    recall = tp / (tp + fn) if (tp + fn) > 0 else 0
    f1 = (2 * precision * recall) / (precision + recall) if (precision + recall) > 0 else 0

    return precision, recall, f1
```

```
############################################################
#Z-VALUES
############################################################
start_time = time.time()

window = 20
data['rolling_mean'] = data['value'].rolling(window=window).mean()
data['rolling_std'] = data['value'].rolling(window=window).std()

threshold = 3
data['is_anomaly'] = (data['value'] - data['rolling_mean']).abs() > (threshold *
data['rolling_std'])

end_time = time.time()

detected_anomalies = data.loc[data['is_anomaly'], ['timestamp', 'value']]
intersection = pd.merge(
    anomaly_points,
    detected_anomalies,
    on=['timestamp', 'value']
)

fig = px.line(data, x="timestamp", y="value", title='[Z-Значення] Попит таксі
Нью-Йорк Сіті')

fig.data[0].line.color = '#454647'
```

```
fig.add_scatter(
    x=anomaly_points['timestamp'],
    y=anomaly_points['value'],
    mode='markers',
    marker=dict(size=10, color='red'),
    name='Real Anomalies'
)

fig.add_scatter(
    x=detected_anomalies['timestamp'],
    y=detected_anomalies['value'],
    mode='markers',
    marker=dict(size=10, color='blue'),
    name='Detected Anomalies'
)

fig.add_scatter(
    x=intersection['timestamp'],
    y=intersection['value'],
    mode='markers',
    marker=dict(size=10, color='#ce42f5'),
    name='Intersection (True Positives)'
)

fig.show()
```

```
execution_time = end_time - start_time
print(f"Execution Time: {execution_time:.4f} seconds")

zval_precision, zval_recall, zval_f1 = calculate_metrics(anomaly_points,
detected_anomalies)

############################################################
#CPD
############################################################
start_time = time.time()

values = data['value'].values

model = "l2"

algo = rpt.Pelt(model=model).fit(values)

penalty = 999000000
change_points = algo.predict(pen=penalty)

change_points_timestamps = data['timestamp'].iloc[change_points[:-1]].values

end_time = time.time()

detected_anomalies = pd.DataFrame({
    'timestamp': change_points_timestamps,
    'value': data['value'].iloc[change_points[:-1]].values
```

```
})
intersection = pd.merge(
    anomaly_points,
    detected_anomalies,
    on=['timestamp', 'value']
)

fig = px.line(data, x="timestamp", y="value", title='[CDP l2] Попит таксі Нью-Йорк Сіті')

fig.data[0].line.color = '#454647'

fig.add_scatter(
    x=anomaly_points['timestamp'],
    y=anomaly_points['value'],
    mode='markers',
    marker=dict(size=10, color='red'),
    name='Anomalies'
)

fig.add_scatter(
    x=detected_anomalies['timestamp'],
    y=detected_anomalies['value'],
    mode='markers',
    marker=dict(size=10, color='blue'),
    name='Detected Anomalies'
)
```

```
fig.add_scatter(
    x=intersection['timestamp'],
    y=intersection['value'],
    mode='markers',
    marker=dict(size=10, color='#ce42f5'),
    name='Intersection (True Positives)'
)

fig.show()

execution_time = end_time - start_time
print(f"Execution Time: {execution_time:.4f} seconds")

cpd_precision, cpd_recall, cpd_f1 = calculate_metrics(anomaly_points,
detected_anomalies)

############################################################
#ISOLATION FOREST
############################################################
start_time = time.time()

model = create_model('iforest', fraction = 0.05)

model_results = assign_model(model)

data['Anomaly'] = model_results['Anomaly']
```

```
end_time = time.time()

detected_anomalies = data[data['Anomaly'] == 1]
intersection = pd.merge(
    anomaly_points,
    detected_anomalies,
    on=['timestamp', 'value']
)

fig = px.line(data, x="timestamp", y="value", title='[IForest] Попит таксі Нью-Йорк Сіті')

fig.data[0].line.color = '#454647'

fig.add_scatter(
    x=anomaly_points['timestamp'],
    y=anomaly_points['value'],
    mode='markers',
    marker=dict(size=10, color='red'),
    name='Anomalies'
)

fig.add_scatter(
    x=detected_anomalies['timestamp'],
    y=detected_anomalies['value'],
    mode='markers',
```

```
    marker=dict(size=10, color='blue'),
    name='Detected Anomalies'
)
fig.add_scatter(
    x=intersection['timestamp'],
    y=intersection['value'],
    mode='markers',
    marker=dict(size=10, color='#ce42f5'),
    name='Intersection (True Positives)'
)

fig.show()

execution_time = end_time - start_time
print(f"Execution Time: {execution_time:.4f} seconds")

iforest_precision, iforest_recall, iforest_f1 = calculate_metrics(anomaly_points,
detected_anomalies)


############################################################
#LOF
############################################################
start_time = time.time()

model = create_model('lof', fraction = 0.01)
```

```
model_results = assign_model(model)

data['Anomaly'] = model_results['Anomaly']
end_time = time.time()

detected_anomalies = data[data['Anomaly'] == 1]
intersection = pd.merge(
    anomaly_points,
    detected_anomalies,
    on=['timestamp', 'value']
)

fig = px.line(data, x="timestamp", y="value", title='[LOF] Попит таксі Нью-Йорк Сіті')

fig.data[0].line.color = '#454647'

fig.add_scatter(
    x=anomaly_points['timestamp'],
    y=anomaly_points['value'],
    mode='markers',
    marker=dict(size=10, color='red'),
    name='Anomalies'
)

fig.add_scatter(
    x=detected_anomalies['timestamp'],
```

```
    y=detected_anomalies['value'],
    mode='markers',
    marker=dict(size=10, color='blue'),
    name='Detected Anomalies'
)
fig.add_scatter(
    x=intersection['timestamp'],
    y=intersection['value'],
    mode='markers',
    marker=dict(size=10, color='#ce42f5'),
    name='Intersection (True Positives)'
)

fig.show()

execution_time = end_time - start_time
print(f"Execution Time: {execution_time:.4f} seconds")

lof_precision, lof_recall, lof_f1 = calculate_metrics(anomaly_points,
detected_anomalies)

############################################################
#PROPHET
############################################################
start_time = time.time()

prophet_data = data[['timestamp', 'value']].rename(columns={'timestamp': 'ds',
```

```
'value': 'y'})

model = Prophet(
    seasonality_mode="multiplicative",
    changepoint_prior_scale=0.05
)
model.fit(prophet_data)

future = model.make_future_dataframe(periods=0)
forecast = model.predict(future)

prophet_data['yhat'] = forecast['yhat']
prophet_data['residuals'] = prophet_data['y'] - prophet_data['yhat']

threshold = 2.6 * prophet_data['residuals'].std()
prophet_data['is_anomaly'] = abs(prophet_data['residuals']) > threshold
end_time = time.time()

detected_anomalies = prophet_data[prophet_data['is_anomaly']]
detected_anomalies = detected_anomalies.rename(columns={'ds': 'timestamp', 'y':
'value'})
intersection = pd.merge(
    anomaly_points,
    detected_anomalies,
    on=['timestamp', 'value']
)
```

```
forecast = forecast.rename(columns={'ds': 'timestamp', 'yhat': 'forecast'})

fig = px.line(data, x="timestamp", y="value", title='[Prophet] Попит таксі Нью-Йорк Сіті')

fig.data[0].line.color = '#454647'

fig.add_scatter(
    x=anomaly_points['timestamp'],
    y=anomaly_points['value'],
    mode='markers',
    marker=dict(size=10, color='red'),
    name='Anomalies'
)

fig.add_scatter(
    x=detected_anomalies['timestamp'],
    y=detected_anomalies['value'],
    mode='markers',
    marker=dict(size=10, color='blue'),
    name='Detected Anomalies'
)

fig.add_scatter(
    x=forecast['timestamp'],
    y=forecast['forecast'],
    mode='lines',
```

```
    line=dict(color='green', dash='dash'),
    name='Forecast'
)
fig.add_scatter(
    x=intersection['timestamp'],
    y=intersection['value'],
    mode='markers',
    marker=dict(size=10, color='#ce42f5'),
    name='Intersection (True Positives)'
)

fig.show()

execution_time = end_time - start_time
print(f"Execution Time: {execution_time:.4f} seconds")

prophet_precision, prophet_recall, prophet_f1 = calculate_metrics(anomaly_points,
detected_anomalies)

##############################
# Results
##############################

print(f"Z-Values - Precision: {zval_precision:.4f}, Recall: {zval_recall:.4f}, F1
Score: {zval_f1:.4f}")
print(f"CPD - Precision: {cpd_precision:.4f}, Recall: {cpd_recall:.4f}, F1 Score:
{cpd_f1:.4f}")
```

```
print(f"Isolation Forest - Precision: {iforest_precision:.4f}, Recall:
{iforest_recall:.4f}, F1 Score: {iforest_f1:.4f}")
print(f"LOF - Precision: {lof_precision:.4f}, Recall: {lof_recall:.4f}, F1 Score:
{lof_f1:.4f}")
print(f"Prophet - Precision: {prophet_precision:.4f}, Recall: {prophet_recall:.4f},
F1 Score: {prophet_f1:.4f}")
```