

Факультет Навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

2026 p.

Харківський національний університет радіоелектроніки

Факультет Навчально-науковий центр заочної форми навчання

Кафедра Штучного інтелекту

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Штучний інтелект
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Грибу Іллі Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтелектуальної системи виявлення аномалій у даних на основі методів машинного навчання

затверджена наказом університету від 18 травня 20 26 р. № 75Стз

2. Термін подання здобувачем роботи до екзаменаційної комісії 10 липня 20 26 р.

3. Вихідні дані до роботи наукові публікації з тематики виявлення аномалій у часових рядах та методів машинного навчання, документація бібліотек Python (TensorFlow, Keras, NumPy, Pandas, Scikit-learn, Streamlit), набір даних Numenta Anomaly Benchmark (NAB) для навчання, тестування та оцінювання моделей виявлення аномалій

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі дослідження

2) Методи виявлення аномалій у даних

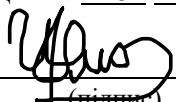
3) Розробка інтелектуальної системи виявлення аномалій у часових рядах

4) Експериментальне дослідження ефективності системи

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	18.05.2026	виконано
2	Аналіз предметної галузі та постановка задачі	18.05.2026–22.05.2026	виконано
3	Теоретичне дослідження методів виявлення аномалій у часових рядах	23.05.2026–28.05.2026	виконано
4	Аналіз наборів даних, вибір та підготовка даних	29.05.2026–03.06.2026	виконано
5	Розробка та навчання моделей виявлення аномалій	04.06.2026–16.06.2026	виконано
6	Проведення експериментальних досліджень та оцінювання результатів	17.06.2026–24.06.2026	виконано
7	Розробка інтерфейсу користувача та інтеграція компонентів системи	25.06.2026–27.06.2026	виконано
8	Оформлення пояснювальної записки	28.06.2026–01.07.2026	виконано
9	Нормоконтроль	02.07.2026–03.07.2026	виконано
10	Попередній захист	04.07.2026	виконано
11	Рецензування	06.07.2026	виконано
12	Захист перед ЕК	10.07.2026	

Дата видачі завдання 18 травня 2026 р.

Здобувач 
(підпис)

Керівник роботи _____
(підпис)

доц. Олег ЗОЛОТУХІН
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 98 с., 57 рис., 1 дод., 31 джерело.

ВИЯВЛЕННЯ АНОМАЛІЙ, ГЛИБОКЕ НАВЧАННЯ, ІЗОЛЯЦІЙНИЙ ЛІС, МАШИННЕ НАВЧАННЯ, STREAMLIT, TEMPORAL TRANSFORMER, ЧАСОВІ РЯДИ, LSTM AUTOENCODER, NUMENTA ANOMALY BENCHMARK, PR-AUC.

Об'єкт дослідження – процес виявлення аномалій у часових рядах.

Предмет дослідження – методи виявлення аномалій у часових рядах із використанням алгоритмів машинного навчання та нейронних мереж.

Мета роботи – розробка інтелектуальної системи виявлення аномалій у даних на основі методів машинного навчання.

Методи дослідження – алгоритм Isolation Forest, моделі LSTM Autoencoder та Transformer Autoencoder, методи попередньої обробки та нормалізації даних, віконне представлення часових рядів, перцентильний підхід до визначення порогових значень, метрики оцінювання якості моделей (Window Recall, PR-AUC, кількість хибних спрацювань).

У роботі розроблено інтелектуальну систему виявлення аномалій у часових рядах, що забезпечує автоматизовану обробку даних, навчання моделей, оцінювання результатів та візуалізацію виявлених аномалій. Проведено порівняльний аналіз трьох підходів до виявлення аномалій: Isolation Forest, LSTM Autoencoder та Transformer Autoencoder. Досліджено вплив набору ознак, довжини часового вікна та значення перцентильного порогу на якість виявлення аномалій. Для перевірки працездатності розробленої системи проведено експериментальні дослідження на часових рядах набору даних NAB та виконано візуалізацію результатів виявлення аномалій.

ABSTRACT

Bachelor's thesis contains: 98 pp., 57 fig., 1 ann., 31 references.

ANOMALY DETECTION, DEEP LEARNING, ISOLATION FOREST, LSTM AUTOENCODER, MACHINE LEARNING, NUMENTA ANOMALY BENCHMARK, PR-AUC, STREAMLIT, TEMPORAL TRANSFORMER, TIME SERIES.

The object of research is the process of anomaly detection in time series.

The subject of research is methods for anomaly detection in time series using machine learning algorithms and neural networks.

The purpose of the study is to develop an intelligent system for anomaly detection in data based on machine learning methods.

Research methods include the Isolation Forest algorithm, LSTM Autoencoder and Transformer Autoencoder models, data preprocessing and normalization techniques, sliding window representation of time series, a percentile-based approach for threshold selection, and model evaluation metrics (Window Recall, PR-AUC, and the number of false positives).

The study presents the development of an intelligent system for anomaly detection in time series that provides automated data processing, model training, performance evaluation, and visualization of detected anomalies. A comparative analysis of three anomaly detection approaches, namely Isolation Forest, LSTM Autoencoder, and Transformer Autoencoder, was conducted. The influence of feature sets, time window length, and percentile threshold values on anomaly detection performance was investigated. To verify the effectiveness of the developed system, experimental studies were carried out using time series from the Numenta Anomaly Benchmark (NAB) dataset, and the results of anomaly detection were visualized.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі та постановка задачі дослідження.....	8
1.1 Особливості виявлення аномалій у часових рядах	8
1.2 Аналіз існуючих підходів та рішень.....	13
1.3 Постановка задачі дослідження	27
2 Методи виявлення аномалій у даних.....	29
2.1 Статистичні підходи до виявлення аномалій.....	29
2.2 Методи машинного навчання для виявлення аномалій.....	35
2.3 Нейромережіві підходи до виявлення аномалій	42
2.4 Метрики оцінювання ефективності методів виявлення аномалій.....	49
3 Розробка інтелектуальної системи виявлення аномалій у часових рядах	55
3.1 Опис набору даних та підготовка даних	55
3.2 Реалізація методів виявлення аномалій.....	60
3.2.1 Реалізація Isolation Forest.....	60
3.2.2 Реалізація LSTM Autoencoder	62
3.2.3 Реалізація Transformer Autoencoder	64
3.3 Реалізація програмного інтерфейсу системи	66
4 Експериментальне дослідження ефективності системи	71
4.1 Дослідження ефективності Isolation Forest	71
4.2 Дослідження ефективності LSTM Autoencoder	76
4.3 Дослідження ефективності Transformer Autoencoder.....	81
4.4 Порівняльний аналіз моделей	84
4.5 Демонстрація роботи програмного прототипу	88
Висновки.....	92
Перелік джерел посилання	94
Додаток А Відомість кваліфікаційної роботи.....	98

ВСТУП

У сучасних інформаційних системах обсяги даних, що генеруються різноманітними джерелами, постійно зростають. Значна частина таких даних має часову природу, оскільки формується у вигляді послідовності вимірювань у часі. До таких даних належать показники роботи промислового обладнання, мережеві метрики, параметри функціонування серверів, а також інші сигнали, що відображають стан складних технічних систем.

Аналіз часових рядів є важливим інструментом для моніторингу стану систем і своєчасного реагування на відхилення від нормального режиму роботи. Особливу роль у цьому процесі відіграє задача виявлення аномалій, яка полягає у знаходженні нетипових значень або ділянок часового ряду, що можуть свідчити про збої, помилки вимірювання або інші небажані події.

На відміну від задач класифікації, де кожному об'єкту заздалегідь відповідає певний клас, у задачі виявлення аномалій такі відхилення часто не мають чітко визначеної структури та можуть проявлятися у різних формах. Це ускладнює побудову ефективних моделей та потребує використання спеціалізованих методів аналізу даних.

Складність задачі також обумовлена наявністю шуму, сезонності, трендів та інших властивостей часових рядів, які можуть маскувати аномальні прояви або, навпаки, створювати хибні спрацювання. У зв'язку з цим актуальним є дослідження сучасних методів виявлення аномалій, зокрема підходів, що базуються на методах машинного навчання та глибокого навчання.

Застосування таких методів дозволяє автоматизувати процес аналізу даних, враховувати складні залежності у часових рядах та підвищувати точність виявлення аномальних ситуацій. Водночас різні підходи можуть демонструвати різну ефективність залежно від характеру даних, що зумовлює необхідність їх порівняльного аналізу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Зростання обсягів даних, що формуються у вигляді часових послідовностей, зумовлює необхідність їх ефективного аналізу з метою виявлення відхилень у поведінці досліджуваних процесів. Такі дані широко використовуються для моніторингу технічних систем, аналізу навантаження на обчислювальні ресурси, контролю параметрів навколишнього середовища та в інших прикладних задачах. У цьому контексті особливої актуальності набуває задача виявлення аномалій, яка дозволяє своєчасно ідентифікувати нетипові зміни у даних та реагувати на потенційно небезпечні або небажані події.

1.1 Особливості виявлення аномалій у часових рядах

Часовий ряд визначається як впорядкована у часі послідовність значень деякої змінної, у якій кожне спостереження відповідає певному моменту або інтервалу часу. Такий тип даних використовується для опису процесів, що змінюються у часі, зокрема показників датчиків, метрик функціонування серверів, фінансових індикаторів та інших динамічних систем [1].

Формально часовий ряд може бути представлений у вигляді послідовності

$$X = \{x_t\}_{t=1}^T, \quad (1.1)$$

де t – момент часу;

x_t – значення досліджуваного показника у цей момент.

Такий запис підкреслює, що кожне значення пов'язане з конкретною часовою позицією та не може розглядатися поза нею.

На відміну від звичайних наборів даних, у часових рядах порядок елементів є принципово важливим. Перестановка спостережень призводить до втрати інформації про динаміку процесу, що робить неможливим коректний аналіз. Це пов'язано з тим, що між значеннями існують часові залежності, які визначають поведінку системи.

Однією з ключових властивостей часових рядів є наявність залежностей між послідовними спостереженнями. Значення у певний момент часу, як правило, формується під впливом попередніх значень, що обумовлює існування автокореляційних зв'язків. У результаті часовий ряд не може розглядатися як набір незалежних спостережень, а потребує методів аналізу, що враховують внутрішню часову структуру даних.

Окрім локальних залежностей між послідовними спостереженнями, часові ряди характеризуються наявністю узагальнених структурних компонентів, які відображають різні аспекти поведінки процесу у часі. До основних компонентів часового ряду відносять тренд, сезонність та шум [2].

Тренд описує довгострокову зміну рівня показника та відображає загальний напрям розвитку процесу. Він може проявлятися у вигляді поступового зростання, зниження або більш складної нелінійної динаміки, що формується під впливом систематичних факторів.

Сезонність характеризує регулярні коливання значень, які повторюються через однакові проміжки часу. Такі коливання зумовлені періодичними впливами, наприклад добовими, тижневими або річними циклами, і є типовими для багатьох прикладних задач, зокрема аналізу навантаження на сервери або споживання ресурсів.

Окрім тренду та сезонності, у часовому ряді присутня випадкова складова (шум), яка відображає нерегулярні та непередбачувані коливання значень. Така компонента не має чіткої структури та може виникати під впливом випадкових факторів або неврахованих змін у системі.

Наявність зазначених компонентів означає, що поведінка часового ряду формується як поєднання детермінованих і випадкових факторів, що

ускладнює його аналіз та потребує використання відповідних методів обробки даних.

Структура часового ряду та його основні компоненти наведені на рисунку 1.1.

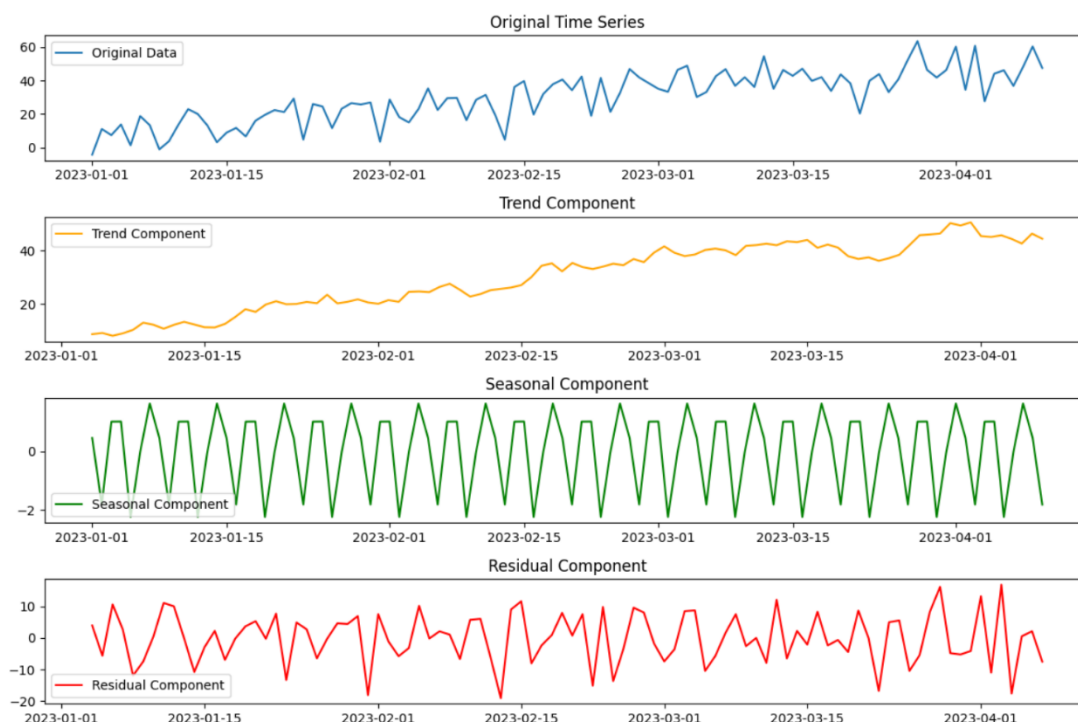


Рисунок 1.1 – Декомпозиція часового ряду [3]

У межах аналізу часових рядів особливу увагу приділяють виявленню аномалій, тобто спостережень, які суттєво відрізняються від очікуваної поведінки даних. Під аномалією розуміється відхилення від закономірностей, що характеризують типовий стан процесу, сформований на основі його попередньої динаміки [4].

На відміну від задач класифікації, де кожному об'єкту відповідає наперед визначена категорія, у задачі виявлення аномалій ключовим є встановлення факту відхилення від нормальної поведінки без жорстко заданих класів. Це означає, що межа між нормальними та аномальними значеннями не є фіксованою та визначається з урахуванням структури даних.

Важливою особливістю аномалій у часових рядах є їх контекстна залежність. Одне й те саме значення може вважатися нормальним або аномальним залежно від моменту часу та умов, у яких воно спостерігається. Наприклад, підвищене навантаження на сервер може бути типовим у пікові години та розглядатися як відхилення у періоди низької активності.

Таким чином, аномалія у часовому ряді визначається не лише абсолютним значенням показника, а й його відповідністю загальній структурі даних, включаючи тренд, сезонні закономірності та попередню динаміку процесу.

Аномалії у часових рядах можуть мати різну природу прояву, тому їх доцільно класифікувати залежно від характеру відхилення. Найбільш поширеним є поділ на точкові, контекстні та колективні аномалії. Приклади різних типів аномалій у часових рядах наведено на рисунку 1.2.

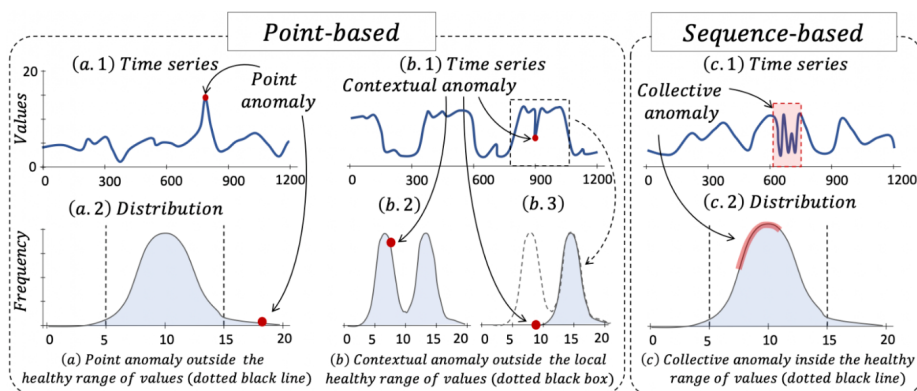


Рисунок 1.2 – Порівняння точкових, контекстних та колективних аномалій у часових рядах [4]

Точкові аномалії (point anomalies) представляють собою окремі спостереження, значення яких істотно відрізняється від інших значень у часовому ряді. Такі аномалії, як правило, легко виявляються, оскільки вони явно виходять за межі характерного діапазону значень.

Контекстні аномалії (contextual anomalies) визначаються лише з урахуванням умов, у яких вони виникають. Значення може не виглядати

аномальним саме по собі, проте стає таким у певному часовому контексті. Це пов'язано з наявністю сезонних або часових закономірностей, які визначають очікувану поведінку процесу у різні періоди.

Колективні аномалії (collective anomalies) проявляються у вигляді послідовності значень, які разом формують нетипову поведінку процесу. У цьому випадку окремі спостереження можуть не мати явних ознак відхилення, проте їх сукупність порушує загальну структуру часового ряду. Виявлення таких аномалій потребує аналізу підпослідовностей та врахування взаємозв'язків між значеннями.

З рисунка 1.2 видно, що різні типи аномалій відрізняються не лише величиною відхилення, а й залежністю від контексту та структури часової послідовності, що ускладнює їх формальне визначення та виявлення.

Задача виявлення аномалій у часових рядах ускладнюється низкою факторів, пов'язаних із природою даних та особливостями їх формування.

Однією з основних проблем є наявність шуму – випадкових коливань значень, які не відображають реальні зміни процесу. Такий шум може як приховувати аномальні відхилення, так і імітувати їх, що призводить до появи хибнопозитивних або хибнонегативних результатів під час аналізу.

Суттєву складність створює також нестационарність часових рядів, тобто зміна їх статистичних характеристик у часі. У таких умовах відсутнє єдине стабільне значення або розподіл, від якого можна оцінювати відхилення, оскільки поведінка процесу може змінюватися поступово або періодично під впливом зовнішніх факторів.

Додатковим ускладненням є різноманітність проявів аномалій. Вони можуть відрізнятися за тривалістю, інтенсивністю та структурою: від одиничних різких відхилень до складних змін поведінки на певному інтервалі часу. Така варіативність унеможлиблює використання єдиної універсальної моделі для їх виявлення.

Крім того, характер аномалій може змінюватися з часом, що пов'язано з еволюцією досліджуваного процесу або зміною зовнішніх умов. У

результаті моделі, побудовані на історичних даних, можуть втрачати актуальність, що вимагає врахування ефекту зміни розподілу даних (concept drift) [5].

Таким чином, задача виявлення аномалій у часових рядах характеризується високим рівнем складності та потребує застосування підходів, здатних враховувати шум, змінність структури даних та різні типи відхилень.

1.2 Аналіз існуючих підходів та рішень

Задача виявлення аномалій у часових рядах є активно досліджуваною у сучасній науковій літературі, що зумовлено широким спектром її практичних застосувань – від моніторингу технічних систем до аналізу фінансових та інформаційних потоків. Існуючі підходи до розв’язання цієї задачі відрізняються як за використовуваними методами, так і за рівнем складності моделей: від класичних статистичних і алгоритмічних рішень до сучасних методів машинного та глибокого навчання.

У роботі [6] запропоновано систему Greenhouse для виявлення аномалій у часових рядах, яка орієнтована на ситуації, коли відсутні приклади аномальних даних для навчання. Основна ідея підходу полягає у використанні лише нормальних даних для побудови моделі, здатної визначати відхилення без попереднього знання про типи аномалій.

Запропонована система базується на використанні LSTM-моделі для прогнозування значень часового ряду. На етапі навчання модель формує уявлення про нормальну поведінку процесу, після чого обчислюється розподіл векторів похибки прогнозу. Додатково використовується метрика M-distance для оцінювання відхилень та визначення порогового значення, що дозволяє відокремити нормальні спостереження від аномальних.

На рисунку 1.3 наведено загальну схему роботи системи, яка включає етап навчання з використанням кількох наборів нормальних даних,

побудову статистичних характеристик похибки та етап застосування моделі до нових даних з подальшим визначенням аномалій.

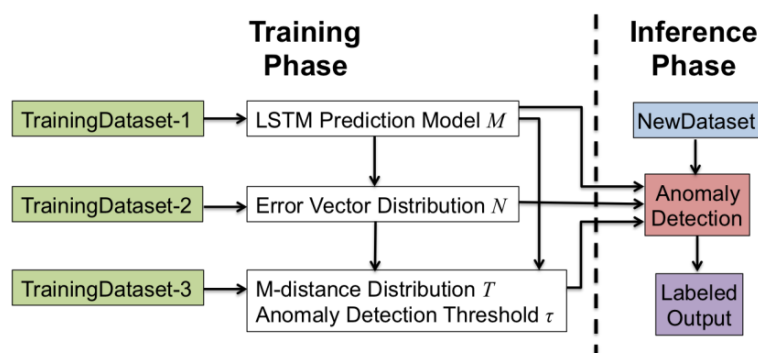


Рисунок 1.3 – Схема етапів навчання та застосування системи Greenhouse для виявлення аномалій у часових рядах [6]

Результати експериментів показали, що запропонований підхід дозволяє ефективно виявляти аномалії у часових рядах навіть за відсутності розмічених аномальних прикладів, що є важливою перевагою у практичних задачах. Разом з тим, ефективність методу залежить від якості представлення нормальної поведінки та правильного вибору порогових значень.

Подальший розвиток підходів до виявлення аномалій у часових рядах представлено у роботі [7], де запропоновано модель, що поєднує варіаційний автоенкодер (VAE), генеративно-змагальну мережу (GAN) та LSTM-архітектуру. Основна ідея підходу полягає у поєднанні можливостей відновлення даних та генерації нових послідовностей для більш точного виявлення аномалій.

Запропонована модель складається з кількох компонентів: енкодера, який перетворює вхідну часову послідовність у латентне представлення; генератора, що відновлює дані з латентного простору; та дискримінатора, який оцінює відповідність згенерованих послідовностей реальним. Для

врахування часових залежностей у всіх основних модулях використано LSTM-мережі.

На рисунку 1.4 наведено архітектуру запропонованої моделі, яка включає етап навчання, де відбувається спільна оптимізація генератора та дискримінатора, а також етап застосування, під час якого аномалії визначаються на основі поєднання похибки відновлення та результатів дискримінатора.

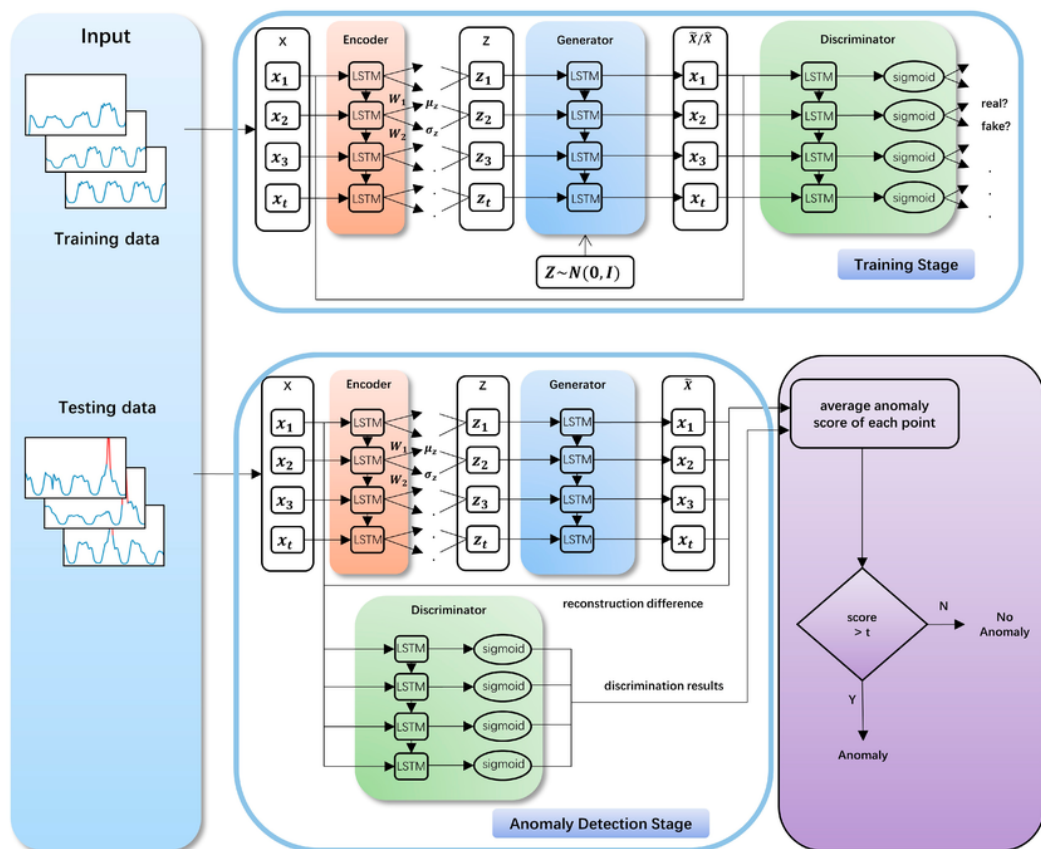


Рисунок 1.4 – Архітектура моделі LSTM-based VAE-GAN для виявлення аномалій у часових рядах [7]

Отримані результати показали, що використання комбінованої архітектури дозволяє більш точно виявляти складні аномалії у часових рядах, зокрема колективні відхилення. Водночас така модель характеризується високою обчислювальною складністю та складністю налаштування, що може ускладнювати її практичне застосування.

Інший підхід до виявлення аномалій у часових рядах представлено у роботі [8], де запропоновано використання методу Matrix Profile для аналізу багатовимірних часових даних. Основна ідея полягає у пошуку підпоследовностей, які істотно відрізняються від інших частин часового ряду, що дозволяє виявляти аномалії не лише на рівні окремих значень, а й на рівні структурних фрагментів.

Метод ґрунтується на обчисленні матриці схожості між підпоследовностями часових рядів, що дозволяє визначити ступінь їх подібності. Аномалії ідентифікуються як підпоследовності з найменшою схожістю до інших, тобто такі, що не повторюють типову поведінку процесу. У випадку багатовимірних даних додатково враховується взаємозв'язок між окремими вимірами.

На рисунку 1.5 наведено приклад багатовимірного часового ряду з вбудованою аномальною підпоследовністю. Показано, що аномалія проявляється не лише як локальне відхилення, а й як порушення кореляції між вимірами, що дозволяє ефективно виявляти складні аномалії у багатовимірних даних.

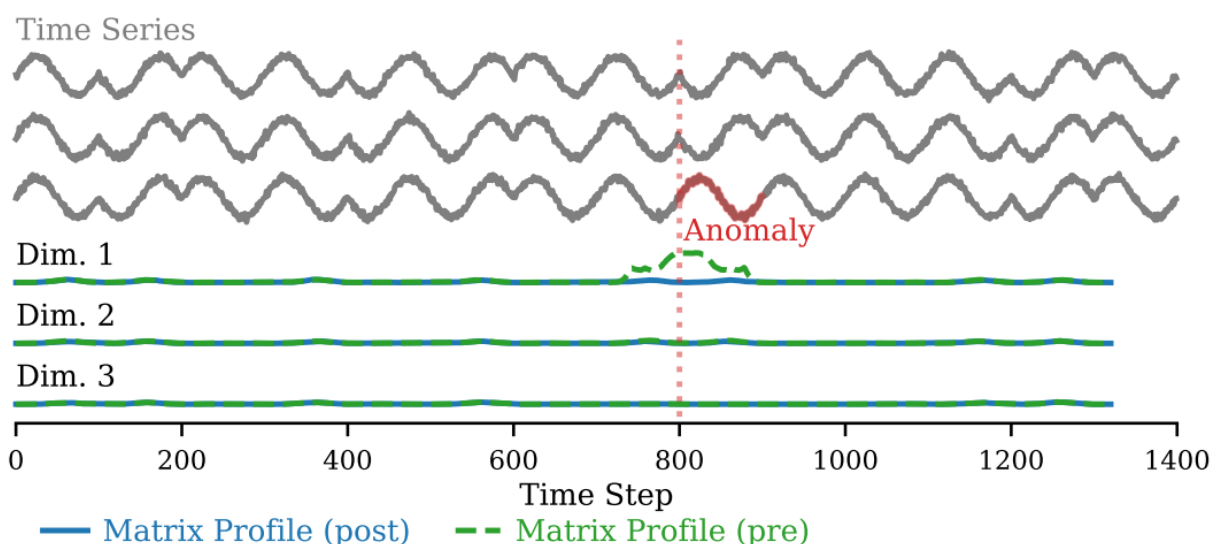


Рисунок 1.5 – Виявлення аномальної підпоследовності у багатовимірному часовому ряді за допомогою методу Matrix Profile [8]

Отримані результати свідчать, що використання Matrix Profile дозволяє точно виявляти аномальні підпоследовності без необхідності попереднього навчання моделі. Водночас ефективність підходу залежить від вибору параметрів підпоследовностей та обчислювальних витрат при обробці великих обсягів даних.

У роботі [9] запропоновано підхід до виявлення аномалій у часових рядах із використанням автоенкодерів, а також досліджено можливість застосування квантових автоенкодерів для покращення якості виявлення відхилень. Основна ідея методу полягає у відновленні вхідної часової последовності та визначенні аномалій на основі величини похибки реконструкції: значні відхилення між вхідними та відновленими даними інтерпретуються як аномальні.

Архітектура моделі включає енкодер, який перетворює вхідну последовність у компактне латентне представлення, та декодер, що відновлює початкові дані з цього представлення. Для врахування часових залежностей у моделі використано LSTM-шари, що дозволяє ефективно працювати з последовними даними. У процесі навчання модель оптимізується таким чином, щоб мінімізувати похибку реконструкції для нормальних даних, формуючи узагальнене представлення типової поведінки часового ряду.

На рисунку 1.6 наведено структуру автоенкодера, яка використовується як базова модель у дослідженні. Показано последовність шарів енкодера та декодера, а також процес перетворення вхідної последовності у латентний простір та її подальшого відновлення.

Під час застосування моделі до нових даних для кожної підпоследовності обчислюється похибка реконструкції, яка використовується як показник аномальності. У роботі також розглянуто використання квантових автоенкодерів, що дозволяє потенційно підвищити ефективність стиснення даних та виявлення складних залежностей у часових рядах.

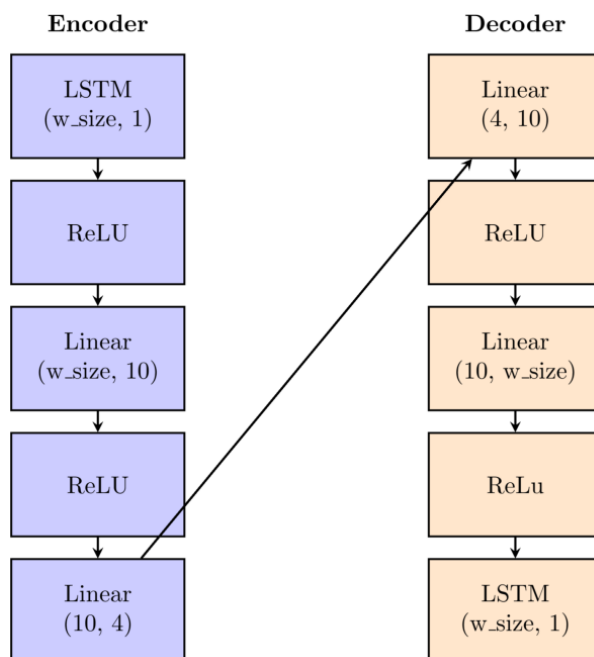


Рисунок 1.6 – Архітектура автоенкодера для виявлення аномалій у часових рядах [9]

Результати експериментів показали, що автоенкодерні моделі здатні ефективно виявляти аномалії у часових рядах без необхідності використання розмічених аномальних даних. Водночас застосування квантових підходів є обмеженим сучасним рівнем розвитку обчислювальних технологій та потребує значних ресурсів.

Подальший розвиток автоенкодерних підходів до виявлення аномалій у часових рядах представлено у роботі [10], де запропоновано модель Donut, що базується на варіаційному автоенкодері (VAE) та орієнтована на аналіз сезонних часових рядів, зокрема КРІ веб-застосунків. Основна ідея полягає у ймовірнісному моделюванні нормальної поведінки даних та визначенні аномалій як відхилень від очікуваного розподілу.

На відміну від класичних автоенкодерів, у моделі використовується варіаційний підхід, що дозволяє отримати розподіл у латентному просторі, а не фіксоване представлення. Це забезпечує більш гнучке моделювання

складних закономірностей у часових рядах, зокрема сезонних коливань та їх варіацій.

Архітектура моделі включає етап попередньої обробки даних, зокрема нормалізацію, обробку пропущених значень та формування підпоследовностей за допомогою ковзного вікна. На етапі навчання використовується модифікована функція втрат (ELBO), що дозволяє ефективно враховувати невизначеність у даних. Для визначення аномалій застосовується процедура реконструкції та подальшого оцінювання відхилень із використанням стохастичних методів.

На рисунку 1.7 наведено загальну архітектуру моделі Donut, яка включає етапи підготовки даних, навчання та виявлення аномалій. Показано, що модель поєднує глибоке представлення даних із імовірнісним підходом до оцінювання відхилень.

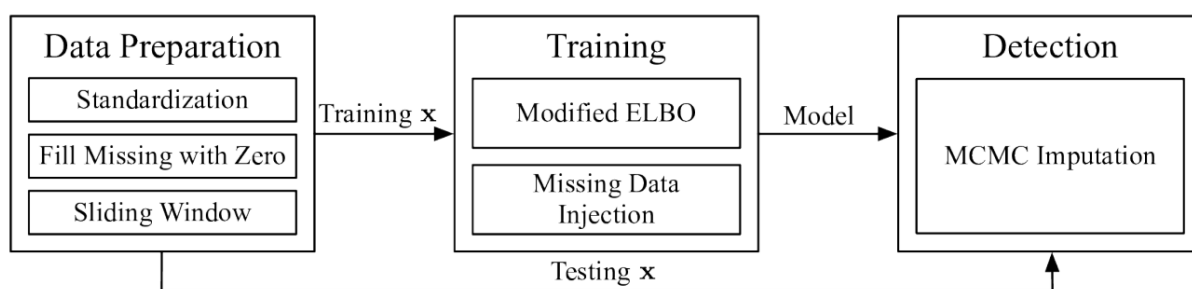


Рисунок 1.7 – Загальна архітектура моделі Donut для виявлення аномалій у часових рядах [10]

Результати експериментів показали, що модель досягає значень F1-міри в діапазоні від 0,75 до 0,9 на реальних даних KPI, перевищуючи як базові автоенкодерні моделі, так і деякі ансамблеві підходи. Важливою особливістю є також наявність теоретичної інтерпретації процесу реконструкції через оцінювання щільності розподілу, що підвищує пояснюваність моделі.

Водночас ефективність підходу залежить від коректного налаштування параметрів моделі та якості попередньої обробки даних, що може впливати на результати виявлення аномалій.

Сучасні підходи до виявлення аномалій у часових рядах активно використовують архітектуру Transformer, що продемонстровано у роботі [11], де запропоновано модель AnomalyBERT. Основна ідея підходу полягає у використанні механізму самоуваги для виявлення аномалій без необхідності розмічених даних, що досягається за рахунок самонавчання.

У запропонованому підході використовується спеціальна схема деградації даних: на етапі навчання частина вхідної часової послідовності штучно змінюється або замінюється, після чого модель навчається визначати пошкоджені фрагменти. Таким чином формується здатність моделі виявляти відхилення від нормальної структури даних без явного використання аномальних прикладів.

Архітектура моделі включає шар лінійного перетворення вхідних даних, блок Transformer, що складається з багатоголової самоуваги та повнозв'язних шарів, а також блок прогнозування, який визначає рівень аномальності для кожного фрагмента послідовності. Використання механізму самоуваги дозволяє враховувати як локальні, так і глобальні залежності у часовому ряді.

На рисунку 1.8 наведено загальну схему запропонованого підходу, яка включає процес генерації деградованих даних, навчання моделі та визначення аномалій на основі порівняння фрагментів послідовності.

Результати експериментів показали, що модель AnomalyBERT демонструє високу точність виявлення аномалій у різних типах часових рядів, перевищуючи ряд існуючих підходів. Водночас складність архітектури та значні обчислювальні витрати можуть обмежувати її використання у задачах з обмеженими ресурсами.

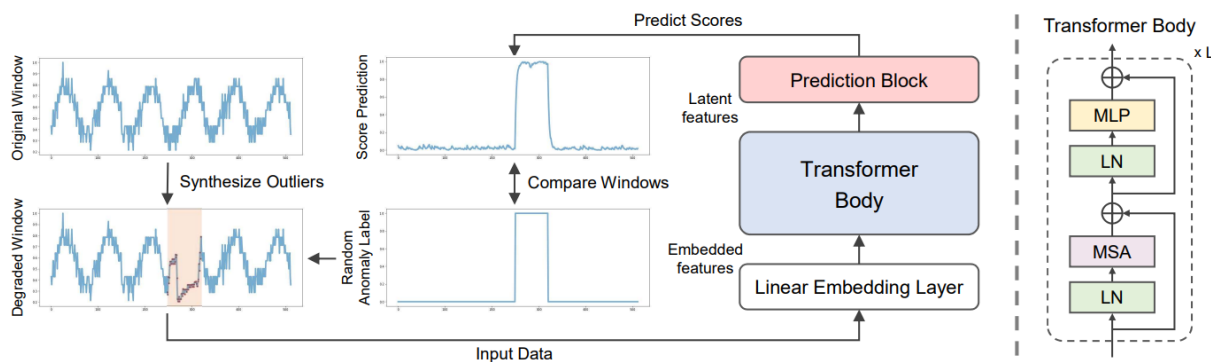


Рисунок 1.8 – Архітектура моделі AnomalyBERT для виявлення аномалій у часових рядах [11]

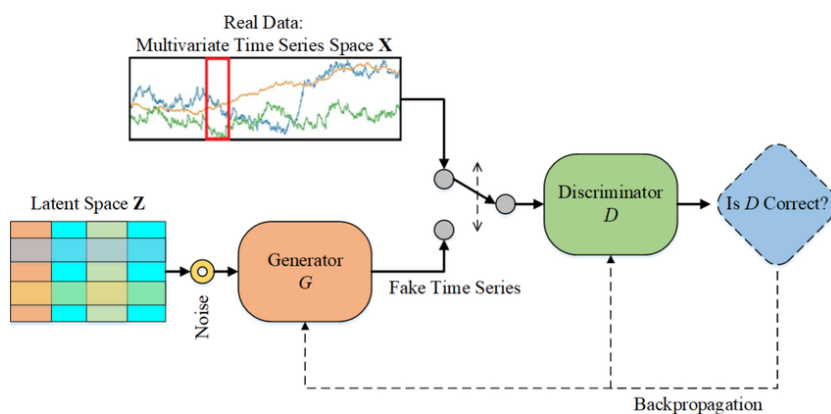
У роботі [12] запропоновано підхід ALGAN для виявлення аномалій у часових рядах, який базується на використанні генеративно-змагальних мереж у поєднанні з LSTM. Основна ідея полягає у моделюванні нормальної поведінки часових рядів за допомогою генератора та подальшому визначенні аномалій через оцінювання відмінності між реальними та згенерованими даними.

Запропонована модель включає генератор, який формує часові послідовності з латентного простору, та дискримінатор, що оцінює їх подібність до реальних даних. Особливістю підходу є використання модифікованої архітектури adjusted-LSTM, яка дозволяє підвищити якість згенерованих часових послідовностей та більш точно відображати характерні закономірності даних. Додатково використовується відображення реальних даних у латентний простір, що дає змогу обчислювати показник аномальності як ступінь відхилення між реальними та реконструйованими послідовностями.

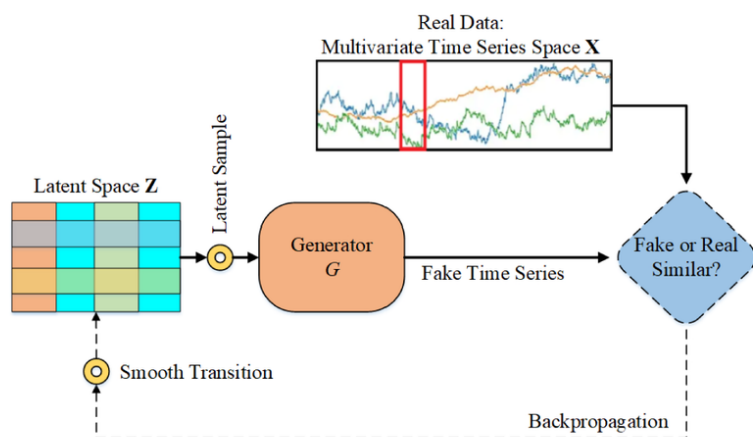
На рисунку 1.9 наведено загальну схему роботи моделі, яка включає етап навчання генератора та дискримінатора, а також етап визначення аномалій шляхом порівняння реальних і згенерованих часових рядів. Показано, що аномалії визначаються як ділянки, для яких модель не може відтворити нормальну структуру даних.

Ефективність запропонованого підходу оцінювалася на 75 реальних наборах даних з різних предметних галузей. Зокрема, модель досягла середнього значення F1-міри 0,771 для одновимірних часових рядів (NAB), 0,851 для багатовимірних даних (SMD) та 0,677 для великомасштабного набору SWaT. Отримані результати свідчать про здатність моделі ефективно виявляти аномалії як в одновимірних, так і в багатовимірних часових рядах.

Водночас підхід характеризується складністю навчання, високими обчислювальними витратами та потребує ретельного налаштування параметрів моделі, що може обмежувати його використання у практичних задачах.



(a) Representing Normal Time Series with Generator G



(b) Mapping Real-Data to the Latent Space and computing Anomaly Scores

Рисунок 1.9 – Схема роботи моделі ALGAN для виявлення аномалій у часових рядах [12]

У роботі [13] запропоновано гібридну модель для виявлення аномалій у часових рядах, що поєднує статистичний підхід SARIMA та нейромережеву модель LSTM. Основна ідея полягає у використанні переваг обох підходів: SARIMA застосовується для моделювання лінійних та стаціонарних компонентів часового ряду, тоді як LSTM використовується для виявлення складних нелінійних залежностей та нестаціонарних характеристик даних.

Запропонована модель передбачає попередній аналіз часових даних, зокрема перевірку стаціонарності та застосування диференціювання у разі необхідності. Далі будується модель SARIMA, яка дозволяє отримати залишки, що використовуються як вхідні дані для LSTM. Для підвищення ефективності навчання у моделі LSTM застосовано механізм багатоголової уваги з розрідженою регуляризацією, що дозволяє виділяти найбільш значущі часові залежності.

На рисунку 1.10 наведено загальну схему роботи гібридної моделі, яка включає етап попередньої обробки даних, побудову SARIMA-моделі, навчання LSTM та прийняття рішення щодо наявності аномалії на основі порогового значення.

Експериментальні результати показали, що запропонована модель досягає точності 91,2% та F1-міри 0,904 на даних KPI, що перевищує показники окремого використання SARIMA або LSTM. Це свідчить про ефективність поєднання статистичних і нейромережевих підходів для задачі виявлення аномалій.

Разом з тим, гібридна модель характеризується складністю побудови та налаштування, оскільки потребує узгодження параметрів двох різних підходів.

У роботі [14] запропоновано підхід MadSGM для виявлення аномалій у багатовимірних часових рядах, що базується на використанні score-based генеративних моделей. Основна ідея полягає у моделюванні розподілу

нормальних даних за допомогою стохастичного процесу, що дозволяє більш точно оцінювати ймовірність появи нових спостережень.

Запропонована модель навчається відновлювати дані, поступово додаючи та видаляючи шум у процесі навчання, що дозволяє сформувати узагальнене представлення нормальної поведінки часового ряду. Аномалії визначаються як спостереження, що мають низьку ймовірність у змодельованому розподілі або суттєво відрізняються від відновлених значень. Особливістю підходу є використання умовної score-мережі та механізму очищення (purification), що дозволяє зменшити вплив шуму та підвищити контраст між нормальними та аномальними даними.

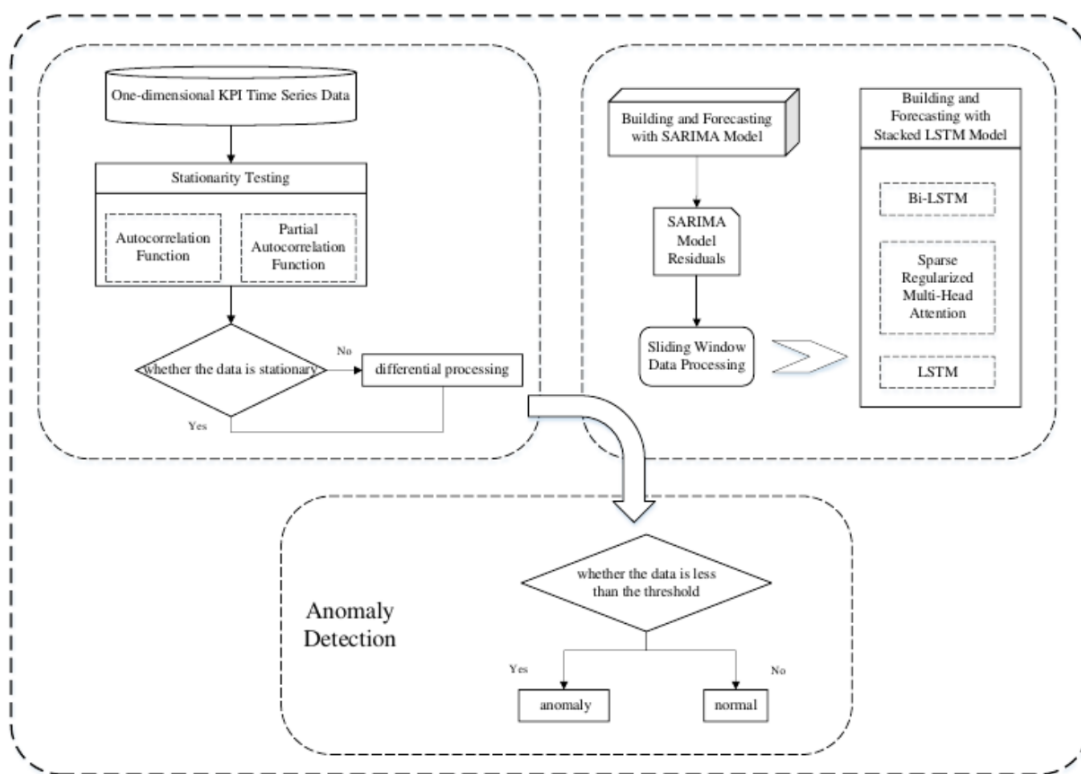


Рисунок 1.10 – Загальна схема гібридної моделі SARIMA–LSTM для виявлення аномалій у часових рядах [13]

Ефективність підходу оцінювалася на п'яти бенчмарк-наборах даних, зокрема SWaT, SMAP, MSL, PSM та SMD. Отримані результати показали, що модель MadSGM досягає значення F1-міри до 0,98 на окремих наборах

даних, що перевищує результати ряду існуючих підходів, таких як LSTM-VAE, MAD-GAN та OmniAnomaly. Це свідчить про високу ефективність запропонованого методу при виявленні складних аномалій у багатовимірних часових рядах.

Водночас застосування таких моделей пов'язане з високими обчислювальними витратами та складністю навчання, що може обмежувати їх використання у практичних задачах.

У роботі [15] запропоновано підхід до виявлення аномалій у часових рядах на основі автоенкодерів із використанням згорткових архітектур, зокрема 1D-CNN та Temporal Convolutional Network (TCN). Основна ідея полягає у навчанні моделей відтворювати нормальні патерни часових рядів та визначенні аномалій на основі похибки реконструкції.

У дослідженні розглянуто застосування моделей до аналізу енергоспоживання побутових приладів у розумних будинках. Для цього використовуються часові ряди з набору даних REFIT, що дозволяє оцінити здатність моделей виявляти відхилення у реальних сценаріях. Модель TCN забезпечує ефективне врахування довгострокових залежностей у часових рядах завдяки використанню розширених згорток.

На рисунку 1.11 наведено архітектуру автоенкодера на основі TCN, яка включає енкодер із послідовністю згорткових шарів та декодер, що відновлює початкову часову послідовність. Така структура дозволяє ефективно виділяти характерні ознаки нормальної поведінки даних.

Результати експериментів показали, що модель на основі TCN перевершує класичну 1D-CNN за точністю виявлення аномалій у задачах аналізу енергоспоживання. Це свідчить про доцільність використання згорткових підходів для аналізу часових рядів.

Водночас ефективність методу залежить від вибору архітектури моделі та параметрів згорткових шарів, що може впливати на якість виявлення аномалій.

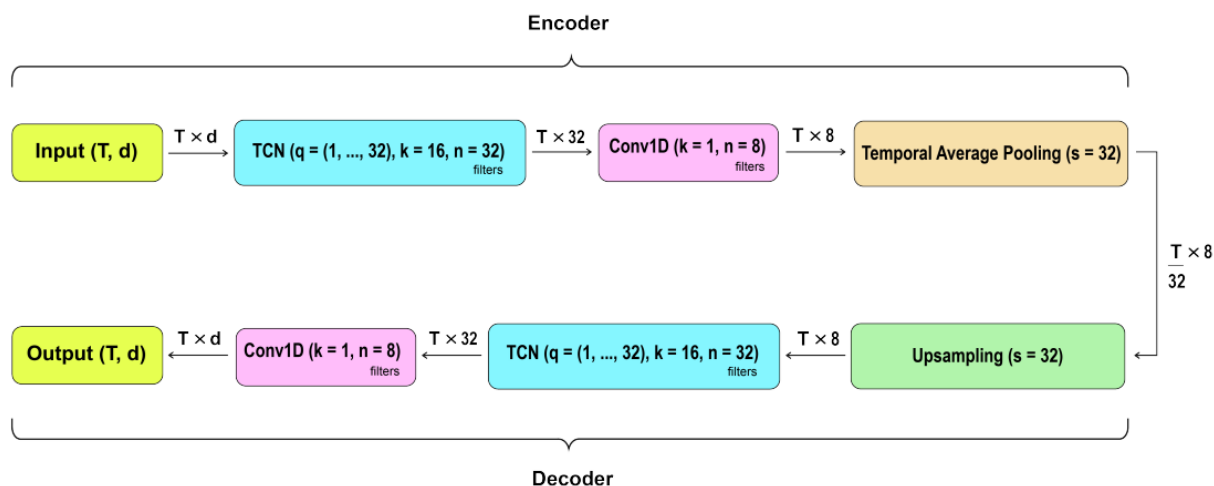


Рисунок 1.11 – Архітектура автоенкодера на основі Temporal Convolutional Network для виявлення аномалій у часових рядах [15]

У роботі [16] запропоновано підхід до виявлення аномалій у часових рядах на основі методу One-Class Support Vector Machine, адаптованого для аналізу підпоследовностей. Основна ідея полягає у побудові моделі, яка описує лише нормальну поведінку даних та визначає аномалії як відхилення від цієї області.

Для врахування часової структури даних часовий ряд розбивається на підпоследовності фіксованої довжини, які далі використовуються як вхідні об'єкти для моделі. Це дозволяє аналізувати не окремі значення, а локальні фрагменти ряду, що є важливим для виявлення складних аномалій.

Моделю One-Class SVM навчається на нормальних даних, формуючи межу, яка відокремлює типові підпоследовності від нетипових. Під час застосування нові підпоследовності оцінюються відносно цієї межі, і ті з них, що виходять за її межі, визначаються як аномальні.

Результати експериментів показали, що запропонований підхід дозволяє ефективно виявляти аномальні підпоследовності у часових рядах, при цьому не потребуючи складних архітектур або значних обчислювальних ресурсів. Водночас точність методу залежить від вибору

параметрів моделі та довжини підпоследовностей, що може впливати на якість виявлення аномалій.

У роботі [17] запропоновано гібридний підхід до виявлення аномалій у часових рядах, що поєднує автоенкодер із механізмом пам'яті (Memory-Augmented Autoencoder) та метод One-Class Support Vector Machine. Основна ідея полягає у поєднанні ефективного представлення ознак за допомогою глибокої нейронної мережі та побудови чіткої межі розділення нормальних і аномальних даних у просторі ознак.

Запропонована модель використовує автоенкодер з механізмом пам'яті для формування більш дискримінативних представлень нормальної поведінки, що дозволяє уникнути проблеми надмірного узагальнення, характерної для класичних автоенкодерів. Отримані ознаки використовуються як вхід для моделі One-Class SVM, яка будує оптимальну межу розділення у просторі ознак. Таким чином реалізується адаптивна «глибока» ядрова функція без необхідності ручного вибору ядра.

Особливістю підходу є використання спільного end-to-end навчання, що забезпечує узгоджену оптимізацію як процесу виділення ознак, так і задачі виявлення аномалій. Додатково застосовується багатофакторне представлення даних, що дозволяє враховувати складні характеристики часових рядів.

Результати експериментів на публічних наборах даних показали, що запропонований підхід досягає середнього значення F1-міри 0,934 та recall 0,958, що перевищує результати існуючих моделей у середньому на 3–4%. Це свідчить про високу ефективність моделі при роботі з високовимірними та незбалансованими даними.

1.3 Постановка задачі дослідження

Аналіз особливостей часових рядів та існуючих підходів до виявлення аномалій показав, що дана задача характеризується значною складністю та

потребує використання методів, здатних враховувати динаміку змін у даних. Різні підходи демонструють ефективність за певних умов, однак їх результати можуть суттєво відрізнятися залежно від характеру даних та способу їх обробки. Це зумовлює доцільність проведення порівняльного аналізу методів з метою визначення найбільш придатного підходу для практичного застосування.

У зв'язку з цим метою даної роботи є розробка інтелектуальної системи виявлення аномалій у часових рядах на основі методів машинного навчання та проведення порівняльного аналізу різних підходів до розв'язання цієї задачі.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- проаналізувати особливості часових рядів та підходи до виявлення аномалій;
- розглянути основні методи виявлення аномалій у даних;
- підготувати та дослідити набір даних часових рядів, зокрема виконати його попередню обробку та формування ознак;
- реалізувати методи виявлення аномалій на основі класичних та нейромережових підходів;
- провести експериментальне дослідження ефективності реалізованих методів на обраному наборі даних;
- виконати порівняльний аналіз отриманих результатів та визначити найбільш ефективний підхід;
- розробити програмну реалізацію системи виявлення аномалій із можливістю візуалізації результатів.

2 МЕТОДИ ВИЯВЛЕННЯ АНОМАЛІЙ У ДАНИХ

Виявлення аномалій є одним із важливих напрямів аналізу даних, оскільки дозволяє своєчасно виявляти нетипові події, збої, відмови обладнання, помилки вимірювань або потенційно небезпечні ситуації. Особливої актуальності ця задача набуває під час роботи з часовими рядами, де дані надходять послідовно в часі та можуть характеризуватися складною структурою, наявністю шуму, трендів і сезонних коливань. За таких умов ефективно виявлення аномалій потребує застосування спеціалізованих методів аналізу, здатних враховувати особливості часової залежності між спостереженнями.

Сучасні підходи до виявлення аномалій охоплюють широкий спектр методів – від простих статистичних моделей до складних алгоритмів машинного навчання та нейронних мереж. Кожен із цих підходів має власні переваги, обмеження та сфери застосування, що обумовлює необхідність їх детального аналізу.

2.1 Статистичні підходи до виявлення аномалій

Статистичні методи є одним із найперших підходів, що застосовуються для виявлення аномалій у даних. Їхня основна ідея полягає у визначенні спостережень, поведінка яких суттєво відрізняється від переважної частини даних. Такі методи базуються на аналізі статистичних характеристик вибірки та припущенні, що нормальні спостереження формують певну закономірність або розподіл, тоді як аномалії проявляються у вигляді значних відхилень від нього. Незважаючи на появу більш складних алгоритмів машинного навчання та нейромережевих моделей, статистичні підходи залишаються важливим інструментом аналізу завдяки простоті реалізації, невисоким обчислювальним витратам та зрозумілій інтерпретації результатів. Особливо ефективними вони є у

випадках, коли структура даних є відносно простою, а характер нормальної поведінки може бути описаний за допомогою базових статистичних показників.

В основі багатьох статистичних методів виявлення аномалій лежить аналіз характеристик розподілу даних. Передбачається, що більшість спостережень описує нормальну поведінку досліджуваного процесу, тоді як аномалії зустрічаються відносно рідко та суттєво відрізняються від типових значень. Для оцінювання таких відхилень використовуються статистичні показники, зокрема середнє значення, дисперсія та стандартне відхилення.

Середнє значення характеризує центральну тенденцію даних та відображає їх типовий рівень. Водночас лише цього показника недостатньо для аналізу, оскільки різні набори даних можуть мати однакове середнє значення, але суттєво відрізнятися за ступенем розсіювання. Для оцінювання варіативності використовується стандартне відхилення, яке показує, наскільки окремі спостереження відхиляються від середнього значення. Чим більшим є значення стандартного відхилення, тим ширше розподілені дані відносно свого центру (рисунк 2.1).

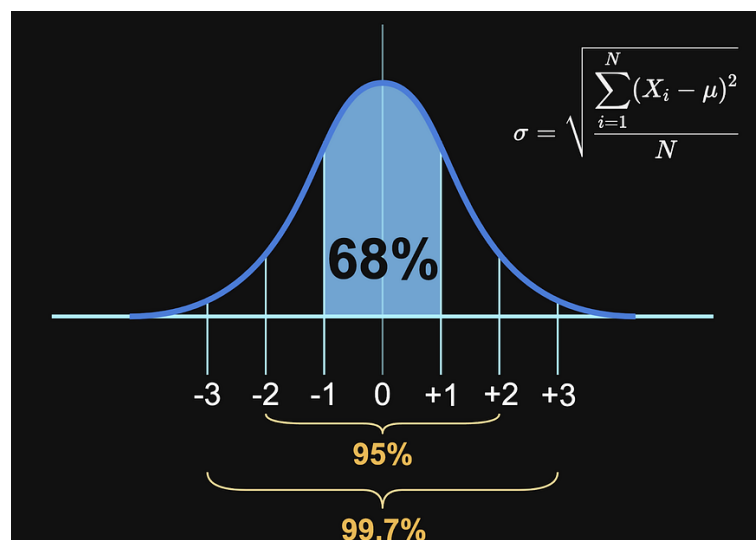


Рисунок 2.1 – Розподіл даних відносно середнього значення та стандартного відхилення [18]

У багатьох практичних задачах припускається, що дані наближено підпорядковуються нормальному розподілу. Для такого розподілу характерне зосередження більшості значень поблизу середнього значення, тоді як імовірність появи дуже великих або дуже малих значень поступово зменшується. Для такого розподілу характерне зосередження більшості значень поблизу середнього значення: приблизно 68% спостережень знаходяться в межах одного стандартного відхилення від середнього значення, близько 95% – у межах двох стандартних відхилень, а майже 99,7% – у межах трьох стандартних відхилень. Саме ця властивість часто використовується для виявлення потенційних аномалій, які розташовані далеко від основної маси даних [18].

Одним із найбільш поширених статистичних підходів до виявлення аномалій є метод Z-score. Його використання ґрунтується на припущенні, що більшість спостережень зосереджена поблизу середнього значення, тоді як аномальні значення характеризуються значним відхиленням від нього. Метод дозволяє оцінити положення кожного спостереження відносно загального розподілу даних та визначити ступінь його віддаленості від центру вибірки [19].

Для цього кожне значення стандартизується шляхом віднімання середнього значення та подальшого ділення на стандартне відхилення. У результаті отримується показник Z-score, який відображає відстань від спостереження до середнього значення в одиницях стандартного відхилення. Розрахунок виконується за формулою (2.1):

$$z = \frac{X - \mu}{\sigma}, \quad (2.1)$$

де X – значення досліджуваного спостереження;

μ – середнє значення вибірки;

σ – стандартне відхилення вибірки;

z – стандартизована оцінка.

Отримане значення дозволяє оцінити ступінь нетиповості окремого спостереження. Значення, близькі до нуля, свідчать про розташування поблизу середнього значення вибірки, тоді як збільшення абсолютного значення Z-score вказує на зростання відхилення від типової поведінки даних. Для інтерпретації результатів часто використовується правило трьох сигм, відповідно до якого спостереження з абсолютним значенням Z-score більше трьох розглядаються як потенційні аномалії.

Принцип виявлення аномалій за допомогою методу Z-score проілюстровано на рисунку 2.2. Центральна частина розподілу відповідає типовим значенням, тоді як області на периферії містять спостереження, що суттєво відрізняються від більшості даних. Саме такі значення можуть бути класифіковані як аномальні та потребувати додаткового аналізу.

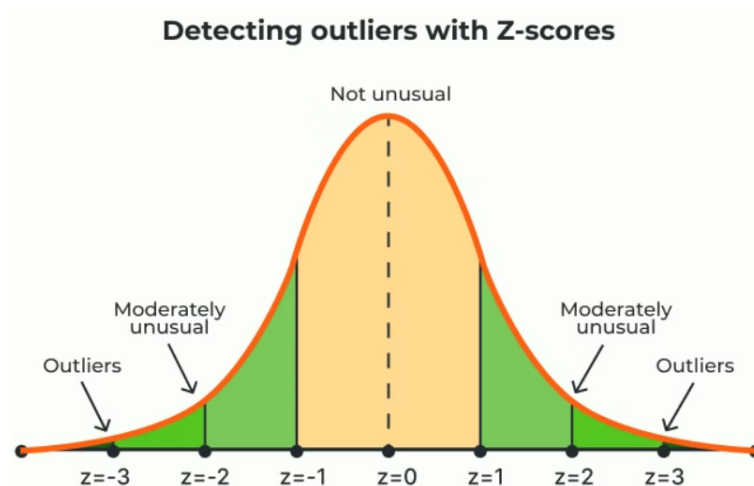


Рисунок 2.2 – Інтерпретація значень Z-score при виявленні аномалій [19]

Основними перевагами методу є простота реалізації, висока швидкість обчислень та зрозуміла інтерпретація результатів. Водночас його застосування є найбільш ефективним для даних, розподіл яких наближений до нормального. За наявності складних нелінійних залежностей, значного рівня шуму або великої кількості викидів ефективність методу може знижуватися.

Незважаючи на простоту та ефективність методу Z-score, його застосування є доцільним переважно для стаціонарних даних, статистичні характеристики яких суттєво не змінюються з часом. У випадку часових рядів середнє значення та дисперсія можуть змінюватися під впливом трендів, сезонності або інших динамічних процесів. За таких умов використання глобальних статистичних характеристик може призводити до зниження точності виявлення аномалій. Для врахування локальних змін у часових рядах застосовуються методи ковзної статистики.

Найпростішим представником цієї групи є метод ковзного середнього (Moving Average, MA). Його основна ідея полягає у послідовному обчисленні середнього значення для фіксованого вікна спостережень, яке переміщується вздовж часового ряду [20]. У результаті формується згладжена послідовність, що відображає загальну тенденцію зміни даних та зменшує вплив випадкових коливань. Розрахунок ковзного середнього виконується за формулою (2.2):

$$MA(t) = \frac{1}{n} \sum_{i=0}^{n-1} x(t - i), \quad (2.2)$$

де $x(t)$ – значення часового ряду в момент часу (t) ;

n – розмір ковзного вікна;

$MA(t)$ – значення ковзного середнього в момент часу (t) .

Порівняння фактичних значень часового ряду з відповідними значеннями ковзного середнього дозволяє виявляти локальні відхилення від типової поведінки. Якщо різниця між реальним значенням та згладженою оцінкою перевищує встановлений поріг, таке спостереження може розглядатися як потенційна аномалія.

Подальшим розвитком цього підходу є експоненціальне ковзне середнє (Exponential Moving Average, EMA), яке враховує різну важливість спостережень. На відміну від звичайного ковзного середнього, де всі значення в межах вікна мають однакову вагу, EMA надає більшу вагу більш

новим спостереженням, що дозволяє швидше реагувати на зміни в даних [20]. Розрахунок експоненціального ковзного середнього виконується за формулою (2.3):

$$EMA(t) = \alpha x(t) + (1 - \alpha)EMA(t - 1), \quad (2.3)$$

де $x(t)$ – значення часового ряду в момент часу (t);

α – коефіцієнт згладжування, що визначає вплив поточного спостереження на результат;

$EMA(t - 1)$ – попереднє значення експоненціального ковзного середнього.

Окрім середнього значення, у практичних задачах часто використовуються й інші ковзні статистики, зокрема ковзна дисперсія та ковзне стандартне відхилення. Їх застосування дозволяє оцінювати не лише зміну рівня сигналу, а й зміну його варіативності в часі. Раптове зростання або зменшення локального розсіювання значень також може свідчити про виникнення аномальних подій у досліджуваному процесі.

Перевагою методів ковзної статистики є їх здатність адаптуватися до локальних особливостей часового ряду та враховувати поступові зміни його характеристик. Водночас ефективність таких методів значною мірою залежить від правильного вибору розміру вікна та порогових значень. Надто мале вікно робить оцінки нестабільними та чутливими до шуму, тоді як надто велике може приховувати локальні аномалії та знижувати швидкість реагування на зміни.

Розглянуті статистичні методи є базовим інструментом виявлення аномалій та широко використовуються завдяки простоті реалізації, високій швидкості обчислень і зрозумілій інтерпретації результатів. Такі підходи не потребують тривалого навчання моделей та можуть ефективно застосовуватися для аналізу даних із відносно простою структурою. Крім того, статистичні методи часто використовуються як початковий етап

аналізу або як базові моделі для порівняння з більш складними алгоритмами.

Водночас їх ефективність значною мірою залежить від припущень щодо властивостей даних. Наявність нелінійних залежностей, зміна статистичних характеристик у часі, складні сезонні закономірності та високий рівень шуму можуть суттєво знижувати якість виявлення аномалій. Особливо помітними ці обмеження стають під час аналізу сучасних часових рядів, які формуються в результаті роботи складних технічних систем, промислового обладнання або мережевої інфраструктури.

2.2 Методи машинного навчання для виявлення аномалій

На відміну від статистичних методів, які зазвичай використовують фіксовані порогові значення, алгоритми машинного навчання оцінюють ступінь подібності або відмінності між об'єктами на основі їхніх характеристик. Це дозволяє враховувати складні взаємозв'язки між ознаками та виявляти аномалії навіть у тих випадках, коли їх неможливо визначити за окремими статистичними показниками.

Серед найбільш поширених методів машинного навчання для виявлення аномалій можна виділити Elliptic Envelope, Isolation Forest, One-Class SVM та Local Outlier Factor [21]. Незважаючи на спільну мету, кожен із цих підходів використовує власний механізм оцінювання аномальності об'єктів та має специфічні особливості застосування.

Одним із методів машинного навчання, що застосовується для виявлення аномалій, є Elliptic Envelope. Його робота ґрунтується на припущенні, що нормальні дані утворюють компакту область у просторі ознак, яка може бути апроксимована багатовимірним еліпсоїдом. Об'єкти, що розташовані всередині цієї області, розглядаються як нормальні, тоді як спостереження за її межами класифікуються як потенційні аномалії.

Для побудови межі нормальних даних метод використовує робастну оцінку середнього значення та коваріаційної матриці. На основі цих параметрів формується еліпсоїдна область, яка охоплює більшість спостережень вибірки. Принцип роботи методу проілюстровано на рисунку 2.3. Зеленими точками позначено об'єкти, що належать до області нормальної поведінки, тоді як червоні точки розташовані за межами побудованої границі та розглядаються як аномальні [22].

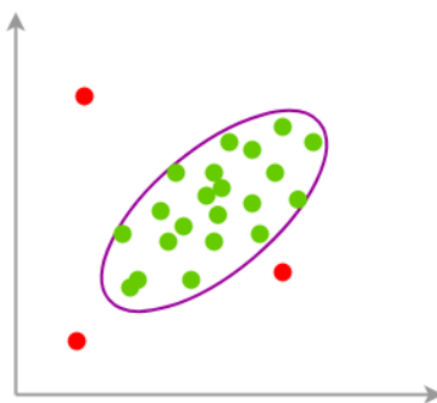


Рисунок 2.3 – Принцип роботи методу Elliptic Envelope [22]

Метод Elliptic Envelope є ефективним у випадках, коли дані мають відносно компактний розподіл і не містять складних нелінійних залежностей. Його перевагами є простота інтерпретації результатів та невисока обчислювальна складність. Крім того, побудована модель дозволяє оцінювати відстань окремих спостережень від центру розподілу та використовувати цю інформацію для прийняття рішення щодо їх аномальності.

Водночас застосування методу пов'язане з певними обмеженнями. Основним із них є припущення про еліпсоїдну форму розподілу нормальних даних. У випадку складних багатомодальних розподілів або наявності виражених нелінійних залежностей якість виявлення аномалій може суттєво знижуватися. Саме тому в сучасних задачах аналізу даних Elliptic Envelope

часто використовується як базовий підхід або як один із методів порівняння з більш складними алгоритмами машинного навчання.

На відміну від Elliptic Envelope, який базується на побудові області нормальних даних, метод Isolation Forest використовує принципово інший підхід до виявлення аномалій. Його основна ідея полягає не в описі типових спостережень, а в безпосередній ізоляції об'єктів, які суттєво відрізняються від більшості даних.

Ефективність методу ґрунтується на двох характерних особливостях аномалій. По-перше, такі спостереження зазвичай становлять лише невелику частину вибірки. По-друге, їхні значення помітно відрізняються від нормальних даних. Саме тому аномальні об'єкти, як правило, можна відокремити від решти вибірки значно швидше, ніж типові спостереження.

Для реалізації цього підходу будується ансамбль спеціальних дерев, які називаються деревами ізоляції. На кожному кроці випадково обирається одна з ознак та випадкове порогове значення, за яким виконується поділ даних. Процес повторюється рекурсивно до моменту, коли окреме спостереження буде повністю відокремлене від інших об'єктів [23]. Схематичне порівняння ізоляції аномального та нормального спостережень наведено на рисунку 2.4.

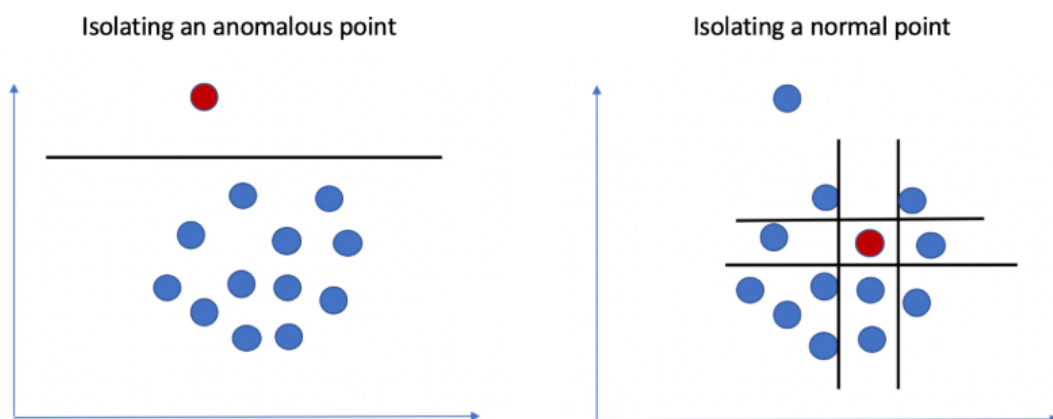


Рисунок 2.4 – Принцип ізоляції аномального та нормального спостережень у методі Isolation Forest [23]

Як видно з рисунка 2.4, для ізоляції аномальної точки достатньо виконати лише один поділ простору ознак, оскільки вона розташована далеко від основної групи даних. Натомість нормальне спостереження знаходиться всередині щільного скупчення об'єктів, тому для його повного відокремлення необхідно виконати значно більшу кількість послідовних поділів. Таким чином, основним показником аномальності в Isolation Forest виступає довжина шляху від кореня дерева до відповідного вузла: чим коротший цей шлях, тим вищою є ймовірність того, що спостереження є аномальним.

На відміну від багатьох алгоритмів, що базуються на обчисленні відстаней між об'єктами або аналізі густини розподілу, Isolation Forest не потребує виконання складних математичних обчислень для кожної пари спостережень. Завдяки цьому метод добре масштабується для великих наборів даних та може ефективно працювати у багатовимірних просторах ознак.

До переваг алгоритму належать висока швидкодія, здатність працювати з великими обсягами даних та відсутність припущень щодо форми їх розподілу. Водночас ефективність методу може знижуватися у випадках, коли межа між нормальними та аномальними спостереженнями є нечіткою або аномалії утворюють окремі щільні групи. Незважаючи на це, Isolation Forest залишається одним із найбільш популярних і практично орієнтованих методів виявлення аномалій у сучасних задачах машинного навчання.

Ще одним поширеним підходом до виявлення аномалій є One-Class SVM. На відміну від Isolation Forest, який оцінює складність ізоляції окремих об'єктів, цей метод ґрунтується на побудові межі, що відокремлює нормальні дані від потенційно аномальних спостережень. Основною метою алгоритму є формування такої області простору ознак, усередині якої зосереджена переважна більшість об'єктів, що відповідають нормальній поведінці системи.

Метод є модифікацією алгоритму Support Vector Machine (SVM), адаптованою для задач, у яких доступні переважно або виключно приклади нормальних даних [24]. У процесі навчання модель аналізує структуру вибірки та визначає межу, яка охоплює основну частину спостережень, залишаючи за її межами лише незначну кількість об'єктів. Після завершення навчання всі нові спостереження оцінюються відносно сформованої межі: об'єкти всередині області вважаються нормальними, а об'єкти за її межами розглядаються як потенційні аномалії.

Принцип роботи One-Class SVM наведено на рисунку 2.5. Червона крива відображає межу, сформовану моделлю на основі навчальних даних. Спостереження, розташовані всередині цієї області, відповідають типовій поведінці системи, тоді як точки за її межами можуть бути класифіковані як аномальні. На відміну від методів, що базуються на аналізі відстаней або статистичних характеристик, One-Class SVM здатний формувати складні нелінійні межі та враховувати взаємозв'язки між ознаками.

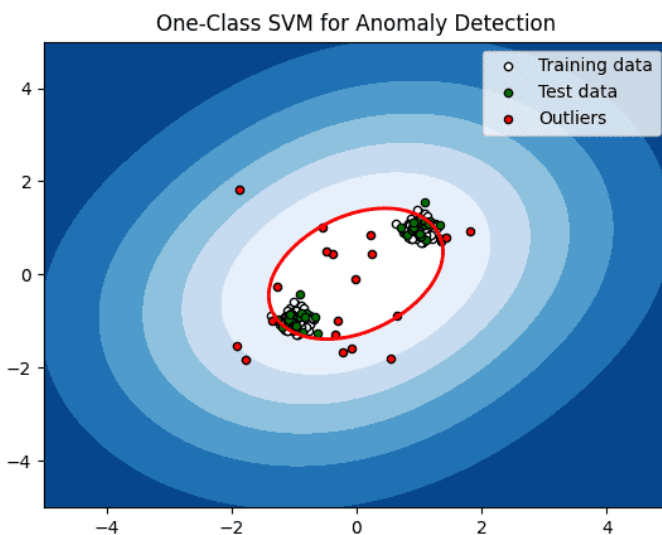


Рисунок 2.5 – Виявлення аномалій за допомогою методу One-Class SVM [24]

Важливою особливістю методу є використання ядерних функцій, які дозволяють виконувати перетворення даних у простір більшої розмірності.

Завдяки цьому модель може ефективно працювати навіть у випадках, коли нормальні та аномальні спостереження неможливо розділити простою лінійною межею. Найчастіше для задач виявлення аномалій застосовується радіальна базисна функція (RBF), що забезпечує достатню гнучкість під час побудови межі нормальних даних.

До переваг One-Class SVM належать здатність працювати за відсутності прикладів аномалій, можливість моделювання складних нелінійних залежностей та висока точність для наборів даних середнього розміру. Водночас ефективність алгоритму значною мірою залежить від вибору параметрів моделі та ядерної функції. Крім того, зі збільшенням обсягу даних зростають обчислювальні витрати на навчання, що може ускладнювати застосування методу для великих наборів даних.

Значну популярність у задачах виявлення аномалій також отримав метод Local Outlier Factor (LOF), який базується на аналізі локальної щільності розташування об'єктів. На відміну від попередніх підходів, LOF оцінює не абсолютне положення спостереження в просторі ознак, а його взаємне розташування відносно найближчих сусідів. Завдяки цьому метод здатний виявляти аномалії навіть у наборах даних зі складною структурою та нерівномірною щільністю розподілу.

Основна ідея алгоритму полягає у порівнянні локальної щільності окремого об'єкта із середньою щільністю його найближчого оточення. Якщо щільність навколо певної точки істотно нижча за щільність сусідніх об'єктів, така точка розглядається як потенційна аномалія. Відповідно, об'єкти, розташовані в межах щільних груп даних, отримують низькі значення показника аномальності та класифікуються як нормальні.

Принцип роботи методу наведено на рисунку 2.6. Розмір червоних кіл відображає ступінь аномальності спостережень: чим більшим є значення Local Outlier Factor, тим більш нетиповим вважається відповідний об'єкт. При цьому навіть точки, що знаходяться відносно близько до інших

спостережень, можуть бути визначені як аномальні, якщо їх локальна щільність суттєво відрізняється від щільності навколишніх даних.

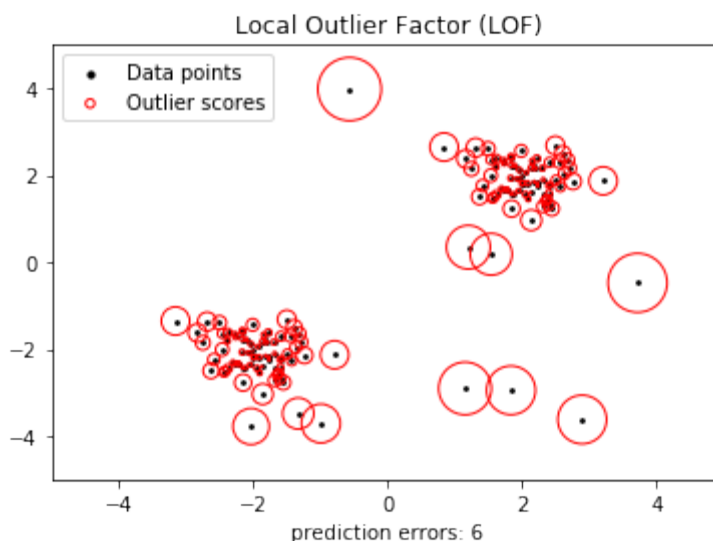


Рисунок 2.6 – Оцінювання аномальності об'єктів за допомогою методу Local Outlier Factor

Важливою перевагою LOF є здатність виявляти локальні аномалії. Наприклад, об'єкт може знаходитися на невеликій відстані від щільного кластера та все одно розглядатися як аномальний, тоді як інший об'єкт на більшій відстані від центру розрідженого кластера може вважатися нормальним. Така особливість дозволяє алгоритму ефективно працювати з даними, що містять області різної щільності, де використання глобальних критеріїв часто призводить до помилкових результатів.

Метод успішно застосовується для аналізу просторових даних, моніторингу мережевого трафіку, обробки відеопотоків та інших задач, у яких структура даних є неоднорідною. Крім того, принцип локальної оцінки щільності може використовуватися в різних модифікаціях алгоритмів пошуку аномалій та кластеризації.

До переваг LOF належать здатність виявляти локальні аномалії, відсутність припущень щодо форми розподілу даних та ефективна робота з

наборами даних, що містять кластери різної щільності. Водночас інтерпретація отриманих значень Local Outlier Factor не завжди є однозначною, оскільки порогові значення можуть відрізнятися залежно від особливостей конкретного набору даних. Крім того, зі збільшенням розмірності простору ознак точність алгоритму може знижуватися, що є характерною проблемою для багатьох методів, заснованих на аналізі відстаней між об'єктами [25].

2.3 Нейромережеві підходи до виявлення аномалій

Подальший розвиток методів виявлення аномалій пов'язаний із використанням нейронних мереж, які здатні автоматично виявляти складні закономірності в даних та враховувати нелінійні залежності між ознаками. На відміну від багатьох класичних алгоритмів машинного навчання, нейромережеві моделі не потребують явного опису характеристик даних і можуть самостійно формувати внутрішні представлення, необхідні для розв'язання поставленої задачі.

Особливої актуальності нейронні мережі набули у задачах аналізу часових рядів, де поведінка системи визначається не лише поточними значеннями, а й їх зміною в часі. У таких випадках аномалії можуть проявлятися як короткочасні відхилення, поступові зміни тенденції або складні послідовності подій, які важко виявити за допомогою статистичних методів чи алгоритмів машинного навчання, що аналізують окремі спостереження незалежно одне від одного.

Однією з ключових переваг нейронних мереж є здатність працювати з високорозмірними даними та автоматично виділяти найбільш інформативні ознаки. Це дозволяє зменшити залежність від ручного проєктування характеристик даних та підвищити ефективність аналізу складних процесів. Крім того, сучасні архітектури нейронних мереж можуть враховувати

довгострокові залежності між спостереженнями, що є важливим для аналізу часових рядів і прогнозування майбутньої поведінки системи.

У задачах виявлення аномалій нейронні мережі найчастіше навчаються на даних, які відповідають нормальному режиму роботи системи. Під час навчання модель формує уявлення про типову поведінку процесу, а в подальшому оцінює ступінь відхилення нових спостережень від вивчених закономірностей. Чим більшим є таке відхилення, тим вищою вважається ймовірність того, що відповідне спостереження є аномальним.

Залежно від архітектури моделі оцінювання аномальності може здійснюватися різними способами. Найпоширенішими підходами є аналіз помилки реконструкції даних, прогнозування майбутніх значень часового ряду та порівняння отриманих результатів із фактичними спостереженнями. Саме на цих принципах базується робота сучасних нейромережевих моделей виявлення аномалій, серед яких найбільшого поширення набули автоенкодер, рекурентні нейронні мережі типу LSTM та архітектури Transformer.

Одним із найбільш поширених нейромережевих підходів до виявлення аномалій є автоенкодер. Вони являють собою спеціальний тип нейронних мереж, призначений для відновлення вхідних даних після їх попереднього стискування. Основна ідея такого підходу полягає у навчанні моделі відтворювати власний вхід, використовуючи при цьому приховане представлення меншої розмірності. Завдяки цьому мережа змушена виділяти найбільш важливі закономірності та характеристики даних.

Типова структура автоенкодера складається з двох основних компонентів: енкодера та декодера (рисунок 2.8). Енкодер виконує перетворення вхідних даних у компактне приховане представлення, яке часто називають латентним простором [26]. Декодер, своєю чергою, використовує це представлення для відновлення початкових даних. У результаті роботи мережі формується реконструйована версія вхідного об'єкта, яка повинна бути максимально близькою до оригіналу.

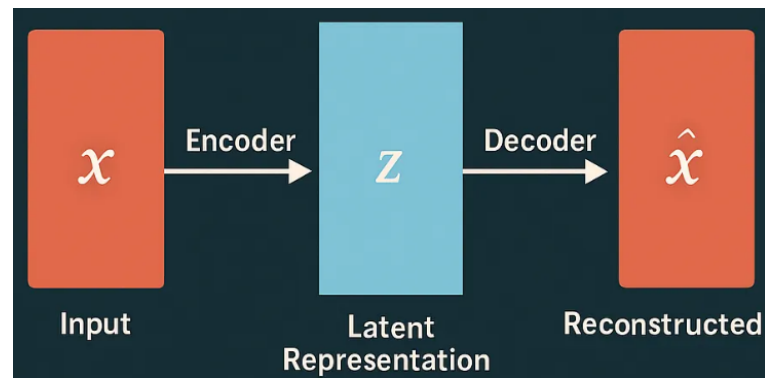


Рисунок 2.7 – Структура автоенкодера та процес реконструкції вхідних даних [26]

Математично роботу автоенкодера можна описати співвідношенням:

$$\hat{x} = g_{\phi}(f_{\theta}(x)) = x, \quad (2.4)$$

де x – вхідні дані;

f_{θ} – функція енкодера;

g_{ϕ} – функція декодера;

$z = f_{\theta}(x)$ – латентне представлення даних;

$\hat{x}$ – реконструйований вихід мережі.

У задачах виявлення аномалій автоенкодери зазвичай навчаються на даних, що відповідають нормальному режиму роботи системи. У процесі навчання модель поступово формує внутрішнє представлення типової поведінки та навчається якісно відновлювати саме ті структури, які характерні для нормальних спостережень. Коли на вхід мережі подається аномальний об'єкт, його характеристики відрізняються від закономірностей, засвоєних під час навчання, тому якість реконструкції погіршується.

Для оцінювання аномальності використовується помилка реконструкції, яка визначає ступінь відмінності між початковими даними та їх відновленою версією. Чим більшою є ця помилка, тим менш типовим вважається відповідне спостереження. Таким чином, аномалії визначаються

не безпосередньо, а через нездатність моделі коректно відтворити об'єкти, які не відповідають вивченим закономірностям.

До переваг автоенкодерів належать здатність працювати з високорозмірними даними, автоматичне виділення прихованих ознак та можливість виявлення складних нелінійних аномалій. Водночас ефективність моделі залежить від архітектури мережі, обсягу навчальних даних та правильного вибору порогу помилки реконструкції. Крім того, за недостатньо якісного налаштування модель може частково відтворювати навіть аномальні спостереження, що ускладнює їх виявлення.

Важливим напрямом розвитку нейромережових методів для аналізу часових рядів стали рекурентні нейронні мережі (Recurrent Neural Networks, RNN). На відміну від звичайних багатoshарових нейронних мереж, які розглядають кожне спостереження незалежно, рекурентні архітектури враховують послідовний характер даних та можуть використовувати інформацію про попередні стани системи під час обробки нових спостережень. Завдяки цьому RNN здатні моделювати часові залежності та аналізувати динаміку зміни даних у часі.

У задачах виявлення аномалій рекурентні мережі використовуються для вивчення типової поведінки часового ряду та прогнозування його подальшого розвитку. Якщо фактичні значення суттєво відрізняються від прогнозованих моделлю, таке відхилення може свідчити про наявність аномалії. Проте класичні рекурентні нейронні мережі мають суттєве обмеження, пов'язане зі складністю збереження інформації про події, що відбулися багато часових кроків тому. Унаслідок цього модель поступово втрачає здатність враховувати довгострокові залежності, які часто мають важливе значення під час аналізу часових рядів.

Для подолання цього недоліку була запропонована архітектура Long Short-Term Memory (LSTM). На відміну від класичних рекурентних мереж, LSTM використовує спеціальні механізми керування інформаційними потоками, які визначають, які дані необхідно зберегти, оновити або

видалити з пам'яті мережі [27]. Крім того, до архітектури вводиться окремий стан комірки (cell state), який виконує роль довгострокової пам'яті та дозволяє передавати важливу інформацію між віддаленими часовими кроками.

Структуру LSTM-комірки наведено на рисунку 2.8. Вона містить три основні механізми керування, які називаються воротами: вхідні ворота (input gate), ворота забування (forget gate) та вихідні ворота (output gate). Ворота забування визначають, яку частину інформації з попереднього стану необхідно зберегти, вхідні ворота відповідають за додавання нової інформації до пам'яті мережі, а вихідні ворота формують інформацію, яка буде передана на наступний часовий крок.

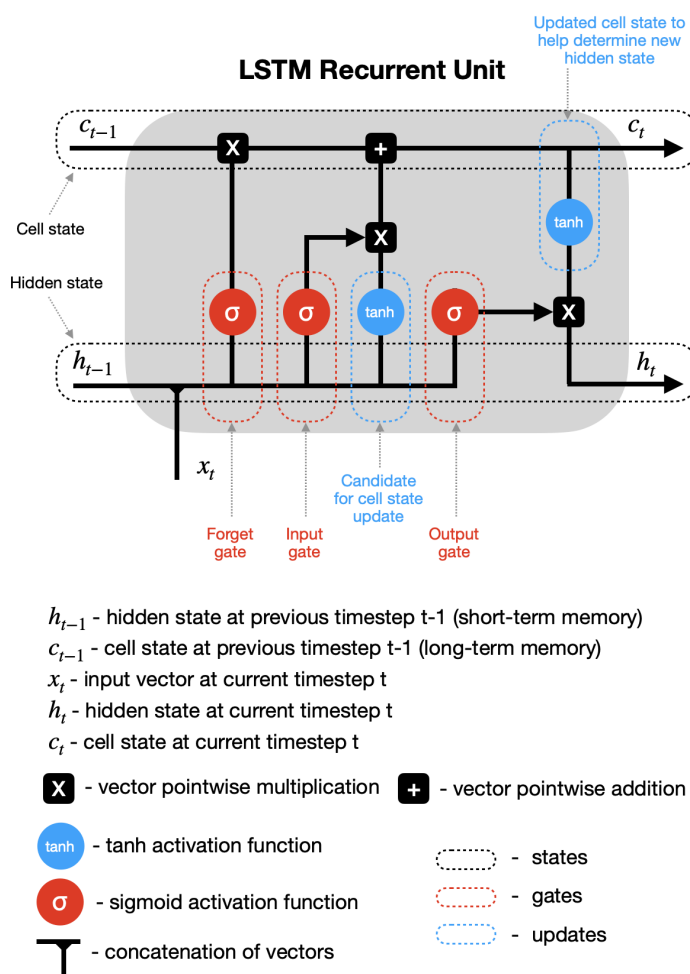


Рисунок 2.8 – Архітектура LSTM-комірки та механізм оновлення пам'яті мережі [27]

Завдяки наявності механізму довгострокової пам'яті LSTM здатна ефективно враховувати як короткострокові, так і довгострокові залежності в часових рядах. Це робить такі мережі особливо придатними для аналізу сенсорних даних, моніторингу технічних систем, фінансових часових рядів та інших задач, у яких аномалії можуть бути пов'язані не лише з окремими значеннями, а й зі зміною поведінки процесу протягом тривалого часу.

Основними перевагами LSTM є здатність працювати з послідовними даними, ефективне врахування довгострокових залежностей та висока якість моделювання часових процесів. Водночас такі моделі характеризуються складнішою архітектурою, потребують значних обчислювальних ресурсів та більшого обсягу навчальних даних порівняно з класичними алгоритмами машинного навчання.

Подальшим розвитком нейромережевих підходів до аналізу послідовних даних стали моделі Transformer. На відміну від рекурентних мереж, які обробляють дані послідовно та передають інформацію між часовими кроками через приховані стани, архітектура Transformer використовує механізм самоуваги (self-attention), що дозволяє аналізувати взаємозв'язки між усіма елементами послідовності одночасно.

Основна ідея механізму self-attention полягає у визначенні важливості окремих елементів вхідної послідовності відносно один одного. Завдяки цьому модель може автоматично виділяти найбільш значущі залежності в даних незалежно від того, наскільки далеко один від одного розташовані відповідні спостереження в часовому ряді. Такий підхід дозволяє уникнути поступової втрати інформації, яка може виникати під час обробки довгих послідовностей рекурентними мережами.

Особливо ефективними Transformer-моделі є у задачах, де необхідно враховувати складні довгострокові залежності між подіями. До таких задач належать аналіз природної мови, прогнозування часових рядів, моніторинг технічних систем та виробничих процесів. У багатьох практичних сценаріях аномалія може бути пов'язана не лише з поточним значенням сигналу, а й із

подіями, що відбулися значно раніше. Завдяки механізму самоуваги модель здатна враховувати подібні взаємозв'язки та використовувати їх під час прийняття рішення [28].

Типова архітектура Transformer складається з двох основних частин: енкодера та декодера. Енкодер відповідає за формування внутрішнього представлення вхідної послідовності, тоді як декодер використовує отриману інформацію для формування вихідних даних. Приклад архітектури Transformer для аналізу часових рядів наведено на рисунку 2.9. Центральним елементом моделі є багатоголовковий механізм уваги (multi-head attention), який дозволяє одночасно аналізувати різні типи залежностей між елементами послідовності.

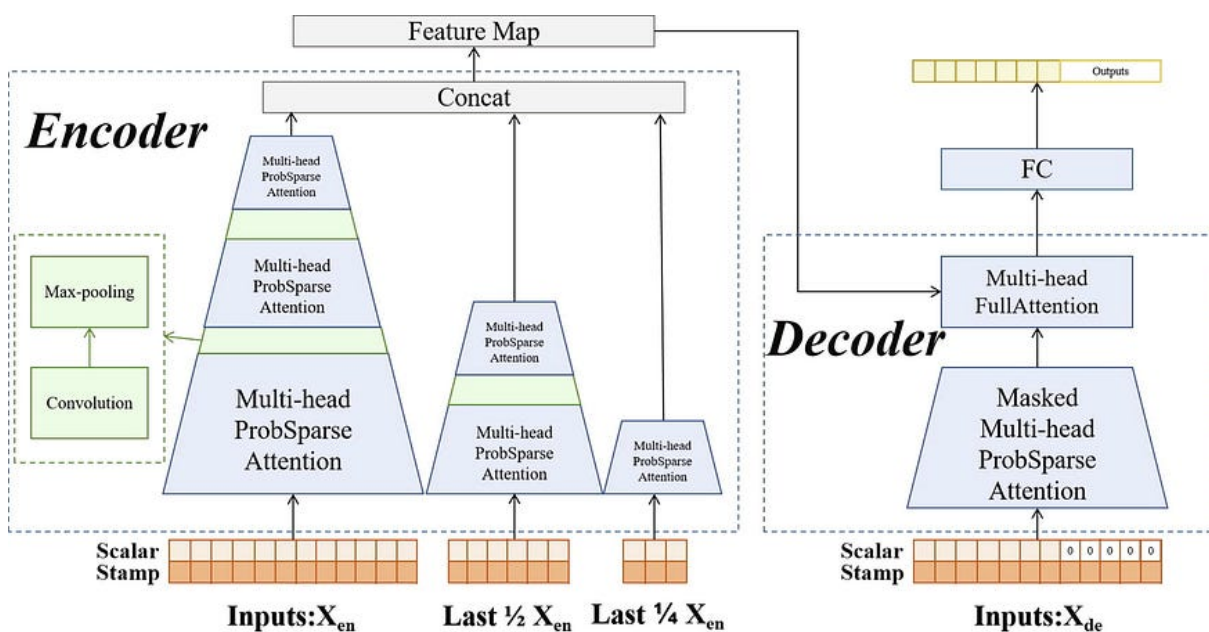


Рисунок 2.9 – Архітектура Transformer-моделі для аналізу часових рядів [28]

У задачах виявлення аномалій Transformer може використовуватися для прогнозування майбутніх значень часового ряду або реконструкції вхідної послідовності. Після навчання на нормальних даних модель формує уявлення про типову поведінку системи. Якщо фактичні значення суттєво

відрізняються від прогнозованих або реконструйованих, таке відхилення може розглядатися як ознака аномалії.

До переваг Transformer належать здатність ефективно працювати з довгими послідовностями, можливість врахування складних нелінійних залежностей та висока якість моделювання часових процесів. Крім того, на відміну від рекурентних мереж, архітектура допускає паралельну обробку даних, що дозволяє суттєво прискорити процес навчання. Водночас такі моделі характеризуються значною кількістю параметрів, високими вимогами до обчислювальних ресурсів та потребують великих обсягів даних для досягнення стабільних результатів.

Розглянуті неймережеві підходи демонструють різні способи розв'язання задачі виявлення аномалій у часових рядах. Автоенкодери орієнтовані на аналіз помилки реконструкції даних, LSTM-моделі забезпечують ефективне врахування часових залежностей завдяки механізму довгострокової пам'яті, тоді як Transformer-архітектури використовують механізм самоуваги для виявлення складних взаємозв'язків між елементами послідовності.

Незважаючи на відмінності в принципах роботи, усі розглянуті підходи дозволяють виявляти аномалії, які важко або неможливо визначити за допомогою статистичних методів чи класичних алгоритмів машинного навчання. Водночас оцінювання ефективності таких моделей потребує використання спеціалізованих метрик, здатних враховувати особливості задачі виявлення аномалій та дисбаланс між нормальними й аномальними спостереженнями.

2.4 Метрики оцінювання ефективності методів виявлення аномалій

Оцінювання ефективності методів виявлення аномалій є важливим етапом аналізу, оскільки дозволяє об'єктивно порівнювати різні підходи та визначати якість їх роботи. На відміну від багатьох задач класифікації, у

задачах anomaly detection аномальні спостереження зазвичай становлять лише незначну частину вибірки, що призводить до суттєвого дисбалансу класів. У таких умовах використання лише однієї метрики часто не дає повного уявлення про ефективність моделі, тому для оцінювання результатів застосовується комплекс показників, кожен з яких характеризує окремий аспект якості виявлення аномалій.

Основою для розрахунку більшості метрик класифікації є матриця помилок (Confusion Matrix), яка відображає співвідношення між фактичними та прогнозованими результатами моделі (рисунок 2.10).

	Predicted Positive	Predicted Negative
Actual Positive	TP <i>True Positive</i>	FN <i>False Negative</i>
Actual Negative	FP <i>False Positive</i>	TN <i>True Negative</i>

Рисунок 2.10 – Матриця помилок для задач бінарної класифікації [29]

Матриця помилок складається з чотирьох основних компонентів [29].

До істинно позитивних результатів (True Positive, TP) належать випадки, коли модель правильно виявила аномалію. Істинно негативні результати (True Negative, TN) відповідають ситуаціям, коли нормальне спостереження було правильно класифіковане як нормальне. Хибно позитивними результатами (False Positive, FP) називають випадки, коли нормальний об'єкт помилково визначається як аномальний. Відповідно, хибно негативні результати (False Negative, FN) виникають тоді, коли реальна аномалія залишається невиявленою та класифікується як нормальне спостереження.

Однією з основних особливостей задач виявлення аномалій є значний дисбаланс між кількістю нормальних та аномальних спостережень. У більшості практичних сценаріїв аномалії виникають рідко та становлять лише невелику частину всіх доступних даних. Наприклад, під час моніторингу технічних систем переважна більшість записів відповідає нормальному режиму роботи, тоді як аварійні ситуації або збої трапляються відносно нечасто.

За таких умов метрика Ассигасу не є достатньо інформативною, оскільки вона показує загальну частку правильних прогнозів, але не відображає, які саме об'єкти були правильно класифіковані. Якщо аномалії становлять лише 1% вибірки, модель, яка завжди визначає всі спостереження як нормальні, може мати близько 99 % точності, але при цьому не виявити жодної реальної аномалії. Тому високе значення Ассигасу у задачах anomaly detection може створювати хибне враження про якість моделі.

Саме тому під час оцінювання методів виявлення аномалій особлива увага приділяється метрикам, які дозволяють окремо аналізувати якість виявлення аномального класу. До таких показників належать Precision, Recall, F1-міра, а також метрики, що оцінюють співвідношення між правильними та помилковими спрацюваннями моделі.

Однією з ключових метрик оцінювання в задачах виявлення аномалій є Precision. Вона характеризує частку правильно виявлених аномалій серед усіх спостережень, які модель класифікувала як аномальні:

$$Precision = \frac{TP}{TP+FP}. \quad (2.5)$$

Високе значення Precision свідчить про низьку кількість хибно позитивних спрацювань та високу достовірність виявлених аномалій.

Метрика Recall відображає здатність моделі знаходити реальні аномалії. Вона показує, яка частка фактично наявних аномальних спостережень була успішно виявлена:

$$Recall = \frac{TP}{TP+FN}. \quad (2.6)$$

Високе значення Recall означає низьку кількість пропущених аномалій, що є особливо важливим у задачах моніторингу та контролю безпеки.

Оскільки між Precision та Recall часто існує компроміс, для комплексного оцінювання якості моделі використовується F1-міра. Вона являє собою гармонійне середнє значень Precision та Recall і дозволяє враховувати обидва показники одночасно:

$$F1 = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}. \quad (2.7)$$

Високе значення F1-міри свідчить про збалансовану роботу моделі як щодо виявлення аномалій, так і щодо мінімізації помилкових спрацювань.

Для комплексного оцінювання якості моделей виявлення аномалій широко використовуються ROC-крива та показник ROC-AUC. ROC-крива (Receiver Operating Characteristic) відображає залежність між часткою правильно виявлених аномалій (True Positive Rate) та часткою хибних спрацювань (False Positive Rate) за різних значень порогу класифікації. Площа під ROC-кривою, яка позначається як ROC-AUC (Area Under the Curve), характеризує здатність моделі відрізнити аномальні спостереження від нормальних. Значення ROC-AUC може змінюватися в межах від 0,5 для випадкової класифікації до 1,0 для ідеальної моделі.

Однак у задачах виявлення аномалій, де кількість аномальних спостережень є значно меншою за кількість нормальних, ROC-AUC не

завжди повністю відображає реальну якість моделі. У таких випадках більш інформативною вважається PR-крива (Precision-Recall Curve), яка показує залежність між значеннями Precision та Recall за різних порогів класифікації. Відповідний показник PR-AUC визначає площу під PR-кривою та характеризує здатність моделі одночасно виявляти аномалії та мінімізувати кількість помилкових спрацювань.

Порівняння ROC- та PR-кривих наведено на рисунку 2.11. Якщо ROC-AUC дозволяє оцінити загальну здатність моделі розрізняти класи, то PR-AUC більшою мірою зосереджується на якості виявлення позитивного класу. Саме тому в задачах anomaly detection, де аномалії є рідкісними подіями, показник PR-AUC часто розглядається як більш інформативний критерій оцінювання ефективності моделі [30].

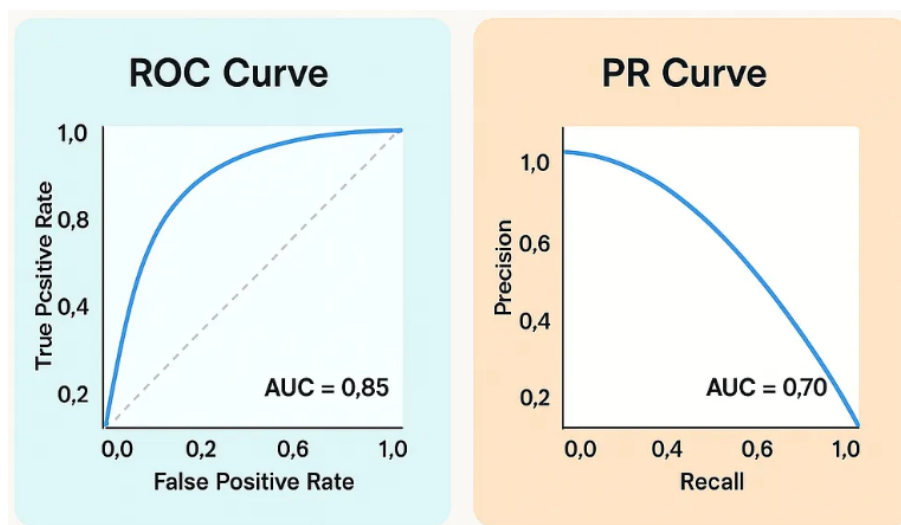


Рисунок 2.11 – Порівняння ROC-кривої та PR-кривої [30]

Під час аналізу часових рядів оцінювання якості виявлення аномалій має певні особливості. На відміну від звичайної класифікації, аномалія в часовому ряді часто являє собою не окреме спостереження, а певний інтервал часу, протягом якого система перебуває в аномальному стані. У таких випадках важливим є не лише факт виявлення аномалії, а й момент її виявлення.

Однією з характерних проблем є затримка виявлення. Навіть якщо модель правильно визначила аномальний стан, надмірно пізнє спрацювання може суттєво знизити практичну цінність такого результату. З цієї причини під час оцінювання часових рядів часто використовується поняття *anomaly window* – часового інтервалу, у межах якого спрацювання моделі вважається коректним. Якщо модель виявила аномалію всередині такого вікна, результат зараховується як успішне виявлення незалежно від конкретного моменту спрацювання.

Подібний підхід використовується в спеціалізованих наборах даних для дослідження методів виявлення аномалій у часових рядах, зокрема в NAB (Numenta Anomaly Benchmark). Це дозволяє враховувати не лише точність виявлення аномалій, а й своєчасність реакції моделі на зміну поведінки системи.

3 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ У ЧАСОВИХ РЯДАХ

Розроблення інтелектуальної системи виявлення аномалій у часових рядах виконано з використанням мови програмування Python, яка забезпечує широкий набір інструментів для обробки даних, машинного навчання та створення програмних застосунків. Реалізацію, навчання та експериментальне дослідження моделей здійснено у хмарному середовищі Google Colab, що надало можливість використовувати графічні процесори для прискорення обчислень.

Для побудови та навчання нейромережових моделей використано бібліотеки TensorFlow і Keras, які забезпечують засоби створення, налаштування та оптимізації архітектур глибокого навчання. Реалізацію методів класичного машинного навчання, попередню обробку даних і оцінювання результатів виконано за допомогою бібліотеки scikit-learn.

Для створення програмного інтерфейсу системи використано фреймворк Streamlit, який забезпечує можливість інтерактивної взаємодії користувача із розробленою системою, завантаження часових рядів, виконання виявлення аномалій та візуалізації отриманих результатів.

Використання графічних процесорів типу NVIDIA T4 у середовищі Google Colab дозволило суттєво скоротити час навчання нейромережових моделей, зокрема LSTM Autoencoder та Transformer Autoencoder, що характеризуються високою обчислювальною складністю порівняно з класичними методами машинного навчання.

3.1 Опис набору даних та підготовка даних

Для проведення експериментального дослідження було використано набір даних Numenta Anomaly Benchmark (NAB) [31], який є одним із найбільш поширених відкритих наборів даних для оцінювання алгоритмів

виявлення аномалій у часових рядах. Особливістю NAB є наявність часових рядів різної природи, що дозволяє дослідити ефективність моделей у різних сценаріях застосування.

Кожен часовий ряд у наборі даних представлений у форматі CSV та містить два атрибути: часову мітку timestamp і числове значення value. Розмітка аномалій зберігається окремо у файлі combined_windows.json у вигляді часових інтервалів, що визначають аномальні ділянки ряду.

На відміну від класичних задач класифікації, у наборі даних NAB аномалії задаються не окремими точками, а часовими інтервалами (anomaly windows). Аномалія вважається успішно виявленою, якщо модель позначила хоча б одну точку всередині відповідного інтервалу. Такий підхід дозволяє врахувати можливу затримку спрацювання алгоритму та відповідає специфіці практичних систем моніторингу.

Первинний аналіз набору даних показав наявність 58 часових рядів, об'єднаних у шість категорій: artificialNoAnomaly, artificialWithAnomaly, realAWScloudwatch, realAdExchange, realKnownCause, realTraffic та realTweets (рисуюнок 3.1). Для проведення експериментів використовувалися лише ряди, що містять аномальні інтервали.

```
NAB already exists
NAB directory: /content/NAB
Data directory exists: True
Labels directory exists: True

Data categories:
- artificialNoAnomaly
- artificialWithAnomaly
- realAWScloudwatch
- realAdExchange
- realKnownCause
- realTraffic
- realTweets

Label files:
- combined_labels.json
- combined_windows.json
- combined_windows_tiny.json

Total CSV files found: 58
```

Рисуюнок 3.1 – Аналіз набору даних

Формування експериментальної вибірки здійснювалося за такими критеріями:

- довжина часового ряду не менше 2000 спостережень;
- перша аномалія виникає не раніше 20 % довжини ряду;
- наявність щонайменше одного аномального інтервалу;
- достатній обсяг нормальних даних для формування часових вікон максимального розміру.

Вибір порогу у 20% обумовлений необхідністю забезпечення достатньої кількості нормальних спостережень для навчання моделей, зокрема автоенкодерів, які потребують переважно безаномальних даних.

За результатами аналізу було відібрано 10 часових рядів різної природи (рисунок 3.2), які охоплюють сценарії моніторингу інфраструктури, температурних показників, транспортних потоків, соціальної активності та штучно згенерованих процесів.

```
SELECTED_SERIES = [
    "artificialWithAnomaly/art_increase_spike_density.csv",
    "artificialWithAnomaly/art_daily_jumpup.csv",

    "realAWScloudwatch/rds_cpu_utilization_e47b3b.csv",
    "realAWScloudwatch/ec2_network_in_257a54.csv",

    "realKnownCause/ambient_temperature_system_failure.csv",
    "realKnownCause/nyc_taxi.csv",

    "realTraffic/occupancy_t4013.csv",
    "realTraffic/occupancy_6005.csv",

    "realTweets/Twitter_volume_GOOG.csv",
    "realTweets/Twitter_volume_IBM.csv"
]
```

Рисунок 3.2 – Відібрані ряди

Для уникнення витоку інформації та забезпечення коректного навчання моделей формування навчальної та тестової вибірок виконувалося окремо для кожного часового ряду. Межа навчальної вибірки визначалась індивідуально на основі положення першого аномального інтервалу.

Кінець навчальної вибірки обчислювався за формулою:

$$\text{train_end} = \text{first_anomaly_index} - \text{window_size}, \quad (3.1)$$

де `first_anomaly_index` – індекс початку першого аномального інтервалу;
`window_size` – розмір часового вікна.

Такий підхід гарантує відсутність аномальних спостережень у навчальній вибірці та виключає потрапляння фрагментів аномалій до часових вікон, що використовуються для навчання моделей.

Для експериментів із нейромережевими моделями було обрано розміри часових вікон 10, 20, 30 та 50 спостережень. Додаткова перевірка показала, що для всіх відібраних часових рядів кількість навчальних вікон перевищує 500 навіть при максимальному значенні `window_size = 50`, що є достатнім для стабільного навчання моделей (рисунок 3.3).

	series	window_size	train_points	train_windows	valid
0	artificialWithAnomaly/art_increase_spike_densi...	10	1805	1795	True
1	artificialWithAnomaly/art_increase_spike_densi...	20	1805	1785	True
2	artificialWithAnomaly/art_increase_spike_densi...	30	1805	1775	True
3	artificialWithAnomaly/art_increase_spike_densi...	50	1805	1755	True
4	artificialWithAnomaly/art_daily_jumpsup.csv	10	2786	2776	True
5	artificialWithAnomaly/art_daily_jumpsup.csv	20	2786	2766	True
6	artificialWithAnomaly/art_daily_jumpsup.csv	30	2786	2756	True
7	artificialWithAnomaly/art_daily_jumpsup.csv	50	2786	2736	True
8	realAWSCloudwatch/rds_cpu_utilization_e47b3b.csv	10	845	835	True
9	realAWSCloudwatch/rds_cpu_utilization_e47b3b.csv	20	845	825	True
10	realAWSCloudwatch/rds_cpu_utilization_e47b3b.csv	30	845	815	True
11	realAWSCloudwatch/rds_cpu_utilization_e47b3b.csv	50	845	795	True
12	realAWSCloudwatch/ec2_network_in_257a54.csv	10	1437	1427	True
13	realAWSCloudwatch/ec2_network_in_257a54.csv	20	1437	1417	True
14	realAWSCloudwatch/ec2_network_in_257a54.csv	30	1437	1407	True

Рисунок 3.3 – Фрагмент результату перевірки

Для забезпечення можливості оцінювання якості моделей часові інтервали аномалій було перетворено на точкові мітки. Кожному спостереженню часового ряду присвоювалося значення $is_anomaly = 1$, якщо його часова позначка потрапляла в межі відповідного аномального інтервалу, та $is_anomaly = 0$ в іншому випадку.

Перед формуванням часових вікон виконано масштабування числових значень часового ряду. Необхідність цього етапу обумовлена використанням нейромережових моделей LSTM Autoencoder та Transformer Autoencoder, чутливих до масштабу вхідних даних.

Для нормалізації значень застосовано метод стандартизації, реалізований за допомогою класу StandardScaler бібліотеки scikit-learn. Параметри масштабування обчислювалися виключно на навчальній вибірці, після чого отримані значення середнього та стандартного відхилення використовувалися для перетворення тестових даних.

Такий підхід дозволяє уникнути витоку інформації з тестової вибірки та забезпечує коректність експериментального дослідження.

У результаті масштабування значення навчальної вибірки було приведено до нульового середнього та одиничного стандартного відхилення.

Для врахування часових залежностей у даних вхідні послідовності було сформовано за допомогою методу ковзного вікна.

Кожне часове вікно являє собою послідовність із фіксованої кількості попередніх спостережень, що використовується моделлю для аналізу локального контексту часового ряду.

У межах дослідження довжину вікна встановлено на рівні 30 спостережень. Для кожного моменту часу формувалася послідовність, що містила 30 попередніх масштабованих значень ознаки `value`, тоді як відповідна мітка визначалася за поточним спостереженням.

Формування часових вікон здійснювалося окремо для навчальної та тестової вибірок. Навчальна вибірка містила лише нормальні

спостереження, що забезпечило можливість навчання автоенкодерів на прикладах штатної поведінки системи.

Результати формування часових вікон наведено на рисунку 3.4. Після перетворення навчальну вибірку представлено у вигляді 1745 часових вікон, а тестову – 2227 вікон. Отримані масиви мають розмірність $(N, 30, 1)$, де N – кількість сформованих послідовностей, 30 – довжина вікна, а 1 – кількість вхідних ознак.

```
Train windows: (1745, 30, 1)
Test windows: (2227, 30, 1)

Train anomalies: 0
Test anomalies: 403
```

Рисунок 3.4 – Результати формування часових вікон для навчальної та тестової вибірок

Серед сформованих тестових вікон 403 відповідають аномальним спостереженням, тоді як навчальна вибірка не містить аномалій.

3.2 Реалізація методів виявлення аномалій

3.2.1 Реалізація Isolation Forest

Для підвищення якості виявлення аномалій методом Isolation Forest вихідний часовий ряд було доповнено набором додаткових ознак, що характеризують локальні зміни та статистичні властивості сигналу.

Формування ознак здійснювалося на основі числового атрибута value. До набору ознак включено ковзне середнє (rolling_mean), ковзне стандартне відхилення (rolling_std) та різницю між сусідніми значеннями (diff).

Ковзне середнє обчислювалося в межах вікна довжиною 10 спостережень і використовувалося для відображення локального тренду

часового ряду. Ковзне стандартне відхилення, розраховане в межах того самого вікна, характеризувало рівень варіативності сигналу на окремих часових інтервалах.

Ознака `diff` визначалася як різниця між поточним і попереднім значеннями часового ряду та дозволяла враховувати швидкість зміни досліджуваного показника.

Під час обчислення похідних ознак для перших спостережень часового ряду виникали відсутні значення, що пов'язано з особливостями розрахунку ковзного стандартного відхилення та різниці між сусідніми значеннями. Такі значення замінювалися нулем, що дозволило зберегти початкову довжину часового ряду та забезпечити коректне формування вхідних даних для моделі.

Після формування додаткових ознак виконано повторну перевірку набору даних на наявність пропущених значень. Пропусків у вихідному часовому ряді не виявлено, тому додаткова імпутація або видалення записів не застосовувалися.

У результаті для моделі `Isolation Forest` сформовано розширений набір ознак, який містить атрибути `value`, `rolling_mean`, `rolling_std` та `diff`.

Навчання моделі виконувалося на навчальній вибірці, що містила нормальні спостереження. Для моделі задано 200 базових дерев рішень (`n_estimators = 200`). Параметр `contamination` встановлено в значення `auto`, для забезпечення відтворюваності результатів використано `random_state = 42`, а для прискорення обчислень – `n_jobs = -1`.

Після навчання для кожного спостереження обчислювався показник аномальності за допомогою методу `score_samples()`. Оскільки в реалізації `scikit-learn` менші значення цього показника відповідають більш аномальним об'єктам, отримані значення було інвертовано. У результаті більші значення `anomaly score` відповідали вищому ступеню аномальності.

Для перетворення числового показника аномальності у бінарну мітку використовувався перцентильний поріг. У роботі перевірялися пороги, що

відповідають 95-му, 97-му та 99-му перцентилям розподілу anomaly score. Спостереження, значення яких перевищувало відповідний поріг, позначалися як аномальні.

Подальше оцінювання результатів виконувалося з урахуванням часових інтервалів аномалій із набору NAB: аномальне вікно вважалося виявленим, якщо хоча б одна точка всередині нього була позначена моделлю як аномальна.

3.2.2 Реалізація LSTM Autoencoder

Для виявлення аномалій у часових рядах реалізовано модель LSTM Autoencoder, яка використовує механізм реконструкції послідовностей. На відміну від методу Isolation Forest, модель працює не з окремими спостереженнями, а з часовими вікнами фіксованої довжини.

На вхід моделі подавалися послідовності розмірності (window_size, 1), де window_size визначає кількість послідовних спостережень, а 1 відповідає числовій ознаці value. Перед формуванням часових вікон значення часового ряду масштабувалися за допомогою методу стандартизації.

Архітектура моделі складалася з енкодера, шару повторення прихованого представлення та декодера. Енкодер реалізовано за допомогою LSTM-шару з 32 нейронами, який перетворював вхідну послідовність у компактне приховане представлення. Далі шар RepeatVector повторював отриманий вектор відповідно до довжини вхідного вікна. Декодер також реалізовано на основі LSTM-шару з 32 нейронами з параметром return_sequences=True, що забезпечувало відновлення послідовності у початковому часовому форматі. Вихідний шар TimeDistributed(Dense(1)) формував реконструйоване значення для кожного моменту часу.

Архітектуру реалізованої моделі наведено на рисунку 3.5.

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 30, 1)	0
lstm (LSTM)	(None, 32)	4,352
repeat_vector (RepeatVector)	(None, 30, 32)	0
lstm_1 (LSTM)	(None, 30, 32)	8,320
time_distributed (TimeDistributed)	(None, 30, 1)	33

Total params: 12,705 (49.63 KB)
 Trainable params: 12,705 (49.63 KB)
 Non-trainable params: 0 (0.00 B)

Рисунок 3.5 – Архітектура моделі LSTM Autoencoder

Модель навчалася відновлювати вхідні послідовності, тобто вхідні та цільові дані були ідентичними. Для навчання використовувалася лише навчальна вибірка, що містила нормальні спостереження.

Як функцію втрат застосовано середньоквадратичну помилку (mse), а оптимізацію параметрів виконано за допомогою алгоритму Adam із коефіцієнтом навчання 0,001.

Максимальна кількість епох навчання становила 20, розмір пакета дорівнював 64 послідовностям. Для контролю процесу навчання 20% навчальної вибірки використовувалося як валідаційна підвибірка. Для запобігання перенавчанню застосовано механізм ранньої зупинки EarlyStopping із параметром patience = 5 та відновленням ваг моделі, що забезпечили найменше значення функції втрат на валідаційній вибірці.

Після завершення навчання модель використовувалася для реконструкції часових вікон тестової вибірки. Для кожної послідовності обчислювалася помилка реконструкції (reconstruction error) як середнє значення квадрата різниці між вхідними та відновленими даними.

Отримане значення reconstruction error використовувалося як показник аномальності (anomaly score). Чим більшою була помилка реконструкції, тим вищою вважалася ймовірність наявності аномалії у відповідному часовому вікні.

Для формування бінарних міток аномалій застосовувався перцентильний підхід. Часові вікна, значення anomaly score яких перевищували встановлений поріг, позначалися як аномальні. Оцінювання якості моделі виконувалося з урахуванням часових інтервалів аномалій набору даних NAV.

3.2.3 Реалізація Transformer Autoencoder

Для виявлення аномалій у часових рядах також реалізовано модель Transformer Autoencoder. Як і LSTM Autoencoder, ця модель працює з часовими вікнами фіксованої довжини та навчається відновлювати вхідні послідовності.

На вхід моделі подавалися послідовності розмірності (window_size, 1), де window_size визначає кількість послідовних спостережень, а 1 відповідає числовій ознаці value. Перед поданням до моделі значення часового ряду масштабувалися за допомогою методу стандартизації.

На першому етапі вхідні значення перетворювалися у векторне представлення розмірності d_model = 32 за допомогою повнозв'язного шару Dense. Для збереження інформації про порядок елементів у часовому вікні використовувалося позиційне кодування, реалізоване через окремий шар PositionalEmbedding.

Основним елементом моделі був блок самоуваги MultiHeadAttention з двома головами уваги (num_heads = 2). Після шару уваги застосовувалися Dropout, залишкове з'єднання Add та нормалізація LayerNormalization. Далі використовувався feed-forward блок, що складався з двох повнозв'язних шарів із розмірністю ff_dim = 64 та d_model = 32.

Після обробки послідовності Transformer-блоком отримане представлення вирівнювалося за допомогою шару Flatten і стискалося до латентного простору розмірності latent_dim = 16. Декодер відновлював представлення до розмірності window_size × d_model, після чого шар

TimeDistributed(Dense(1)) формував реконструйоване значення для кожного моменту часу.

Архітектуру реалізованої моделі Transformer Autoencoder наведено на рисунку 3.6.

Model: "functional_6"

Layer (type)	Output Shape	Param #	Connected to
input_layer_6 (InputLayer)	(None, 50, 1)	0	-
dense_18 (Dense)	(None, 50, 32)	64	input_layer_6[0]...
positional_embeddi... (PositionalEmbeddi...	(None, 50, 32)	1,600	dense_18[0][0]
multi_head_attenti... (MultiHeadAttentio...	(None, 50, 32)	4,224	positional_embed... positional_embed...
dropout_5 (Dropout)	(None, 50, 32)	0	multi_head_atten...
add_4 (Add)	(None, 50, 32)	0	positional_embed... dropout_5[0][0]
layer_normalizatio... (LayerNormalizatio...	(None, 50, 32)	64	add_4[0][0]
dense_19 (Dense)	(None, 50, 64)	2,112	layer_normalizat...
dense_20 (Dense)	(None, 50, 32)	2,080	dense_19[0][0]
dropout_6 (Dropout)	(None, 50, 32)	0	dense_20[0][0]
add_5 (Add)	(None, 50, 32)	0	layer_normalizat... dropout_6[0][0]
layer_normalizatio... (LayerNormalizatio...	(None, 50, 32)	64	add_5[0][0]
flatten (Flatten)	(None, 1600)	0	layer_normalizat...
dense_21 (Dense)	(None, 16)	25,616	flatten[0][0]
dense_22 (Dense)	(None, 1600)	27,200	dense_21[0][0]
reshape (Reshape)	(None, 50, 32)	0	dense_22[0][0]
time_distributed_6 (TimeDistributed)	(None, 50, 1)	33	reshape[0][0]

Total params: 63,057 (246.32 KB)

Trainable params: 63,057 (246.32 KB)

Non-trainable params: 0 (0.00 B)

Рисунок 3.6 – Архітектура моделі Transformer Autoencoder

Модель навчалася відновлювати вхідні часові послідовності, тому вхідні та цільові дані були ідентичними. Як функцію втрат використано

середньоквадратичну помилку (mse), а оптимізацію параметрів виконано за допомогою алгоритму Adam із коефіцієнтом навчання 0,001.

Максимальна кількість епох навчання становила 30, розмір пакета дорівнював 64 послідовностям. Для контролю навчання 20% навчальної вибірки використовувалося як валідаційна підвибірка. Для запобігання перенавчанню застосовано механізм ранньої зупинки EarlyStopping із параметром patience = 5 та відновленням найкращих ваг моделі.

Після навчання модель виконувала реконструкцію тестових часових вікон. Для кожної послідовності обчислювалася помилка реконструкції як середнє значення квадрата різниці між вхідною та відновленою послідовністю.

Отримана помилка реконструкції використовувалася як показник аномальності. Чим більшою була помилка реконструкції, тим вищим вважався рівень відхилення відповідного часового вікна від нормальної поведінки.

Для формування бінарних міток аномалій застосовувався перцентильний поріг. Часові вікна, значення anomaly score яких перевищували встановлений поріг, позначалися як аномальні.

3.3 Реалізація програмного інтерфейсу системи

Для забезпечення взаємодії користувача з розробленою системою виявлення аномалій було реалізовано прототип системи на основі фреймворку Streamlit.

Загальну архітектуру взаємодії користувача із системою наведено на рисунку 3.7. Система реалізує послідовний конвеєр обробки даних, який охоплює етапи завантаження часового ряду, підготовки даних, аналізу часового ряду моделлю машинного навчання, визначення аномальних точок та візуалізації результатів.



Рисунок 3.7 – Схема взаємодії користувача з програмним інтерфейсом системи

Взаємодія користувача із системою здійснюється за таким сценарієм:

- користувач завантажує CSV-файл із часовим рядом через вебінтерфейс;
- система виконує зчитування та перевірку структури вхідних даних;
- здійснюється попередня обробка даних, зокрема масштабування значень та формування часових вікон;
- підготовлені дані передаються до попередньо навченої моделі машинного навчання;
- на основі отриманих результатів обчислюється показник аномальності та визначаються аномальні точки;
- результати аналізу відображаються у вигляді інтерактивних графіків і статистичних показників.

Вхідні дані подаються у форматі CSV та містять два стовпці: часову позначку `timestamp` і числове значення `value`. Після завантаження файлу система автоматично виконує перевірку коректності структури даних, перетворення типів, обробку пропущених значень та сортування записів за часовою ознакою.

Для забезпечення стабільності роботи програмного прототипу реалізовано механізм автоматичного завантаження попередньо збережених параметрів моделі та засобів попередньої обробки даних. Це дозволяє виконувати аналіз нових часових рядів без необхідності повторного навчання моделі та додаткового налаштування системи.

Інтерфейс системи виявлення аномалій наведено на рисунку 3.8. Розроблений інтерфейс забезпечує виконання повного циклу аналізу часового ряду: від завантаження вхідних даних до візуалізації результатів та експорту отриманих даних.

Інтелектуальна система виявлення аномалій у часових рядах

Система використовує попередньо навчену модель LSTM Autoencoder для автоматичного виявлення аномалій у часових рядах.

Завантажте CSV-файл із часовим рядом



art_dai...umpsup.csv
137.3KB



Кількість точок: 4032

Поріг: 97.0-й перцентиль

Статистика результатів

Кількість аномальних точок

121

Частка аномальних точок

3.00%

Максимальний anomaly score

2899.08

Виявлені аномалії (121 точка)



> Попередній перегляд даних

> Таблиця результатів

Завантажити результати у CSV

Рисунок 3.8 – Інтерфейс системи

Інтерфейс містить такі функціональні елементи:

- область завантаження CSV-файлу з часовим рядом;
- інформаційний блок, що відображає основні характеристики вхідних даних, зокрема кількість точок у часовому ряді та значення порогу виявлення аномалій;
- блок статистичних показників, який містить кількість виявлених аномальних точок, частку аномалій у часовому ряді та максимальне значення показника аномальності;
- інтерактивний графік часового ряду з візуальним виділенням аномальних точок;
- панелі попереднього перегляду вхідних даних і результатів аналізу (рисунки 3.9);
- засіб експорту результатів у форматі CSV.

Попередній перегляд даних	
	value
0	19.7613
1	20.5008
2	19.9616
3	21.4903
4	20.1877
5	19.9231
6	21.6984
7	20.8788
8	18.4462
9	18.7108

Таблиця результатів					
	timestamp	value	anomaly_score	is_anomaly	label
2998	2014-04-11 09:50:00	129.4379	740.9326	☐	normal
2999	2014-04-11 09:55:00	125.21	804.9584	☐	normal
3000	2014-04-11 10:00:00	138.1185	884.3689	☒	anomaly
3001	2014-04-11 10:05:00	159.2597	989.66	☒	anomaly
3002	2014-04-11 10:10:00	147.3992	1077.4396	☒	anomaly
3003	2014-04-11 10:15:00	150.0462	1170.1243	☒	anomaly
3004	2014-04-11 10:20:00	150.0825	1262.6382	☒	anomaly
3005	2014-04-11 10:25:00	154.7985	1360.9763	☒	anomaly
3006	2014-04-11 10:30:00	130.6001	1429.1202	☒	anomaly
3007	2014-04-11 10:35:00	136.6984	1506.4049	☒	anomaly

Завантажити результати у CSV

Рисунок 3.9– Відображення вхідних даних і результатів аналізу в інтерфейсі програмного прототипу системи

Для побудови графіків використано бібліотеку Plotly, що забезпечує інтерактивну взаємодію з результатами аналізу. Користувач має можливість змінювати масштаб графіка, виділяти окремі часові інтервали, переміщувати область перегляду, приховувати або відображати окремі елементи легенди, а також зберігати сформовані візуалізації.

Виявлені аномалії відображаються на графіку у вигляді окремих маркерів, що дозволяє швидко оцінити характер відхилень та їх локалізацію в часовому ряді. Панелі перегляду даних забезпечують доступ до детальної інформації про вхідний часовий ряд і результати аналізу, зокрема часові позначки, значення досліджуваного показника, обчислені значення показника аномальності та результати класифікації кожної точки.

Реалізований механізм експорту дозволяє зберігати результати аналізу у форматі CSV для подальшого опрацювання, формування звітів або використання в інших інформаційних системах.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

Оцінювання якості моделей виконувалося з урахуванням специфіки набору даних NAB, у якому аномалії представлені часовими інтервалами, а не окремими точками. Аномальний інтервал вважався виявленим, якщо модель позначала як аномальну хоча б одну точку в його межах.

Для моделі Isolation Forest досліджувався вплив набору ознак і значення перцентильного порогу на результати виявлення аномалій. Для моделей LSTM Autoencoder та Transformer Autoencoder додатково аналізувався вплив довжини часового вікна.

Як основний показник повноти виявлення використовувався Window Recall, що визначає частку виявлених аномальних інтервалів від їх загальної кількості. Для оцінювання здатності моделей ранжувати спостереження за рівнем аномальності використовувався показник PR-AUC, який обчислювався на основі значень anomaly score та не залежав від обраного порогу.

Додатково аналізувалася кількість хибних спрацювань, тобто кількість точок, помилково позначених як аномальні поза межами аномальних інтервалів.

Для перетворення значень anomaly score у бінарні мітки застосовувався перцентильний підхід. У межах дослідження перевірялися пороги, що відповідали 95-му, 97-му та 99-му перцентилям розподілу anomaly score.

4.1 Дослідження ефективності Isolation Forest

Для дослідження ефективності моделі Isolation Forest проведено серію експериментів на 10 часових рядах набору даних NAB. У межах

дослідження аналізувався вплив набору ознак та значення перцентильного порогу на результати виявлення аномалій.

Загалом виконано 120 експериментів, що відповідає всім комбінаціям часових рядів, наборів ознак та порогових значень.

Агреговані результати дослідження впливу набору ознак наведено на рисунку 4.1. Найвищі значення Window Recall продемонстрували набори value_mean_std та value_mean_std_diff, тоді як використання лише вихідної ознаки value призвело до значного збільшення кількості хибних спрацювань.

	feature_set	mean_window_recall	mean_pr_auc	mean_false_positive_points
1	value_mean	0.907780	0.31272	338.366667
2	value_mean_std	0.971113	0.31117	104.766667
3	value_mean_std_diff	0.975557	0.30187	78.266667
0	value	0.912223	0.29982	464.900000

Рисунок 4.1 – Середні результати моделі Isolation Forest залежно від набору ознак

Отримані результати свідчать, що використання виключно поточних значень часового ряду не забезпечує моделі достатньої інформації для відокремлення аномальних відхилень від природних коливань сигналу. Унаслідок цього короточасні зміни амплітуди або локальні піки можуть помилково інтерпретуватися як аномалії.

Вплив набору ознак на середню кількість хибних спрацювань наведено на рисунку 4.2. Додавання статистичних характеристик часового ряду дозволило зменшити середню кількість хибних спрацювань із 464,9 для ознаки value до 78,3 для набору value_mean_std_diff.

Найбільше зниження кількості хибних спрацювань спостерігається при послідовному розширенні набору ознак. Додавання ковзного середнього дозволяє враховувати локальний рівень сигналу, ковзного

стандартного відхилення – його мінливість, а ознаки diff – швидкість зміни значень між сусідніми моментами часу.

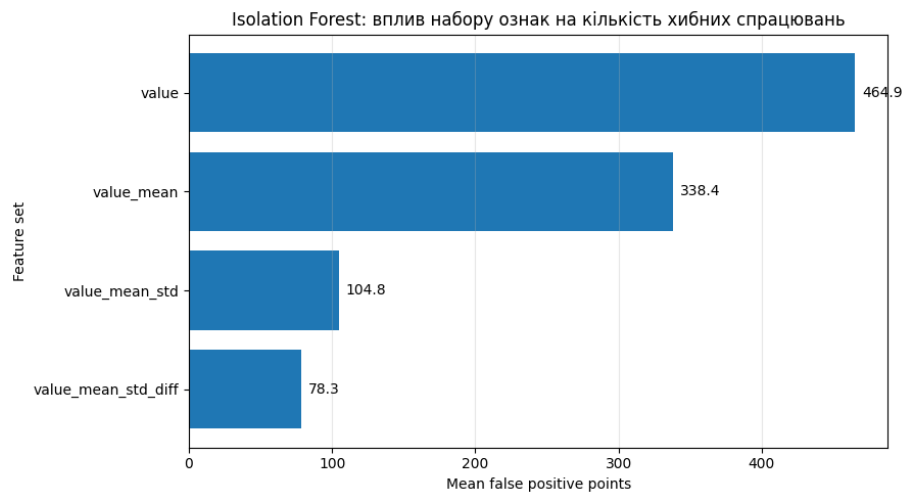


Рисунок 4.2 – Вплив набору ознак на кількість хибних спрацювань моделі Isolation Forest

Отримані результати свідчать, що використання додаткових ознак, які враховують локальну динаміку часового ряду, підвищує стійкість моделі до випадкових коливань. Найкращі результати продемонстрував набір ознак value_mean_std_diff, який забезпечив найвище значення Window Recall при найменшій кількості хибних спрацювань.

Агреговані результати дослідження впливу перцентильного порогу наведено на рисунку 4.3.

	percentile	mean_window_recall	mean_pr_auc	mean_false_positive_points
0	95	0.975000	0.306395	313.050
1	97	0.948335	0.306395	260.975
2	99	0.901670	0.306395	165.700

Рисунок 4.3 – Середні результати моделі Isolation Forest залежно від значення перцентильного порогу

Зі збільшенням значення перцентильного порогу кількість хибних спрацювань зменшується, що підтверджується даними рисунка 4.4. Зокрема, підвищення порогу від 95-го до 99-го перцентилля дозволило знизити середню кількість помилкових спрацювань із 313,1 до 165,7.

Така залежність пояснюється тим, що зі зростанням порогу модель позначає як аномальні лише спостереження з найвищими значеннями показника аномальності, відсікаючи менш виражені відхилення.

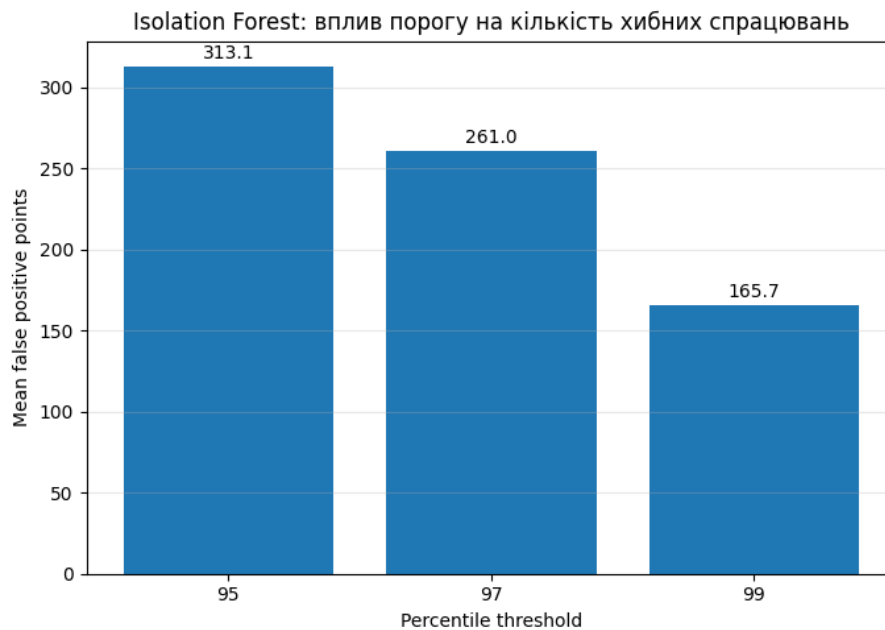


Рисунок 4.4 – Вплив перцентильного порогу на кількість хибних спрацювань моделі Isolation Forest

Водночас підвищення порогу супроводжується зниженням повноти виявлення аномалій. Як показано на рисунку 4.5, середнє значення Window Recall зменшується з 0,975 для 95-го перцентилля до 0,902 для 99-го перцентилля.

Отримані результати свідчать про наявність компромісу між повнотою виявлення аномалій та кількістю хибних спрацювань. З огляду на це для подальших досліджень обрано 97-й перцентиль, який забезпечує збалансоване співвідношення між цими показниками.

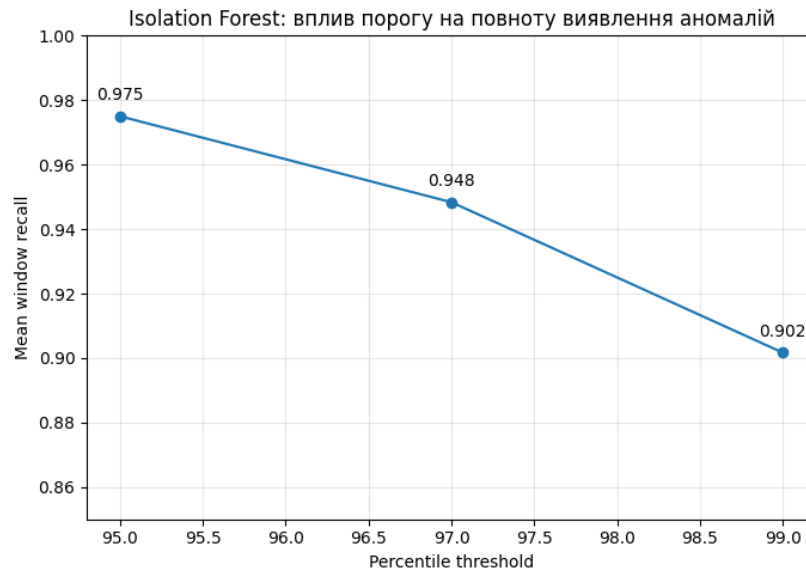


Рисунок 4.5 – Вплив перцентильного порогу на повноту виявлення аномалій моделі Isolation Forest

Приклад роботи моделі для часового ряду `pus_taxi.csv` наведено на рисунку 4.6.

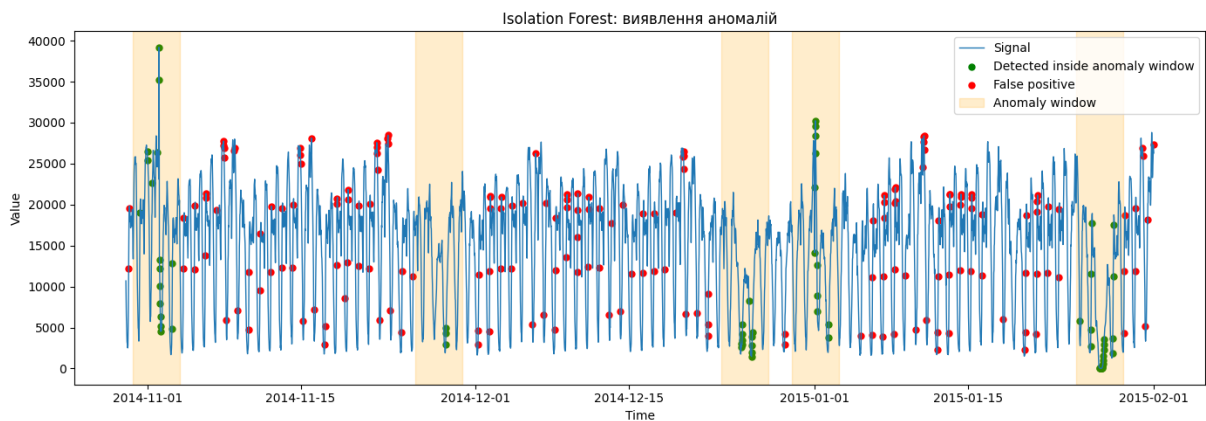


Рисунок 4.6 – Приклад виявлення аномалій моделлю Isolation Forest

На рисунку зеленими маркерами позначено точки, правильно ідентифіковані в межах аномальних інтервалів, червоними – хибні спрацювання, а зафарбованими областями – аномальні інтервали відповідно до розмітки набору даних NAB.

Найкращі результати модель Isolation Forest продемонструвала при використанні набору ознак `value_mean_std_diff` у поєднанні з 97-м перцентилем. Це дозволило забезпечити високу повноту виявлення аномалій при помірній кількості хибних спрацювань.

4.2 Дослідження ефективності LSTM Autoencoder

Для дослідження ефективності моделі LSTM Autoencoder проведено серію експериментів на 10 часових рядах набору даних NAB. У межах дослідження аналізувався вплив розміру часового вікна та значення перцентильного порогу на результати виявлення аномалій.

Динаміка функції втрат під час навчання моделі наведена на рисунку 4.7. Протягом усього процесу навчання спостерігається поступове зменшення значень як навчальної, так і валідаційної помилки без ознак перенавчання. Близькі значення функції втрат на навчальній і валідаційній вибірках свідчать про здатність моделі узагальнювати закономірності часового ряду та коректно відтворювати його нормальну поведінку.

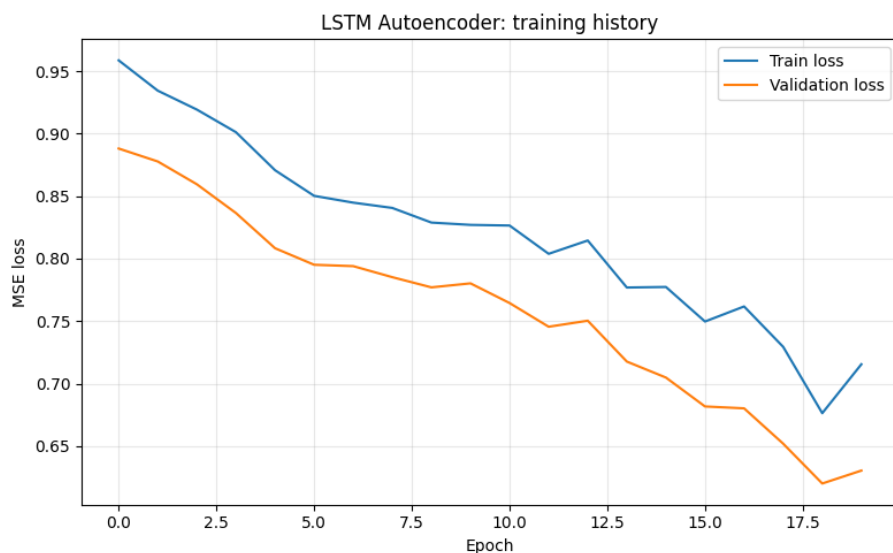


Рисунок 4.7 – Динаміка функції втрат моделі LSTM Autoencoder під час навчання

Для визначення оптимального розміру часового вікна проведено серію експериментів із використанням вікон довжиною 10, 20, 30 та 50 спостережень. Узагальнені результати наведено на рисунку 4.8.

	model	series	window_size	percentile	epochs	final_loss	final_val_loss	detected_windows	total_windows	window_recall	false_positive_points	pr_auc
0	LSTM Autoencoder	artificialWithAnomaly/art_increase_spike_densi...	10	97	20	0.094744	0.073047	1	1	1.0	56	0.2219
1	LSTM Autoencoder	artificialWithAnomaly/art_increase_spike_densi...	20	97	20	0.622355	0.605210	1	1	1.0	37	0.2465
2	LSTM Autoencoder	artificialWithAnomaly/art_increase_spike_densi...	30	97	20	0.669342	0.583128	1	1	1.0	37	0.2673
3	LSTM Autoencoder	artificialWithAnomaly/art_increase_spike_densi...	50	97	20	0.793258	0.889499	1	1	1.0	18	0.3136

Рисунок 4.8 – Результати дослідження впливу розміру часового вікна на якість роботи моделі LSTM Autoencoder

Залежність значення PR-AUC від розміру часового вікна наведено на рисунку 4.9. Отримані результати свідчать про зростання значення PR-AUC зі збільшенням розміру вікна. Зокрема, збільшення довжини вікна від 10 до 50 спостережень дозволило підвищити значення PR-AUC з 0,2219 до 0,3136. Це свідчить про важливість урахування часового контексту під час виявлення аномалій у послідовних даних.

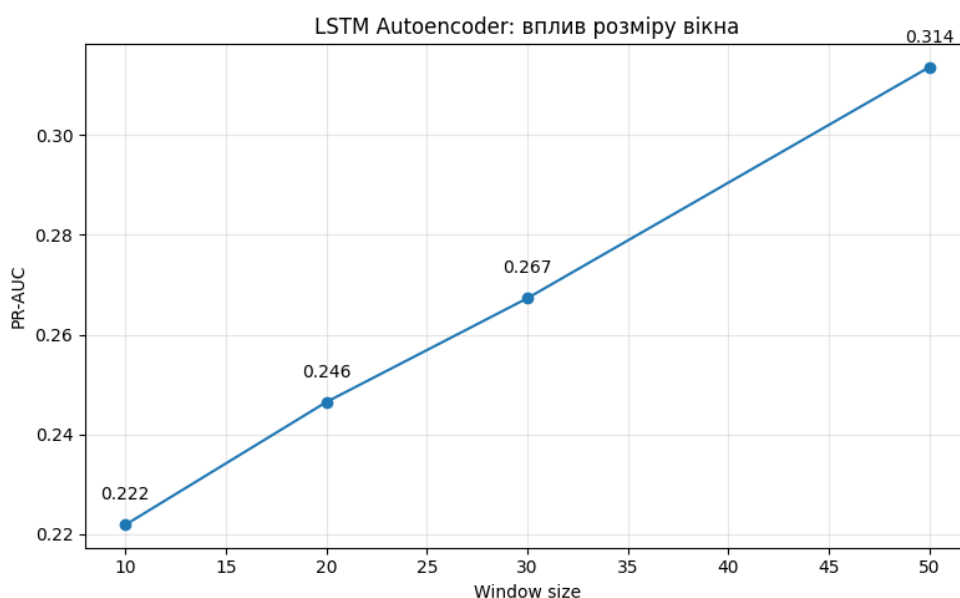


Рисунок 4.9 – Вплив розміру часового вікна на значення PR-AUC моделі LSTM Autoencoder

Використання коротких послідовностей обмежує обсяг контекстної інформації, доступної моделі, через що вона гірше розрізняє аномальні та нормальні фрагменти сигналу. Збільшення довжини вікна дозволяє враховувати локальну структуру та періодичність часового ряду, що позитивно впливає на якість виявлення аномалій. За результатами експерименту для подальших досліджень обрано розмір вікна, що дорівнює 50.

Узагальнені результати оцінювання моделі для різних значень перцентильного порогу наведено на рисунку 4.10.

	percentile	mean_window_recall	mean_pr_auc	mean_false_positive_points	mean_epochs
0	95	0.79667	0.43671	92.7	19.6
1	97	0.69667	0.43671	45.3	19.6
2	99	0.58667	0.43671	6.7	19.6

Рисунок 4.10 – Середні результати моделі LSTM Autoencoder залежно від значення перцентильного порогу

Аналогічно до результатів моделі Isolation Forest, збільшення значення перцентильного порогу призводить до зменшення кількості хибних спрацювань, однак супроводжується зниженням повноти виявлення аномалій. Зі збільшенням порогу від 95-го до 99-го перцентилія середнє значення Window Recall зменшилося з 0,797 до 0,587, тоді як середня кількість хибних спрацювань скоротилася з 92,7 до 6,7.

Отримані результати свідчать про наявність компромісу між повнотою виявлення аномалій та кількістю хибних спрацювань. Для подальших досліджень обрано 97-й перцентиль, який забезпечує збалансоване співвідношення між цими показниками.

Результати оцінювання моделі на окремих часових рядах наведено на рисунку 4.11.

	series	window_recall	false_positive_points	pr_auc	epochs
10	realAWScloudwatch/ec2_network_in_257a54.csv	1.0000	0	0.8336	20
19	realTraffic/occupancy_t4013.csv	0.5000	0	0.7610	20
4	artificialWithAnomaly/art_daily_jumpsup.csv	1.0000	0	0.6508	20
16	realKnownCause/nyc_taxi.csv	0.8000	8	0.5339	20
13	realKnownCause/ambient_temperature_system_fail...	1.0000	36	0.3275	20
1	artificialWithAnomaly/art_increase_spike_densi...	1.0000	18	0.3191	20
25	realTweets/Twitter_volume_GOOG.csv	0.6667	107	0.2985	20
22	realTraffic/occupancy_6005.csv	0.0000	23	0.2930	20
28	realTweets/Twitter_volume_IBM.csv	1.0000	165	0.2559	20
7	realAWScloudwatch/rds_cpu_utilization_e47b3b.csv	0.0000	96	0.0938	16

Рисунок 4.11 – Результати моделі LSTM Autoencoder для окремих часових рядів

Аналіз результатів показав суттєву залежність ефективності моделі від характеру аномалій. Найкращі результати отримано для часових рядів із вираженими локальними відхиленнями, зокрема для рядів `ambient_temperature_system_failure.csv`, `ec2_network_in_257a54.csv` та `art_daily_jumpsup.csv`, для яких значення Window Recall становило 1,0.

Водночас для рядів `rds_cpu_utilization_e4f7b3.csv` та `occupancy_6005.csv` модель не змогла виявити жодного аномального інтервалу. Це свідчить про те, що ефективність LSTM Autoencoder суттєво залежить від ступеня відмінності аномальних ділянок від нормальної поведінки сигналу.

Для детальнішого аналізу причин пропуску аномалій досліджено часовий ряд `occupancy_6005.csv`. Як видно з рисунка 4.12, значення `reconstruction error` в межах аномального інтервалу не перевищують встановлений поріг. Це свідчить про те, що модель успішно відтворює структуру аномальної ділянки, сприймаючи її як нормальну поведінку.

Така особливість є характерною для автоенкодерів і може призводити до пропуску аномалій, які не супроводжуються різкими змінами форми сигналу.

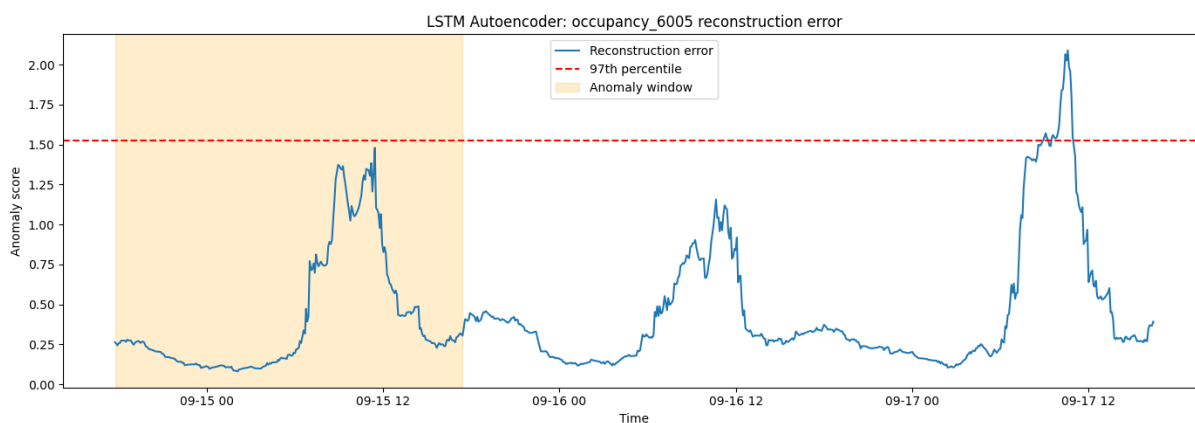


Рисунок 4.12 – Значення reconstruction error для часового ряду
occupancy_6005.csv

Приклад роботи моделі LSTM Autoencoder наведено на рисунку 4.13.

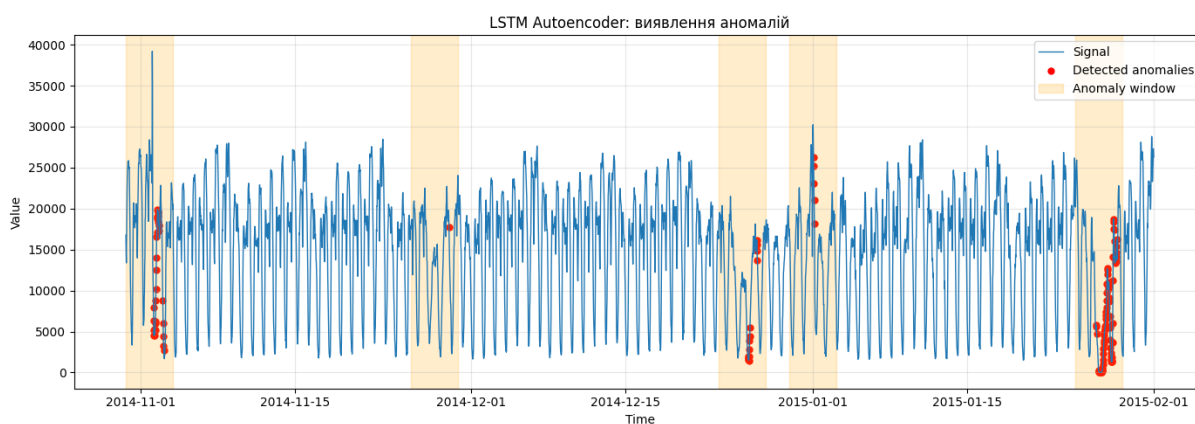


Рисунок 4.13 – Приклад виявлення аномалій моделлю LSTM Autoencoder

На рисунку аномальні інтервали, визначені відповідно до розмітки набору даних NAB, виділено зафарбованими областями, а точки, класифіковані моделлю як аномальні, позначено окремими маркерами.

Отримані результати показали, що LSTM Autoencoder ефективно враховує часовий контекст і забезпечує високу якість виявлення локальних відхилень. Водночас продуктивність моделі залежить від характеру аномалії. У випадках, коли аномальний інтервал зберігає загальну структуру

сигналу, модель може не виявити його, оскільки значення reconstruction error залишається нижчим за порогове значення.

4.3 Дослідження ефективності Transformer Autoencoder

Для дослідження ефективності моделі Transformer Autoencoder проведено експерименти на тих самих 10 часових рядах набору даних NAB. Як і для LSTM Autoencoder, використовувався розмір часового вікна 50 спостережень, а якість виявлення аномалій оцінювалася для 95-го, 97-го та 99-го перцентилів.

Динаміку функції втрат під час навчання Transformer Autoencoder наведено на рисунку 4.14. Протягом навчання спостерігається поступове зменшення значень функції втрат на навчальній та валідаційній вибірках. Значного розходження між кривими не виявлено, що свідчить про відсутність вираженого перенавчання.

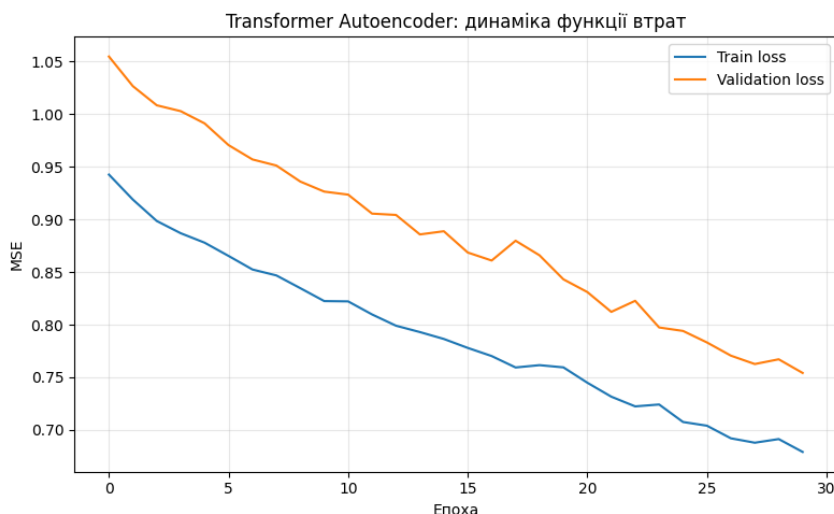


Рисунок 4.14 – Динаміка функції втрат моделі Transformer Autoencoder під час навчання

Агреговані результати оцінювання моделі для різних значень перцентильного порогу наведено на рисунку 4.15.

	percentile	mean_window_recall	mean_pr_auc	mean_false_positive_points	mean_epochs	mean_final_loss	mean_final_val_loss
0	95	0.79667	0.41016	87.0	27.1	0.264566	0.286829
1	97	0.67667	0.41016	43.6	27.1	0.264566	0.286829
2	99	0.58667	0.41016	5.8	27.1	0.264566	0.286829

Рисунок 4.15 – Середні результати моделі Transformer Autoencoder залежно від значення перцентильного порогу

Зі збільшенням порогу кількість хибних спрацювань зменшується: від 87,0 для 95-го перцентилі до 5,8 для 99-го перцентилі. Водночас середнє значення Window Recall знижується з 0,797 до 0,587. Це підтверджує той самий компроміс, що спостерігався для попередніх моделей: жорсткіший поріг зменшує кількість помилкових спрацювань, але підвищує ризик пропуску аномальних інтервалів.

Для подальшого аналізу використано 97-й перцентиль, який забезпечує проміжне співвідношення між повнотою виявлення та кількістю хибних спрацювань.

Результати Transformer Autoencoder на окремих часових рядах наведено на рисунку 4.16.

	series	window_recall	false_positive_points	pr_auc	epochs
19	realTraffic/occupancy_t4013.csv	0.5000	0	0.7566	21
4	artificialWithAnomaly/art_daily_jumpsup.csv	1.0000	0	0.6502	30
16	realKnownCause/nyc_taxi.csv	0.6000	0	0.6085	30
10	realAWSCloudwatch/ec2_network_in_257a54.csv	1.0000	0	0.4206	30
13	realKnownCause/ambient_temperature_system_fail...	1.0000	33	0.3694	19
25	realTweets/Twitter_volume_GOOG.csv	0.6667	127	0.3225	30
1	artificialWithAnomaly/art_increase_spike_densi...	1.0000	18	0.3219	30
22	realTraffic/occupancy_6005.csv	0.0000	23	0.3024	30
28	realTweets/Twitter_volume_IBM.csv	1.0000	139	0.2558	28
7	realAWSCloudwatch/rds_cpu_utilization_e47b3b.csv	0.0000	96	0.0937	23

Рисунок 4.16 – Результати моделі Transformer Autoencoder для окремих часових рядів

Найкращі результати модель продемонструвала для рядів `ambient_temperature_system_failure.csv`, `art_increase_spike_density.csv`, `ec2_network_in_257a54.csv` та `Twitter_volume_IBM.csv`, для яких значення Window Recall становило 1,0. Водночас для рядів `occupancy_6005.csv` та `rds_cpu_utilization_e4f7b3.csv` модель не виявила жодного аномального інтервалу.

Такі результати свідчать, що Transformer Autoencoder, як і LSTM Autoencoder, краще виявляє аномалії з вираженими локальними відхиленнями. Якщо аномальна ділянка зберігає загальну структуру сигналу, помилка реконструкції може залишатися недостатньо високою для перевищення порогового значення.

Зміну reconstruction error для тестового часового ряду наведено на рисунку 4.17.

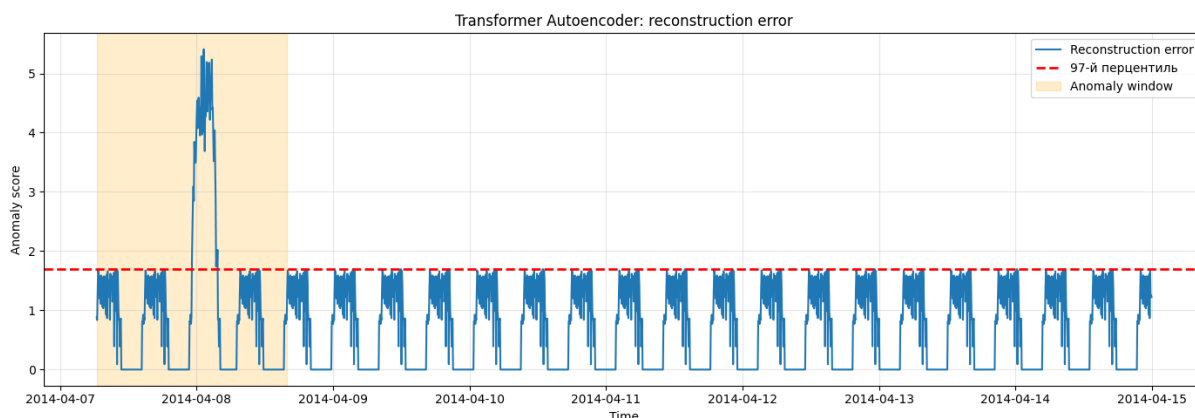


Рисунок 4.17 – Значення reconstruction error моделі Transformer Autoencoder

На рисунку видно, що найбільше зростання reconstruction error спостерігається всередині аномального інтервалу. Це означає, що модель гірше відновлює фрагменти сигналу, які відрізняються від нормальної поведінки, і саме ці ділянки отримують вищі значення anomaly score.

Приклад виявлення аномалій моделлю Transformer Autoencoder наведено на рисунку 4.18.

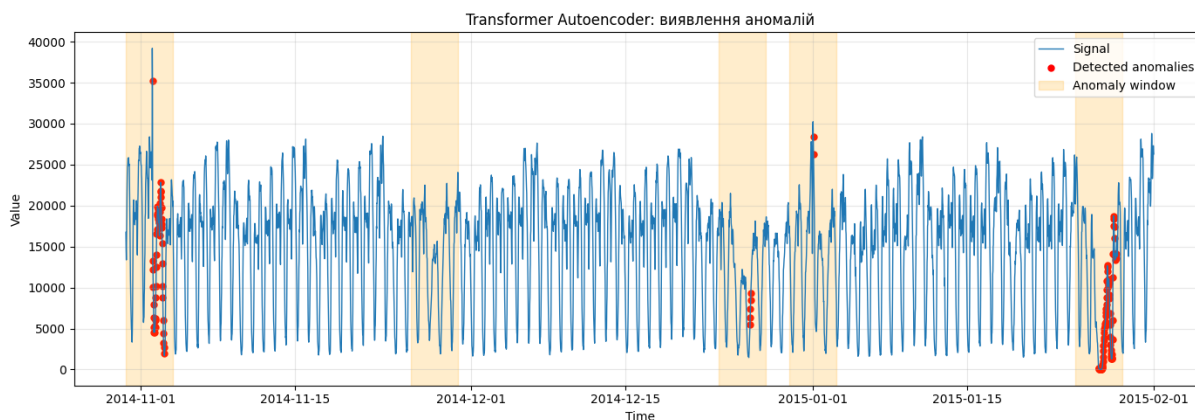


Рисунок 4.18 – Приклад виявлення аномалій моделлю Transformer Autoencoder

На рисунку аномальні інтервали виділено зафарбованими областями, а точки, позначені моделлю як аномальні, показано окремими маркерами. Модель коректно реагує на частину аномальних ділянок, однак також демонструє залежність від характеру сигналу та вибраного порогового значення.

Отримані результати показали, що Transformer Autoencoder забезпечує якість, близьку до LSTM Autoencoder, однак не усуває проблему пропуску аномалій, які не мають різко вираженого відхилення від нормальної структури часового ряду. При цьому модель є складнішою за архітектурою та потребує більшого обсягу обчислень.

4.4 Порівняльний аналіз моделей

Порівняльний аналіз моделей Isolation Forest, LSTM Autoencoder та Transformer Autoencoder проводився для фінальних конфігурацій, обраних за результатами попередніх експериментів. Для всіх моделей

використовувався 97-й перцентиль як порогове значення, оскільки він забезпечував збалансоване співвідношення між повнотою виявлення аномалій та кількістю хибних спрацювань.

Узагальнені результати дослідження наведено на рисунку 4.19.

	model	mean_window_recall	mean_false_positive_points	mean_pr_auc
0	Isolation Forest	1.0000	76.5	0.3019
1	LSTM Autoencoder	0.6967	45.3	0.4367
2	Transformer Autoencoder	0.6767	43.6	0.4102

Рисунок 4.19 – Підсумкові результати порівняння моделей виявлення аномалій

Порівняння моделей за показником Window Recall наведено на рисунку 4.20.

Найвище значення Window Recall продемонструвала модель Isolation Forest, для якої середнє значення показника становило 1,0. Це свідчить про здатність моделі виявляти принаймні одну аномальну точку в кожному аномальному інтервалі. Для моделей LSTM Autoencoder та Transformer Autoencoder значення Window Recall становили 0,6967 та 0,6767 відповідно.

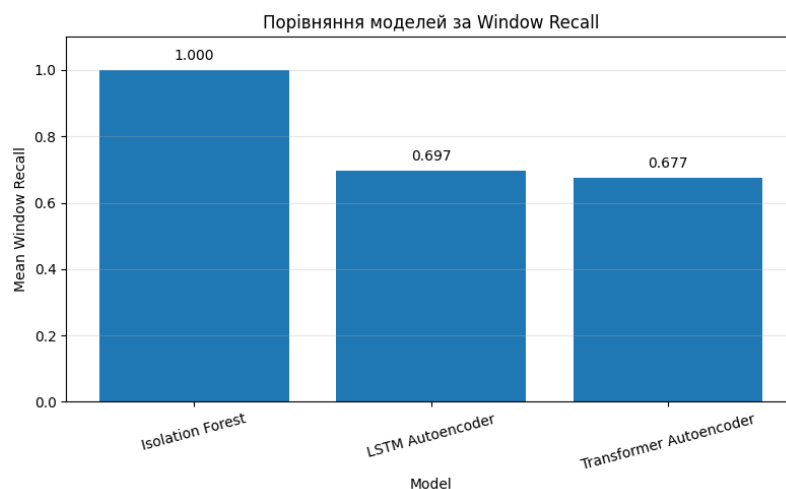


Рисунок 4.20 – Порівняння моделей за показником Window Recall

Однак високе значення Window Recall не гарантує високої точності виявлення аномалій. Порівняння моделей за середньою кількістю хибних спрацювань наведено на рисунку 4.21.

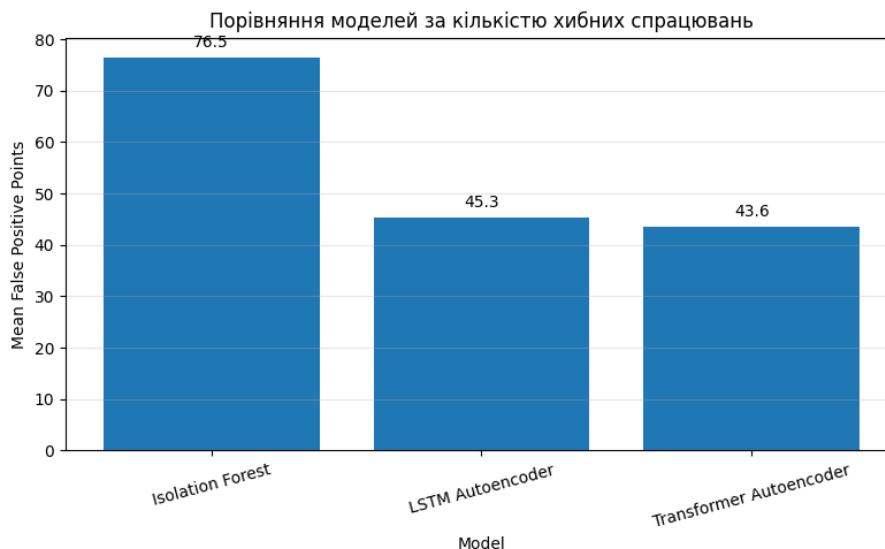


Рисунок 4.21 – Порівняння моделей за кількістю хибних спрацювань

Найменшу кількість хибних спрацювань продемонструвала модель Transformer Autoencoder – 43,6 точки в середньому. Для LSTM Autoencoder цей показник становив 45,3, тоді як для Isolation Forest – 76,5. Отримані результати свідчать, що нейромережеві моделі краще відокремлюють нормальні та аномальні патерни часового ряду, зменшуючи кількість помилкових спрацювань.

Порівняння моделей за показником PR-AUC наведено на рисунку 4.22.

Найвище середнє значення PR-AUC отримано для моделі LSTM Autoencoder – 0,4367. Transformer Autoencoder продемонстрував близький результат – 0,4102, тоді як Isolation Forest отримав найнижче значення – 0,3019.

Це свідчить про те, що нейромережеві моделі забезпечують якісніше ранжування спостережень за рівнем аномальності. Використання часового

контексту дозволяє їм краще враховувати локальну структуру сигналу та виявляти складні закономірності, недоступні для моделей, що працюють лише з набором ознак.

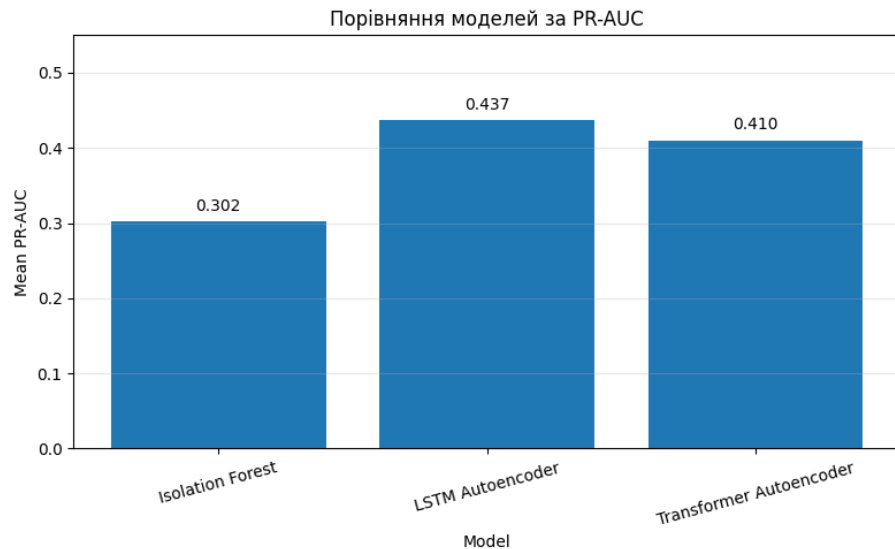


Рисунок 4.22 – Порівняння моделей за показником PR-AUC

Результати порівняння PR-AUC для окремих часових рядів наведено на рисунку 4.23.

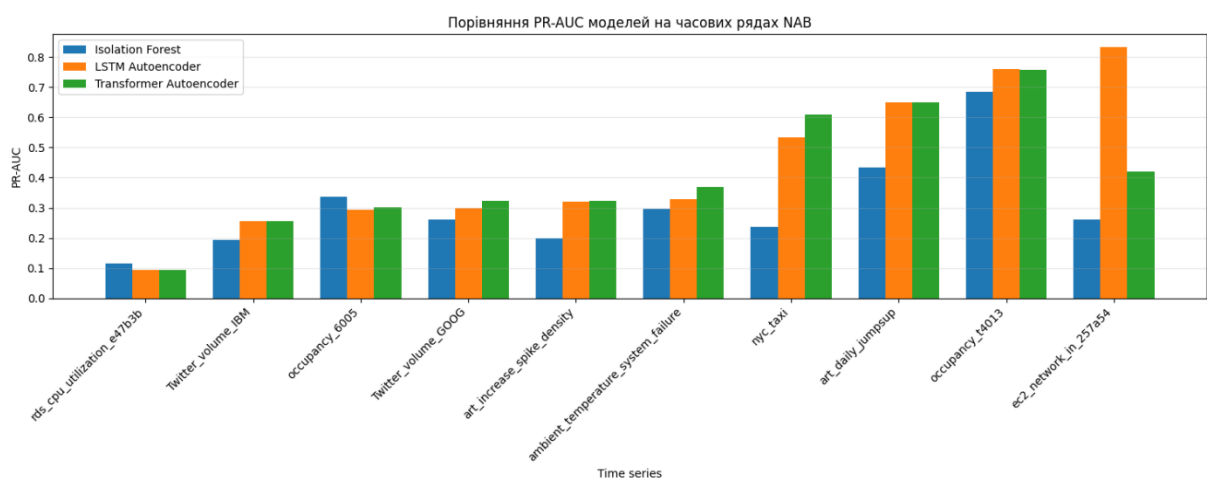


Рисунок 4.23 – Порівняння значень PR-AUC моделей на часових рядах набору даних NAB

Аналіз результатів показав, що ефективність моделей суттєво залежить від характеру часового ряду та типу аномалій. Для рядів `art_daily_jumpsup`, `occupancy_t4013`, `nyc_taxi` та `ec2_network_in_257a54` найкращі результати продемонстрували нейромережеві моделі.

Водночас для ряду `rds_cpu_utilization_e47b3b` усі моделі показали низькі значення PR-AUC, що свідчить про складність виявлення аномалій цього типу. Для ряду `occupancy_6005` модель Isolation Forest продемонструвала вищий результат порівняно з автоенкодерами.

Загалом отримані результати показали, що Isolation Forest забезпечує максимальну повноту виявлення аномалій, однак супроводжується більшою кількістю хибних спрацювань. LSTM Autoencoder продемонстрував найкращий баланс між показниками Window Recall, PR-AUC та кількістю помилкових спрацювань. Transformer Autoencoder забезпечив результати, близькі до LSTM Autoencoder, проте не продемонстрував переваги, яка б компенсувала вищу складність архітектури та більші обчислювальні витрати.

4.5 Демонстрація роботи програмного прототипу

Для оцінювання здатності розробленої системи узагальнювати закономірності та виявляти аномалії в нових даних проведено додаткове тестування на часових рядах, які не використовувалися під час формування навчальної та тестової вибірок.

Для експерименту обрано три часові ряди з набору даних NAV, що належать до різних предметних галузей:

- `Twitter_volume_PFE.csv` – часовий ряд, який відображає інтенсивність згадувань компанії Pfizer у соціальній мережі Twitter;
- `Twitter_volume_CVS.csv` – часовий ряд із даними про кількість згадувань компанії CVS Health;

– ec2_request_latency_system_failure.csv – часовий ряд, що характеризує затримки обробки запитів у хмарній інфраструктурі Amazon EC2.

Використання даних, які не входили до початкового набору SELECTED_SERIES, дозволяє об'єктивно оцінити здатність моделі виявляти аномалії в умовах, наближених до реального використання.

Результати аналізу часового ряду Twitter_volume_PFE.csv наведено на рисунку 4.24. Досліджуваний ряд характеризується значною кількістю локальних коливань та короткочасних піків активності.

Система виявила 475 аномальних точок, що становить 3,0% від загальної кількості спостережень. Більшість виявлених аномалій відповідає різким локальним зростанням кількості згадувань компанії в соціальних мережах. Максимальне значення показника аномальності становило 6,21, що свідчить про помірний рівень відхилень від типової поведінки сигналу.

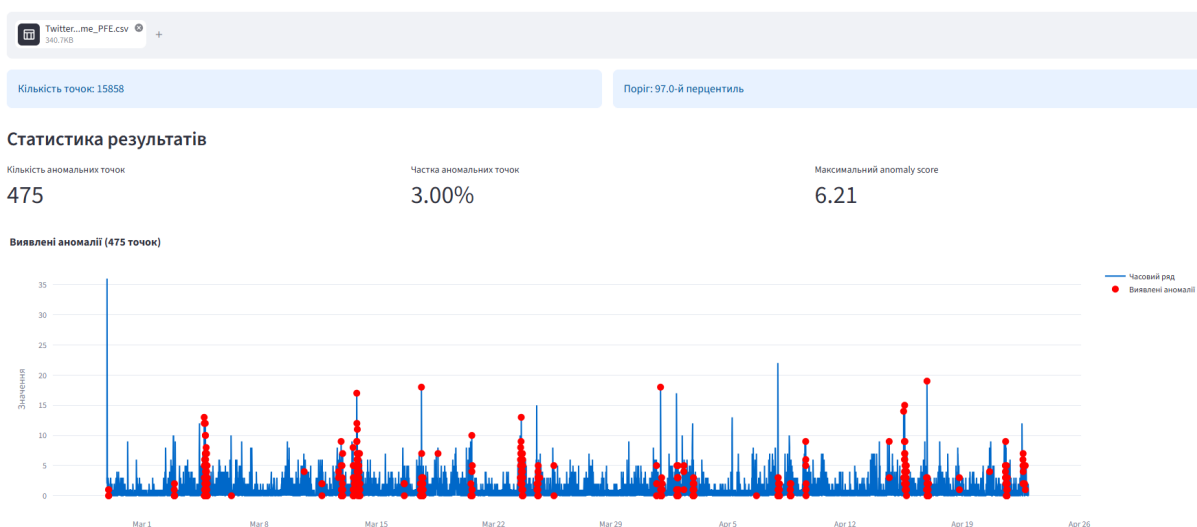


Рисунок 4.24 – Результати виявлення аномалій у часовому ряді
Twitter_volume_PFE.csv

На рисунку 4.25 наведено результати аналізу часового ряду Twitter_volume_CVS.csv. На відміну від попереднього прикладу, цей ряд містить окремі виражені піки, амплітуда яких суттєво перевищує фоновий

рівень активності. Система успішно ідентифікувала як поодинокі аномальні сплески, так і групи аномалій, пов'язані з короткостроковими змінами інтенсивності повідомлень.

Загальна кількість виявлених аномальних точок також становила 475, проте максимальне значення показника аномальності досягло 19,17, що вказує на наявність більш виражених відхилень.

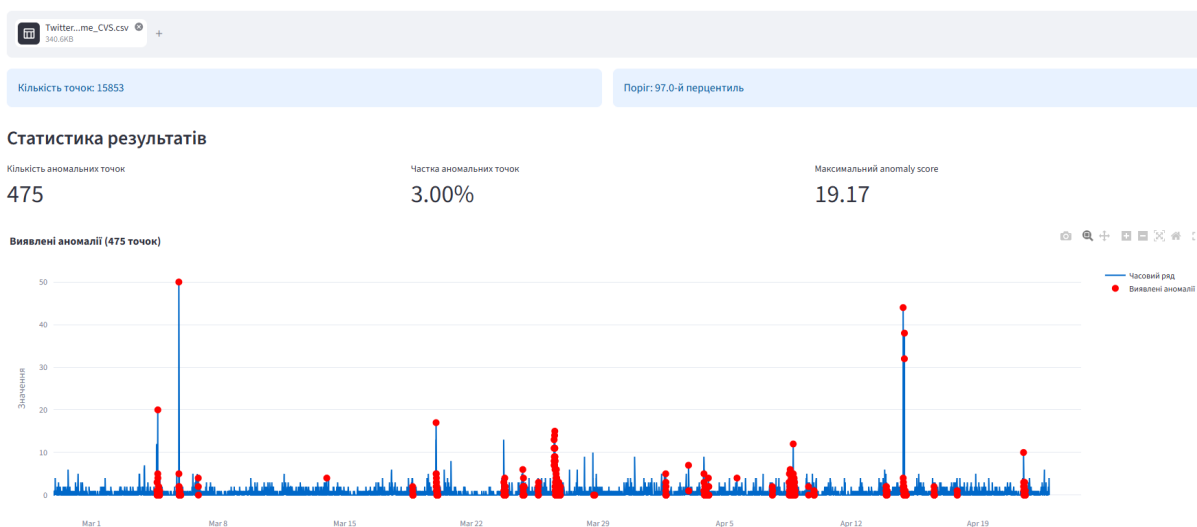


Рисунок 4.25 – Результати виявлення аномалій у часовому ряді
Twitter_volume_CVS.csv

Результати аналізу часового ряду `es2_req_aiure.csv` наведено на рисунку 4.26. Даний ряд характеризується відносно стабільною поведінкою з незначними коливаннями навколо середнього значення. Попри це система коректно виявила групи аномальних спостережень, пов'язаних зі зміною режиму роботи сервісу, а також одиничний виражений пік.

Загальна кількість виявлених аномальних точок становила 121, або 3,0% від загального обсягу даних. Максимальне значення показника аномальності досягло 298,16, що свідчить про суттєве відхилення від нормального режиму функціонування.

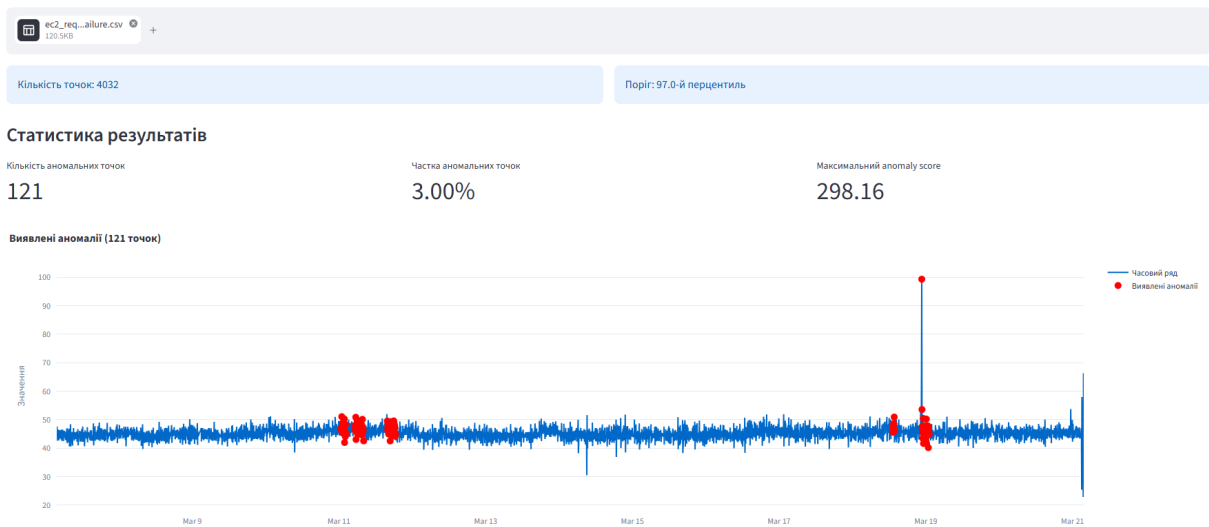


Рисунок 4.26 – Результати виявлення аномалій у часовому ряді
ec2_req_aiure.csv

Отримані результати демонструють, що розроблена система здатна ефективно виявляти аномалії різної природи: локальні короткочасні відхилення, групові аномалії та поодинокі пікові значення. Успішне застосування моделі до часових рядів, які не використовувалися під час навчання, підтверджує її здатність до узагальнення та можливість використання для аналізу нових даних із різних предметних галузей.

ВИСНОВКИ

У кваліфікаційній роботі розв'язано актуальне завдання розробки інтелектуальної системи виявлення аномалій у часових рядах на основі методів машинного навчання.

У ході виконання роботи проаналізовано предметну галузь виявлення аномалій, досліджено особливості часових рядів як об'єкта аналізу та розглянуто основні підходи до виявлення аномалій, зокрема статистичні методи, алгоритми машинного навчання та нейромережеві моделі. Проведено аналіз сучасних наборів даних для оцінювання алгоритмів виявлення аномалій та обґрунтовано вибір набору даних NAB, який містить реальні та синтетичні часові ряди з розміченими аномальними інтервалами.

На основі проведеного аналізу розроблено інтелектуальну систему виявлення аномалій, реалізовану засобами мови програмування Python із використанням бібліотек Pandas, NumPy, Scikit-learn, TensorFlow, Keras та Streamlit. Архітектура системи включає модулі підготовки даних, формування ознак, навчання моделей, оцінювання результатів та візуалізації виявлених аномалій.

У межах практичної частини реалізовано та досліджено три підходи до виявлення аномалій: Isolation Forest, LSTM Autoencoder та Transformer Autoencoder. Для моделі Isolation Forest досліджено вплив різних наборів ознак і значень порогу прийняття рішення. Встановлено, що використання розширеного набору ознак, який включає поточне значення, ковзне середнє, стандартне відхилення та різницю між сусідніми спостереженнями, дозволяє суттєво зменшити кількість хибних спрацювань. Для нейромережевих моделей досліджено вплив довжини часового вікна на якість виявлення аномалій. Визначено, що збільшення розміру вікна до 50 спостережень забезпечує найкращі результати завдяки врахуванню локальної структури та періодичності сигналу.

Оцінювання моделей виконувалося з використанням метрик Window Recall, PR-AUC та кількості хибних спрацювань. Встановлено, що Isolation Forest забезпечує найвищий рівень повноти виявлення аномалій із середнім значенням Window Recall, що дорівнює 1,0, проте характеризується більшою кількістю хибних спрацювань. Нейромережеві моделі продемонстрували нижчі значення Window Recall, але забезпечили кращу здатність до ранжування спостережень за рівнем аномальності. Найвище середнє значення PR-AUC отримано для LSTM Autoencoder – 0,4367, тоді як Transformer Autoencoder продемонстрував значення 0,4102.

Проведений порівняльний аналіз показав, що ефективність моделей суттєво залежить від характеру аномалій та структури часового ряду. Isolation Forest краще виявляє локальні відхилення та забезпечує максимальну повноту виявлення, тоді як автоенкодера ефективніше працюють зі складними часовими залежностями та забезпечують меншу кількість помилкових спрацювань.

Розроблену систему апробовано на часових рядах, які не використовувалися під час навчання та налаштування моделей. Результати експериментів підтвердили її здатність виявляти аномалії різної природи, зокрема короткочасні сплески, групові відхилення та поодинокі пікові значення, а також продемонстрували можливість узагальнення на нові дані.

Практична цінність роботи полягає у створенні програмного засобу, який може бути використаний для автоматизованого моніторингу часових рядів у різних предметних галузях, зокрема для аналізу телеметричних даних, моніторингу ІТ-інфраструктури, промислових процесів, мережевого трафіку та фінансових показників.

Перспективами подальших досліджень є розширення переліку моделей виявлення аномалій, використання ансамблевих підходів, адаптивне налаштування порогових значень, урахування багатовимірних часових рядів, а також дослідження методів онлайн-виявлення аномалій у потокових даних у реальному часі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Sneiderman R. A quick introduction to time series analysis. *Towards Data Science*. URL: <https://towardsdatascience.com/a-quick-introduction-to-time-series-analysis-d86e4ff5fdd/> (date of access: 18.05.2026).
2. Time series decomposition: understand trends, seasonality, and noise. *DataCamp*. URL: <https://www.datacamp.com/tutorial/time-series-decomposition> (date of access: 18.05.2026).
3. Gulati J. How to decompose time series data into trend, seasonal, and residual components in python. *Statology*. URL: <https://www.statology.org/how-to-decompose-time-series-data-into-trend-seasonal-and-residual-components-in-python/> (date of access: 18.05.2026).
4. Boniol P., Palpanas T. Time series anomaly detection. *ACM SIGMOD Blog*. URL: <https://wp.sigmod.org/?p=3739> (date of access: 18.05.2026).
5. Park J., Chiang F., Milani M. Adaptive anomaly detection in the presence of concept drift: extended report. *arXiv*. 2025. URL: <https://doi.org/10.48550/arXiv.2506.15831> (date of access: 18.05.2026).
6. Greenhouse: a zero-positive machine learning system for time-series anomaly detection / T. J. Lee et al. *arXiv*. 2018. URL: <https://doi.org/10.48550/arXiv.1801.03168> (date of access: 19.05.2026).
7. Niu Z., Yu K., Wu X. LSTM-Based VAE-GAN for Time-Series Anomaly Detection. *Sensors*. 2020. Vol. 20, no. 13. P. 3738. URL: <https://doi.org/10.3390/s20133738> (date of access: 19.05.2026).
8. Matrix profile for anomaly detection on multidimensional time series / C.-C. M. Yeh et al. *arXiv*. 2026. URL: <https://doi.org/10.48550/arXiv.2409.09298> (date of access: 19.05.2026).
9. Frehner R., Stockinger K. Applying quantum autoencoders for time series anomaly detection. *arXiv*. 2024. URL: <https://doi.org/10.48550/arXiv.2410.04154> (date of access: 21.05.2026).

10. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications / H. Xu et al. *The 2018 world wide web conference*, Lyon, France, 23–27 April 2018. New York, New York, USA, 2018. URL: <https://doi.org/10.1145/3178876.3185996> (date of access: 21.05.2026).
11. AnomalyBERT: self-supervised transformer for time series anomaly detection using data degradation scheme / Y. Jeong et al. *arXiv*. 2023. URL: <https://doi.org/10.48550/arXiv.2305.04468> (date of access: 21.05.2026).
12. Bashar M. A., Nayak R. ALGAN: time series anomaly detection with adjusted-lstm GAN. *International journal of data science and analytics*. 2025. URL: <https://doi.org/10.1007/s41060-025-00810-2> (date of access: 22.05.2026).
13. Time series anomaly detection hybrid model based on SARIMA and LSTM / W. Wu et al. *Journal of computing and electronic information management*. 2025. Vol. 16, no. 1. P. 63–69. URL: <https://doi.org/10.54097/vnmprq69> (date of access: 22.05.2026).
14. MadSGM: multivariate anomaly detection with score-based generative models / H. Lim et al. *CIKM '23: the 32nd ACM international conference on information and knowledge management*, Birmingham United Kingdom. New York, NY, USA, 2023. URL: <https://doi.org/10.1145/3583780.3614956> (date of access: 23.05.2026).
15. Zamani S., Talebi H., Stevens G. Time series anomaly detection in smart homes: a deep learning approach. *arXiv*. 2023. URL: <https://doi.org/10.48550/arXiv.2302.14781> (date of access: 23.05.2026).
16. Amine M. A., Batton-Hubert M., Casoetto N. One-Class Support Subsequence Machine: abnormal time series subsequences detection using One-Class SVM. *Research Square*. 2023. URL: <https://doi.org/10.21203/rs.3.rs-2378527/v2> (date of access: 23.05.2026).
17. Multi-feature unsupervised time series anomaly detection based on Memory-Augmented Autoencoder - One-Class Support Vector Machine / G. Hao et al. *Engineering applications of artificial intelligence*. 2026. Vol. 171.

P. 114124. URL: <https://doi.org/10.1016/j.engappai.2026.114124> (date of access: 23.05.2026).

18. Anomaly detection in time series using statistical analysis. *Medium*. URL: <https://medium.com/booking-com-development/anomaly-detection-in-time-series-using-statistical-analysis-cc587b21d008> (date of access: 24.05.2026).

19. Using statistical methods in anomaly detection. *Apriorit*. URL: <https://www.apriorit.com/dev-blog/anomaly-detection-with-statistical-methods> (date of access: 24.05.2026).

20. Eka E. P. A., Zakaria M. Z. Real-Time outlier detection in fast-moving data streams. *International journal of advances in artificial intelligence and machine learning*. 2024. Vol. 1, no. 1. P. 19–27. URL: <https://doi.org/10.58723/ijaauml.v1i1.287> (date of access: 24.05.2026).

21. Usman N., Utami E., Hartanto A. D. Comparative analysis of elliptic envelope, isolation forest, one-class SVM, and local outlier factor in detecting earthquakes with status anomaly using outlier. *2023 international conference on computer science, information technology and engineering (iccosite)*, Jakarta, Indonesia, 16 February 2023. 2023. URL: <https://doi.org/10.1109/iccosite57641.2023.10127748> (date of access: 24.05.2026).

22. Pramoditha R. 4 machine learning techniques for outlier detection in Python. *Towards Data Science*. 2021. URL: <https://towardsdatascience.com/4-machine-learning-techniques-for-outlier-detection-in-python-21e9cfacb81d/> (date of access: 26.05.2026).

23. Mavuduru A. How to perform anomaly detection with the Isolation Forest algorithm. *Towards Data Science*. URL: <https://towardsdatascience.com/how-to-perform-anomaly-detection-with-the-isolation-forest-algorithm-e8c8372520bc/> (date of access: 26.05.2026).

24. How to implement anomaly detection with One-Class SVM in python. *Spot Intelligence*. URL: <https://spotintelligence.com/2024/05/27/anomaly-detection-one-class-svm/> (date of access: 26.05.2026).

25. Local outlier factor. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/machine-learning/local-outlier-factor/> (date of access: 27.05.2026).

26. Dey R. Autoencoders for anomaly detection. *Medium*. URL: <https://medium.com/@roshmitadey/autoencoders-for-anomaly-detection-215dd7db59ae> (date of access: 27.05.2026).

27. Dobilas S. LSTM recurrent neural networks – how to teach a network to remember the past. *Towards Data Science*. URL: <https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e/> (date of access: 28.05.2026).

28. Orabi M. Enable transformers for anomaly detection in multivariate time series data. *Medium*. URL: <https://medium.com/@moussab.orabi/enable-transformers-with-anomaly-detection-in-high-order-multivariate-time-series-data-509a5df39151> (date of access: 28.05.2026).

29. Understanding the confusion matrix. *Medium*. URL: <https://medium.com/@danyal.wainstein1/understanding-the-confusion-matrix-b9bc45ba2679> (date of access: 28.05.2026).

30. Saxena S. How good is your AI? ROC vs PR-AUC explained simply. *Medium*. URL: <https://medium.com/@samaishsax/how-good-is-your-ai-roc-vs-pr-auc-explained-simply-341908ac2e4c> (date of access: 28.05.2026).

31. Numenta anomaly benchmark (NAB). *Kaggle: The World's AI Proving Ground*. URL: <https://www.kaggle.com/datasets/boltzmannbrain/nab> (date of access: 29.05.2026).