

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки**

До захисту допущено
Завідувач кафедри

_____ Дмитро ЛАНДЕ

(підпис)

«_____» _____ 2024 р.

**Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системи, технології та математичні
методи кібербезпеки»
спеціальності 125 «Кібербезпека»**

на тему: «Виявлення аномалій у часових рядах методами машинного навчання без
використання патернів»

Виконав (-ла): здобувач вищої освіти **IV** курсу, групи **ФБ-01**
(шифр групи)

Полянський Ян Валентинович

(прізвище, ім'я, по батькові)

(підпис)

Керівник **Ткач Володимир Миколайович**

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові) (підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Здобувач вищої освіти _____

(підпис)

Київ – 2024 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 125 «Кібербезпека»

Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Дмитро ЛАНДЕ
(підпис)

« ____ » _____ 2024 р.

ЗАВДАННЯ
на дипломну роботу здобувачу вищої освіти

Полянському Яну Валентиновичу
(прізвище, ім'я, по батькові)

1. Тема роботи: «Виявлення аномалій у часових рядах методами машинного навчання без використання патернів», керівник роботи Ткач Володимир Миколайович, доцент кафедри інформаційної безпеки, (прізвище, ім'я, по батькові, науковий ступінь, вчене звання) затверджені наказом по університету від «26» травня 2023 р. № 2028-С
2. Термін подання здобувачем вищої освіти роботи 7 червня 2024 р.
3. Вихідні дані до роботи: Наукова література по виявленню аномалій у часових рядах, сучасні методи машинного навчання, зокрема рекурентні нейронні мережі (RNN), LSTM, та дані для експериментів, включаючи часові ряди таких з різними параметрами.
4. Зміст роботи: огляд методів виявлення аномалій у часових рядах, опис математичних моделей для прогнозування та виявлення аномалій, створення додаткових ознак для аналізу часових рядів, розробка та навчання моделі LSTM для виявлення аномалій, здійснення експерименту на реальних даних та аналіз результатів.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): Презентація до захисту дипломної роботи.
6. Дата видачі завдання: 22.09.2023

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Отримання завдання та узгодження теми роботи	23.09.2023	Виконано
2	Огляд наукової літератури по темі	23.02.2024 - 10.03.2024	Виконано
3	Огляд методів виявлення аномалій у часових рядах	11.03.2024 - 31.03.2024	Виконано
4	Опис математичних моделей для прогнозування та виявлення аномалій	01.04.2024 - 07.04.2024	Виконано
5	Створення додаткових ознак для аналізу часових рядів	08.04.2024 - 20.04.2024	Виконано
6	Розробка та навчання моделі LSTM для виявлення аномалій	21.04.2024 - 05.05.2024	Виконано
7	Здійснення експерименту на реальних даних та аналіз результатів	06.05.2024 - 26.05.2024	Виконано
8	Написання дипломної роботи	26.05.2023 - 3.06.2023	Виконано

Здобувач вищої освіти

(підпис)

Ян ПОЛЯНСЬКИЙ
(Власне ім'я, ПРІЗВИЩЕ)

Керівник роботи

(підпис)

Володимир ТКАЧ
(Власне ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Обсяг дипломної роботи 53 сторінки, 23 ілюстрацій, 1 таблиця, 1 додаток і 12 джерел літератури.

Об'єкт дослідження: Часові ряди даних, що характеризують поведінку різних систем, інтернет-трафік або інші динамічні процеси, які потребують моніторингу та виявлення аномалій

Предмет дослідження: Методи та алгоритми для виявлення аномалій у часових рядах, зокрема використання додаткових ознак та моделей LSTM.

Мета дослідження: Розробка ефективного методу виявлення аномалій у часових рядах шляхом створення та використання додаткових ознак і навчання моделі LSTM для підвищення точності прогнозування та ідентифікації аномалій.

Методи дослідження: Вивчення існуючих методів виявлення аномалій у часових рядах, опис та застосування математичних моделей для обробки та аналізу часових рядів, використання статистичних характеристик, навчання моделей LSTM для прогнозування та виявлення аномалій у часових рядах. Проведення експериментів.

Отримані результати: Розроблено та реалізовано метод виявлення аномалій у часових рядах з використанням додаткових ознак та моделі LSTM, що дозволило підвищити точність виявлення аномалій. Проведено тестування на реальних даних, що підтвердило ефективність запропонованого методу.

Ключові слова: Часові ряди, виявлення аномалій, LSTM, додаткові ознаки, машинне навчання.

ABSTRACT

The volume of the thesis is 53 pages, 23 illustrations, 1 table, 1 appendix and 12 sources of literature.

Object of research: Time series of data characterising the behaviour of various systems, Internet traffic or other dynamic processes that require monitoring and anomaly detection.

Subject of research: Methods and algorithms for detecting anomalies in time series, including the use of additional features and LSTM (Long Short-Term Memory) models.

Research objective: To develop an effective method for detecting anomalies in time series by creating and using additional features and training the LSTM model to improve the accuracy of forecasting and identifying anomalies.

Research methods: Study of existing methods for detecting anomalies in time series, description and application of mathematical models for processing and analysing time series, use of statistical characteristics, training of LSTM models for forecasting and detecting anomalies in time series. Experiments.

Results: A method for detecting anomalies in time series using additional features and LSTM model has been developed and implemented, which has improved the accuracy of anomaly detection. Testing on real data was conducted, which confirmed the effectiveness of the proposed method.

Keywords: Time series, anomaly detection, LSTM, additional features, machine learning.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	7
ВСТУП.....	8
1 ТЕОРЕТИЧНІ ВІДОМОСТІ.....	10
1.1 Часовий ряд	10
1.2 Аномалії.....	11
1.3 Методи машинного навчання для виявлення аномалій.....	12
1.4 Порівняння методів машинного навчання для виявлення аномалій.....	16
1.5 Використання методів виявлення аномалій у SIEM системах.....	19
Висновки до розділу 1	21
2 РОЗРОБКА МЕТОДУ ВИЯВЛЕННЯ АНОМАЛІЙ.....	23
2.1 Прогнозування.....	23
2.2 Навчання моделі LSTM.....	25
Висновки до розділу 2.....	30
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	32
3.1 Підхід 1: Виявлення загальної аномалії	32
3.2 Підхід 2: Виявлення аномалій в зашумлених даних	35
3.3 Порівняння отриманих результатів з класичним методом.....	41
Висновки до розділу 3.....	43
ВИСНОВКИ.....	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	46
ДОДАТОК А.....	48

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ML / МН – Machine Learning (Машинне навчання)

AI / ІІІ – Artificial Intelligence (Штучний інтелект)

NP-AD(Non-Pattern Anomaly Detection) – детектування аномалій без використання шаблонів

LSTM (Long Short-Term Memory) – довготривала короткочасна пам'ять; тип рекурентної нейронної мережі, що використовується для обробки та прогнозування часових рядів.

RNN - Вид нейронних мереж, де зв'язки між елементами утворюють спрямовану послідовність.

NAB (Numenta Anomaly Benchmark) - бенчмарк для оцінки алгоритмів виявлення аномалій у потокових онлайн-додатках.

SIEM — Security Information and Event Management (управління інформацією і подіями безпеки)

ВСТУП

Виявлення аномалій у критичній інфраструктурі є не лише ключовим кроком до виявлення загроз, але й попередженням про ймовірність потенційних кібератак, системних помилок або збоїв в інфраструктурі.

Більшість існуючих підходів до виявлення аномалій стосуються початкового профілювання, використання патернів, тобто визначення того, яка поведінка, представлена часовими рядами, є «нормальною», тоді як все, що не відповідає критеріям «нормальності», визначається як «ненормальне» або аномальне. Таке визначення не сповна відображає складність природи аномалії. У різних умовах однакова поведінка може бути як аномальною, так і ні. Оскільки набір правил є фіксованим, системи виявлення аномалій залежать від експерта і не сприйнятливі до нових типів вторгнень, поки вони не будуть описані в нових правилах. Однією з найбільших проблем кіберзахисту є відсутність реальних ефективних механізмів запобігання атакам, якщо не визначені певні ознаки таких атак. Цей метод спрямований на вирішення цієї проблеми шляхом виявлення аномалій без використання патернів.

Тому в цій роботі досліджується необхідність застосування Non-Pattern підходу як кроку до демонстрації релевантності відхилення або невідповідності очікуваній поведінці. Виходячи з експериментів, проведених у цьому дослідженні, часові ряди NP-AD є потенційно важливим підходом, заснованим на несезонних викидах, полях потоку даних, що використовуються у формі часових рядів для наборів даних різного характеру. Цей підхід може стати кроком до передбачення аномалій у різних кіберінфраструктурах, критичних інфраструктурах та інших складних середовищах.

Актуальність роботи базується на факті, що аномалії в часових рядах даних є важливим аспектом аналізу для багатьох галузей, зокрема кібербезпеки. Традиційні методи аналізу аномалій часто мають обмежену ефективність у виявленні складних закономірностей через їхню здатність враховувати лише обмежену кількість ознак та правил. Використання рекурентних нейронних

мереж, зокрема Long Short-Term Memory (LSTM), дозволяє враховувати більш складні залежності та тенденції в даних, що робить ці моделі більш придатними для виявлення аномалій у часових рядах. Враховуючи зростаючі обсяги даних та складність їх структури, особливо для України, яка активно впроваджує сучасні технології в різних галузях, актуальність даної роботи полягає в розробці більш точних та ефективних методів виявлення аномалій.

Метою роботи є розробка та реалізація LSTM-моделі для виявлення аномалій у часових рядах з додатковими можливостями, які дозволять підвищити точність та надійність виявлення аномалій.

Об'єктом дослідження є процеси та явища в часових рядах, які можуть містити аномалії, що потребують виявлення для забезпечення безпеки, точності прогнозування та ефективності управління.

Предметом дослідження є характеристики та особливості часових рядів, що впливають на ефективність виявлення аномалій за допомогою LSTM-моделей.

Методом дослідження є аналіз типів аномалій на основі часових рядів систем та огляд методів виявлення аномалій

Наукова новизна роботи полягає в розробці LSTM-моделі для виявлення аномалій у часових рядах з додатковими ознаками, які підвищують точність і надійність прогнозів.

Практичне значення наявних результатів полягає в можливості впровадження розробленої моделі в різних галузях, зокрема кібербезпеки, де точне та надійне виявлення аномалій є критично важливим.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Часовий ряд

Ряд даних — це впорядкована послідовність точок даних[1]. Точки даних представляють властивості об'єкта або процесу на основі безперервних вимірювань, наприклад: температура, маса, швидкість та ін. Порядок, заснований на одиницях часу, зазвичай називають хронологічним. Незалежно від впорядкування, запис точок даних зазвичай слідує за дискретними інтервалами рівних відрізків часу. Саме через те, а також оскільки більшість алгоритмів аналізу рядів даних не залежать від джерела, будемо визначати терміни «ряд даних», «часовий ряд» і «послідовність» як синоніми.

Загалом, будь-яке спостереження за певний проміжок часу можна вважати часовим рядом. Ці серії зазвичай використовуються для подальшого аналізу. Основна мета аналізу часових рядів полягає у виявленні та дослідженні конкретних моделей, тенденцій, періодичності, сезонності та інших закономірностей у цих даних, а також у виявленні аномалій.

Аналіз складається з різних алгоритмів або методів, що використовуються для отримання певної статистичної інформації та характеристик даних з метою прогнозування майбутніх значень на основі даних минулих часових рядів.

Аналіз часових рядів корисний для спостереження за тим, як певний актив, вартість або економічна змінна змінюються з часом. Його також можна використовувати для вивчення впливу певного показника порівняно з іншими показниками за той самий період.

Часовий ряд передбачає повторний збір даних за певний період часу. На відміну від перехресних даних, час також є порівняльним методом аналізу. Перехресні дослідження вивчають один момент часу. Це корисно для порівняння та аналізу впливу різних факторів один на одного та для пояснення зразків. На практиці зазвичай використовуються обидві форми аналізу, але в цьому випадку це не обов'язково.

Оскільки кожен момент потрібно аналізувати в реальному часі, часові ряди ідеально відповідають потребам цього завдання. Розгляд окремих елементів для більш детального аналізу корисний для попередніх досліджень для визначення стану «за замовчуванням» системи та відповідного вибору коефіцієнтів. Поперечні розрізи дуже корисні в цьому контексті. Однак передбачається, що діапазон «нормальних» значень вже відомий. Оскільки кожен момент потрібно аналізувати в реальному часі, часові ряди ідеально відповідають потребам цього завдання. Розгляд окремих елементів для більш детального аналізу корисний для попередніх досліджень для визначення стану «за замовчуванням» системи та відповідного вибору коефіцієнтів. Поперечні розрізи дуже корисні в цьому контексті. Однак передбачається, що діапазон «нормальних» значень вже відомий.

1.2 Аномалії

Оскільки аномалії трапляються в різних областях, вони є предметом інтенсивних досліджень у різних сферах застосування, таких як мережева безпека, Інтернет речей і виробничі системи. Ключовою особливістю, спільною для всіх доменів, є розуміння аномалій як відхилень від правил або порушень, які не вважаються частиною нормальної роботи системи. Аномальна динаміка переважно невідома та випадкова, що призводить до нестабільності та, як наслідок, підвищення неефективності та системних помилок.

Аномалія в часовому ряді, також відома як викид, — це значення, яке відхиляється від очікуваної поведінки або норми. Тобто це точка або послідовність точок, яка не відповідає нормальній поведінці. Вони важливі для аналізу даних, оскільки можуть вказувати на потенційні проблеми або зміни в системі, які потребують уваги. В аналізі даних виявлення аномалій визначається як пошук й ідентифікація елементів, подій та спостережень, що не відповідають очікуваній поведінці або іншим елементам набору даних.

В часових рядах аномалії можуть бути точковими, контекстними або колективними[2][3]:

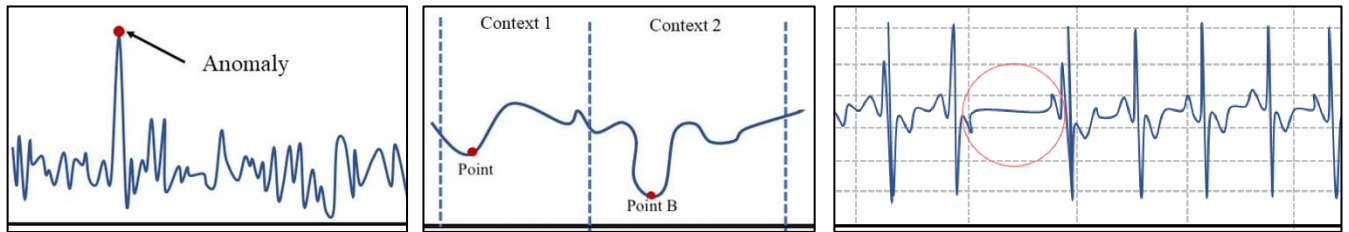


Рисунок 1.1 — Точкова аномалія (ліворуч), контекстна аномалія (посередині), колективна аномалія (праворуч)

- 1) Точкова аномалія - окремий екземпляр, який є аномальним у порівнянні з рештою точок даних у ряді даних. Точкові аномалії часто являють собою екстремальне, нерегулярне або випадкове явище і не мають особливого значення.
- 2) Контекстуальна аномалія - це випадок, який вважається аномальним у певному контексті. Це означає, що спостереження однієї і тієї ж точки в різних контекстах не завжди дасть нам сигнали про аномальну поведінку.
- 3) Колективна аномалія - групи значень, які самі по собі не є аномальними, але їхня спільна поява є аномальною.

1.3 Методи машинного навчання для виявлення аномалій

Методи машинного навчання (ML) стають все більш популярними для виявлення аномалій, оскільки вони можуть ефективно обробляти великі обсяги даних і виявляти складні моделі. Ці методи можуть бути керованими, некерованими або напівкерованими.

Керовані методи вимагають мічених даних для навчання, поділяючись на «нормальні» та «аномальні», тоді як некеровані методи можуть вивчати нормальні моделі даних та виявляти аномалії без мічених даних, виходячи з припущення, що більшість значень є «нормальними», тоді як «аномальні» будуть тими, що найменше відповідають решті множини.

При напівкерованому виявленні аномалій модель тренується на нормальних (або «невизначених») даних, а потім використовується для

виявлення нових спостережень, які відрізняються від навчального набору даних. Це може бути особливо корисним, коли доступні лише невеликі набори мічених даних або коли аномалії рідкісні.

У цій роботі використовується метод виявлення аномалій з напівкеруваним спостереженням. На вході — часовий ряд параметрів, який містить аномалії. Для кожного елемента цього ряду ми прогнозуємо наступний елемент за поточними значеннями та минулими додатковими можливостями. Для цього ми використовуємо різновид рекурентної нейронної мережі LSTM, особливості якої будуть згадані нижче, з урахуванням додаткових можливостей, таких як середнє, стандартне відхилення, мінімальне та максимальне значення, а також процентилі у вікні даних. Таким чином, ми отримуємо часовий ряд з прогнозованими значеннями, які, як вважається, є типовою поведінкою для ряду, щоб порівняти його з реальним. Це дозволяє виявити відхилення та аномалії, які можуть свідчити про аномальні події або зміни в даних.

Нейронні мережі використовуються для широкого спектру завдань короткого терміну, такого як прогнозування та попередження. З еволюцією задач складність систем і моделей, що використовуються для вирішення цих завдань, за допомогою нейронної мережі, завжди буде змінюватися з таким спрямуванням, щоб забезпечити бажаний результат для практичного застосування у контексті аналізу даної задачі. Складність математичних програм, з якими ми маємо справу, привела, вирішивши цю проблему, до того, що нейронні мережі модифікувались. Ці модифікації, у свою чергу, змінюються задля вирішення конкретної проблеми, або задля розв'язання проблем, в яких природа проблеми має характеристики необхідного рішення. Однак це в жодному раз не означає, що яка-небудь модифікація нейронної мережі не може використовуватися для широкого спектру застосувань, а навпаки, деякі з цих модифікацій створюють більш універсальні нейронні мережі, які, в свою чергу, можуть бути використані для вирішення широкого кола завдань, з якими щодня стикаються аналітики та дослідники.

Перш за все, вони вказують, якою важливою може бути класифікація цих завдань для характеризування змін, які потрібно внести до звичайних нейронних мереж на тлі постійного розвитку інформаційних технологій. При правильному використанні останніх розробок, які значно спрощують взаємодію та обмін сигналами всередині самої мережі, спростити або більш ефективно формувати тренувальні, повторювані або згорткові шари, і з додаванням модифікацій, які можуть розрізняти ці функції нейронної мережі й зміцнювати їх, можна отримати ефективний і швидкий інструмент для вирішення необхідних завдань.

У звичайних нейронних мереж немає можливості аналізувати події, що відбувалися у минулому. Одним з причин модифікації нейронних мереж є спроба виправити цю проблему. Ця проблема може бути розв'язана здійсненням модифікацій, які дозволять нейронній мережі зберігати інформацію, попри проблеми, які виникли при перетворенні інформації у зворотному напрямку через велику кількість епох навчання мережі. Такі нейронні мережі є чутливими до обчислювального середовища і зазвичай потребують більше часу на навчання на комп'ютерах, що не можуть швидко повністю розв'язати поставлену задачу. Як вже раніше зазначалося, рекурентні нейронні мережі мають кілька модифікацій, які можуть використовуватися для розв'язання цієї основної проблеми, зумовленої необхідністю аналізу попередніх подій, а також гарантування адекватності рішень та можливості їх аналізу.

Модифікація нейронних мереж, яка не має проблем, властивих звичайним нейронним мережам, полягає в тому, що нейрони даної мережі можуть запам'ятовувати інформацію, отриману нещодавно, але не можуть довго зберігати в пам'яті те, що було оброблено багато ітерацій тому, незалежно від якості та важливості такої інформації. Така модифікація рекурентних нейронних мереж називається нейронними мережами з довгою короткочасною пам'яттю (LSTM)[4]. У таких мережах внутрішні нейрони являють собою складну систему, яка побудована досить складним і витонченим чином.

Основна ідея таких мереж полягає в передачі пам'яті між шарами, і для цього кожен нейрон містить спеціальний індикатор, який називається «стан клітини». Він проходить через всю мережу, зберігаючи інформацію і передаючи її на наступну ітерацію виконання і навчання. Саме завдяки цій особливості вибір в роботі пав на цю модель: LSTM було розроблено для вирішення такої проблеми звичайних RNN, як зникаючий градієнт, що унеможлиблює вивчення довготривалих зв'язків. Проблема зникаючого градієнта описує сценарій, в якому частина ваг у RNN має тенденцію припиняти змінюватися в процесі навчання. Таким чином, приділення уваги поточній інформації може призвести до нехтування минулими подіями. Отже, залежності, які повторюються протягом тривалого періоду часу (довгострокові залежності), не можуть бути ефективно засвоєні. Роль LSTM полягає в тому, щоб контролювати всі інформаційні потоки, що відбуваються всередині кожного нейрона. Для цього вводиться механізм забування, який контролює процес додавання і видалення інформації зі стану клітини, який ітеративно поширюється. Таким чином можна регулювати процес забування і реалізувати специфічну поведінку пам'яті для моделювання як короткострокових, так і довгострокових залежностей. Саме це й робить їх особливо корисними для аналізу часових рядів.

Тож, такий метод цікаво розглянути й можна вважати ефективним з таких причин:

Врахування складних залежностей: модель LSTM (Long Short-Term Memory) спеціально розроблена для обробки послідовних даних, що дозволяє їй враховувати довготривалі залежності в часовому ряді. Це особливо важливо для виявлення аномалій, де відхилення можуть бути залежними від попередніх значень, що складно виявити за допомогою традиційних методів.

Інтеграція додаткових ознак: додавання таких ознак, як середнє значення, стандартне відхилення, мінімальні та максимальні значення, а також процентилі, дозволяє моделі LSTM мати більш глибоке розуміння структури даних. Ці ознаки

забезпечують додаткову інформацію про статистичні характеристики даних, що покращує здатність моделі до виявлення аномалій.

Поліпшена точність прогнозування: завдяки використанню додаткових ознак, модель може краще розрізняти нормальні коливання та реальні аномалії. Це зменшує кількість хибних позитивних результатів (коли нормальні значення помилково визначаються як аномальні) та хибних негативних результатів (коли справжні аномалії не виявляються).

1.4 Порівняння методів машинного навчання для виявлення аномалій

Як було зазначено раніше, вибір моделі LSTM замість класичних методів пояснений тим, що вона дозволяє враховувати довготривалі залежності в часовому ряді, що важливо для виявлення аномалій. Зроблене це з ряду причин та порівняльних характеристик, що розглядалися впродовж аналізу й довгого вибору підходу до детектування аномалій. Класичні методи виявлення аномалій включають в себе ряд статистичних методів і методів МН, які використовуються для аналізу часових рядів і виявлення відхилень від нормальної поведінки. У цьому розділі ми розглянемо декілька популярних статистичних методів для виявлення аномалій, включаючи метод 3-сигма, метод IQR (Interquartile Range), метод Z-Score та метод Tukey's Fences.

Метод 3-Сигма

Метод 3-сигма (3σ) базується на статистичних принципах нормального розподілу. Цей метод використовує середнє значення (mean) і стандартне відхилення (standard deviation) набору даних для визначення меж нормальних значень. Аномаліями вважаються ті дані, що виходять за межі трьох стандартних відхилень від середнього значення.

Формула:

$$\text{Верхня межа: } \mu + 3\sigma \quad (1.1)$$

$$\text{Нижня межа: } \mu - 3\sigma \quad (1.2)$$

Де μ – середнє значення, а σ – стандартне відхилення.

Серед переваг такого методу можна виділити простоту у реалізації та ефективний для даних, що мають нормальний розподіл. Відповідно, метод не підходить для даних, що не мають нормального розподілу, бо чутливий до викидів, які можуть сильно вплинути на середнє значення та стандартне відхилення. Для даних, які не мають нормального розподілу, розраховане середнє значення та стандартне відхилення можуть не відображати істинний розподіл значень. Наприклад, у асиметричних розподілах велика частина даних може бути зосереджена на одному боці середнього, що робить середнє та стандартне відхилення менш репрезентативними для всієї вибірки. Це може призвести до неправильних висновків щодо аномальних значень. А у випадках, коли розподіл даних має більше варіацій або неоднорідності, метод 3-Сигма може недооцінювати кількість аномалій, оскільки стандартне відхилення може бути занадто великим і включати в себе значення, які насправді є аномальними.

Метод інтерквартильного діапазону (IQR)

Метод IQR використовує середній розмах (interquartile range) даних для виявлення аномалій. Він визначає межі аномалій, відштовхуючись від першого (Q1) і третього кuartилів (Q3), що охоплюють центральні 50% даних.

Формула:

$$IQR = Q3 - Q1 \quad (1.3)$$

$$\text{Верхня межа: } Q3 + 1,5 \times IQR \quad (1.4)$$

$$\text{Нижня межа: } Q1 - 1,5 \times IQR \quad (1.5)$$

Цей метод вже підходить для даних без нормального розподілу, але водночас, може не враховувати зміни в розподілі даних, тим самим не найкращим чином підходить для виявлення аномалій у сильно змінюваних даних.

Метод Z-Score

Метод Z-Score використовує відхилення кожної точки даних від середнього значення, нормалізоване на стандартне відхилення. Він дозволяє оцінити, наскільки кожна точка даних відхиляється від середнього значення в термінах стандартних відхилень.

Формула:

$$Z = \frac{X - \mu}{\sigma} \quad (1.6)$$

Де X — значення точки даних, μ — середнє значення, а σ — стандартне відхилення

Цей метод добре себе показує для виявлення аномалій в нормальних розподілах, але також не підходить для даних без нормального розподілу. Z-Score передбачає, що дані мають нормальний розподіл. Це означає, що більшість значень повинні розташовуватися поблизу середнього. У випадку ненормального розподілу, Z-Score може неправильно оцінювати розподіл даних і, відповідно, пороги для визначення аномалій. Наприклад, у даних з сильним перекосом стандартне відхилення може бути ненадійним показником варіацій.

Таблиця 1.1 — Порівняння методів

Метод	Переваги	Недоліки
3-сигма	Простий і ефективний для нормальних розподілів	Чутливий до викидів і не підходить для ненормальних розподілів
IQR	Стійкий до викидів і підходить для ненормальних розподілів	Може не враховувати зміни в розподілі даних
Z-Score	Добре підходить для нормальних розподілів і кількісної оцінки	Чутливий до викидів і не підходить для ненормальних розподілів

Статистичні методи, такі як 3-сигма, IQR, Z-Score, є простими у використанні і добре підходять для базових аналізів або коли дані мають прості структури або стабільні розподіли. Вони особливо корисні для швидкого та зрозумілого виявлення аномалій у невеликих наборах даних або у даних з передбачуваними розподілами.

Розроблений метод на основі LSTM з додатковими ознаками, який застосований у цій роботі, має переваги над класичними методами. По-перше, LSTM моделі здатні запам'ятовувати інформацію на довгі проміжки часу, що є важливим для часових рядів з довготривалими залежностями. Це дозволяє моделі більш точно прогнозувати майбутні значення та виявляти аномалії. По-друге, інтеграція додаткових ознак, дозволяє моделі краще розуміти структуру даних та підвищувати точність виявлення аномалій. Ці додаткові ознаки надають моделі більше інформації про поведінку часових рядів, що допомагає знижувати кількість хибних позитивних та хибних негативних результатів. По-третє, метод LSTM з додатковими ознаками є високою адаптивністю. Він може бути адаптований до різних типів даних та різних застосувань, що робить його більш універсальним. Це особливо важливо для застосувань у різних галузях, де часові ряди можуть мати різні характеристики.

1.5 Використання методів виявлення аномалій у SIEM системах

Системи SIEM (Security Information and Event Management) є ключовим компонентом сучасних архітектур кібербезпеки, забезпечуючи централізований збір, аналіз і моніторинг даних про безпеку з різних джерел. [5] Ці системи об'єднують інформацію про події з операційних систем, мережевих пристроїв, додатків, баз даних та інших елементів IT-інфраструктури. Основною метою SIEM-систем є виявлення потенційних загроз та інцидентів безпеки шляхом співставлення та аналізу зібраних даних, що дозволяє швидко реагувати на підозрілу активність та запобігати атакам.[6] Вони також допомагають організаціям відповідати нормативним вимогам і надавати звітність про безпеку.

Існує кілька типів SIEM-систем, кожна з яких має свої особливості. [7] Традиційні SIEM-системи покладаються на заздалегідь визначені правила кореляції для виявлення загроз, що добре підходить для організацій з відносно стабільним і передбачуваним середовищем. У той же час SIEM-системи нового покоління використовують ML, AI та аналітику великих даних, щоб забезпечити більш гнучке та адаптивне реагування на нові загрози. Хмарні SIEM-рішення забезпечують масштабованість і доступність для організацій, що використовують хмарні сервіси, в той час як керовані SIEM дозволяють стороннім провайдерам взяти на себе моніторинг і управління безпекою, що особливо корисно для організацій з обмеженими внутрішніми ресурсами. [8][9]

Розроблений метод виявлення аномалій на основі LSTM має значний потенціал для підвищення ефективності SIEM-систем. По-перше, LSTM-моделі, які здатні запам'ятовувати і враховувати довгострокові залежності, є хорошим інструментом для аналізу даних часових рядів, які часто зустрічаються в журналах подій і мережевих даних. Це дозволяє виявляти зміни в поведінці, які відбуваються поступово або мають довгострокову кореляцію, що важливо для виявлення складних загроз.

По-друге, така модель LSTM може ефективно виявляти складні та приховані аномалії, які не завжди можна ідентифікувати за допомогою традиційних підходів. Задавання додаткових ознак дозволяють моделі краще розуміти контекст даних, що допомагає зменшити кількість хибних спрацьовувань і підвищує точність виявлення загроз. [10] Це особливо важливо для зменшення навантаження на аналітиків безпеки та покращення реагування на інциденти.

Крім того, ваш метод є доволі адаптивним і універсальним. Його можна легко адаптувати до різних типів даних і умов, що робить його корисним інструментом для інтеграції в різні SIEM-системи. Нарешті, використання методів МН, таких як LSTM, для виявлення аномалій в режимі реального часу дозволяє швидко реагувати на нові загрози. Завдяки здатності обробляти великі

обсяги даних, ваш метод може бути ефективно інтегрований в SIEM-системи, які працюють з великими обсягами інформації з численних джерел.

Таким чином, інтеграція методу виявлення аномалій на основі LSTM з додатковими функціями в SIEM-системи може значно підвищити їх ефективність в контексті кібербезпеки. Це допоможе краще виявляти складні та динамічні загрози, зменшити кількість хибних спрацьовувань та адаптуватися до нових викликів безпеки.

Висновки до розділу 1

Переглянувши основні дані про доступні методи виявлення аномалій у часових рядах, стає очевидним, що це залишається актуальним завданням для аналітиків і дослідників у різних галузях. У цьому розділі були розглянуті поняття часових рядів і аномалій, а також проаналізовані різні підходи до виявлення аномалій, включаючи як класичні статистичні методи, так і методи машинного навчання.

Висвітлено переваги та недоліки традиційних методів, які часто використовуються для аналізу часових рядів. Ці методи характеризуються простотою реалізації і швидкими розрахунками, але мають свої обмеження в можливості враховувати складні залежності і інтегрувати додаткові характеристики даних.

Також були розглянуті сучасні методи машинного навчання, зокрема рекурентні нейронні мережі та їх модифікація — довгострокова короткочасна пам'ять (LSTM). Мережі LSTM були детально проаналізовані завдяки своїй здатності моделювати довгострокові залежності часових рядів і використовувати додаткові функції для підвищення точності виявлення аномалій. Порівняння цих методів показало, що модель LSTM з додатковими можливостями може значно перевершити класичні підходи в складних завданнях виявлення аномалій завдяки своїй гнучкості та потужності, і є перспективною для подальших досліджень і впровадження в різні сфери, зокрема SIEM системи нового покоління. Обраний

підхід пропонує ефективні методи виявлення аномалій, які можуть бути використані в майбутніх дослідженнях і експериментах.

2 РОЗРОБКА МЕТОДУ ВИЯВЛЕННЯ АНОМАЛІЙ

2.1 Прогнозування

Прогнозування є одним із важливих аспектів у виявленні аномалій у часових рядах. Воно передбачає використання моделей, які базуються на історичних даних, для передбачення майбутніх значень. Важливою особливістю прогнозування є те, що майбутнє недоступне, і його слід оцінювати лише на основі того, що вже сталося.

Прогнозування — послідовність дій, яка виражає спосіб побудови моделі прогнозування часових рядів. Модель прогнозування — це функціональне представлення, яке адекватно описує часовий ряд і служить основою для отримання майбутніх значень процесу. Процес прогнозування включає визначення моделі для конкретного часового ряду та дії з оцінки якості прогнозних значень. Прогнозування може застосовуватися за умови виконання двох основних умов: доступність числової інформації про минуле, і припущення, що деякі аспекти минулих подій продовжуватимуться і в майбутньому.

Прогнозування можна розділити на два види: однозмінне і багатозмінне. Однозмінне прогнозування використовує одиночні ряди даних, які можуть бути безперервними або категоричними даними. Багатозмінне прогнозування включає декілька споріднених серій, що дозволяє отримати спільну оцінку по ряду даних. Існує багато методів прогнозування, кожен метод має свою специфіку, точність і вартість, розроблених для різних дисциплін і цілей. Більшість методів прогнозування використовують або дані часових рядів, або перехресні дані.

Для поставленої задачі ми вибрали підхід, який поєднує ML з додатковими ознаками, що покращують якість прогнозування і виявлення аномалій.

Підготовка даних

Для підготовки даних ми нормалізуємо часовий ряд до інтервалу $[0, 1]$, щоб уникнути впливу різних масштабів на навчання моделі. Додатково ми генеруємо кілька ознак, які допомагають моделі краще зрозуміти контекст даних.

Визначення 2.1. Середнє значення: обчислюється для кожного вікна даних і дозволяє моделі зрозуміти загальний рівень значень у цьому вікні.

Формула середнього значення:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.1)$$

— x_i — значення вікна

— n — кількість значення у вікні

Середнє значення є ефективним, оскільки воно надає модель інформацію про центральну тенденцію даних у конкретному вікні.

Визначення 2.2. Стандартне відхилення: відображає варіативність даних у вікні і показує, наскільки значення розподілені навколо середнього.

Формула стандартного відхилення:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.2)$$

— x_i — значення вікна

— $\bar{x}$ — середнє значення

— n — кількість значення у вікні

Стандартне відхилення ефективно показує рівень розсіювання даних, що може бути індикатором стабільності або нестабільності у часі.

Визначення 2.3. Мінімальне та максимальне значення: дозволяють моделі зрозуміти межі значень у кожному вікні, що допомагає визначити екстремальні точки.

Формули мінімального та максимального значень:

$$x_{min} = \min(x_1, x_2 \dots, x_n) \quad (2.3)$$

$$x_{max} = \max(x_1, x_2 \dots, x_n) \quad (2.4)$$

Мінімальні та максимальні значення ефективні, оскільки вони допомагають моделі зрозуміти амплітуду коливань даних і ідентифікувати потенційні аномалії.

Визначення 2.4. Перцентилі (5-й та 95-й): Перцентилі допомагають виявляти крайні значення і можливі аномалії, показуючи межі, в яких знаходиться певний відсоток даних.

Формули для 5-го та 95-го перцентилів визначаються як значення, нижче яких знаходиться 5% та 95% даних відповідно:

$$P_5 = \text{перцентиль}(x_1, x_2 \dots, x_n, 5) \quad (2.5)$$

$$P_{95} = \text{перцентиль}(x_1, x_2 \dots, x_n, 95) \quad (2.6)$$

Перцентилі ефективні для виявлення аномалій, оскільки вони вказують на відхилення від звичайного діапазону значень.

Створення навчальних та тестових вибірок

Ми використовуємо метод ковзного вікна для формування навчальних і тестових вибірок. Це означає, що для кожного вікна значень створюються вхідні дані та відповідні цільові значення для навчання моделі. Ковзне вікно дозволяє моделі вивчати динаміку зміни даних на різних часових проміжках.

2.2 Навчання моделі LSTM

Модель LSTM (Long Short-Term Memory) тренується на підготовлених даних. LSTM-мережі ефективно працюють з послідовними даними та здатні запам'ятовувати довгострокові залежності. Процес навчання полягає у

мінімізації середньоквадратичної помилки між прогнозованими та фактичними значеннями, що дозволяє моделі навчитися передбачати майбутні значення на основі поточних вхідних даних.

Визначення 2.5. Середньоквадратична помилка (Mean Squared Error, MSE) є одним із основних показників для оцінки якості моделі прогнозування. Вона обчислюється як середнє арифметичне квадратів різниць між фактичними та прогнозованими значеннями. Формула для обчислення середньоквадратичної помилки виглядає наступним чином:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.7)$$

- y_i — фактичне значення для i -го спостереження
- $\bar{y}$ — прогнозоване значення для i -го спостереження
- n — загальна кількість спостережень

MSE є корисним для визначення середнього рівня помилки між прогнозованими та фактичними значеннями. Чим менше значення MSE, тим точнішою є модель.

Прогнозування:

Нейронні мережі з довгою короткостроковою пам'яттю призначені для розв'язання проблем, з якими зіштовхуються звичайні рекурентні нейронні мережі. Описаний вид мережі було розроблений для розв'язування проблеми тривалих залежностей. Це модифікація рекурентної нейронної мережі, у якій передбачено можливість та вміння зберігати та обробляти інформацію на порівняно тривалий проміжок часу. Основною ознакою рекурентної нейронної мережі є ланцюгова структура, побудована з повторюваних модулів мережі. Звичайна RNN складається з модулів мережі, структура яких виявляє досить просто, для її побудови, наприклад, може вистачити лише одного шару з функцією активації тангенс гіперболічний

Після навчання модель використовується для прогнозування значень на тестовій вибірці. Прогнозовані значення порівнюються з фактичними для оцінки

точності моделі. Важливою особливістю є те, що модель надає більше значення новим даним, що підвищує її адаптивність до змін. Приклад такої мережі можемо побачити на Рисунку 2.1:

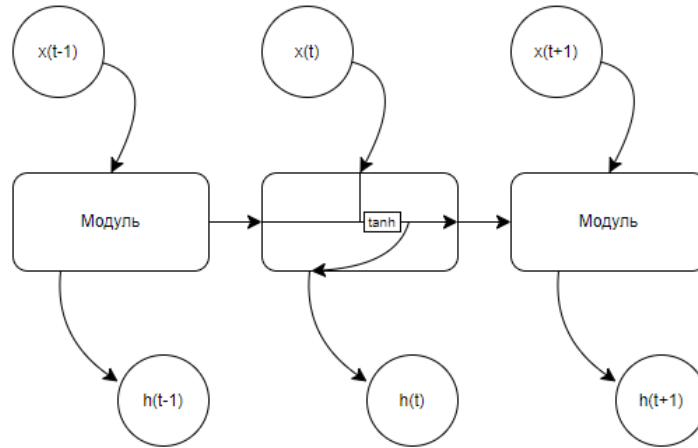
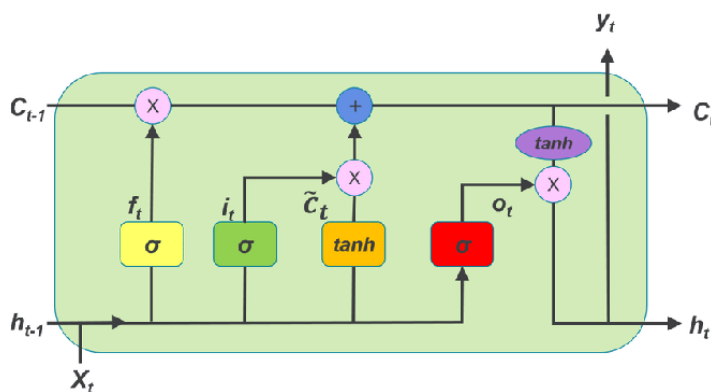


Рисунок 2.1 – Рекурентна нейронна мережа з функцією активації

$$\tanh(x)$$

Детальне вивчення модуля звичайної рекурентної нейронної мережі показує, що він не є дуже складним. Це тому, що сканування модуля містить одношаровий перцептрон з активаційною функцією. Нейронні мережі з довготривалою короткочасною пам'яттю використовують модуль, який є набагато складнішим (рис. 2.2).[\[11\]](#) Такий модуль містить декілька функцій активації й велику кількість різноманітних математичних перетворень:



$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ \tilde{c}_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\ h_t &= \tanh(o_t \odot c_t) \end{aligned}$$

Рисунок 2.2 – Структура LSTM клітинки в нейромережі

Прогнозування у часових рядах за допомогою моделей, таких як LSTM (Long Short-Term Memory), базується на використанні попередніх значень для передбачення майбутніх. Після навчання на тренувальних даних модель використовується для прогнозування наступних значень на тестовій вибірці.

Для моделі LSTM процес прогнозування може бути описаний наступним чином[12]:

1. Нехай X_t — вектор вхідних значень у момент часу t .
2. Модель прогнозує наступне значення $\hat{y}_{t+1}$ на основі поточного вхідного вектора X_t та попередніх внутрішніх станів.

Основне рівняння прогнозування можна подати у вигляді:

$$\hat{y}_{t+1} = f(X_t) \quad (2.8)$$

де f — це функція, що представляє модель LSTM.

Формально, модель LSTM може бути описана наступними рівняннями:

1. Вхідний гейт: $i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i)$ (2.9)

2. Забуваючий гейт: $f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f)$ (2.10)

3. Гейт виходу: $o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o)$ (2.11)

4. Оновлення стану комірки: $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, X_t] + b_C)$ (2.12)

5. Стан комірки: $C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$ (2.13)

6. Вихідний стан: $h_t = o_t \cdot \tanh(C_t)$ (2.14)

де:

- σ — сигмоїдна функція активації,
- $\tanh$ — гіперболічний тангенс,
- W_i, W_f, W_o, W_C — ваги відповідних гейтів,
- b_i, b_f, b_o, b_C — зміщення (біаси) відповідних гейтів,

- h_t — прихований стан у момент часу t ,
- C_t — стан комірки у момент часу t .

Прогнозоване значення $\hat{y}_{t+1}$ обчислюється на основі вихідного стану h_t :

$$\hat{y}_{t+1} = W_y \cdot h_t + b_y \quad (2.15)$$

де:

- W_y — ваги вихідного шару,
- b_y — зміщення вихідного шару.

Ці рівняння показують, як LSTM використовує попередні значення та внутрішні стани для прогнозування наступного значення. Модель адаптується до нових даних, що дозволяє їй ефективно справлятися зі змінами у часових рядах.

Виявлення аномалій:

Для визначення аномалій у часових рядах використовується метод обчислення залишків між фактичними та прогнозованими значеннями. Цей метод дозволяє оцінити, наскільки фактичні значення відхиляються від передбачених. Якщо залишок перевищує певний поріг, це вважається аномалією. У нашому методі поріг визначається як 96-й перцентиль розподілу залишків.

1. Обчислення залишків:

Залишок e_t обчислюється як різниця між фактичним значенням y_t та прогнозованим значенням $\hat{y}_t$:

$$e_t = y_t - \hat{y}_t \quad (2.16)$$

2. Розподіл залишків:

Для визначення порогу необхідно зібрати всі залишки $\{e_t\}$ та обчислити їх розподіл. Це може бути зроблено шляхом сортування залишків та визначення перцентилі.

3. Визначення порогу аномалії:

Поріг аномалії P обирається, наприклад, як 96-й перцентиль розподілу залишків. Для цього необхідно знайти значення залишку, яке перевищує 96% всіх інших залишків. Якщо залишки впорядковані за зростанням, то:

$$P = e_{[0.96 \times N]} \quad (2.17)$$

де N — кількість залишків, а $[\cdot]$ — операція округлення до більшого цілого числа.

4. Виявлення аномалій:

Якщо залишок e_t для певного часу t_t перевищує визначений поріг P , це значення вважається аномалією:

$$\text{Аномалія}(t) = \begin{cases} 1, & \text{якщо } e_t > P \\ 0, & \text{якщо } e_t \leq P \end{cases} \quad (2.18)$$

Таким чином, процес виявлення аномалій можна підсумувати наступними кроками:

1. Обчислення залишків e_t для всіх точок даних.
2. Визначення порогу аномалії P як певного значення перцентилу розподілу залишків.
3. Класифікація кожного залишку e_t як аномалії, якщо e_t перевищує поріг P .

Цей підхід дозволяє виявляти значні відхилення від нормальної поведінки у часових рядах, що може вказувати на аномальні події або зміни в даних.

Висновки до розділу 2

У цьому розділі було детально розглянуто методику розробки та реалізації моделі виявлення аномалій у часових рядах. Були визначені основні теоретичні аспекти прогнозування та особливості роботи з часовими рядами, що є важливим для розуміння процесу виявлення аномалій.

Початковий етап включав збір та підготовку даних, де ми відібрали та обґрунтували вибір ключових характеристик, таких як середнє, стандартне відхилення, мінімальне та максимальне значення, а також 5-й та 95-й процентилі. Ці характеристики були обрані через їхню здатність ефективно відображати структуру та мінливість даних у часових рядах.

На основі зібраних даних було побудовано модель на основі LSTM, яку було використано для прогнозування майбутніх значень. Основний підхід моделі полягав у тому, щоб приділяти більше уваги новим даним, що дозволило забезпечити більшу адаптивність до змін у часі.

Після побудови моделі прогнозування залишки між фактичними та прогнозованими значеннями були використані для виявлення аномалій. Розглянуто та обґрунтовано метод розрахунку порогу аномалії з використанням 96-го процентиля залишкового розподілу, що дозволило ефективно виявляти значні відхилення від нормальної поведінки.

Таким чином, у цьому розділі розроблено комплексний підхід до виявлення аномалій, який включає всі етапи від підготовки даних до побудови та використання моделі для прогнозування та виявлення аномалій. Запропонований метод дозволяє надійно і точно виявляти аномалії в часових рядах, що може бути корисним для широкого кола застосувань в різних галузях.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

Аби перевірити здійсненність та доказовість запропонованого підходу для виявлення аномалій у часових рядах, було проведено його реалізацію. Для реалізації методу був написаний застосунок на мові Python.

Реалізовані експерименти, зосереджені на наступному: проведення загального виявлення аномалій NP-AD у часових рядах різної природи.

А саме: експеримент над часовими рядами, аби визначити, чи була якась аномалія в історичній зміні даних; експеримент, аби визначити аномалії у більш зашумлених даних.

Використано різні типи рядів: несезонні часові ряди з викидами та датасети Numenta Anomaly Benchmark (NAB), щоб показати аномалії в часових рядах.

Також провів порівняння розробленого методу з класичним методом 3-сигма.

Відображення результатів наведене в Розділах 3.1, 3.2 та 3.3 відповідно.

3.1 Підхід 1: Виявлення загальної аномалії

Для часових рядів роль експерименту, проведеного в рамках першого підходу, полягала в тому, щоб визначити, чи є якась аномалія у історичних даних. На рисунку показано графік часових рядів, що містить дані та виявлені аномалії. Сіра лінія представляє собою значення часових рядів, а червоні точки – виявлені аномалії.



Рисунок 3.1 — Виявлення точкових аномалій

Перші точкові аномалії знаходяться приблизно на 150-ій позначці часу. Хоча вони й виглядають як значні відхилення візуально, було важливу спершу перевірити роботу алгоритму, який визначив їх як аномалії через перевищення встановленого порогу, який розраховувався на основі порогового значення визначення аномалій у 0,98. Остання аномалії також виявляється через значне відхилення від «нормальних» значень. Ці точки показують ще більш різке збільшення показників, що відзначається алгоритмом як аномалії.

NAB_art_daily_perfect_square_wave

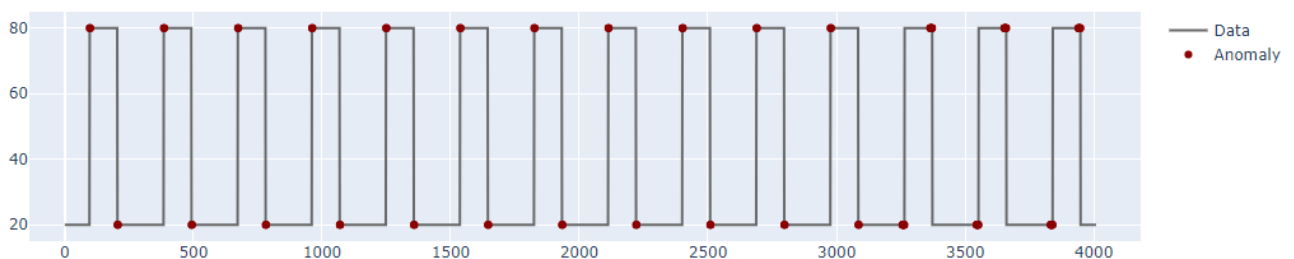


Рисунок 3.2 — Виявлення точкових аномалій у циклічних даних

З рисунку 3.1.2 видно, як модель розпізнає аномалії в NAB часовому ряду без очевидно виявлених аномалій. Але, на даному рисунку виявлені аномалії дійсно є перехідними між двома різними, доволі стабільними станами: значення, близькі до 20, і значення, близькі до 80. Цікаво, що в певний момент часу (3000-3500-ті позначки) точка виявлення аномалії змінює своє звичне розташування, тобто модель помітила якусь мінімальну зміну поведінки навіть у настільки циклічних даних, і виявляє місце переходу інакше.

rds_cpu_utilization_cc0c53

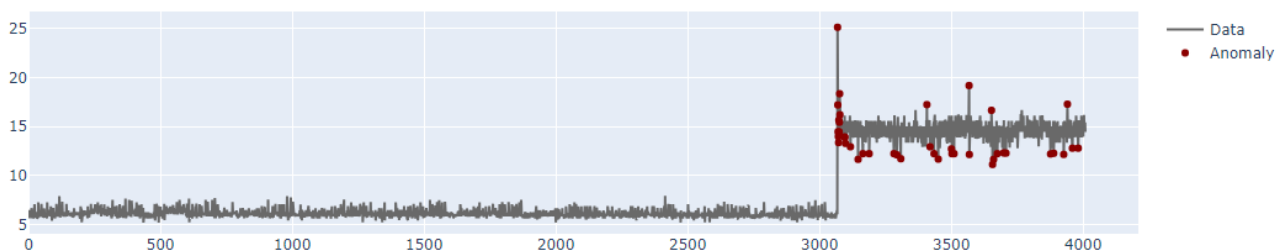


Рисунок 3.3 — Виявлення аномалій CPU даних у RDS

rds_cpu_utilization_cc0c53

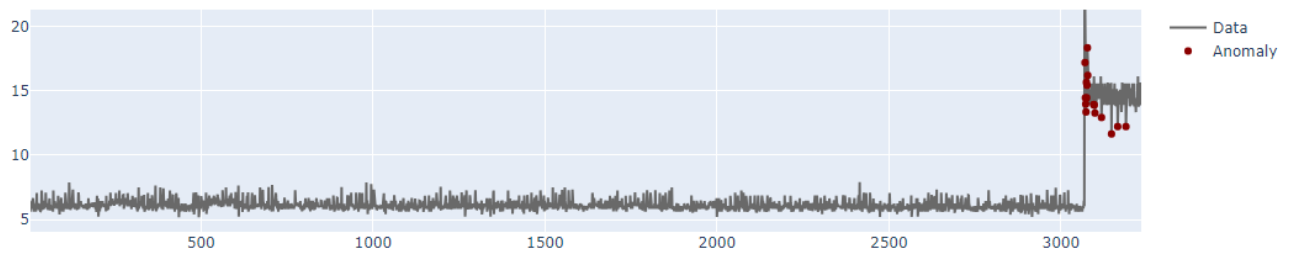


Рисунок 3.4 — Масштабований 3.3

На рисунках 3.3 та 3.4 (масштабований 3.3) NAB часовому ряду видно, як модель визначає аномалії в ряді, коли після тривалого часу малої активності (~5-7), з'явилося різке зростання значень, після якого дуже велика кількість точок визнається як аномалія. Однак пізніше цей новий діапазон поведінки стає нормальним, і вже на ньому модель розпізнає велику кількість викидів, що виходять за рамки перценталів.

ec2_network_in_5abac7

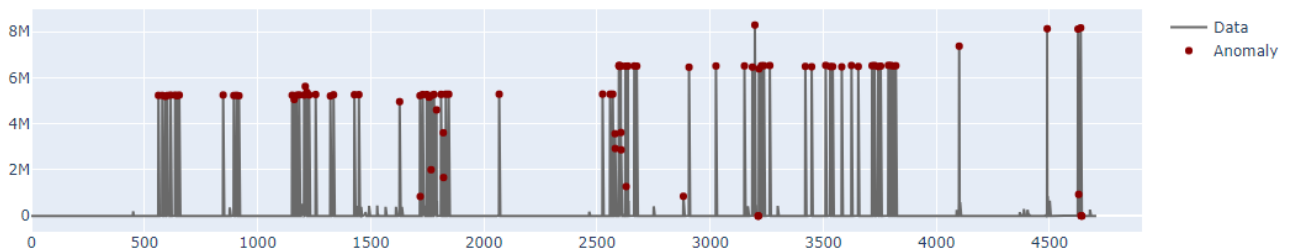


Рисунок 3.5 — Точкові аномалії EC2 вебсервісу

Графік відображає аналіз мережевого трафіку EC2 (імовірно, вхідного трафіку) для конкретного серверу або групи серверів. На графіку показано, як змінюється обсяг мережевого трафіку з часом, а також вказано, де виявлені аномалії. У кількох місцях графіку можна спостерігати цілі кластери аномалій. Це може свідчити про певний період або подію, що викликає постійні аномальні збої або різкі зміни в обсязі трафіку.

Такі кластери можуть вказувати на інциденти, які варто досліджувати окремо, наприклад, потенційні атаки, технічні збої або нестандартні операції. Існують також періоди, де трафік виглядає стабільним і без аномалій. Це означає, що в ці моменти обсяг мережевого трафіку був у межах норми, і модель не виявила відхилень.

3.2. Підхід 2: Виявлення аномалій в зашумлених даних

Цей підхід в основному показує кроки, зроблені при виявленні нетипових аномалій. Розглядаючи більш складні та зашумлені дані з очевидними точками аномалій, ніж ті, що були використані в Розділі 3.1

NAB_art_daily_flatmiddle

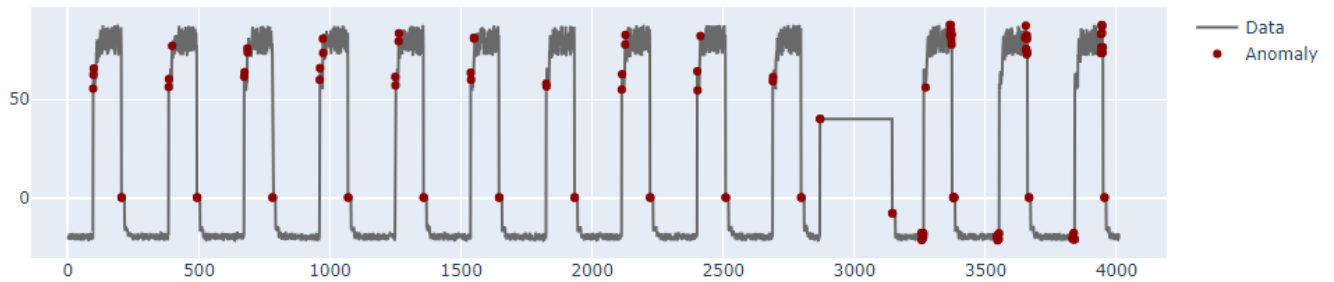


Рисунок 3.6 — Контекстні аномалій

NAB_art_daily_jumpsdown

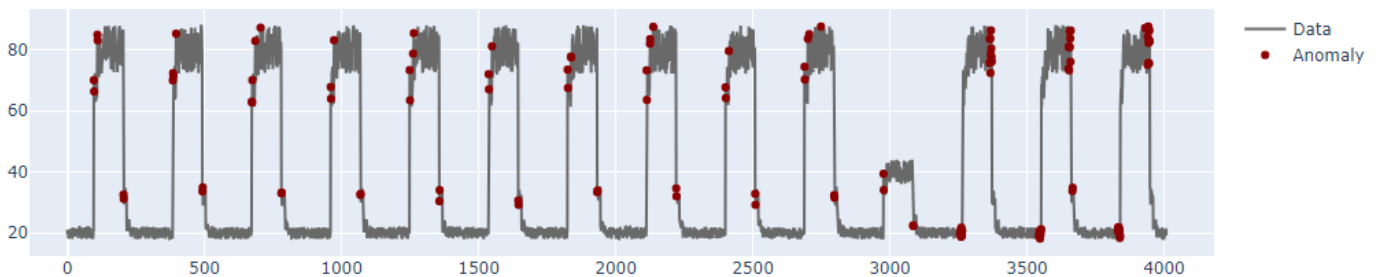


Рисунок 3.7 — Контекстні аномалій приклад №2

На рисунках 3.2.1 та 3.2.2 видно, що модель виявляє аномалії в місцях різких змін значень часового ряду. Це особливо помітно в ділянках, де дані різко знижуються і піднімаються, що є характерною ознакою аномальних поведінок. В діапазоні (500, 1000) модель ідентифікує кілька аномалій під час різкого підвищення і зниження значень. В діапазоні (1500, 2000) модель також виявляє аномалії під час різких змін, підтверджуючи її здатність виявляти контекстні аномалії.

Модель виявляє контекстні аномалії на циклічно повторюваних змінах, замість виявлення колективних в діапазоні (2900, 3100). Їх модель теж розглядає лише як контекстні. У підсумку, ці графіки демонструють здатність моделі ефективно виявляти контекстні аномалії, що виникають у результаті змін патернів в даних, але не завжди можуть ідентифікувати колективні аномалії. Це може бути наслідком використання перцентилів як ознак і як порогу для виявлення аномалій. Перцентилі в ознаках допомагають моделі розуміти розподіл даних в кожному вікні, а перцентиль для порогу (у даному випадку 0.96) виявлення аномалій визначає значні відхилення від нормальної поведінки.

NAB_speed_6005

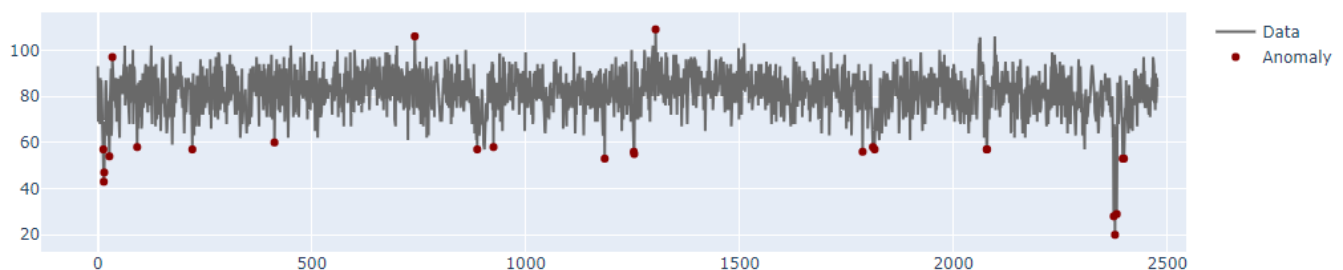


Рисунок 3.8 — Точкові аномалії

На рисунку видно результати роботи моделі виявлення аномалій для набору даних чергового ряду NAB. Графік демонструє, як модель ідентифікує аномалії в часовому ряді даних, що характеризуються високою варіативністю значень.

На цьому графіку модель виявляє аномалії в різних точках часового ряду, де відбуваються значні відхилення від середнього значення. Зокрема:

В діапазоні (0, 500) модель виявляє кілька аномалій у точках, де значення часового ряду суттєво перевищують або падають нижче середнього рівня. Це свідчить про високу чутливість моделі до локальних відхилень.

В діапазоні (1000, 1500) видно, що модель продовжує ідентифікувати аномалії, що вказує на її здатність до виявлення змін у змінах даних, навіть якщо вони незначні.

В наступному діапазоні (2000, 2500) модель виявляє декілька аномалій, особливо помітних в точках, де відбуваються різкі падіння значень. Це свідчить про ефективність моделі у виявленні значних відхилень від норми.

Цей графік підтверджує, що модель ефективно ідентифікує аномалії в даних з високою варіативністю, де зміни можуть бути як поступовими, так і різкими. Перцентилі, використані як ознаки, допомагають моделі розуміти розподіл значень у кожному вікні, що дозволяє їй краще визначати аномальні точки. Перцентиль, використовуваний для визначення порогу аномалій (0,98 у цьому випадку), забезпечує ефективне відсіювання значних відхилень, що гарантує високу адаптивність моделі до змін у даних.

remaining_behavior sequence_length(count)

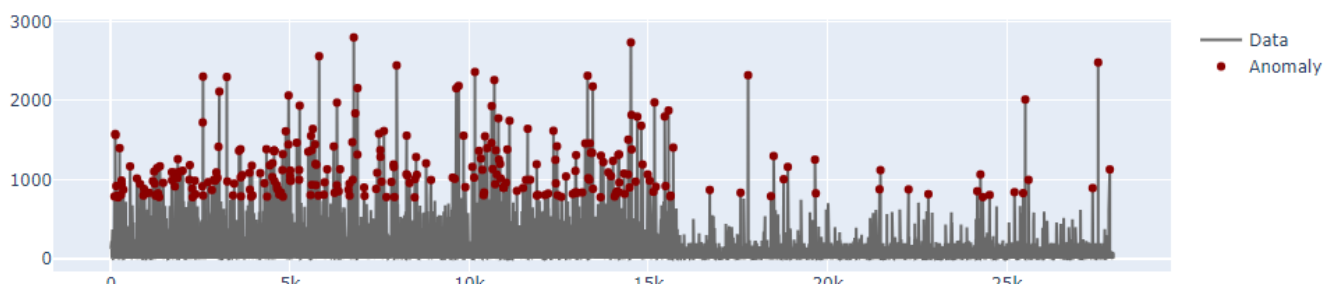


Рисунок 3.9 — Групові та точкові аномалії

remaining_behavior sequence_length(count)

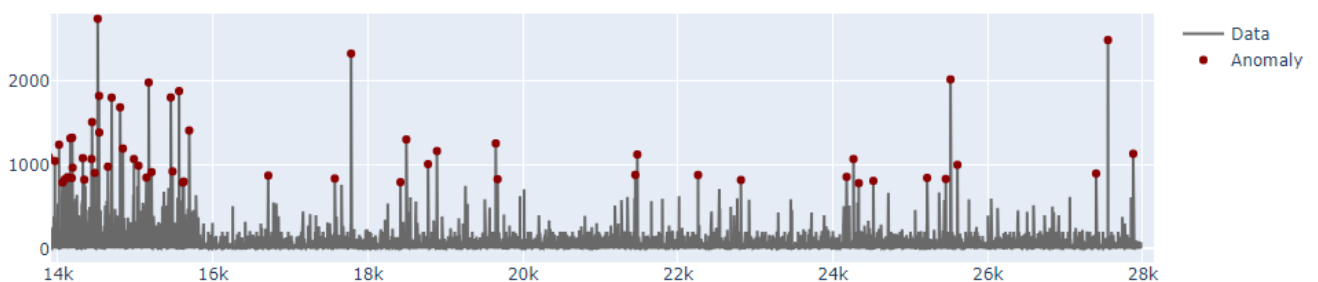


Рисунок 3.10 — Масштабований 3.9

На рисунках наведено обрахунки моделі для набору даних чергового ряду НАВ. Графік демонструє, як модель ідентифікує аномалії в часовому ряді даних, що характеризуються високою варіативністю значень.

На рисунку видно, як модель виявляє аномалії в різних точках часового ряду контекстні аномалії: модель виявляє аномалії в місцях різких змін значень часового ряду.

Це помітно в діапазонах, де дані різко змінюються, наприклад, в діапазоні (0, 5000): ідентифікує аномалії під час різкого підвищення значень. Часті випадки аномалій, що вказує на високу чутливість моделі до змін й викидів даних, що виходять за межі перцентилів.

В діапазоні (15000, 28000) відзначаються рідкісні випадки аномалій, що вказує на знижену активність та більшу стабільність даних в цьому інтервалі.

Тож модель ефективно виявляє контекстні аномалії, які виникають у зв'язку із різкою зміною у даних. Загалом, графік демонструє здатність моделі ідентифікувати аномалії в широкому діапазоні значень, підтримуючи точність виявлення аномальної поведінки.

remaining_behavior num_sessions

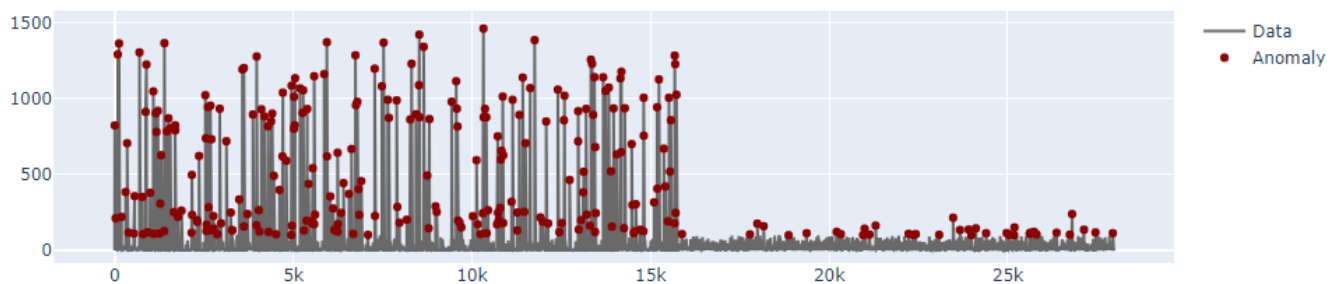


Рисунок 3.11 — Групові та точкові аномалії приклад №2

remaining_behavior num_sessions

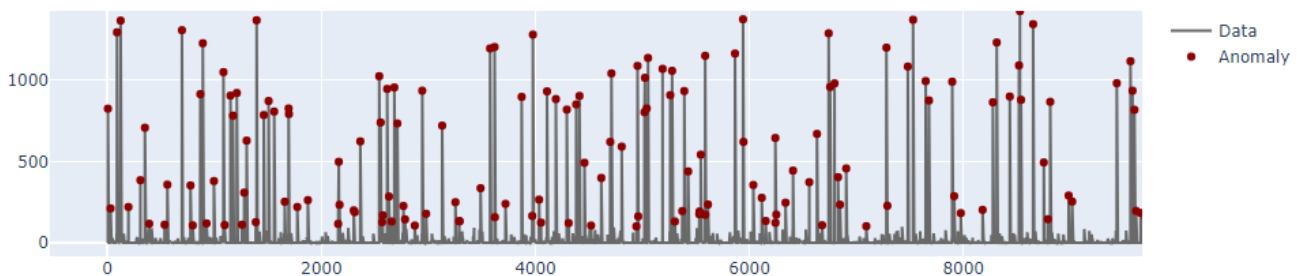


Рисунок 3.12 — Масштабований 3.11

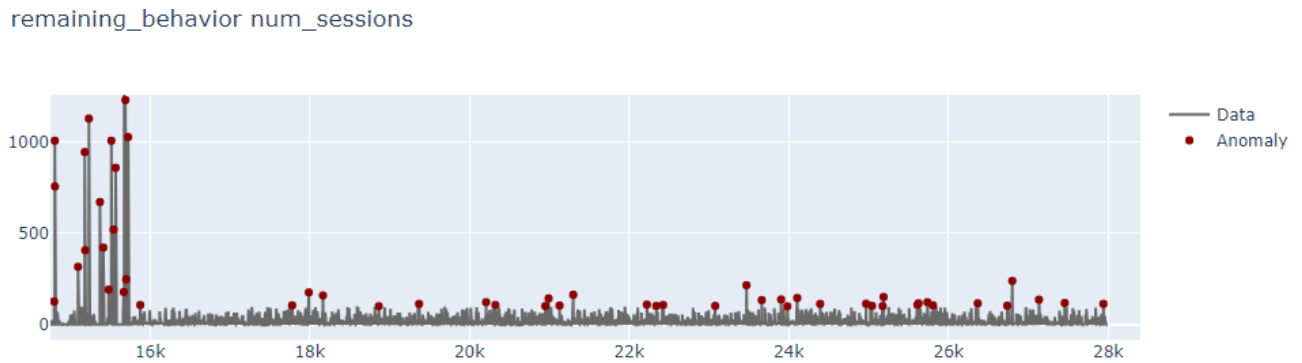


Рисунок 3.13 — Масштабований 3.11

На графіку показані результати роботи моделі виявлення аномалій для набору даних чергового ряду NAV. Графік демонструє, як модель ідентифікує аномалії в часовому ряді даних, що характеризуються високою варіативністю значень. Модель виявляє як точкові, так і контекстні аномалії в різних точках часового ряду, де відбуваються значні відхилення від середнього значення.

В діапазоні (0, 5000) модель виявляє аномалії у точках, де значення часового ряду суттєво перевищують або падають нижче середнього рівня. В діапазоні (10000, 15000) модель виявляє декілька контекстних аномалій, особливо помітних у точках, де відбуваються різкі падіння значень. Це свідчить про ефективність моделі у виявленні значних відхилень від норми. В діапазоні (15000, 30000) аномалії з'являються рідше, що може свідчити про стабільність даних або знижену активність.

Цей графік підтверджує, що модель ефективно ідентифікує як точкові, так і контекстні аномалії в даних з високою варіативністю, де зміни можуть бути як поступовими, так і різкими. Графіки демонструють здатність моделі ефективно виявляти як контекстні, так і точкові аномалії, що виникають у результаті змін у даних. Модель чутлива до різких змін значень, що дозволяє їй ідентифікувати несподівані відхилення у даних.

ec2_request_latency_system_failure

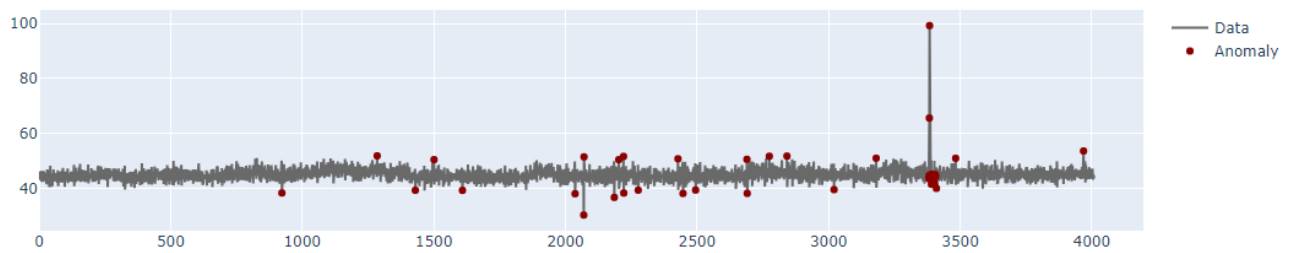


Рисунок 3.14 — Аномалії у кількості запитів EC2 вебсервіс

Графік демонструє аналіз затримки обробки запитів (latency) для системи EC2. На графіку відображено, як змінюється затримка обробки запитів з часом. Більшу частину часу затримка обробки запитів коливається навколо певного середнього рівня, що вказує на стабільну роботу системи. Невеликі коливання навколо цього рівня є нормальними і не розглядаються як аномалії. Виявлені аномалії вказують на моменти часу, коли затримка значно відхиляється від звичайного рівня. Наприклад, є кілька виражених піків (особливо близько 3500), які суттєво перевищують середнє значення затримки і можуть вказувати на серйозні збої або перевантаження системи.

Окрім одного великого піка, є й інші менш помітні, але все ж суттєві сплески в затримці, які також виявлені як аномалії. Це може свідчити про періодичні проблеми або несподівані навантаження на систему.

Наприклад, на інтервалах 1000, 1500 та 2000 видно незначні, але постійні коливання, які також відзначені як аномалії.

Також видно кілька точок, де затримка раптово знижується до значення нижче середнього рівня (наприклад, на інтервалі часу близько 2000). Це теж розцінюється як аномальна поведінка, яка може вказувати на можливі зміни в мережевих умовах або поведінці системи.

3.3 Порівняння отриманих результатів з класичним методом

У підрозділі наведені деякі порівняння результатів виявлення аномалій методом 3-сигма і розробленою моделлю LSTM. Графіки, які показує результати методу 3-сигма для визначення аномалій, демонструють, як цей класичний метод справляється з аналізом наявних датасетів.

NAB_art_daily_flatmiddle

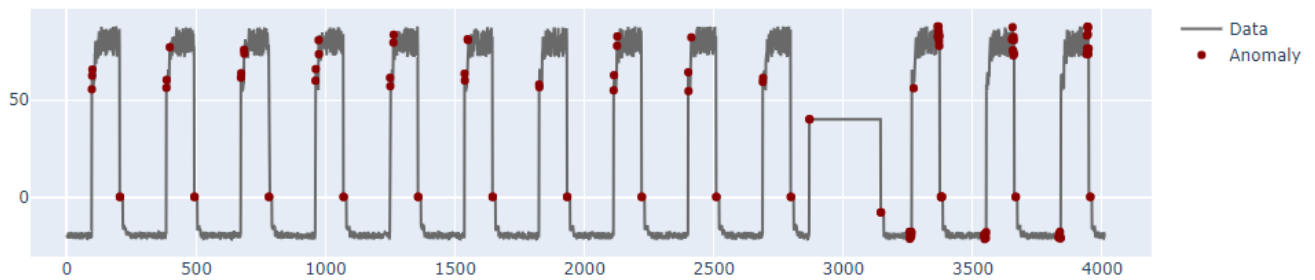


Рисунок 3.15 — Результат розробленої моделі

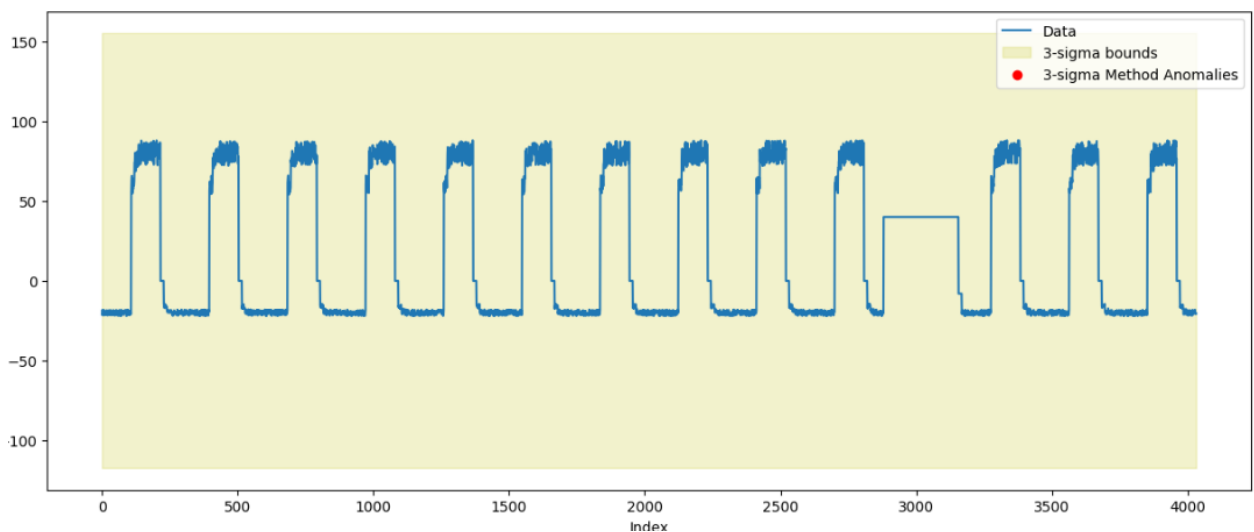


Рисунок 3.16 — Результат 3-сигма методу

Розроблений метод виявив кілька аномалій у часовому ряді. Це свідчить про його здатність виявляти навіть незначні відхилення в даних. Наприклад, зміни в інтервалі 2700-3000 могли б бути критичними, і модель їх розпізнала. В той самий час 3-сигма метод не виявив жодної аномалії, оскільки всі значення знаходяться в межах трьох стандартних відхилень від середнього значення. Це може означати, що метод недостатньо чутливий до специфічних змін у даних або до періодичних сигналів з різкими переходами, як на цьому графіку.

NAB_speed_6005

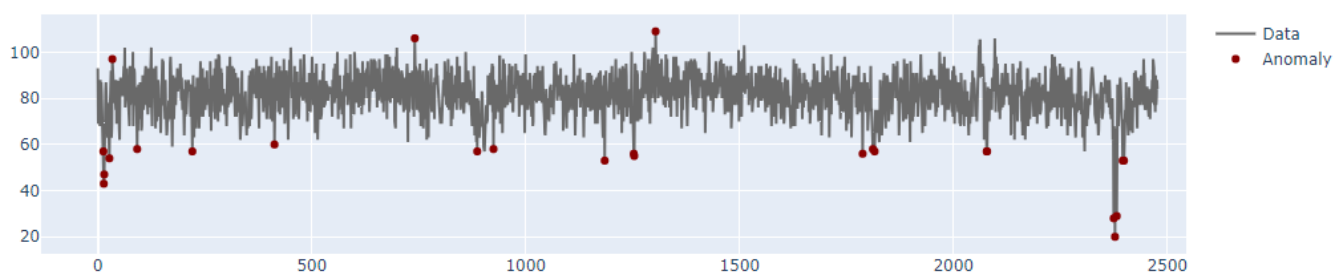


Рисунок 3.17 — Результат розробленої моделі приклад №2

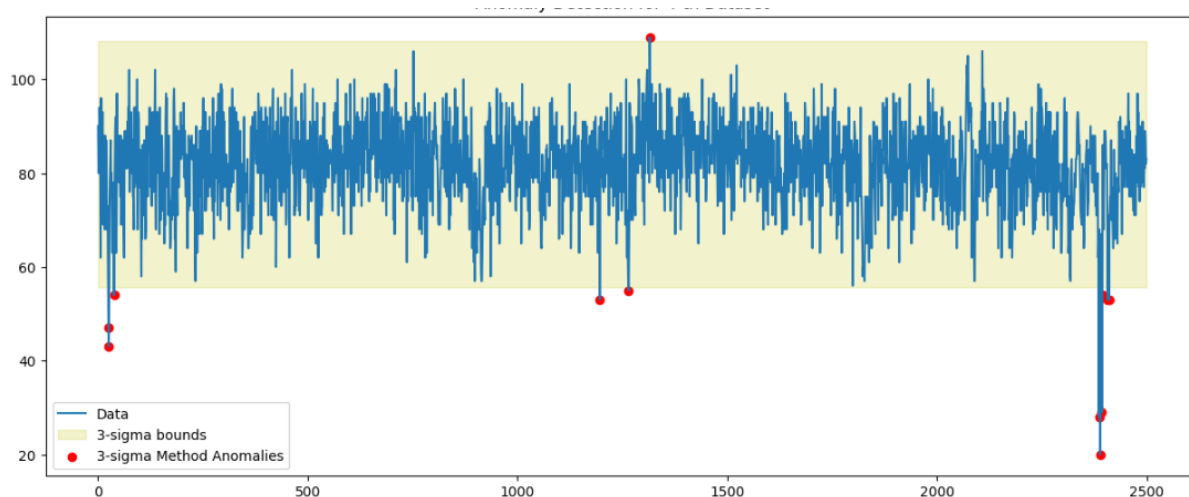


Рисунок 3.18 — Результат 3-сигма методу приклад №2

В цьому випадку розроблена модель МН впоралася схоже до класичного рішення 3-сигма. У результаті, що отримали з використанням розробленої моделі маємо деяку кількість значень, які не визначилися методом 3-сигма, як аномалії, але які виходять за межі перцентилів, що задані у моделі, тобто модель показує більш чутливий результат щодо змін у даних, аніж метод 3-сигма.

rds_cpu_utilization_cc0c53

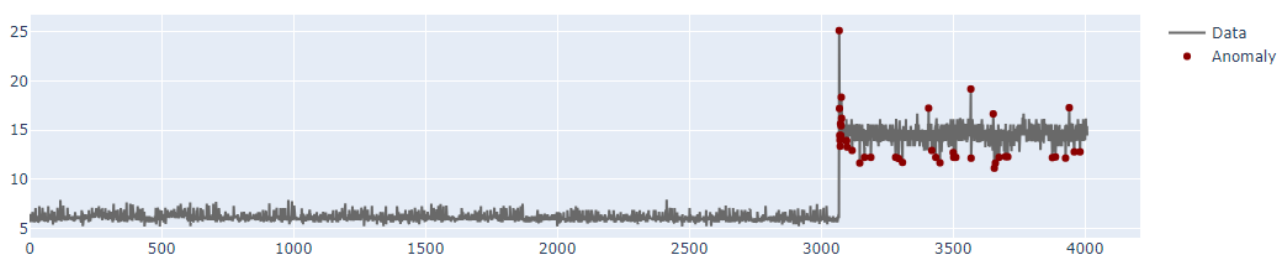


Рисунок 3.19 — Результат розробленої моделі приклад №3

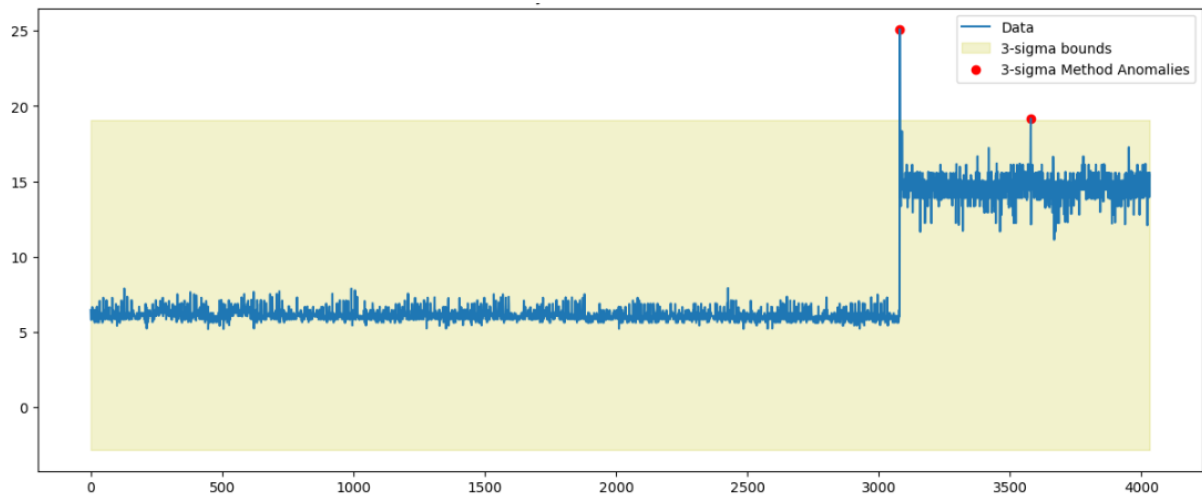


Рисунок 3.20 — Результат 3-сигма методу приклад №3

Рисунок 3.19 та 3.20 наглядно показують кардинальну різницю в детекції викидів між методом 3-сигма та розробленому методі на основі МН. Підхід класичного методу 3-сигма, базується на відхиленні значень більше ніж на три стандартних відхилення від середнього, які він і визначає як аномалії, але це ніяк не може розкрити суті аномалій в «живих» даних без нормального розподілу, коли запропонований метод, за допомогою ковзного вікна, може допомогти вирішити цю проблему і визначає більшу кількість контекстних та точкових викидів, що можуть сигналізувати про ранні збої чи атаки.

Висновки до розділу 3

У розділі «Розробка методу виявлення аномалій» розглянуто процес роботи, реалізований у розробленому методі виявлення аномалій у даних часових рядів. Середнє значення, стандартне відхилення, мінімальні та максимальні значення та проценти допомагають моделі ефективно оцінювати мінливість та розподіл даних у кожному вікні.

У рамках цього підходу модель навчалась на історичних даних, після чого застосовувалась для прогнозування значень на тестовій вибірці. Обчислення залишків між фактичними та прогнозованими значеннями дозволяє ідентифікувати аномалії, коли залишок перевищує поріг, визначений як 96-й перцентиль розподілу залишків. Це забезпечує ефективне виявлення значних відхилень від нормальної поведінки.

Представлені графіки показують здатність моделі ідентифікувати як контекстуальні, так і значні аномалії в різних типах даних. Модель визначає

аномалії в даних високої змінності, що є підтвердженням її адаптивності та точності. Таким чином, розроблений метод виявлення аномалій є надійним інструментом аналізу часових рядів, оскільки забезпечує найвищу якість виявлення відхилень і можливість вчасно реагувати на потенційні проблеми.

ВИСНОВКИ

Розглянуто процес створення, тренування та використання моделі ідентифікації аномалій у даних часових рядів. Метою цієї роботи було розробити потужний і гнучкий підхід до автоматичного виявлення аномалій, який буде застосовуватися в самих різних типах даних і забезпечить найвищу точність при зміні умов.

На початку роботи було представлено огляд теоретичної бази аналізу часових рядів. Розглянуто методи обробки даних: нормалізація, створення ознак для підвищення якості моделювання. Описується попередня обробка даних, яка включає нормалізацію, створення ознак та розбиття на зразки поїздів та тестів. Використання розсувного вікна при створенні особливостей дозволяє моделі враховувати локальні властивості даних, що досить важливо при роботі з часовими рядами.

Також описано процес розробки та тренування моделі прогнозування часових рядів. Використання середніх значень, стандартних відхилень, мінімальних і максимальних значень, а також процентиль кожного вікна даних дозволяє отримати глибоке уявлення про поведінку даних. Це дозволяє більш точно прогнозувати майбутні значення по моделі з урахуванням мінливості і розподілу даних в кожному вікні.

Детальна процедура розробки методу виявлення аномалій представлена в третьому розділі. Обчислення залишків між фактичними та прогнозними значеннями дозволяє визначити аномалії, коли залишок більший за поріг, встановлений на процентилі залишкового розподілу. Це дозволяє ефективно виявляти важливі відхилення від нормальної поведінки. Модель може ідентифікувати як контекстні, так і важливі аномалії в різних типах даних. Модель добре визначає аномалії навіть у даних з високою дисперсією, що доводить її адаптивність і точність.

Тобто метод, розроблений для виявлення аномалій, є надійним для аналізу часових рядів. Це дозволяє отримати якісне виявлення відхилень і своєчасно реагувати на потенційні проблеми. Запропонована попередня обробка даних та методи побудови ознак дозволяють моделі враховувати локальні аномалії, значно підвищуючи її точність та ефективність. Представлені результати підтверджують, що розроблений підхід має великий потенціал для успішного застосування до великого класу проблем, пов'язаних з аналізом часових рядів і виявленням аномалій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Marco Peixeiro // Time Series Forecasting in Python
2. Mohiuddin Ahmed Abdun Naser Mahmood Jiankun Hu // A survey of network anomaly detection techniques [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/science/article/abs/pii/S1084804515002891>
3. Chandola V., Banerjee A., Kumar V. // Anomaly detection: A survey [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/220565847_Anomaly_Detection_A_Survey
4. "Long Short-Term Memory" by S. Hochreiter and J. Schmidhuber [Електронний ресурс] – Режим доступу: <https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735>
5. Josh Fruhlinger, Tim Ferrill // What is SIEM? [Електронний ресурс] – Режим доступу: <https://www.csoononline.com/article/524286/what-is-siem-security-information-and-event-management-explained.html>
6. What is SIEM software? How it works and how to choose the right tool [Електронний ресурс] — Режим доступу: <https://www.csoononline.com/article/2124604/what-is-siem-software-how-it-works-and-how-to-choose-the-right-tool.html>.
7. Ellis Stewart // "Top 10 SIEM Tools to Consider in 2023." [Електронний ресурс] – Режим доступу: <https://em360tech.com/top-10/SIEM-tools>
8. Sumo Logic. "Cloud SIEM." [Електронний ресурс] – Режим доступу: <https://www.sumologic.com/solutions/cloud-siem/>
9. What is managed SIEM? [Електронний ресурс] – Режим доступу: <https://www.crowdstrike.com/cybersecurity-101/managed-siem/>
10. Joshua Gordon // Practical Guide for Feature Engineering of Time Series Data [Електронний ресурс] – Режим доступу: <https://dotdata.com/blog/practical-guide-for-feature-engineering-of-time-series-data/>

11. Antonio Elia Pascarella // Neural Sequence Analysis Toolbox [Электронный ресурс] – Режим доступа:
https://www.researchgate.net/publication/367464385_Neural_Sequence_Analysis_Toolbox
12. C. Olah // Understanding LSTM Networks [Электронный ресурс] – Режим доступа: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

ДОДАТОК А ТЕКСТ ПРОГРАМ

```

def create_dataset(data, time_step=1):

    dataX, dataY = [], []
    for i in range(len(data) - time_step - 1):
        a = data[i:(i + time_step)]
        dataX.append(a)
        dataY.append(data[i + time_step, 0])
    return np.array(dataX), np.array(dataY)

def add_features(data, window_size=3):

    features = []
    for i in range(len(data)):
        if i < window_size:
            features.append([
                data[i, 0],
                np.mean(data[:i+1]),
                np.std(data[:i+1]),
                np.min(data[:i+1]),
                np.max(data[:i+1]),
                np.percentile(data[:i+1], 5),
                np.percentile(data[:i+1], 95)
            ])
        else:
            features.append([
                data[i, 0],
                np.mean(data[i-window_size+1:i+1]),
                np.std(data[i-window_size+1:i+1]),
                np.min(data[i-window_size+1:i+1]),
                np.max(data[i-window_size+1:i+1]),
                np.percentile(data[i-window_size+1:i+1], 5),
                np.percentile(data[i-window_size+1:i+1], 95)
            ])
    return np.array(features)

```

```
scaler = MinMaxScaler(feature_range=(0, 1))
```

```
def train_lstm_model(timeseries, time_step=10, epochs=100):
```

```
    timeseries = timeseries.reshape(-1, 1)
```

```
    timeseries = scaler.fit_transform(timeseries)
```

```
    # Додавання додаткових характеристик
```

```
    timeseries = add_features(timeseries)
```

```
    train_size = int(len(timeseries) * 0.8)
```

```
    train_data = timeseries[0:train_size]
```

```
    test_data = timeseries[train_size:len(timeseries)]
```

```
    X_train, y_train = create_dataset(train_data, time_step)
```

```
    X_test, y_test = create_dataset(test_data, time_step)
```

```
    # Перевірка форми даних
```

```
    print(f"Shape of X_train: {X_train.shape}")
```

```
    print(f"Shape of y_train: {y_train.shape}")
```

```
    print(f"Shape of X_test: {X_test.shape}")
```

```
    print(f"Shape of y_test: {y_test.shape}")
```

```
    model = Sequential()
```

```
    model.add(LSTM(350, return_sequences=True, input_shape=(time_step, X_train.shape[2])))
```

```
    model.add(Dropout(0.25))
```

```
    model.add(LSTM(250, return_sequences=True))
```

```

model.add(Dropout(0.25))

model.add(LSTM(200))

model.add(Dropout(0.25))

model.add(Dense(1))

model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001),
loss='mean_squared_error')


early_stop = EarlyStopping(monitor='val_loss', patience=7)


model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epochs, batch_size=512,
verbose=1, callbacks=[early_stop], shuffle=False)


train_predict = model.predict(X_train)

test_predict = model.predict(X_test)


# Перетворюємо прогнозовані дані назад до початкового масштабу

train_predict = scaler.inverse_transform(train_predict)

y_train = scaler.inverse_transform(y_train.reshape(-1, 1))

test_predict = scaler.inverse_transform(test_predict)

y_test = scaler.inverse_transform(y_test.reshape(-1, 1))


# Повертаємо об'єднані прогнозовані та початкові значення

original_timeseries = scaler.inverse_transform(timeseries[:, 0].reshape(-1, 1)).reshape(-1)

predictions = np.concatenate((train_predict.flatten(), test_predict.flatten()))

```

```

# Обрізаємо оригінальний часовий ряд до розміру прогнозів

original_timeseries = original_timeseries[time_step + 1:time_step + 1 + len(predictions)]

return original_timeseries, predictions

def anomalydetection(original_timeseries, predictions):

    residuals = np.abs(original_timeseries - predictions)

    threshold = np.percentile(residuals, 97)

    anomalies = residuals > threshold

    return anomalies, residuals

def get_anomaly(timeseries, name):

    original_timeseries, predictions = train_lstm_model(timeseries)

    anomaly, variances = anomalydetection(original_timeseries, predictions)

    timerange = list(range(len(original_timeseries)))

    # Create Plotly figure

    fig = go.Figure()

    # Add the main timeseries

    fig.add_trace(go.Scatter(x=timerange, y=original_timeseries, mode='lines',
line=go.scatter.Line(color="dimgray") , name='Data'))

    # Add the anomalies

    fig.add_trace(go.Scatter(x=[i for i in timerange if anomaly[i]], y=[original_timeseries[i] for i in
timerange if anomaly[i]], mode='markers', marker_color="darkred", name='Anomaly'))

    # Update the layout

    fig.update_layout(showlegend=True, title=name)

    fig.show()

```

```

if __name__ == '__main__':

    timeseries = []

    data = pd.read_csv('data/art_daily_perfect_square_wave.csv')

    timeseries.append(data['value'].values)

    data = pd.read_csv('data/art_daily_flatmiddle.csv')

    timeseries.append(data['value'].values)

    data = pd.read_csv('data/art_daily_jumpsdown.csv')

    timeseries.append(data['value'].values)

    data = pd.read_csv('data/speed_6005.csv')

    timeseries.append(data['value'].values)

    data = pd.read_csv('data/rds_cpu_utilization_cc0c53.csv')

    timeseries.append(data['value'].values)

    data = pd.read_csv('data/GSPC.csv')

    timeseries.append(data['Close'].values)

    data = pd.read_csv('data/BTC-USD.csv')

    timeseries.append(data['Close'].values)

    data = pd.read_csv('data/remaining_behavior.csv')

    timeseries.append(data['sequence_length(count)'].values)


    data = pd.read_csv('data/remaining_behavior.csv')

    timeseries.append(data['num_sessions'].values)

    data = pd.read_csv('data/NAB/realKnownCause/ec2_request_latency_system_failure.csv')

    timeseries.append(data['value'].values)

    data = pd.read_csv('data/NAB/realAWScloudwatch/ec2_network_in_5abac7.csv')

    timeseries.append(data['value'].values)

```

```
get_anomaly(timeseries[0], 'NAB_art_daily_perfect_square_wave')  
get_anomaly(timeseries[1], 'NAB_art_daily_flatmiddle')  
get_anomaly(timeseries[2], 'NAB_art_daily_jumpsdown')  
get_anomaly(timeseries[3], 'NAB_speed_6005')  
get_anomaly(timeseries[4], 'rds_cpu_utilization_cc0c53')  
get_anomaly(timeseries[7][2000:30000], 'remaining_behavior sequence_length(count)')  
get_anomaly(timeseries[8][2000:30000], 'remaining_behavior num_sessions')  
get_anomaly(timeseries[9], 'ec2_request_latency_system_failure')  
get_anomaly(timeseries[10], 'ec2_network_in_5abac7')
```