НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

На правах рукопису
УДК 004.852

До захисту допущено
В.о. зав. кафедри ШІ
_____О.І. Чумаченко
«___»_____2022 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 122 «Комп'ютерні науки»

на тему: «Система виявлення аномалій методами інтелектуального аналізу даних»

Виконав:
студент ІІ курсу, групи КІ-11мп
Хомич Олександр Ростиславович _____

Керівник:
д.т.н., проф. Недашківська Надія Іванівна _____

Рецензент:
доцент кафедри СП НН ІПСА
КПІ ім. Ігоря Сікорського, к.т.н., Булах Б. В. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ
2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

Рівень вищої освіти – другий (магістерський)

Спеціальність – 122 «Комп'ютерні науки »

ЗАТВЕРДЖУЮ
В.о. зав. кафедри
_____ О.І. Чумаченко
«___» _____ 2022 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Хомичу Олександру Ростиславовичу

1. Тема дисертації: «Система виявлення аномалій методами інтелектуального аналізу даних», науковий керівник роботи Недашківська Надія Іванівна, д.т.н., проф. кафедри ММСА затверджено наказом по університету від «03» листопада 2022 р. № 4046-с.
2. Термін подання студентом дисертації 15.12.2022
3. Об'єкт дослідження: задача виявлення аномалій на прикладі часових рядів.
4. Предмет дослідження: нейронні мережі з комірками LSTM, графові нейроні мережі, нейронні мережі з кодувальником-декодувальником, ізоляційний ліс та гібридні підходи.
5. Перелік завдань, які потрібно зробити:
   1) здійснити огляд технічної літератури за темою роботи;
   2) дослідити актуальність обраної теми;
   3) ознайомитись з існуючими методами та моделями виявлення аномалій даних;

4) здійснити порівняльний аналіз наявних методів, виявити їх переваги та недоліки;

5) розробити та реалізувати систему, що використовує апарат нейрониих мереж, та вирішує задачу виявлення аномалій у часових рядах;

6) провести експеримент, що засвідчує працеспроможність запропонованої моделі (методу), виконати аналіз результатів;

7) провести аналіз ринкових можливотсей запуску стартап проекту;

8) розробити концептуальні висновки;

9) підготувати ілюстративний матеріал;

10) оформити пояснювальну записку.

6. Перелік ілюстративного матеріалу.

7. Дата видачі завдання: 6 вересня 2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів роботи	Примітка
1.	Вивчення літератури за темою роботи.	06.09.2022 – 19.09.2022	Виконано
2.	Підготовка першого розділу.	20.09.2022 – 26.09.2022	Виконано
3.	Підготовка другого розділу.	27.09.2022 – 03.10.2022	Виконано
4.	Розробка програмного продукту.	04.10.2022 – 17.10.2022	Виконано
5.	Підготовка третього розділу	18.10.2022 – 24.10.2022	Виконано
6.	Підготовка частини стартап-проєкту	25.10.2022 – 31.10.2022	Виконано
9.	Концептуальні висновки. Перспективи розвитку отриманих рішень	01.11.2022 – 14.11.2022	Виконано
10.	Оформлення пояснювальної записки	15.11.2022 – 26.11.2022	Виконано

Студент Олександр ХОМИЧ

Керівник Надія НЕДАШКІВСЬКА

РЕФЕРАТ

Магістерська дисертація: 91 с., 24 табл., 18 рис., 24 джерел, 1 додаток.

ЧАСОВІ РЯДИ, ВИЯВЛЕННЯ АНОМАЛІЙ, ANOMALY DETECTION, LSTM, АВТОКОДУВАЛЬНИКИ, ІЗОЛЯЦІЙНИЙ ЛІС, ІНДЕКС СОНЯЧНОЇ АКТИВНОСТІ.

Об’єктом дослідження є задача виявлення аномалій на прикладі часових рядів.

Предмет дослідження – нейронні мережі з комірками LSTM, графові нейроні мережі, нейронні мережі з кодувальником-декодувальником, ізоляційний ліс та гібридні підходи.

Мета дослідження полягає у аналізі часового ряду, виборі моделей для виявлення аномалій, реалізації методів виявлення аномалій на основі обраних моделей штучного інтелекту та виборі найкращого з них.

Як результат дослідження було запропоновано та розроблено метод виявлення аномалій, що використовує механізми штучного інтелекту, як нейронні мережі з комірками LSTM та ізоляційний ліс.

Проведено порівняння побудованих моделей та вибрано найкращу за метриками оцінки якості. Результат даної роботи можна застосувати для вирішення подібних задач виявлення аномалій часових рядів.

ABSTRACT

Master’s thesis: 91 p., 24 tab., 18 fig., 24 sources, 1 appendix.

TIME SERIES, ANOMALY DETECTION, LSTM, AUTOENCODERS, ISOLATION FOREST, SOLAR ACTIVITY INDEX.

The object of the research is the task of detecting anomalies using the example of time series.

The subject of research is neural networks with LSTM cells, graph neural networks, neural networks with encoder-decoder, isolation forest and hybrid approaches.

The purpose of the study is to analyze the time series, select models for anomaly detection, implement anomaly detection methods based on selected artificial intelligence models, and select the best of them.

As a result of the research, an anomaly detection method using artificial intelligence mechanisms such as neural networks with LSTM cells and isolation forest were proposed and developed.

The built models were compared and the best one according to quality assessment metrics was selected. The result of this work can be applied to solve similar problems of detecting time series anomalies.

ЗМІСТ

ВСТУП 8
РОЗДІЛ 1 ОГЛЯД МЕТОДІВ ВИЯВЛЕННЯ АНОМАЛІЙ ДЛЯ ТЕХНІЧНИХ СИСТЕМ НА ОСНОВІ МЕРЕЖ LSTM 10
1.1. Класифікація аномалій 11
1.2. Розпізнавання аномалій за допомогою LSTM 14
1.2.1. Підходи, засновані на LSTM 16
1.2.2. Підходи, засновані на кодувальниках-декодувальниках 17
1.2.3. Гібридні підходи 20
1.3. Останні тенденції виявлення аномалій на основі глибокого навчання 23
1.3.1. Підходи на основі графів 23
1.3.2. Підходи до трансферного навчання 26
1.4. Аналіз існуючих методів виявлення аномалій 30
1.5. Постановка задачі 31
1.6. Висновки 32
РОЗДІЛ 2 РОЗРОБКА МОДЕЛІ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ 33
1.1. Формулювання задачі виявлення аномалій для часового ряду 33
2.2. Підходи до виявлення аномалій 35
2.2.1. Методи, засновані на прогнозуванні 36
2.2.2. Методи, що засновані на реконструкції 37
2.2.3 Методи, що засновані на кодуванні 37
2.2.4. Методи, засновані на особливостях розподілу 38
2.2.5. Методи дерева ізоляції 40
2.3. Детальний опис обраних методів виявлення аномалій 41

2.3.1. Ізоляційний ліс 41
2.4. Програмні засоби системи виявлення аномалій 46
2.4.1. Scikit-learn 46
2.4.2. TensorFlow 47
2.4.3. Keras 48
2.5. Висновки 49
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ТА АНАЛІЗ РЕЗУЛЬТАТІВ 50
3.1. Вхідні дані та їх обробка 50
3.2. Запропонований алгоритм пошуку аномалій 50
3.3. Застосування ізоляційного лісу 54
3.4. Застосування LSTM з автокодувальником 56
3.5. Висновки 58
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ 60
4.1. Опис ідеї проекту 60
4.2. Технологічний аудит ідеї проекту 62
4.3. Аналіз ринкової стратегії проекту 70
4.4. Розроблення маркетингової програми стартап-проекту 74
4.5. Висновки 78
ВИСНОВКИ 79
ПЕРЕЛІК ПОСИЛАНЬ 80
ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО КОДУ 83

ВСТУП

Аномалії представляють відхилення від передбачуваної роботи системи і можуть призвести до зниження ефективності, а також часткового або повного відмови системи. Оскільки причини аномалій часто невідомі через складну динаміку системи, необхідно ефективне виявлення аномалій. Звичайні підходи до виявлення покладаються на статистичні та інваріантні в часі методи, які не враховують складну та динамічну природу аномалій. Завдяки досягненням у галузі штучного інтелекту та зростаючому значенню для виявлення та запобігання аномаліям у різних областях, підходи штучних нейронних мереж дозволяють виявляти складніші типи аномалій, враховуючи тимчасові та контекстуальні характеристики.

У цій роботі проведено дослідження щодо виявлення найсучасніших методів виявлення аномалій за допомогою LSTM та ізоляційного лісу. Досліджені підходи оцінюються на основі сценарію застосування, даних і типів аномалій, а також інших показників. Щоб підкреслити потенціал майбутніх методів виявлення аномалій, в дослідження також включено підходи на основі графів і трансферне навчання, що дає змогу аналізувати різнорідні дані, а також компенсувати їх недоліки і покращувати обробку динамічних процесів.

В даній роботі пропонується розглянути задачу виявлення аномалій у часових рядах як об’єкт дослідження з метою аналізу даних, підбору моделей для виявлення аномалій, реалізації методів на основі механізмів штучного інтелекту та вибору найкращого на основі метрик якості.

Предметом дослідження є нейронні мережі з комірками LSTM, графові нейроні мережі, нейронні мережі з кодувальником-декодувальником, ізоляційний ліс та гібридні підходи.

Актуальність даного дослідження полягає в тому, що виявлення аномалій у часових рядах є комплексною задачею через його чутливість до

вхідних даних та різного роду економічних, фізичних та біологічних факторів, тому використання нових методів є необхідністю задля кращої результативності для різних вхідних даних.

Наукова новизна роботи полягає в розробці методу виявлення аномалій часових рядів на прикладі даних, що ілюструють роботу сонячних батарей, шляхом попередньої обробки за допомогою нормалізації та використання нейронної мережі на основі LSTM з кодувальником-декодувальником, що дозволяє якісно підвищити рівень виявлення аномалій у порівнянні зі стандартними методами.

РОЗДІЛ 1 ОГЛЯД МЕТОДІВ ВИЯВЛЕННЯ АНОМАЛІЙ ДЛЯ ТЕХНІЧНИХ СИСТЕМ НА ОСНОВІ МЕРЕЖ LSTM

Аномалії створюють проблему в різних областях застосування, таких як виробництво, медичні або комунікаційні системи. Вони часто призводять до зниження продуктивності системи і можуть викликати нестабільність і збої. Часто причинами аномалій є невідомі ефекти в складних системах. Отже, здатність розуміти та виявляти ці основні ефекти за допомогою даних є ключем до забезпечення бажаного результату складних технічних систем. Завдяки прогресу досліджень у сфері машинного навчання останніми роками було запропоновано широкий спектр нових підходів для виявлення аномалій. Різні архітектури глибоких нейронних мереж, зокрема архітектури на основі довготривалої короткострокової пам'яті (LSTM), були розроблені і виявилися здатними вирішувати різноманітні складні задачі розпізнавання [1]

Існуючі високопрофесійні найсучасніші дослідження методів виявлення аномалій лише незначною мірою розглядають підходи, засновані на нейронних мережах. Сучасні дослідницькі розробки щодо глибоких нейронних мереж та архітектур LSTM для виявлення аномалій часто не враховуються. В опитуваннях в основному розрізняють статистичний, класифікаційний, кластеризаційний та інформаційно-теоретичний підходи. Таким чином, основну частину досліджень становлять методи, засновані на аналізі головних компонент (PCA), машині опорних векторів (SVM), алгоритмі k найближчих сусідів (k-NN) або різних типах кореляційного аналізу. Спільним для всіх підходів є їх мета виявлення аномалій на основі статичних та інваріантних у часі моделей [2].

Для виявлення динамічних і змінних у часі аномалій використовуються додаткові методи, такі як ковзні вікна, які поєднуються з вищезгаданими підходами. Як наслідок, підходи не включають моделі для адекватного охоплення змінної в часі динаміки системи і, отже, не можуть характеризувати

аномальні контексти. Для вирішення проблеми виявлення складних контекстних аномалій з динамічними характеристиками та характеристиками, що змінюються в часі, з’явилися нові перспективні архітектури рекурентних нейронних мереж (RNN). Останніми роками були проведені дослідження щодо таких підходів глибокого навчання для виявлення аномалій, переваги та недоліки, а також описані обчислювальна складність (напів-), контрольованих, неконтрольованих та гібридних підходів глибокого навчання [3]. Однак класифікація різних підходів LSTM відсутня, і детальне дослідження архітектури моделей LSTM, описів сценаріїв або механізмів виявлення не є частиною роботи. Дослідження класифікує існуючі підходи виявлення з точки зору характеристик часових рядів, а саме підходи для одно- та багатовимірних даних часових рядів [4]. Підходи LSTM згадуються в усіх існуючих дослідженнях глибокого навчання, але вони не є ані центральними, ані додатково класифікованими та проаналізованими щодо архітектури та механізму виявлення. Однак через недавню появу різних підходів LSTM, які широко використовуються для різних цілей виявлення аномалій, ця стаття має на меті представити детальний огляд виявлення аномалій для технічних систем з чітким акцентом на таких підходах LSTM.

1.1. Класифікація аномалій

Аномалії зустрічаються в різних областях і тому підлягають інтенсивному дослідженню в широкому діапазоні різних областей застосування, таких як мережева безпека, Інтернет речей або виробничі системи [1, 3, 4, 5]. Суттєвою спільною основою для всіх сфер є розуміння аномалії як відхилення від правила або порушення, яке не розглядається як частина нормальної поведінки системи [2]. Аномальна динаміка здебільшого невідома і виникає випадково, призводить до нестабільності і, отже, є причиною підвищення неефективності та системних помилок.

Таксономія аномалій застосовується до всіх досліджуваних областей застосування: її можна охарактеризувати на основі різних аспектів, таких як точка фокусування (наприклад, певний привод виробничої машини), вимірність, (не-) лінійність та тимчасова поведінка. Залежно від застосування можливі різні точки фокусування. Це може означати, наприклад, безпосередній вплив на динаміку всієї системи або перешкоди на рівні (окремих) датчиків і виконавчих механізмів. Аномалії можна або виміряти безпосередньо, або їх потрібно спостерігати за допомогою якоїсь непрямої оцінки стану. Крім того, вони можуть мати лінійні або нелінійні характеристики. В рамках дослідженої літератури динаміка аномальної системи, яка виявляється та моделюється за допомогою мереж LSTM, рідко є інваріантною у часі через властивості осередку LSTM. Стосовно динаміки, що змінюється в часі, її можна розрізнити на стаціонарну та нестаціонарну або короткострокову та довготривалу поведінку [5].

Для виявлення подібних нерівномірностей спочатку основну увагу приділяли стохастичним методам виявлення **викидів**, або так званих **точкових аномалій**. Таким чином, щільності ймовірності були розраховані для цільових параметрів, а визначені процентилі були оголошені як викиди і, таким чином, як аномалії певного ступеня [6].

Огляд стохастичних методів виявлення аномалій [3], представляє інші типи аномалій, які зустрічаються в даних часових рядів технічних систем. Як наслідок, колективні та контекстуальні порушення визначаються як додаткові типи аномалій на додаток до статистично описаних викидів. Рис. 1.1 ілюструє три типи аномалій, які включені в одновимірний часовий ряд [4].

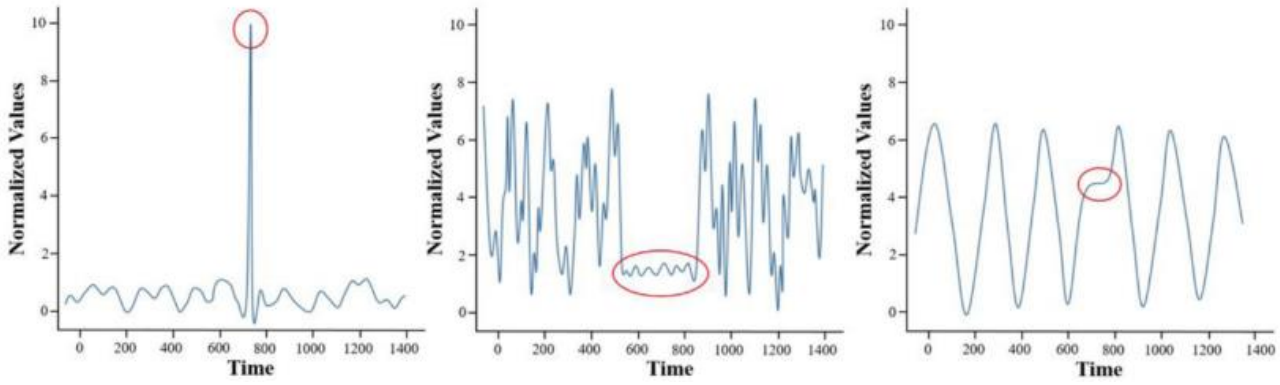

Рисунок 1.1 – Точкова аномалія (ліворуч), колективна аномалія (середина) і контекстна аномалія (праворуч) [3]

Колективні аномалії можна охарактеризувати як групу векторів даних, де кожен окремий вектор даних є допустимим, але склад групи вказує на нерегулярність. Отже, внутрішня структура послідовності даних визначає ступінь відхилення. Визначено колективні аномалії для випадку багатовимірних часових рядів. Для їх виявлення пропонується багатовимірна метрика на основі ступеня відхилення всіх одиничних векторів даних групи. Їхня значимість моделюється змінним у часі способом з експоненціально спадною функцією. Якщо сукупна метрика на основі всіх одиничних відхилень перевищує динамічний поріг, виявляється колективна аномалія [6].

Контекстуальні аномалії можуть бути охарактеризовані як окремі вектори даних (не будучи точковими аномаліями) або групи векторів даних (не будучи колективними аномаліями), які є допустимими, але вказують на нерегулярність у межах конкретних навколишніх векторів даних або груп векторів даних, які називаються контекстом.

Застосування контекстного виявлення аномалії може сприяти кращому виявленню аномалії, підкреслюючи або відкидаючи певну аномалію, тим самим зменшуючи кількість хибних результатів або знаходячи основну причину певної аномалії за допомогою метаданих, які також називають атрибуцією аномалії.

Тому, на відміну від колективних аномалій, які описуються своєю внутрішньою структурою або змістом, виявлення контекстуальних аномалій сильно залежить від короткострокових і довгострокових характеристик навколишніх зовнішніх структур даних. Що стосується багатовимірних часових рядів, контекст вектора даних або групи векторів даних можна інтерпретувати як об'єднання всіх навколишніх векторів даних, які лежать у визначеному часовому горизонті. Показники, засновані на відстані, в першу чергу використовуються для характеристики контекстуальної аномалії. Це може бути реалізовано, наприклад, на основі техніки ковзного вікна, коли метрика відстані до попереднього вікна перераховується з кожною новою вибіркою даних. Перевищення динамічних порогів свідчить про контекстуальні аномалії [7]. В іншому випадку база даних відомих контекстів додатково розширюється. Залежно від домену програми контекст може мати різні визначення, області та розміри. Ними можуть бути будь-які часові, просторові чи просторово-часові ознаки навколишнього середовища. Для області дискретного виробництва він може охоплювати дані про виробничий процес і пов'язані з ним параметри процесу.

1.2. Розпізнавання аномалій за допомогою LSTM

Цей розділ починається з короткого введення в LSTM, щоб забезпечити базове розуміння мережевих архітектур та механізмів виявлення мереж LSTM, які обговорюються в наступних розділах. LSTM була розроблена [2] для вирішення проблеми із зникаючим градієнтом, яка виникає зі звичайними RNN і призводить до неможливості вивчення довгострокових залежностей. Проблема із зникаючим градієнтом описує обставини, коли частини ваг у RNN, як правило, припиняють змінюватися під час процесу навчання. Як наслідок, встановлення пріоритетів поточної інформації може призвести до нехтування минулими подіями. Таким чином, відношенням, які повторюються

протягом тривалого періоду часу (довгострокові залежності), неможливо адекватно навчитися. LSTM сконструйований для управління всім потоком інформації всередині нейронів. Для цієї мети введено механізм забування, який керує процесом додавання та видалення інформації зі стану клітинки, що ітеративно поширюється. Таким чином, процес забування можна контролювати, і реалізується певна поведінка пам’яті для моделювання як короткострокових, так і довгострокових залежностей. Архітектура LSTM зображена на рис. 1.2.

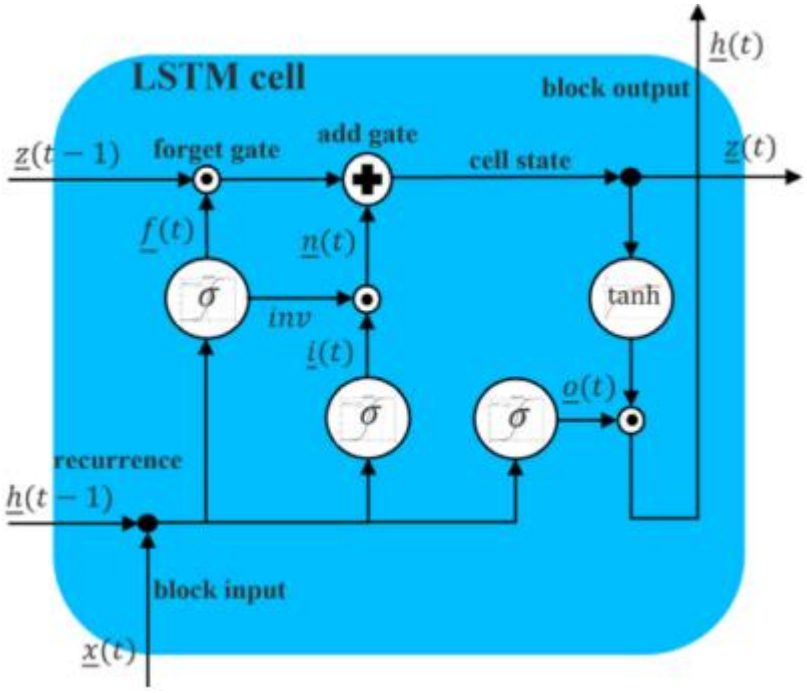

Рис. 1.2 – LSTM зі зворотним відображенням між воротами забування та додавання [2]

На відміну від звичайних RNN, вихід нейрона $h(t)$ не будується безпосередньо з входами $x(t)$ і попередніми виходами $h(t-1)$, а на основі стану клітини $z(t)$. З іншого боку, стан комірки визначається механізмом забування LSTM. Вектор $f(t)$ як вихід елемента забування, а вектор $n(t)$ як вихід елемента додавання ітеративно адаптують стан комірки для керування поведінкою пам'яті. У зображеній архітектурі LSTM для обмеження ємності пам’яті до певної міри використовується зворотний зв’язок між цими двома

вентилями. Отже, з кожною ітерацією інформація додається та видаляється зі стану комірки. Ця процедура зумовлена обставиною, що жодна пам’ять не є нескінченною і що людська пам’ять як взірець також має обмежені можливості. Вихідний вентиль додатково визначає оновлений стан комірки для обчислення бажаного результату. Таким чином, останні вхідні дані не обов’язково домінують у генерації вихідних сигналів, оскільки стан комірки інкапсулює зменшене та зважене представлення історичної вхідної інформації. Ця інформація відображається на виході. Отже, вплив важливих минулих подій включено в проекцію виходу, і недосконаліть поточних вхідних даних з низькою щільністю інформації можлива. Самі ворота створюються на основі поточних входів і минулих виходів.

LSTM можна включити в широкий спектр архітектур нейронних мереж. Огляд досліджуваних підходів наведено в таблиці 1. У подальшому курсі цієї статті ми зосередимося на дослідженні мережевих архітектур, які базуються на LSTM і які розроблені для вирішення визначених завдань виявлення аномалій. Наступний розділ розповість про підходи на основі кодувальника-декодувальника, гібридного, графового та трансферного навчання.

1.2.1. Підходи, засновані на LSTM

Мережі LSTM призначені для виявлення контекстуальних аномалій завдяки їх здатності вивчати тимчасові відносини та фіксувати їх у представленні стану низької розмірності. Ці відносини можуть стосуватися стаціонарної та нестаціонарної динаміки, а також короткострокових і довгострокових залежностей. Мережі LSTM особливо підходять для моделювання багатовимірних часових рядів і систем із зміною часу [8]. Таким чином, відхилення реальних вихідних даних системи від очікуваних вихідних даних, передбачених мережею, можна використовувати для цілей виявлення аномалій. Доведено, що підходи, засновані на LSTM, демонструють чудові

можливості виявлення аномалій. У статті представлена архітектура LSTM для виявлення аномалій у даних часових рядів [9]. На відміну від надійного або шумопоглинаючого LSTM AE, в якості вхідних даних не використовуються елементи зі зменшеним розміром. Виявлення здійснюється шляхом оцінки відхилення прогнозованих результатів на основі аналізу дисперсій. Глибока мережа LSTM використовується як предиктор поведінки звичайного зв'язку шини в транспортних засобах. Значні відхилення виявляються за допомогою динамічного порогу для виявлення аномальної поведінки зв'язку, спричиненої кібератаками. Тут мережа LSTM прогнозує регулярну динаміку системи, а SVM використовується як класифікатор аномалій, щоб реалізувати адаптований і самонавчаючийся механізм виявлення. Таким чином, тимчасові аномалії в багатоваріантних даних можна виявити напів- або без нагляду. Підхід до виявлення колективних аномалій за допомогою мереж LSTM. Новинка полягає в оцінці множинних помилок прогнозування на один крок вперед на відміну від оцінки кожного кроку часу окремо. Мережі LSTM підвищують точність виявлення шляхом прогнозного моделювання стаціонарних і нестаціонарних залежностей від часу. Таким чином реалізується ефективне виявлення структур тимчасових аномалій – підхід виявлення в реальному часі реалізований на основі двох мереж LSTM [7]. Один для моделювання короткострокових характеристик і здатний виявляти окремі наступні аномальні точки даних у часових рядах, а інший для контролю виявлення на основі довгострокових порогових значень.

1.2.2. Підходи, засновані на кодувальниках-декодувальниках

У більшості випадків використання, досліджених у різних сферах застосування, таких як виробництво або комунікація, отримані дані не мають жодних міток, а також доступні інформаційні моделі, що описують контекст даних. Внаслідок цього необхідні методи навчання без нагляду, щоб

реалізувати непряме маркування даних і таким чином виявити аномалії. Зокрема, нові підходи нейронної мережі з архітектурою кодувальник-декодувальник, які були розроблені в останні роки, демонструють чудову застосовність для завдань виявлення без нагляду. Мережі АЕ є прикладом, коли частина кодувальника прагне вивчити представлення вхідних даних у нижньому вимірі, а частина декодувальника націлена на реконструкцію цих стислих функцій [6]. Отже, АЕ навчається з даними, які представляють звичайну динаміку системи, і вчиться стискати та відновлювати ці дані. Навпаки, обробка аномальних даних з навченим АЕ призводить до помилки реконструкції. Динаміка помилок може бути використана для створення механізму виявлення аномалій для прикладу надійного глибокого АЕ. У цьому випадку шари аналізу головних компонентів і регуляризації були інтегровані з АЕ, щоб зменшити шум вхідних даних і реалізувати надійну поведінку виявлення аномалій. Для ефективного вилучення аномальної динаміки метрика реконструкції складається з двох частин. Подібний підхід застосовується і до скорочувального LSTM AE [9].

Одна частина метрики реконструкції оцінює здатність відокремлювати аномалії, такі як викиди, від звичайних даних, а інша оцінює здатність виявляти зв’язки всередині даних. Крім того, шумопоглинаючий АЕ LSTM, яка має на меті оптимізувати точність передбачення та виявлення шляхом виділення основних і непошкоджених зв’язків у нестійких до відхилень даних. Композиція LSTM і АЕ дозволяє вивчати короткострокові та довгострокові залежності з точки зору тимчасових характеристик нижнього виміру і, таким чином, забезпечує знайдену основу для виявлення складних аномалій у часі. На основі описаної роботи були проведені подальші розширення та дослідження щодо виявлення аномалій за допомогою LSTM AE. Для цілей виявлення аномалій введено варіативний АЕ LSTM. Схема використовує механізми імовірнісної проекції в частинах кодувальника та декодувальника. Таким чином, вхідні послідовності перетворюються в розподіли ознак нижчої

розмірності та реконструюються на основі визначеного значення ознак. Цей підхід застосовує виявлення аномалій на основі логарифмічної правдоподібності шляхом обчислення оцінки логарифмічної правдоподібності для реальних і відновлених результатів. Крім того, навчений кодувальник можна використовувати окремо для ймовірнісного зменшення розмірності. Також LSTM AE використовується для моделювання нормальної поведінки системи дискретних виробничих процесів [4]. Навчена декодувальна частина мережі використовується як модель зворотного процесу для виявлення аномалій шляхом порівняння реальних і відновлених змінних активації на основі спостерігача за порушенням. Таким чином, можна виявити ефекти різних характеристик, таких як стаціонарні та нестаціонарні аномалії, які порушують роботу систем управління.

На додаток до AE, мережі LSTM від послідовності до послідовності (sequence-to-sequence – Seq2Seq) мають структуру кодувальник-декодувальник і використовувалися для завдань виявлення аномалій [10]. Цей тип мережі використовується для виявлення аномалій на основі станів осередку, що поширюються через мережу. Невідомі стани комірок і вектори копіювання з великим відхиленням між шаром кодера і декодера вважаються аномаліями. Далі вони оцінюються за допомогою алгоритму кластеризації постобробки. Інший підхід Seq2Seq, де різні типи аномалій можуть бути виявлені шляхом моделювання та прогнозування різноманітних атрибутів. Цей підхід перевершив накопичувальний LSTM у рамках емпіричного дослідження набору контрольних даних. Підхід до підвищення можливостей узагальнення та екстраполяції мереж Seq2Seq LSTM для оптимізованого виявлення аномалій представлений. Запропонована архітектура проілюстрована на рис. 1.3. Вона складається з розріджених кодувальників та декодувальників, що містять з’єднання, які залежать від щільності інформації у вхідних послідовностях і забезпечують більш гнучке поширення стану мережі. Кілька кодувальників використовують один і той самий шар

копіювання для поширення зменшеного вектора ознак до декодувальника. Ця процедура запобігає перенавчанню і призводить до кращих характеристик узагальнення [11]. Функція вартості для виявлення аномалій мінімізує сукупність усіх помилок реконструкції та містить штрафний термін для керування потоком інформації в спільному копіювальному шарі.

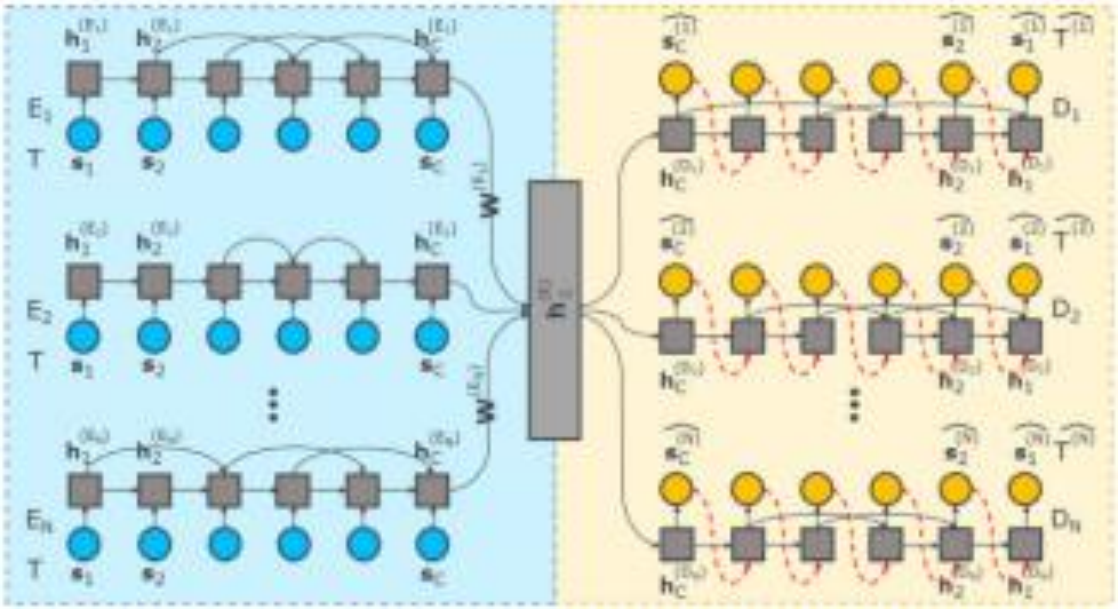

Рисунок 1.3 – Ансамбль LSTM AE з пропускними з’єднаннями [11]

1.2.3. Гібридні підходи

Підходи, описані в цьому розділі, мають спільне використання композиції з двох нейронних мереж в рамках комплексної архітектури виявлення аномалій. Таким чином, ці гібридні підходи складаються з LSTM та другої мережі. Перш за все, один справляється із завданням прогнозування динаміки процесу, а другий виявляє відхилення від реального результату процесу та виявляє аномальну динаміку. Мета гібридних підходів полягає в тому, щоб скористатися перевагами обох типів мереж, одночасно компенсуючи неточності. Виникнення локальних аномалій складається з відхилень щодо одного або кількох екземплярів даних і тимчасово обмежено

фіксованим проміжком часу. Навпаки, глобальні аномалії в першу чергу описують дрейфуючі відхилення від еталону, які явно показують нестаціонарні довгострокові характеристики. Спочатку була представлена нова композиція накопичених AE та LSTM для виявлення аномалій на основі нерозмічених даних та невідомої динаміки системи [7]. Частина кодувальника сконструйована для обробки кількох послідовностей на кожному кроці дискретного часу і може обробляти вихідні дані або зменшені функції введення. Він витягує відносини, максимізуючи ентропію в стиснутій інформації. Детектор аномалій реалізований за допомогою другої мережі в рамках складної архітектури. Отже, мережу LSTM навчають ідентифікувати характеристики відхилення у реконструйованому просторі ознак. Подальший приклад гібридного підходу з LSTM AE [12], де він розширений за допомогою алгоритму кластеризації, який характеризує відновлену динаміку системи за допомогою представлення у просторі станів. Отже, аномальна динаміка виявляється у випадку різкого або дрейфуючого переходу станів або у разі створення нових станів. Для подальшої оптимізації співпраці предиктора і детектора використовуються генеративні змагальні мережі на основі LSTM. Архітектура мережі складається з двох взаємодіючих мереж: генератора, який має на меті відтворення даних реальної системи, і дискримінатора, який має на меті відрізнити ці штучно згенеровані дані від реальних даних. Генератор поводиться як декодувальник і реконструює ряди даних в реальному часі, тоді як дискримінатор класифікує реконструкцію як вихідну з нормальних або аномальних вхідних даних. Це залежить від складності відрізнити реконструкцію від реальних даних [8].

Щоб уможливити виявлення багатовимірних аномалій, згорткові нейронні мережі (CNN) і LSTM об’єднані в новітньому методі, якому здатність ефективно стискати дані великої розмірності дозволяє витягувати залежності в кількох вимірах. Таким чином, комбіноване використання CNN і LSTM дозволяє виявляти аномалії в множинних і взаємопов’язаних вимірах, таких як

просторові, тимчасові або інші виміри, що стосуються конкретної програми. Таким чином, можна виявити складні контекстуальні аномальні структури шляхом кореляції різних вимірів, навіть якщо вони не демонструють аномальної поведінки в усіх вимірах. Класифікація проводиться на основі крос-ентропії. Підхід проілюстрований на рис. 1.4. LSTM поєднується з експоненціально зваженим ковзним середнім (EWMA) і технікою динамічного порогового значення [9].

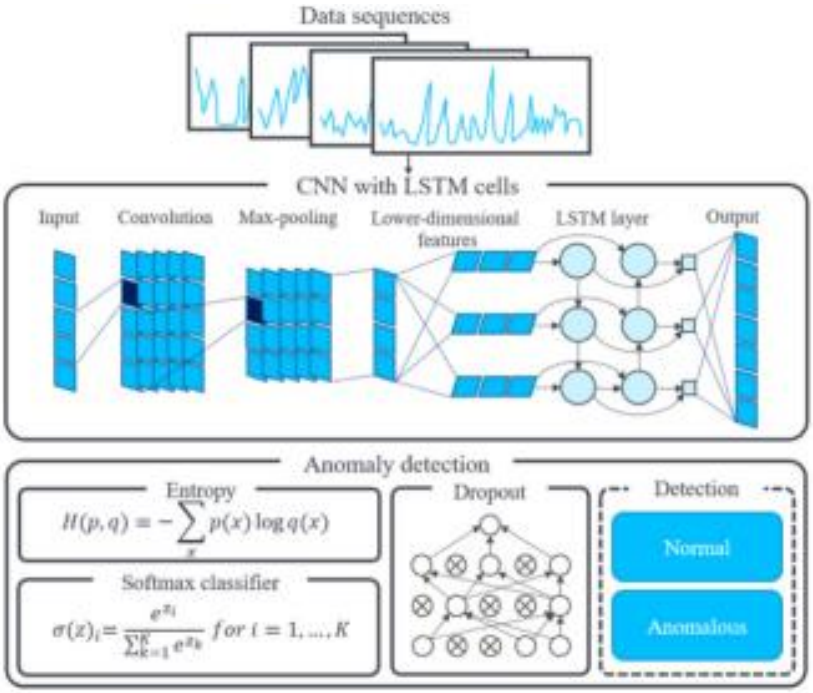

Рис. 1.4 – CNN і LSTM для виявлення аномалій [9]

Проаналізовано тимчасові структури багатовимірних часових рядів для виділення моделей подібних характеристик. Ці закономірності постійно визначаються та оцінюються. EWMA використовується в поєднанні з динамічним порогом для оцінки результатів прогнозування мережі LSTM. Це дозволяє виявляти аномальні структури в даних шляхом дослідження залишків прогнозу за допомогою вищезгаданої методики. Основною перевагою підходу є той факт, що всі контекстуальні аномалії нових часових

послідовностей можуть бути ідентифіковані в рамках одного процесу виявлення, щоб можна було досягти значного підвищення ефективності.

1.3. Останні тенденції виявлення аномалій на основі глибокого навчання

У цьому розділі представлені останні тенденції виявлення аномалій на основі глибокого навчання. Вони спрямовані на розширення універсальності та надійності підходів, описаних у попередньому: підходи на основі графів дозволяють покращити представлення контекстної інформації, тоді як підходи до трансферного навчання фокусуються на кількості даних, необхідних для навчання алгоритмів виявлення аномалій.

1.3.1. Підходи на основі графів

Граф — це мережа вершин і ребер, яка може бути спрямованою або ненаправленою зі зваженими або незваженими ребрами залежно від природи та області даних, що моделюються. Виявлення аномалій у межах графа або використання підходу, заснованого на графі, є подальшими перспективними підходами для виявлення аномалій. Основною перевагою цих підходів є здатність графа моделювати кореляції між точками даних, а не окремо представляти їх. Таким чином, можна виділити та проаналізувати притаманну взаємозалежність між даними, а також з посиланням на інші зовнішні фактори. Іншою перевагою є застосовність графів для виявлення колективних і контекстних аномалій шляхом кластеризації вузлів на основі контекстних атрибутів і виявлення аномальних ребер або вузлів всередині кластерів. Виявлення аномалій на основі графів було присутнє в дослідженнях протягом останніх десятиліть, здебільшого зосередженим на статичному аналізі графів.

Для виявлення аномалій також розглядаються гібридні підходи, які поєднують графи та нові алгоритми машинного та глибокого навчання [13].

По суті, виявлення аномалій у межах графів вимагає двох або трьох кроків залежно від структури вхідних даних. На рис. 1.5 зображено спрощений вигляд процесу. Спочатку або дані вже представлені в графі, або їх потрібно змоделювати в одному з різних гетерогенних баз даних і форматів даних, де схема графа з вузлами та ребрами визначається залежно від варіанту використання. Графові моделі потім зберігаються та керуються в базі даних графів, при цьому вони версійні на основі різних екземплярів часу, щоб уможливити аналіз прогресування аномалії на основі часу. Другий крок зазвичай включає кластеризацію або розбиття графа на підграфи, які можуть базуватися на структурних або часових ознаках, а також на комбінації обох. Цей початковий аналіз графа допомагає визначити структуру мережі, а також слабко або сильно зв’язані вузли та кластери. Крім того, можна визначити й виявити функції контексту та вмісту. Виходячи з цього кроку, третій крок вимагає аналізу на основі часу всередині кластерів для виявлення аномальних вузлів або ребер на основі визначеної моделі графа та контекстних ознак. Щоб позначити ці кластери, ребрам вузлів, тобто зв’язкам, можна ітераційно призначити ваги.

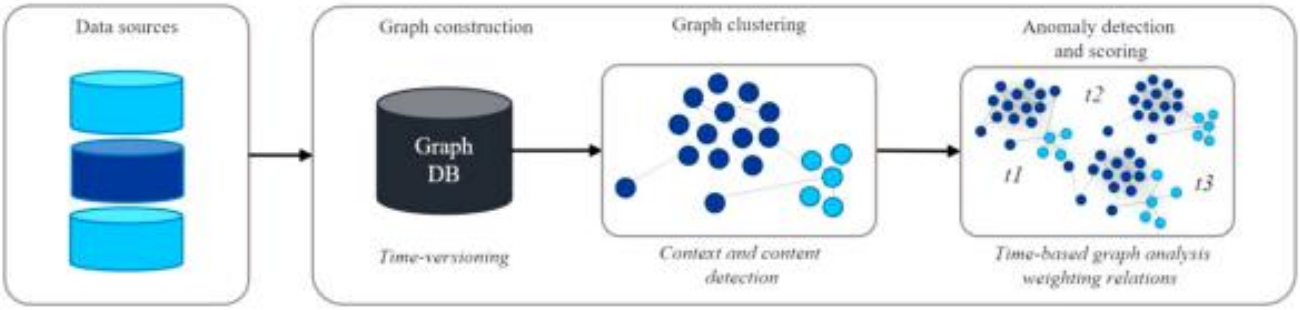

Рисунок 1.5 – Підхід на основі графів для виявлення аномалій як першочерговий крок до підходів на основі LSTM [13]

Результат цього описаного процесу може бути додатково використаний і вдосконалений різними можливими способами. Спочатку виявлені аномальні кластери можуть бути додатково досліджені, застосовуючи вбудовування графів, тобто перетворення структури підграфа у векторне представлення для застосування додаткових алгоритмів машинного навчання. Крім того, отримані функції можна використовувати як вхідні дані для мереж LSTM, що дозволить більш ефективно використовувати їх.

У 2016 році було представлено алгоритм виявлення контекстуальних колективних аномалій. Маючи атрибутований граф як вхідні дані, вузли представляють окремі точки даних, таким чином, поведінкові атрибути та контекстні атрибути ребер. Пропонується ручний вибір об’єктів у межах графа та використовується алгоритм кластеризації Лувена. Усередині кластерів кожному вузлу було присвоєно оцінку аномалій, щоб виявити аномальні вузли.

У більш новому методі контекстні відхилення в даних датчиків виявляються за допомогою підходу на основі графів. Атрибути поведінки задаються значеннями температури та вологості, тоді як контекстний атрибут представлений часом. Із збереженого набору даних сервера створюється граф, що представляє кожну точку даних у вигляді вузла та ребра на основі евклідової відстані та відповідно присвоєної ваги. На основі графа будуються кластери за допомогою ітераційного алгоритму мінімального охоплюючого дерева. Викиди потім виявляються за допомогою ковзного часового вікна, де кластер з найбільшою кількістю даних вважається нормальним, а інші кластери аномальними з вищим призначеним показником викидів. Для цього сценарію не потрібні дані про навчання чи маркування [14].

Аномалії, як правило, мають динамічну природу, і виявлення їх у динамічних графіках є складним завданням, оскільки слід враховувати не лише структурні та змістові, а й часові особливості. Їх можна виявити, фіксуючи довгострокові та короткострокові шаблони вузлів.

В останні роки підходи до виявлення аномалій у динамічних графах розвивалися, використовуючи переваги глибокого навчання. Наприклад, вбудовування графів відображають граф у векторний простір і аналізують вузли на основі їх структурної подібності. Крім того, використання графових згорткових мереж (GCN) дозволяє розширити, витягуючи структурні та змістові особливості вузлів. Використовуючи GCN, аномальні ймовірності вузла можуть бути поширені на сусідні вузли. Однак досі мало уваги приділяється довгостроковому виявленню ознак. Пропонується підхід для виявлення аномальних країв мережі, використовуючи також GRU як варіант LSTM з контекстною моделлю на основі уваги, щоб фіксувати короткострокові та довгострокові подібності вузлів. Таким чином, GCN виводить стан вузла з урахуванням його структурних і змістових особливостей і доповнюється GRU для довгострокового збору інформації [13].

Наостанок можна сказати, що охоплення додаткових доступних метаданих і зовнішніх атрибутів під час виявлення аномалій може привести до кращих результатів виявлення або надати більше розуміння для розуміння причин аномалій. Цей ефект посилюється в поєднанні з методами на основі графів. Якщо дані представлені у вигляді графа, контекстну інформацію можна використовувати для покращення виявлення аномалій. Виявлені та оцінені аномалії потім можуть бути вхідними для LSTM, де високодинамічна поведінка процесу моделюється з невизначеністю.

1.3.2. Підходи до трансферного навчання

Подібно до інших випадків широке використання керованого даними виявлення аномалій у виробництві має подолати серйозну проблему: набори даних, необхідні для навчання таких алгоритмів, повинні бути великими та різноманітними, що ускладнює їх отримання. Щоб подолати цю проблему,

використовується передача знань між кількома навчальними агентами та навчання на незалежних наборах даних.

Область дослідження зосереджена на використанні знань, отриманих під час навчання за попередніми завданнями, для покращення підготовки нового завдання в області трансферного навчання [15]. Тут були визначені різні підходи для вирішення такого роду проблем:

а) перенесення або екземплярів (або їх перетворень) наборів даних старих завдань у новий;

б) (частини) самого алгоритму (або його перетворень).

Хоча перше зазвичай простіше реалізувати, воно не вирішує згадані вище проблеми. Тому далі буде розглядатися лише останній. Він складається з передачі параметрів, (часткового) повторного використання мережі, попередньо навченої для вихідної задачі, та реляційної передачі знань, яка покращує передачу параметрів за допомогою адаптації домену. І те й інше зменшує потребу в навчальних даних щодо цільового завдання (див. рис. 1.6).

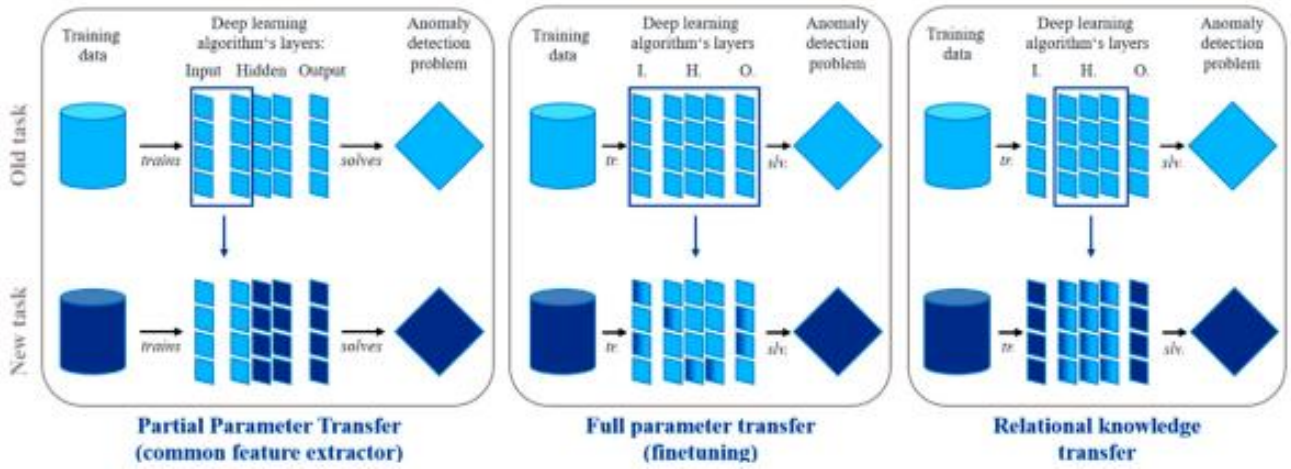

Рисунок 1.6 – Підходи до трансферного навчання: передача параметрів і реляційна передача знань [15]

Незважаючи на те, що практичне застосування методів трансферного навчання все ще знаходиться на ранніх стадіях, можна знайти деякі зразкові

реалізації алгоритмів виявлення аномалій. На жаль, через відсутність контрольних наборів даних їх продуктивність не можна прямо порівняти.

У 2018 році було запропоновано метод навчання з перенесенням, що використовується для покращення виявлення аномалій у споживанні електроенергії на різних машинах для екструзії алюмінію. AE, що знімає шуми, попередньо навчається без нагляду на великому наборі вихідних даних, а потім тонко налаштовується на цільовому наборі даних (рис. 1.6, посередині).

Надалі трансферне навчання використовується для покращення виявлення аномалій у роботі службових ліфтів, враховуючи відмінності у типі та кількості датчиків у різних робочих місцях. Ця реалізація покладається на одновимірні CNN для кожного датчика для вилучення функцій. Ці ознаки потім агрегуються та класифікуються різними типами нейронних мереж, серед яких LSTM, GRU та RNN (див. рис. 1.6, ліворуч). Наступний аналіз методологічно дуже ретельний, враховуючи численні розподіли «нормальних» та «аномальних» вибірок, а також різні алгоритми для цього дуже складного сценарію застосування. Хоча точність, досягнута набором запропонованих алгоритмів, дуже висока, на жаль, на наборі даних не випробовуються жодні конкуруючі підходи.

Також трансферне навчання використовується для покращення виявлення аномалій у послідовностях процесів у різних виробничих модулях. Тут останній рівень кодувальника та декодувальника попередньо підготовленого AE на основі LSTM повторно навчається на відповідних цільових наборах даних. Це дозволяє швидко адаптувати попередньо навчений алгоритм до різних виробничих модулів, для яких доступні лише невеликі, зазвичай недостатні набори даних. Хоча вони показують кращу продуктивність, ніж звичайні алгоритми глибокого навчання, інформативна цінність обмежена через те, що набір даних простої послідовності процесів далекий від реалістичності [15].

Навчання передачі використовується для покращення виявлення викидів у контрольованих мережах (CAN), дозволяючи алгоритму вивчати нові типи викидів при першому зіткненні. Згортковий LSTM (ConvLSTM) використовується для обробки багатоваріантних часових рядів у вигляді двовимірної інформації, тоді як байєсівське одноразове навчання забезпечує функціональні можливості трансферного навчання. Оцінка проводиться на основі даних CAN, зібраних з двох різних типів автомобілів протягом більш ніж 24 годин їзди. Алгоритм перевершує декілька базових моделей, напр. SVM, ізоляційні ліси (IF) або ансамбль RNN (eRNN), на завдання, які не були включені в дані навчання, тобто на нових завданнях. На відомих завданнях він працює трохи гірше, ніж найкращий конкурентний підхід. У порівнянні зі швидкістю даних CAN, алгоритм виявляє викиди в режимі реального часу [16].

Зовсім інший підхід використовується для покращення виявлення аномалій. Так, використовується безперервне навчання, тобто багатозадачне машинне навчання з використанням передачі знань між різними завданнями для підвищення ефективності виконання старих і нових завдань. Випадок використання – це процес формування металу, що включає часті зміни у виготовлених продуктах. Набір LSTM поєднується з розширеною функцією втрат, завдяки чому алгоритм зберігає попередні можливості. Ретельне порівняння різних підходів безперервного навчання та базового традиційного алгоритму показує значні покращення за допомогою так званої еластичної консолідації ваги [17].

У підсумку, було розглянуто лише кілька підходів, які охоплюють лише невелику групу можливих методів і технік. Хоча в цілому цей напрямок є багатообіцяючим, чітких тенденцій поки що не можна помітити, що вимагає додаткових досліджень у цій галузі.

1.4. Аналіз існуючих методів виявлення аномалій

У даному розділі дається огляд мереж LSTM для виявлення аномалій і розділяє існуючі підходи на п’ять категорій. Досліджені звичайні підходи, засновані на нейронних мережах, далі поділяються на звичайні архітектури LSTM, мережі LSTM на основі кодувальника-декодувальника, а також гібридні підходи.

Отже, звичайний LSTM дозволяє точно виявляти колективні та контекстуальні аномалії. У порівнянні з цими чистими підходами LSTM, архітектури, засновані на кодувальнику-декодувальнику, дозволяють LSTM додатково оптимізувати свої можливості виявлення для просторів даних великої розмірності. Різні архітектури, такі як скорочувальний AE та варіаційний AE, були використані для конкретних цілей виявлення. Гібридні підходи насамперед спрямовані на поєднання переваг двох методів в одній архітектурі. Ці підходи часто містять компоненти предиктора і детектора, щоб завдання були точно розділені.

Підходи, засновані на графах, для виявлення аномалій мають перевагу, що дозволяють уніфікувати представлення різнорідних даних і джерел даних. Таким чином можна проаналізувати причину і поширення аномалій, особливо в контексті їх контексту. Таким чином, багато підходів, заснованих на графах, розглядають контекстні аномалії. Як приклад, виявлення аномалій у фізичних процесах може враховувати дані про процес, систему, а також атрибути навколишнього середовища в межах одного графа, щоб зробити висновок про аномалії. Застосування машинного навчання в поєднанні з представленням даних на основі графів і аналітикою графів теоретично було б перспективним підходом для більшої точності виявлення та прогнозування аномалій, особливо в мережевих системах.

У порівнянні з проблемою відсутності мічених даних для машинного навчання, представлення даних у графі та керування вершинами та ребрами на

основі доменної програми може бути складним і трудомістким завданням. Подальшою проблемою є вибір контекстних ознак у межах графа та складність охоплення багатовимірних контекстних функцій у процесі кластеризації та ранжування викидів. Цю проблему називають профілюванням контексту, де багатовимірний контекст не може бути дискретизованим і вимагає багатоваріантного підходу до кластеризації графів.

Трансферне навчання вирішує проблему відсутності достатньо великих і різноманітних наборів даних для навчання алгоритмів глибокого навчання у випадках виявлення аномалій. Послідовне навчання на кількох наборах даних, що представляють різні завдання або різні стани одного завдання, і перенесення знань з одного навчання на інше, допоможе вирішити обидві проблеми. Хоча все ще вони є досить новими, зростає кількість реалізацій, що відображають різні підходи до традиційного навчання на основі нейронних мереж, особливо вони мають широке застосування у промисловості

1.5. Постановка задачі

1. здійснити огляд технічної літератури за темою роботи;
2. дослідити актуальність обраної теми;
3. ознайомитись з існуючими методами та моделями виявлення аномалій даних;
4. здійснити порівняльний аналіз наявних методів, виявити їх переваги та недоліки;
5. розробити алгоритм пошуку аномалій
6. розробити та реалізувати нову систему на основі пропонованого алгоритму, що використовує апарат нейрониих мереж, та вирішує задачу виявлення аномалій у часових рядах;
7. провести експеримент, що засвідчує працеспроможність запропонованої моделі (методу), виконати аналіз результатів;

8. провести аналіз ринкових можливотсей запуску стартап проекту.

1.6. Висновки

Аномалії виникають у широкому діапазоні технічних застосувань і можуть мати значний вплив на продуктивність і стабільність системи, а також на якість її виходу. Їх можна описати різними характеристиками. Мережі LSTM дозволяють особливо виявляти часові характеристики. Отже, в рамках цієї роботи були досліджені різні підходи LSTM для виявлення аномалій у даних часових рядів. Крім того, були досліджені останні досягнення в підходах до виявлення аномалій на основі графів і трансферного навчання, особливо зосереджено на їхньому рівні застосовності до проблем реального життя.

Представлені підходи до виявлення аномалій на основі мереж LSTM, які застосовувалися в різних технічних системах, таких як виробництво або робототехніка. Проведене дослідження показує, що різні архітектури мережі LSTM доступні і здатні точно виявляти різний діапазон складних аномалій, таких як колективні та контекстуальні аномалії.

З представленого дослідження робиться висновок про необхідність подальших досліджень щодо виявлення аномалій не лише в розмежованих системах, а й у мережах взаємодіючих систем. Таким чином, майбутні розробки можуть бути зосереджені на включенні мереж LSTM в підходи на основі графів для оптимізованої характеристики контекстуальних аномалій. Для подальшого підвищення точності виявлення, можна було б більш інтенсивно досліджувати комбінацію мереж LSTM і методів навчання передачі.

РОЗДІЛ 2 РОЗРОБКА МОДЕЛІ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ

Виявлення аномальних підпослідовностей у даних часових рядів є важливим завданням у різних сферах, починаючи від виробничих процесів і фінансових додатків до моніторингу охорони здоров'я. Аномалія може вказувати на важливі події, такі як виробничі збої, вузькі місця доставки, системні дефекти або мерехтіння серця, і тому становить центральний інтерес. Оскільки часові ряди часто є великими та демонструють складні закономірності, спеціалісти з обробки даних розробили різні спеціалізовані алгоритми для автоматичного виявлення таких аномальних моделей. Кількість і різноманітність алгоритмів виявлення аномалій значно зросла в минулому. Оскільки багато з цих рішень були розроблені незалежно та різними дослідницькими спільнотами, немає комплексного дослідження, яке б систематично оцінювало та порівнювало різні підходи. З цієї причини вибір найкращої методики виявлення для певного завдання виявлення аномалії є складним завданням.

1.1. Формулювання задачі виявлення аномалій для часового ряду

Ряд даних — це впорядкована послідовність точок даних. Точки даних описують деякий об'єкт або властивість процесу на основі безперервного вимірювання, наприклад температуру (наприклад, фізика), масу (наприклад, хімія), кут (наприклад, астрономія), положення (наприклад, геологія) або швидкість (наприклад, машинобудування). Якщо порядок заснований на часі, послідовність зазвичай називають часовим рядом. Незалежно від міри впорядкування, запис точок даних зазвичай слідує за дискретними інтервалами з рівним інтервалом. З цієї причини та оскільки більшість алгоритмів аналізу рядів даних не залежать від еталонного показника, ми використовуємо терміни ряд даних, часовий ряд і послідовність як синоніми. Точками даних запису часового ряду є одна або кілька дійсних змінних. Кожна

змінна моделює один канал часового ряду. Якщо точки даних складаються лише з однієї змінної, часовий ряд називається однофакторним; інакше він багатоваріантний. Аномалія в такому часовому ряді – це точка (наприклад, викид) або послідовність точок (наприклад, нерегулярність), яка відхиляється від моделі або вбудовування з регулярних шаблонів послідовності. У багатофакторних часових рядах ці відхилення шаблону можуть мати місце в будь-якому окремому каналі, а також у кореляції каналів (наприклад, перекіс). Аномальні підпослідовності можуть мати різну довжину, і вони можуть знову з’являтися в тому самому часовому ряді. Залежно від області часового ряду, його аномалії можуть описувати важливі події, такі як серцева недостатність у кардіології [4], структурні дефекти в реактивній турбінобудуванні або порушення екосистеми в науках про Землю.

Різні типи аномалій, моделі шаблонів і властивості часових рядів призвели до розробки безлічі різних алгоритмів виявлення аномалій. Багато з них використовують дуже схожі підходи до виявлення, але загальна різноманітність підходів надзвичайно висока, починаючи від простого виявлення викидів і статистичного аналізу, обробки сигналів і аналізу даних до підходів глибокого навчання. Оскільки всі ці підходи мають індивідуальні сильні та слабкі сторони, вибір відповідного алгоритму для певного завдання виявлення аномалії є надзвичайно складним. Наприклад, метод відстані Sub-LOF дуже добре виявляє точкову аномалію, оскільки він чутливий до величин значень і амплітуд змін; однак він не може виявити аномалію підпослідовності. LSTM-AD як метод прогнозування, навпаки, вивчає нормальні закономірності та значною мірою покладається на сезонність/періодичність; з цієї причини він легко ідентифікує аномальну підпослідовність, але оскільки він стійкий до шуму, він ігнорує точкову аномалію [11].

Оскільки немає комплексного наукового дослідження, яке б оцінювало алгоритми виявлення аномалій часових рядів, абсолютно незрозуміло,

наскільки добре вони працюють для різних наборів даних, типів аномалій і налаштування параметрів. Беручи до уваги існуючі оцінки було виявлено, що вони зазвичай розглядають лише крихітну частину пов’язаних робочих алгоритмів і що вони часто базуються на тривіальних, підібраних, упереджених, неправильно позначених, нереалістичних або лише кілька наборів даних [14].

Конкретне визначення аномалії в літературі відрізняється. Індивідуальні аномальні точки даних зазвичай називають викидами [2]. Враховуючи характеристику впорядкування даних часових рядів, аномальні точки даних були класифіковані на точкові, послідовні та контекстуальні аномалії, щоб враховувати інформацію про довжину та контекст у класифікації. Під час поділу часового ряду на підпослідовності фіксованого розміру (наприклад, за допомогою віконного перегляду), (не-) нормальність окремих шаблонів може бути визначена їх відстанню до (найближчих) сусідніх шаблонів.

2.2. Підходи до виявлення аномалій

Розглянуті алгоритми походять з різних областей досліджень і належать до різних сімейств методів. Зокрема, виявлення аномалій у:

- глибокому навчанні;
- стохастичному навчанні;
- класичному машинному навчанні;
- виявленні викидів;
- статистиці (регресія та прогнозування);
- аналізі даних;
- аналізі сигналу.

2.2.1. Методи, засновані на прогнозуванні

Методи прогнозування використовують попередньо навчену модель для прогнозування кількох часових кроків на основі поточного вікна контексту. Значення для прогнозованих точок даних залежать виключно від точок даних часового ряду в попередньому вікні контексту та попередньо вивченої моделі. Прогнозовані точки потім порівнюються зі спостережуваними значеннями в початковому часовому ряді, щоб визначити, наскільки спостережувані значення аномальні. Більшість методів прогнозування використовують ковзне вікно з кроком 1 для створення контекстного вікна та прогнозування по одній точці за раз.

Методи в цьому сімействі найбільше відрізняються типом моделі прогнозування, яку вони використовують (тобто тип моделі нормальної поведінки), способом побудови цієї моделі (тобто підходом до навчання) та метрикою розрахунку для показників аномалії (тобто метрика відстані для прогнозованих і спостережуваних значень). Представниками цієї категорії, які ми розглядаємо в нашій оцінці, є: AD-LTI, ARIMA, RBForest, RForest, SARIMA, та XGBoosting.

Зазвичай методи прогнозування навчаються в напівконтрольований спосіб: навчальний часовий ряд без аномалій використовується для вивчення нормальної моделі даних. Відхилення від цієї нормальної очікуваної поведінки в часовому ряді тесту, тобто значна різниця в спостережених і прогнозованих точках даних вважається аномальною. Аномалії можуть бути виявлені, зокрема, методами RBForest, RForest, XGBoosting, AD-LTI, LSTM-AD (який може передбачати кілька точок).

2.2.2. Методи, що засновані на реконструкції

Методи реконструкції будують модель нормальної поведінки шляхом кодування підпослідовностей нормального навчального часового ряду в (низькомірному) прихованому просторі. Щоб виявити аномалії в тестовому часовому ряді, підпослідовності з тестового ряду реконструюються з прихованого простору, а значення реконструйованих підпослідовностей потім порівнюються з початковими спостережуваними значеннями ряду. Вхідними даними для процесу реконструкції є навчальні вікна (зазвичай створені за допомогою ковзного вікна з кроком 1), які надають часовий контекст моделі. Оскільки модель побудована лише на звичайних даних (напівконтрольованих), модель не може реконструювати аномальні підпослідовності в серії тестів. Отже, бал аномалії можна розрахувати на основі різниці між вихідною та реконструйованою підпослідовностями. Представниками цієї категорії в нашій оцінці є AutoEncoder (AE), DenoisingAutoEncoder (DAE), LSTM-VAE, RobustPCA.

Винятком із напівконтрольованого навчання цих методів є чотири неконтрольовані методи FFT, SR, PCC і PCI: вони кодують вхідні підпослідовності серії тестів у заздалегідь визначений прихований простір і, таким чином, навмисно втрачають інформацію, точність, необхідну для виявлення аномалій. Під час процесу реконструкції не всі деталі оригінальних підпослідовностей можна відтворити. Отже, відмінності між реконструйованими підпослідовностями та оригінальними можна повідомити як бали аномалії.

2.2.3 . Методи, що засновані на кодуванні

Методи кодування подібні до методів реконструкції тим, що вони також кодують підпослідовності часового ряду в низьковимірному прихованому

просторі. Однак вони не намагаються реконструювати підпослідовності з латентного простору, а обчислюють оцінку аномалії безпосередньо з представлень латентного простору. Більш конкретно, бали аномалії приписуються точкам, які відповідають закодованим підпослідовностям у латентному просторі. Найпопулярніші представники цієї категорії GrammarViz та Series2Graph.

LaserDBN, PST і MultiHMM створюють імовірнісні моделі та використовують логарифм правдоподібності підпослідовностей як оцінку аномалії; у той час як MultiHMM будує модель із звичайного часового ряду навчання (напівконтрольованого), LaserDBN і PST розглядають лише тестовий часовий ряд. Series2Graph перетворює підпослідовності тестового часового ряду в низьковимірний простір, з якого підхід будує спрямований циклічний граф. Ребра графа являють собою переходи між групами підпослідовностей. Чим частіше ребро проходить часовий ряд, тим вищий його бал. Таким чином, краї з низькими оцінками є більш аномальними.

2.2.4. Методи, засновані на особливостях розподілу

Методи розподілу оцінюють розподіл даних або пристосовують модель розподілу до даних. Розподіл обчислюється або за точками даних, або за підпослідовностями, отриманими за допомогою вікон. Хоча подібність точок і підпослідовностей може бути фактором відповідності розподілу (дуже подібні шаблони вважаються рівними), аномалія оцінюється за частотою, а не за відстанню в цьому сімействі алгоритмів. Оцінки аномалії зазвичай вимірюються за допомогою ймовірностей, правдоподібностей або відстаней точок або підпослідовностей попередньо обчислених розподілів. Загалом, це неконтрольований підхід, оскільки аномалії можна знайти в екстремумах/хвостах розподілів. У напівконтрольованому випадку розподіл оцінюється протягом навчального часового ряду, який містить лише

нормальну поведінку, тоді як точки або підпослідовності тестового часового ряду потім перевіряються за раніше вивченим розподілом. Представниками методів розподілу в нашій оцінці є: COPOD, DWT-MLEAD, Fast-MCD, HBOS, Normalizing Flows (NF), S-H-ESD, DSPOT і Sub-Fast-MCD.

DWT-MLEAD, Fast-MCD і Sub-Fast-MCD оцінюють розподіл Гауса за часовим рядом. Після цього аномалія точок або підпослідовностей вимірюється їх відстанню до середнього розподілу. Різниця в цих підходах полягає в тому, що DWTMLEAD використовує дискретне вейвлет-перетворення (DWT) як етап попередньої обробки та логарифм правдоподібності підпослідовностей як оцінку аномалії; З іншого боку, Fast-MCD і Sub-Fast-MCD обчислюють оцінку як відстань Махаланобіса між точками (Fast-MCD) або підпослідовностями (Sub-Fast-MCD) і розрахунковий розподіл Гауса нормального часового ряду навчання. (напівнаглядовий). Інший алгоритм, HBOS, оцінює загальний розподіл ймовірностей підпослідовностей за допомогою гістограм; потім він використовує зворотну щільність бінів гістограми підпослідовностей як показник аномалії. Алгоритм COPOD створює емпіричну багатовимірну кумулятивну функцію розподілу (копулу) для оцінки ймовірності хвоста точок; ці ймовірності хвоста потім перетворюються на бали аномалії. S-H-ESD — це детектор точкових аномалій, який спочатку виконує декомпозицію часових рядів STL, а потім застосовує тест Ґраббса, який передбачає розподіл Гауса, до залишків, щоб позначити точки викидів. Інший детектор точкових аномалій, DSPOT, оцінює узагальнений розподіл Парето екстремальних значень часового ряду та застосовує порогове значення до хвостів розподілу, щоб позначити аномальні точки.

Нарешті, NF — це контрольований метод виявлення аномалії послідовності, який перетворює довільний попередній розподіл у розподіл Гауса за допомогою нейронної мережі; потім він позначає підпослідовності, що потрапляють у хвости розподілу, як аномальні.

2.2.5. Методи дерева ізоляції

Методи ізольованого дерева створюють ансамбль випадкових дерев, які розділяють вибірки (точки або підпослідовності) тестового часового ряду. Для побудови дерева методи рекурсивно вибирають випадкові функції та випадкові розділені значення як вузли дерева, щоб зрештою виділити зразки в листках дерева. Кількість поділів, необхідних для виділення зразка, є мірою, яка описується середньою довжиною шляху по всіх випадкових деревах в ансамблі. Оскільки аномальні зразки легше відокремити, ніж звичайні зразки, вони в середньому знаходяться ближче до кореня дерева та мають помітно коротший шлях. З цієї причини довжини шляху є характерними для нормальності зразків і, отже, їх зворотне значення перетворюється на бали аномалії. Представниками цієї категорії є: Extended Isolation Forest (EIF), Hybrid Isolation Fores (HIF), Isolation Forest - Local Outlier Factor (IF-LOF), Isolation Forest (iForest), і Sub-IF.

Загальний метод дерева ізоляції, який також є методом, на якому базуються всі алгоритми в цьому сімействі, є алгоритмом iForest. Контрольованими варіантами цього підходу є EIF і HIF. Алгоритм Sub-IF є розширенням для виявлення аномалії підпослідовності, тобто. наприклад, алгоритм, який може обробляти послідовності замість точок, а IF-LOF є комбінацією iForest і LOF [16].

Ізольовані ліси (IF), подібні до випадкових лісів, будуються на основі дерев рішень. І оскільки тут немає попередньо визначених міток, це модель без нагляду. Ізоляційний ліс були побудовані на основі того факту, що аномалії – це «нечисленні та різні» точки даних.

В ізольованому лісі випадково відібрані дані обробляються в деревовидній структурі на основі випадково вибраних функцій. Зразки, які потрапляють глибше в дерево, мають меншу ймовірність бути аномаліями,

оскільки для їх виділення потрібно більше надрізів. Подібним чином, зразки, які закінчуються коротшими гілками, вказують на аномалії, оскільки дереву було легше відокремити їх від інших спостережень.

2.3. Детальний опис обраних методів виявлення аномалій

2.3.1. Ізоляційний ліс

Як згадувалося раніше, виявлення викидів ізольованих лісів є нічим іншим, як ансамблем бінарних дерев рішень. І кожне дерево в ізольованому лісі називається ізольованим деревом (iTree). Алгоритм починається з навчання даних шляхом генерації ізоляційних дерев.

Розглянемо повний алгоритм крок за кроком:

1. Коли надається набір даних, випадкова підвибірка даних вибирається та призначається бінарному дереву.
2. Розгалуження дерева починається з вибору спочатку випадкової ознаки (з набору всіх N ознак). Потім виконується розгалуження за випадковим порогом (будь-яке значення в діапазоні мінімальних і максимальних значень вибраної функції).
3. Якщо значення точки даних менше вибраного порогу, воно переходить до лівої гілки, інакше до правої. І таким чином вузол розбивається на ліву та праву гілки.
4. Цей процес, починаючи з кроку 2, продовжується рекурсивно, доки кожна точка даних не буде повністю ізольована або до досягнення максимальної глибини (якщо визначено).
5. Наведені вище кроки повторюються для побудови випадкових бінарних дерев.

Після створення ансамблю iTrees (Ізоляційний ліс) навчання моделі завершено. Під час оцінювання точка даних проходить через усі дерева, які

були навчені раніше. Тепер «оцінка аномалії» призначається кожній із точок даних на основі глибини дерева, необхідної для досягнення цієї точки. Ця оцінка є сукупністю глибини, отриманої від кожного з iTrees. Оцінка аномалії -1 присвоюється аномаліям, а 1 — нормам на основі наданого параметра забруднення (відсоток аномалій, присутніх у даних).

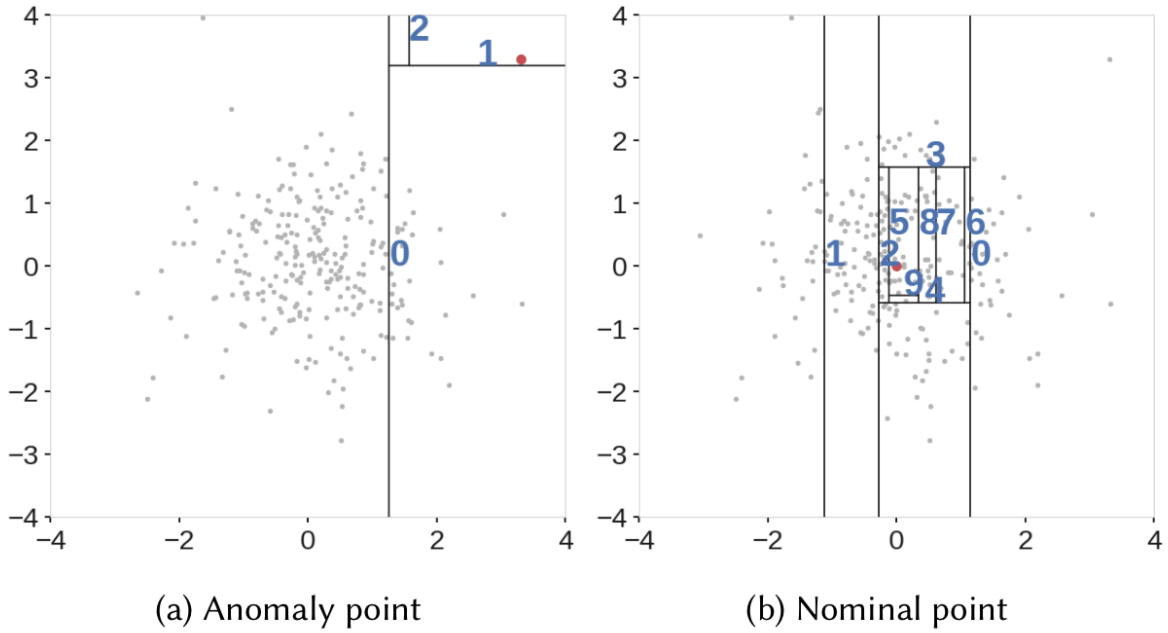

Рисунок 2.1 – Візуалізація того, що легше виділити аномалію порівняно зі звичайним спостереженням за допомогою ізоляційного лісу [24]

Ізоляційні ліси ефективні з точки зору обчислень і, як було доведено, дуже ефективні у виявленні аномалій. Незважаючи на переваги, є кілька обмежень, про які зазначено нижче.

Остаточна оцінка аномалії залежить від параметра забруднення, наданого під час навчання моделі. Це означає, що ми повинні заздалегідь мати уявлення про те, який відсоток даних є аномальними, щоб отримати кращий прогноз.

Крім того, модель страждає від зміщення через те, як відбувається розгалуження. Що ж, щоб зрозуміти другий момент, ми можемо поглянути на наведену нижче карту оцінки аномалій.

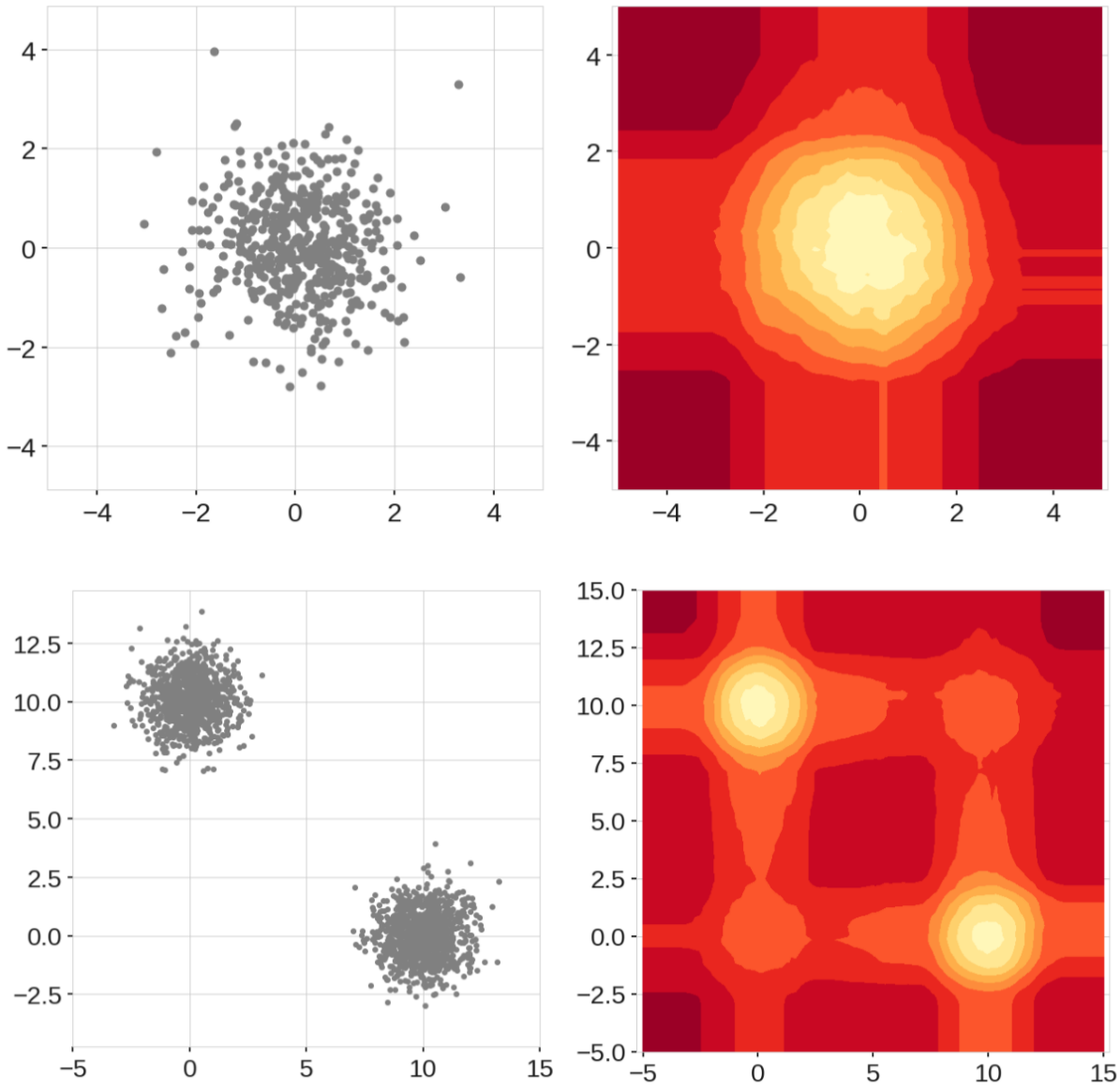

Рисунок 2.2 – Карти оцінки аномалій, отримані за допомогою ізоляційного лісу [24]

Тут, на карті оцінок праворуч, ми бачимо, що точки в центрі отримали найнижчу оцінку аномалії, яка й очікувалася. Однак ми можемо побачити чотири прямокутні області навколо кола з нижчими показниками аномалії. Отже, коли оцінюється нова точка даних у будь-якій із цих прямокутних областей, це може не бути виявлено як аномалія.

Подібним чином на наведеному вище малюнку ми бачимо, що модель призвела до двох додаткових плям (у верхньому правому та нижньому лівому кутах), яких ніколи навіть не існувало в даних.

Щоразу, коли вузол в iTree розбивається на основі порогового значення, дані розбиваються на ліву та праву гілки, що призводить до горизонтальних і вертикальних розрізів гілок. І ці обрізки гілок призводять до цієї моделі.

2.3.2. Нейронна мережа LSTM з автокодувальником

Автокодувальник LSTM — це реалізація автокодувальника для даних послідовності з використанням архітектури кодувальника-декодера LSTM. Для заданого набору даних послідовностей кодер-декодер LSTM налаштований на зчитування вхідної послідовності, кодування, декодування та відтворення. Продуктивність моделі оцінюється на основі здатності моделі відтворювати вхідну послідовність. Коли модель досягне бажаного рівня продуктивності, відтворюючи послідовність, декодерну частину моделі можна видалити, залишивши лише модель кодера. Потім цю модель можна використовувати для кодування вхідних послідовностей у вектор фіксованої довжини.

Отримані вектори потім можна використовувати в різноманітних додатках, не в останню чергу як стиснуте представлення послідовності як вхідних даних для іншої моделі навчання під наглядом. Робота Autoencoder полягає в тому, щоб отримати деякі вхідні дані, пропустити їх через модель і отримати реконструкцію вхідних даних. Реконструкція має максимально відповідати введеним. Хитрість полягає у використанні невеликої кількості параметрів, щоб ваша модель вивчала стисле представлення даних.

У певному сенсі автокодувальники намагаються вивчити лише найважливіші характеристики (стиснуту версію) даних. Тут ми розглянемо, як передавати дані часового ряду в автокодер. Ми будемо використовувати кілька

шарів LSTM (отже, LSTM Autoencoder), щоб зафіксувати часові залежності даних. Щоб класифікувати послідовність як нормальну або аномалію, ми виберемо поріг, вище якого серцебиття вважається ненормальним.

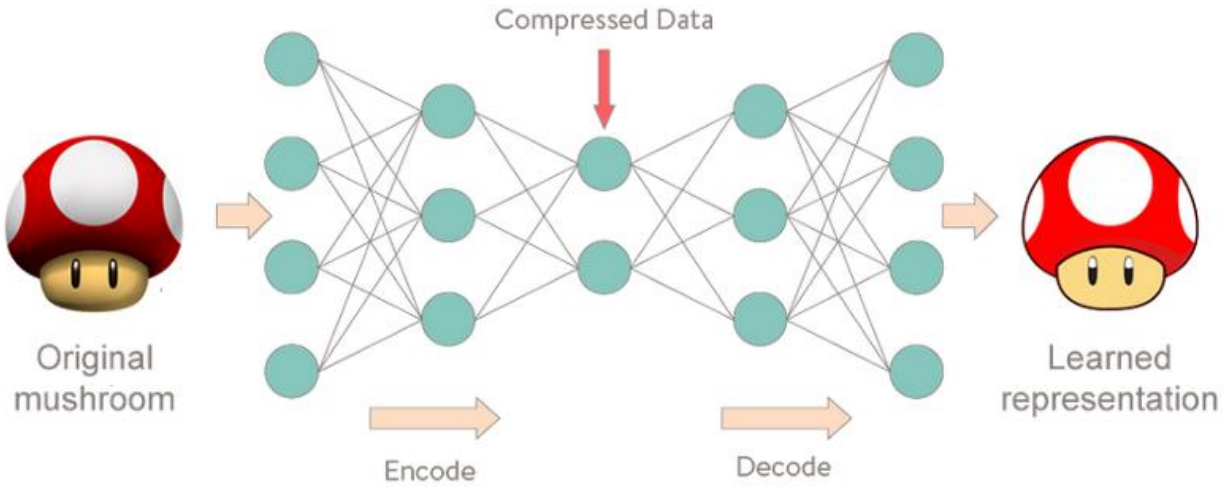

Рисунок 2.3 – Проста візуалізація робота LSTM автокодувальника [23]

Втрати при реконструкції під час навчання автокодувальника мета полягає в тому, щоб реконструювати вхідні дані якомога краще. Це досягається шляхом мінімізації функції втрат (так само, як у контрольованому навчанні). Ця функція відома як втрати при реконструкції. Перехресна втрата ентропії та середня квадратична помилка є типовими прикладами.

Теорія автокодерів полягає в тому, щоб спробувати зменшити розмірність набору даних і збільшити розмірність за допомогою автокодування. Під час фази реконструкції модель намагається нормально реконструювати дані про серцебиття. Оскільки ми надаємо дані про аномалії моделі, помилка реконструкції є високою, оскільки помилка поширюється, показуючи аномалії, які ми шукаємо. Загальна архітектура Autoencoder складається з двох компонентів. Кодер, який стискає вхідні дані, і декодер, який намагається його реконструювати.

2.4. Програмні засоби системи виявлення аномалій

2.4.1. Scikit-learn

Scikit-learn — це ключова бібліотека для мови програмування Python, яка зазвичай використовується в проектах машинного навчання. Scikit-learn зосереджено на інструментах машинного навчання, включаючи математичні, статистичні та алгоритми загального призначення, які є основою для багатьох технологій машинного навчання. Як безкоштовний інструмент Scikit-learn надзвичайно важливий у багатьох різних типах розробки алгоритмів для машинного навчання та пов’язаних технологій.

Деякі з основних ключових елементів Scikit-learn, корисних для машинного навчання, включають алгоритми класифікації, регресії та кластеризації. Наприклад, Scikit-learn підтримує роботу з випадковими лісами, де окремі цифрові дерева містять інформацію про вузли, яка об’єднується в декілька деревних архітектур для досягнення лісового підходу. Інший спосіб поговорити про це полягає в тому, що кожне дерево включає кластеризовані вузли в топології дерева, а аналіз різних дерев додається разом, щоб отримати глобальний підхід, який точніше обробляє дані для візуалізації результатів.

Окрім випадкового лісу, Scikit-learn дозволяє працювати з посиленням градієнта, векторними машинами та іншими елементами машинного навчання, які є ключовими для досягнення результатів. Як основний ресурс Scikit-learn працює з такими інструментами, як SciPy і matplotlib, які забезпечують візуалізацію та багато іншого.

Даний фреймворк в контексті роботи використовується для виконання первинної обробки даних (MinMaxScaler та StandartScaler), а також використання базової моделі ізоляційного лісу (Isolation Forest).

2.4.2. TensorFlow

TensorFlow — це бібліотека з відкритим кодом для швидких чисельних обчислень. Він був створений і підтримується Google і був випущений під ліцензією Apache 2.0 з відкритим кодом. API номінально призначений для мови програмування Python, хоча є доступ до основного API C++.

На відміну від інших числових бібліотек, призначених для використання в Deep Learning, як-от Theano, TensorFlow було розроблено для використання як у дослідженнях і розробках, так і в виробничих системах, серед яких не в останню чергу RankBrain у пошуку Google і цікавий проект DeepDream.

Він може працювати на системах з одним процесором і графічним процесором, а також на мобільних пристроях і великомасштабних розподілених системах із сотень машин. Обчислення описується в термінах потоку даних і операцій у структурі орієнтованого графа:

- вузли виконують обчислення та мають нуль або більше входів і виходів. Дані, які переміщуються між вузлами, відомі як тензори, які є багатовимірними масивами реальних значень.
- ребро визначає потік даних, розгалуження, цикл і оновлення стану. Спеціальні ребра можна використовувати для синхронізації поведінки в межах графа, наприклад, очікування завершення обчислень для кількох вхідних даних.
- операція — це назване абстрактне обчислення, яке може приймати вхідні атрибути та створювати вихідні атрибути. Наприклад, ви можете визначити операцію додавання або множення.

У контексті побудованої системи дана бібліотека застосовувалася для отримання базових шарів нейронних мереж, таких як Input, Dropout, Dense, LSTM, TimeDistributed, RepeatVector. А для регуляризації використовувалися відповідні regularizers.

2.4.3. Keras

Keras — це високорівневий API глибокого навчання, розроблений Google для впровадження нейронних мереж. Він написаний на Python і використовується для полегшення впровадження нейронних мереж. Він також підтримує численні серверні обчислення нейронної мережі.

Keras відносно легко освоїти та працювати з ним, тому що він забезпечує інтерфейс Python з високим рівнем абстракції, маючи можливість використовувати кілька серверних інтерфейсів для обчислень. Це робить Keras повільнішим, ніж інші фреймворки глибокого навчання, але надзвичайно зручним для початківців.

TensorFlow прийняв Keras як офіційний API високого рівня. Keras вбудовано в TensorFlow і може використовуватися для швидкого глибокого навчання, оскільки він надає вбудовані модулі для всіх обчислень нейронної мережі. У той же час, обчислення з використанням тензорів, обчислювальних графів, сеансів тощо можна створювати на замовлення за допомогою Tensorflow Core API, який надає повну гнучкість і контроль над програмою та дозволяє реалізувати ідеї за відносно короткий час.

Keras — це API, створений таким чином, щоб його було легко освоїти. Keras був створений, щоб бути простим. Він пропонує послідовні та прості API, скорочує кількість дій, необхідних для реалізації загального коду, і чітко пояснює помилки користувача.

Час створення прототипу в Keras менше. Це означає, що ідеї можна реалізувати та розгорнути за більш короткий час. Keras також надає різноманітні варіанти розгортання залежно від потреб користувача.

Мови з високим рівнем абстракції та вбудованими функціями працюють повільно, тому створювати спеціальні функції в них може бути важко. Але Keras працює на основі TensorFlow і відносно швидкий. Keras також глибоко

інтегровано з TensorFlow, тому ви можете легко створювати персоналізовані робочі процеси.

Дослідницьке співтовариство Keras є великим і високорозвиненим. Доступна документація та допомога є набагато більшими, ніж інші структури глибокого навчання.

Keras комерційно використовується багатьма компаніями, як-от Netflix, Uber, Square, Yelp тощо, які розгорнули продукти у відкритому доступі, створені за допомогою Keras.

2.5. Висновки

Отже, було виявлено, що жоден з розглянутих алгоритмів не працює краще за інші на всіх наборах даних. Для окремих задач виявлення аномалій прості методи дають майже таку ж продуктивність, як і більш складні методи.

Гарні результати показують гібридні системи виявлення аномалій, які поєднують неявні переваги для виявлення більш різноманітних аномалій у часових рядах із довільними характеристиками.

Встановлено, що лише небагато алгоритмів можуть обробляти всі часові ряди без помилок і в межах загального часу та пам’яті. Тому треба наголосити на важливості подальших досліджень надійності та масштабованості алгоритмів виявлення аномалій часових рядів.

Виконаний аналіз методів показав, що більшість алгоритмів виявлення аномалій можуть бути чутливими до налаштувань своїх параметрів і потребувати додаткових налаштувань. Що погіршує цю проблему, так це те, що більшість практичних випадків використання не мають навчальних даних для конфігурації алгоритму. З цієї причини дуже необхідні подальші дослідження алгоритмів автоматичного налаштування та самоналаштування.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1. Вхідні дані та їх обробка

Обрані початкові дані були зібрані на двох сонячних електростанціях в Індії протягом 34 днів. Він містить дві пари файлів – кожна пара містить один набір даних про виробництво електроенергії та один набір даних про показання датчиків. Набори даних про виробництво електроенергії збираються на рівні інвертора - до кожного інвертора підключено кілька ліній сонячних панелей. Дані датчиків збираються на рівні заводу – один масив датчиків, оптимально розміщений на підприємстві.

На сонячній електростанції є кілька проблемних питань -

1. Чи можемо ми спрогнозувати вироблення електроенергії на наступні пару днів? - це дозволяє краще керувати мережею
2. Чи можемо ми визначити потребу в чищенні/обслуговуванні панелей?
3. Чи можемо ми визначити несправне або неоптимальне обладнання?

На рисунку нижче (рис. 3.1) можна побачити частину вхідних даних у необробленому вигляді.

3.2. Запропонований алгоритм пошуку аномалій

1. Завантаження початкових даних, їх статистична обробка, зокрема побудова розширеної кореляційної матриці та ін;
2. Вибір параметрів для дослідження на предмет аномалій, їх нормалізація;

3. Побудова, візуалізація та аналіз часових рядів обраних на попередньому етапі вхідних параметрів;
4. Побудова і навчання ансамбля моделей iForest на основі ізоляційного лісу (Isolation Forest). Підбір параметрів цього ансамблю методом решітчастого пошуку GridSearch та його оцінка;
5. Виявлення і візуалізація кандидатів в аномалії на основі результатів, отриманих в попередньому пункті;
6. Побудова контура, отриманого в пп. 4 ансамбля моделей iForest;
7. Побудова і навчання альтернативної моделі LSTM з автокодувальником та її оцінка;
8. Графічне відображення функції втрат у часі для моделі LSTM з автокодувальником разом з порогом аномалій;
9. Виявлення та візуалізація кандидатів в аномалії за допомогою LSTM з автокодувальником;
10. Порівняльний аналіз результатів, отриманих обома методами.

Проілюструємо етапи цього алгоритму.

На рисунку 3.2 зображений статистичний огляд даних у вигляді розширеної кореляційної матриці. Виконаємо нормалізацію вхідних параметрів за допомогою логарифмування та StandardScaler з бібліотеки sklearn. Центрування та масштабування відбуваються незалежно для кожної функції шляхом обчислення відповідних статистичних даних на зразках у навчальному наборі. Середнє значення та стандартне відхилення потім зберігаються для подальшого використання в даних за допомогою перетворення.

	DATE_TIME	PLANT_ID	SOURCE_KEY	DC_POWER	AC_POWER	DAILY_YIELD	TOTAL_YIELD
0	2020-05-15 00:00:00	4135001	1BY6WEcLGh8j5v7	0.0	0.0	0.000	6259559.0
1	2020-05-15 00:00:00	4135001	1IF53ai7Xc0U56Y	0.0	0.0	0.000	6183645.0
2	2020-05-15 00:00:00	4135001	3PZuoBAID5Wc2HD	0.0	0.0	0.000	6987759.0
3	2020-05-15 00:00:00	4135001	7JYdWkrLSPkdwr4	0.0	0.0	0.000	7602960.0
4	2020-05-15 00:00:00	4135001	McdE0feGgRqW7Ca	0.0	0.0	0.000	7158964.0
...	...	...	...	...	...	...	...
68773	2020-06-17 23:45:00	4135001	uHbuxQJl8lW7ozc	0.0	0.0	5967.000	7287002.0
68774	2020-06-17 23:45:00	4135001	wCURE6d3bPkepu2	0.0	0.0	5147.625	7028601.0
68775	2020-06-17 23:45:00	4135001	z9Y9gH1T5YWrNuG	0.0	0.0	5819.000	7251204.0
68776	2020-06-17 23:45:00	4135001	zBIq5rxdHJRwDNY	0.0	0.0	5817.000	6583369.0
68777	2020-06-17 23:45:00	4135001	zVJPv84UY57bAof	0.0	0.0	5910.000	7363272.0

Рисунок 3.1 – Початкові дані без обробки

Стандартизація набору даних є загальною вимогою для багатьох оцінювачів машинного навчання: вони можуть поводитися погано, якщо окремі функції більш-менш не схожі на стандартні дані з нормальним розподілом (наприклад, гауссове значення з нульовим середнім і одиничною дисперсією).

Наприклад, багато елементів, які використовуються в цільовій функції алгоритму навчання (таких як ядро RBF опорних векторних машин або регуляризатори L1 і L2 лінійних моделей), припускають, що всі функції зосереджені навколо 0 і мають дисперсію в тому самому порядку. Якщо ознака має дисперсію, яка на порядки більша, ніж інші, вона може домінувати над цільовою функцією та зробити оцінювач нездатним правильно навчатися на інших ознаках, як очікувалося.

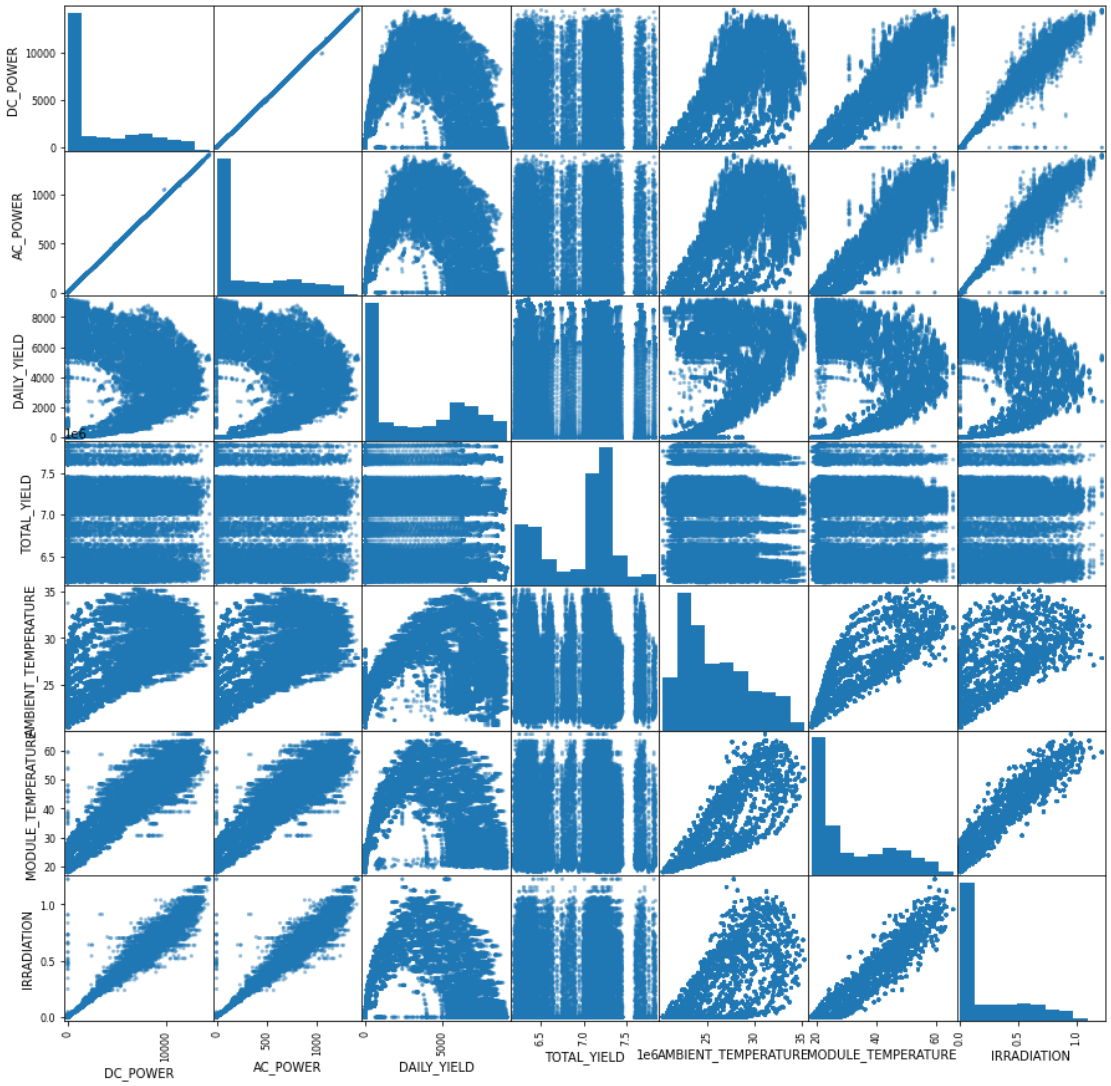

Рисунок 3.2 – Розширена кореляційна матриця вхідних параметрів

Далі оберемо параметри, які ми будемо досліджувати на предмет аномалій. Експериментальним шляхом було визначено, що це будуть об'єми сонячної енергії та виробленої за допомогою конкретної сонячної батареї. Далі на рисунку 3.3 зображено часові ряди цих параметрів. Можна побачити, що 7 червня і 14 червня є деякі області неправильного виробництва, які можна вважати аномаліями у роботі сонячних батарей.

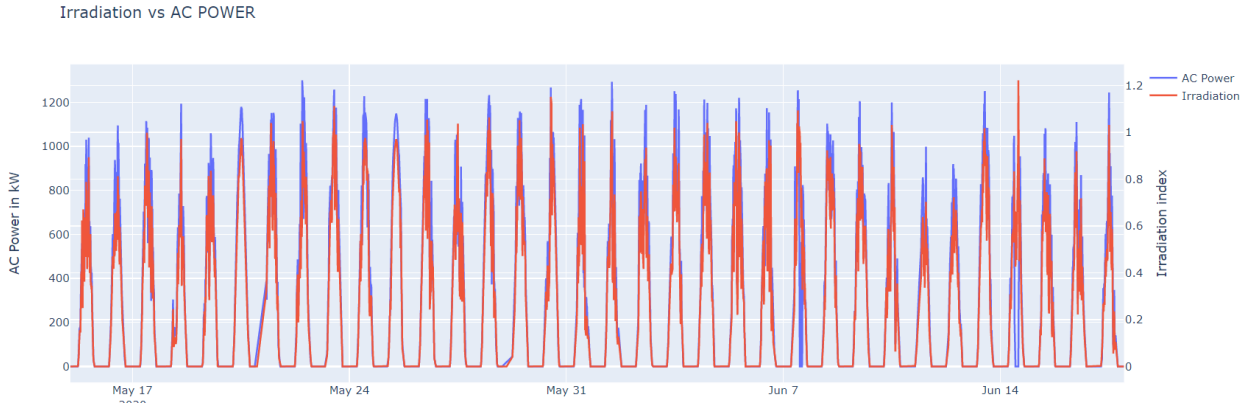

Рисунок 3.3 – Часові ряди обраних вхідних параметрів

3.3. Застосування ізоляційного лісу

Ініціалізуємо ансамбль моделей регресії на основі ізоляційного лісу з наступними параметрами, які були підібрані:

```
clf = IsolationForest(n_estimators=1000,
                      max_samples='auto',
                      contamination=.03,
                      max_features=2,
                      bootstrap=False,
                      n_jobs=-1,
                      random_state=42,
                      verbose=0)
```

Рисунок 3.4 – Параметри моделі ізоляційного лісу

Отже, в результаті отримаємо наступне:

Таблиця 3.1 – Результати застосування методу ізоляційного лісу

Вибірка	Виділені аномалії	Неаномальні значення	% аномалій
Тренувальна	57	1836	3,1%
Тестова	41	1220	3,3%

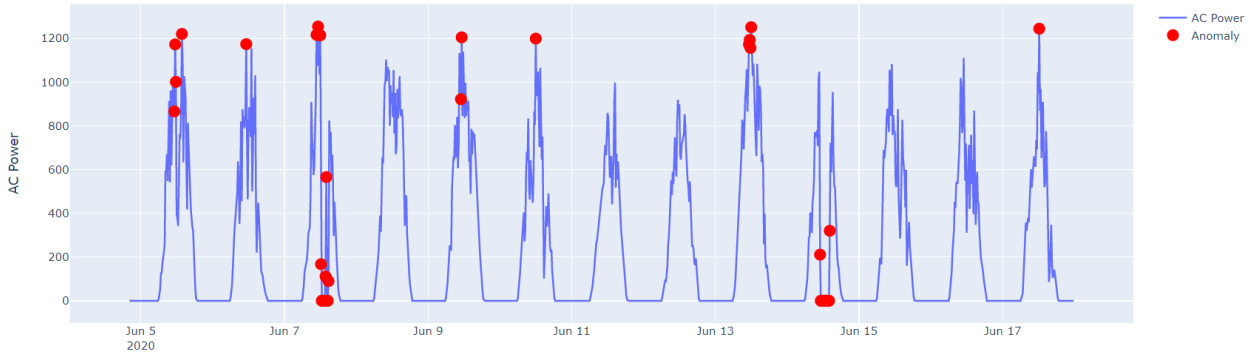

Рисунок 3.4 – Результат визначення аномалій за допомогою iForest

Модель дійсно виявляє області неправильного виробництва 7 і 14 червня, але також вона виявляє аномалії на піках більшості інших днів. Але їх не можна віднести до аномальних, отже, модель не є цілком бажаною до застосування.

Для цього поглянемо на контур отриманого iForest на рис. 3.5.

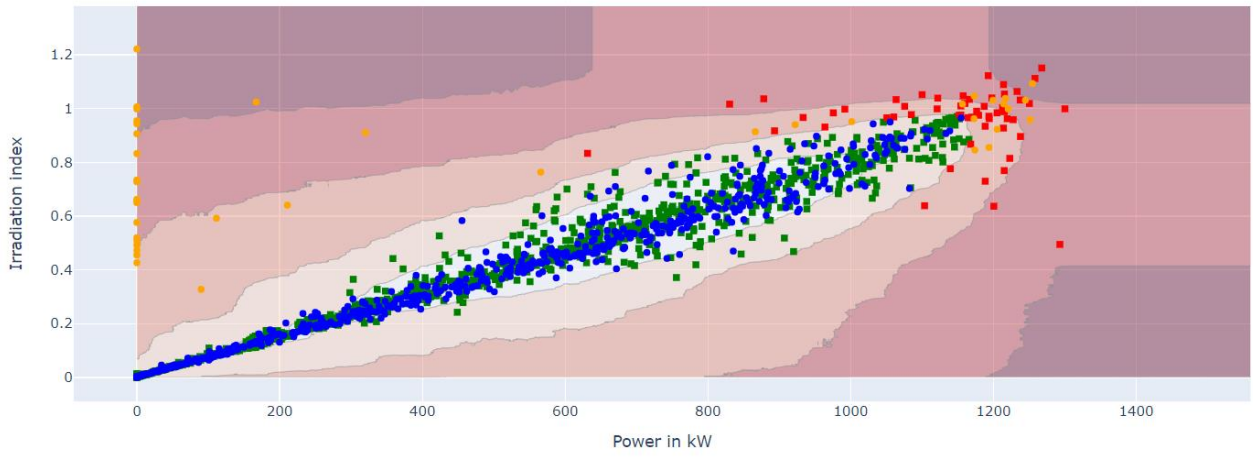

Рисунок 3.5 – Контур отриманого iForest

Оранжевим та червоним кольором виділено аномалії з тестової та навчальної вибірки відповідно. Чітко видно межу, для якої показники стають аномальними. Отже, зелені та сині – це звичайні показники, які не є аномальними. Тут ми наочно бачимо, як працює модель Isolation Forest. Жовті крапки показують аномалії, виявлені в тестовому наборі даних, а також

червоні квадрати, які показують нам аномалії, виявлені в наборі навчальних даних.

Ці точки не відповідають контурному малюнку графіка, і ми чітко бачимо, що жовті точки в крайньому лівому куті є точками від 7 червня і 14 червня.

3.4. Застосування LSTM з автокодувальником

Стислий опис отриманої моделі наведено на рис. 3.6 нижче, де можна побачити кількість шарів та параметрів для тренування.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 1, 4)]            0

 lstm (LSTM)                 (None, 1, 16)             1344

 lstm_1 (LSTM)               (None, 4)                 336

 repeat_vector (RepeatVector  (None, 1, 4)             0
 )

 lstm_2 (LSTM)               (None, 1, 4)              144

 lstm_3 (LSTM)               (None, 1, 16)             1344

 time_distributed (TimeDistr  (None, 1, 4)             68
 ibuted)

=================================================================
Total params: 3,236
Trainable params: 3,236
Non-trainable params: 0
```

Рисунок 3.6 – Стислий опис моделі LSTM з автокодувальником

Після навчання у 100 епох маємо наступні значення функції втрат, що є цілком прийнятними.

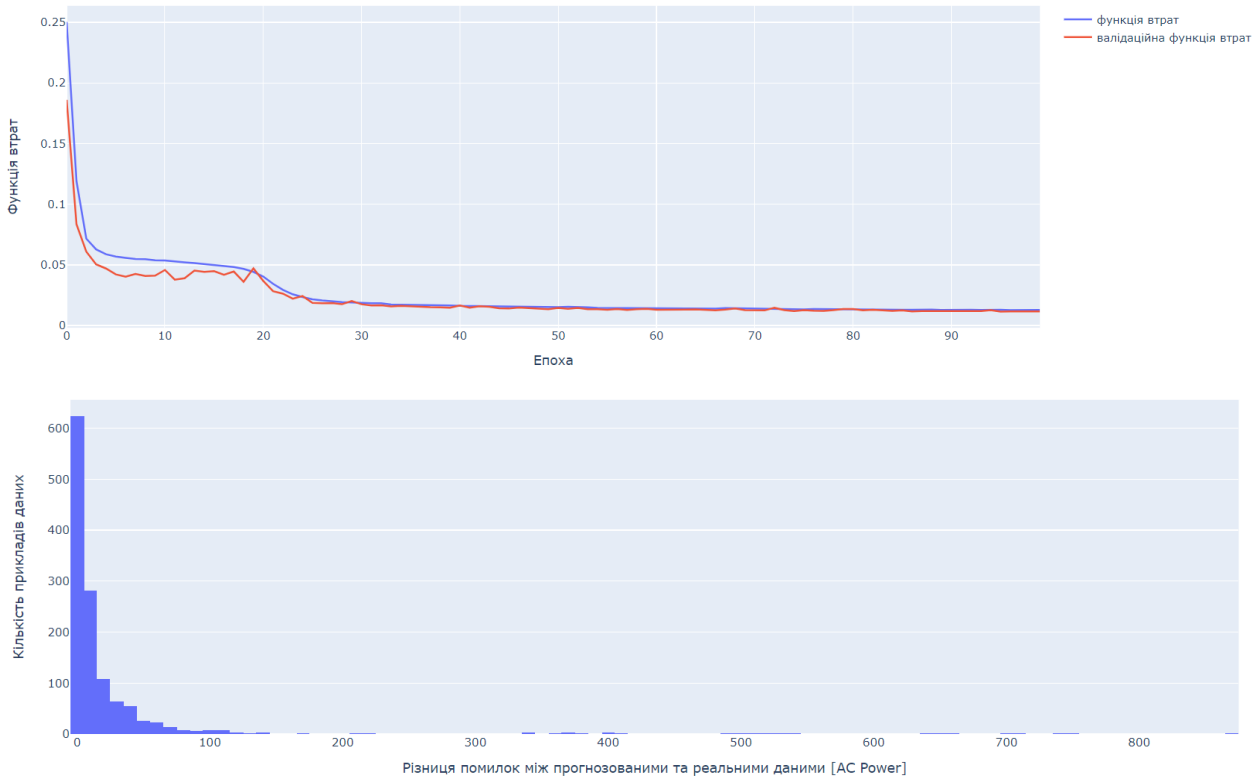

Рисунок 3.7 – Помилка автокодувальника з LSTM та розподіл помилок протягом навчання

Щоб краще зрозуміти принцип розподілу аномалій даним методом, наведемо поріг аномалії на рис. 3.8 нижче.

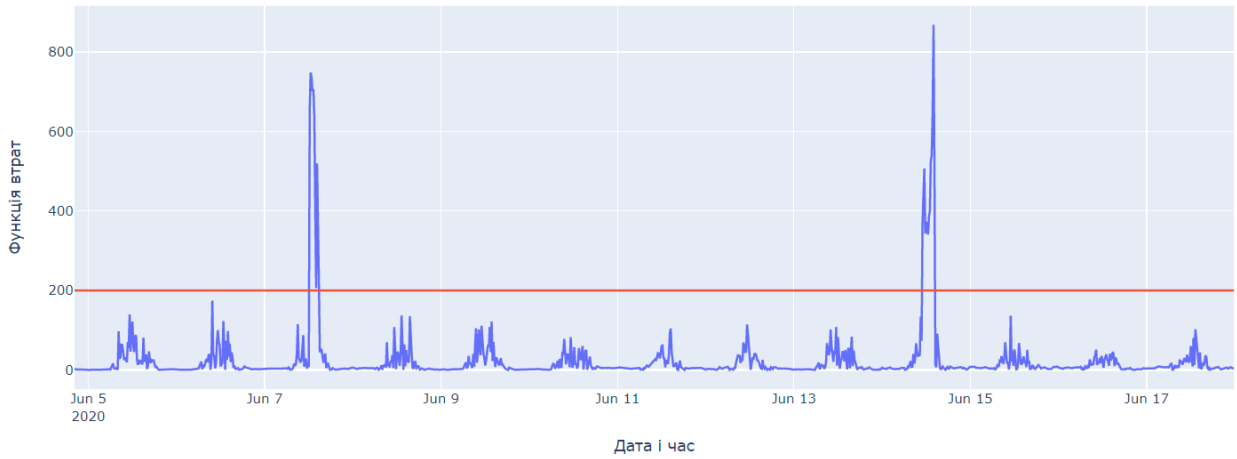

Рисунок 3.8 – Функція втрат у часі та поріг аномалій

Таблиця 3.2 – Результати застосування моделі LSTM з автокодувальником

Виділені аномалії	Неаномальні значення	% аномалій
25	1235	2,02%

Отже, наведемо візуалізацію результатів визначення аномалій даним методом.

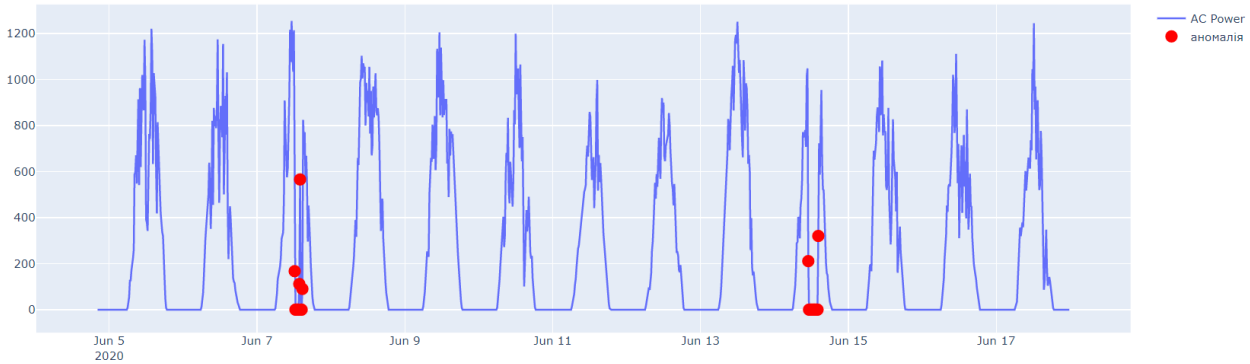

Рисунок 3.9 – Результат визначення аномалій за допомогою LSTM з автокодувальником

3.5. Висновки

У даному розділі було виконано аналіз вхідних даних, що було обрано для виявлення аномалій, реалізовано моделі обраними методами та проведено експерименти, результатом яких є побудовані отримані візуалізації.

У якості моделей для реалізації експерименту було обрано два кандидати – ізоляційний ліс (iForest) та LSTM з автокодувальником. Також вищезгадану нейронну мережу було побудовано з різними конфігураціями кількостей шарів та їх розмірностями. Дивлячись на результати можна зробити висновки, що реалізовані моделі пройшли навчання успішно та є придатними для виявлення аномалій у часових рядах (з деякими уточненнями щодо першої з них). Також з мінімальної різниці відсотку визначених аномалій на тестовій

та навчальній вибірці робимо висновок, що моделі мають гарні узагальнюючі властивості.

Як результат проведення експериментів було встановлено, що підхід, заснований на LSTM автокодувальнику є більш ефективним способом виявлення аномалій, ніж підхід Isolation Forest, можливо, з більшим набором даних Isolation Forest може перевершити автокодувальник, маючи швидшу та досить хорошу модель для виявлення аномалій.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Стартап як форма малого ризикового підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок, і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартаппроекту, взята окремо, не вартує майже нічого. Головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов.

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Далі наведено маркетинговий аналіз стартап проекту.

4.1. Опис ідеї проекту

В межах підпункту було проаналізовано і подано у вигляді таблиць:

а) зміст ідеї;

б) можливі напрямки застосування;

в) основні вигоди, що може отримати користувач товару;

г) чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подані у вигляді таблиці (таблиця 4.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1 — Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Дана комплексна система дозволяє розвязати проблему виявлення аномалій часових рядів	Виявлення аномалій при роботі з часовими рядами	Зменшеня часу, витраченного на подальшу обробку даних
	Аналітика даних	Покращення якості прогнозів даних

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

а) визначення переліку техніко-економічних властивостей та характеристик ідеї;

б) визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

в) проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають

а) гірші значення (W, слабкі);

б) аналогічні (N, нейтральні) значення;

в) кращі значення (S, сильні) (табл. 4.2).

Таблиця 4.2 —Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтр. сторона)	S (сильна сторона)
		Мій проект	Facebook Prophet	Amazon quicksight			
1.	Форма виконання	Надавання послуг	Надавання послуг	Надаванння послуг		+	
2.	Собівартість	Низька	Висока	Середня			+
3.	Функціонал	Вузький	Широкий	Широкий	+		

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

4.2. Технологічний аудит ідеї проекту

В межах даного підрозділу було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 4.3):

1) за якою технологією буде виготовлено товар згідно ідеї проекту?

2) чи існують такі технології, чи їх потрібно розробити/доробити?
3) чи доступні такі технології авторам проекту?

Таблиця 4.3 — Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Створення системи виявлення аномалій часових рядів	Використання мови програмування Python	Наявна	Доступна
		Використання мови програмування C#	Відсутні	Недоступні
		Використання мови C++	Відсутні	Недоступні
Обрана технологія реалізації ідеї проекту: мова програмування Python.				

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як Python через доступність та безкоштовність.

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 4.4).

Таблиця 4.4 — Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Хар-ка
1	Кількість головних гравців, од	50
2	Загальний обсяг продаж, грн/ум.од	100000
3	Динаміка ринку	Зростає
4	Наявність обмежень для входу	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі, %	10%

За результатами аналізу таблиці 4.4 було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики та сформовано орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5 — Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Точне та швидке виявлення аномалій часового ряду	Власник бізнесу	Велика кількість даних	Простота використання, висока точність
Аналіз вхідних даних	Аналітики даних	Цікавить простота у використанні, низька ціна підтримки системи	Швидкість створення, низька ціна

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 4.6, 4.7).

Таблиця 4.6 — Фактори загроз

№ п/ п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Вихід на ринок продуктів з кращими характеристиками	Передбачити додаткові переваги власного програмного продукту (ПП) для того, щоб повідомити про них саме після виходу на ринок конкурентів. Вдосконалення технічних моментів власного продукту. Обрати нову цільову аудиторію і зосередитися на ній: зниження цін.
2	Зміна потреб користувачів	Користувачам необхідний сервіс з більшим/новим функціоналом.	Розроблення гнучкої архітектури програмного забезпечення для легшого впровадження нового функціоналу

Таблиця 4.7 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Гнучкі ціни	Зменшення ціни товару задля збільшення попиту	Введення власних гнучких цін
2	Поява нових методів квантування зображення	З’являться нові методи, що будуть швидше, та більш точно квантувати зображення	Покращити ПП додаванням нового функціоналу, розширення можливостей

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (таблиця 4.8).

Таблиця 4.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1. Тип конкуренції- чиста	Існує величезна кількість конкурентів на ринку.	Якісно провести рекламу.
2. За рівнем конкурентної боротьби - міжнародний	Компанії-конкуренти з інших країн	Створити основу ПП таким чином, щоб можна було легко переробити даний ПП для використання у галузях інших країн.
3. За галузевою ознакою - міжгалузева	Продукт може використовуватись для різних галузей	Постійне вдосконалення продукту, що не має прив’язки до сфери
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між видами ПП, їх особливостями.	Створити ПП, враховуючи недозображення конкурентів
5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення ПП, щоб собівартість була нижчою	Удосконалення моделі. Використання більш дешевих технологій для розробки, ніж використовують конкуренти, але тільки якщо

		ці технології відповідають необхідним вимогам якості.
6. За інтенсивністю - не марочна	Бренд присутній, але його роль незначна	Реклама, участь у конференціях, семінарах.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (табл. 4.9).

Таблиця 4.9 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товарозамінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
	Facebook Prophet	Наявність вже існуючих рішень	-	Контроль якості продукту	Наявність більш широкого функціоналу, зручнішого інтерфейсу
Висновки:	Досить інтенсивна конкурентна боротьба з іншими гравцями	Є можливості виходу на ринок, але є і конкуренти.	-	Клієнти диктують умови роботи на ринку:	Необхідно випускати ПЗ не гірше, ніж у конкурентів та розширяти функціонал.

				зручний інтерфейс	

За результатами аналізу було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію.

Цей висновок був врахований при формулюванні переліку факторів конкурентоспроможності у наступному пункті. На основі аналізу конкуренції, проведеного в таблиці, а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл. 4.6, 4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності.

Аналіз оформлено у (табл. 4.10).

Таблиця 4.10 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Ціна	Один із факторів для вибору продукту клієнтом.
2	Якість	Один із факторів для вибору продукту клієнтом.
3	Зручність роботи з програмою	Дозволяє користувачу легко працювати з програмою

За визначеними факторами конкурентоспроможності (табл. 4.10) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 —Порівняльний аналіз сильних та слабких сторін проекту

№ п/ п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні						
			–3	–2	–1	0	+1	+2	+3
1	Ціна	15					*		
2	Якість	10			*				
3	Зручність роботи з програмою	15					*		

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 4.11).

Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 4.12 — SWOT-аналіз стартап-проекту

Сильні сторони: Якість Простота використання Висока швидкодія	Слабкі сторони: Дуже насичений ринок, мала кількість функціоналу, відсутня кросплатформеність.
Можливості: насичення ринку новим підходом до прогнозування; різноманітна клієнтура, вдосконалення системи	Загрози: Конкуренція

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 4.13).

Таблиця 4.13 — Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	PR, просування бренду	50%	6 місяців
2	Перехід на безкоштовне розповсюдження	75%	3 місяців
3	Партнерство для об’єднання продукції	65%	2 місяці

Після аналізу було обрано альтернативу №2.

4.3. Аналіз ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 4.14).

Таблиця 4.14 — Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Власники бізнесу	Висока	Високий	Сильна	Просто
2	Поодинокі користувачі	Середня	Нізька	Сильна	Складно
3	Великі компанії	Середня: велика конкуренція і можливість власних аналіт.	Середній	Сильна	Складно
4	Маленькі компанії.	Середня	Низький	Слабка	Середня
Які цільові групи обрано:1,2,4					

За результатами аналізу потенційних груп споживачів було обрано цільові групу, для яких буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 4.15).

Таблиця 4.15 — Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Постійне оновлення і покращення продукту	Ринкове позиціонування на індивіуальних користувачів	Швидкодія, якість продукту	Концентрований маркетинг

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 4.16).

Таблиця 4.16 — Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні.	Компанія буде шукати нових споживачів та забирати існуючих у конкурентів	Буде копіювати, удосконалювати та створювати свої унікальні пропозиції	Зайняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.15) та стратегії конкурентної поведінки (таблиця 4.16) розроблено стратегію позиціонування (таблиця 4.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.17 — Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартаппроекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Легкість розуміння, зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для створення палітри кольорів.	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Економія часу; Зручність застосування; Практичність та точність результату

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап- компанії на ринку.

4.4. Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього підсумовано результати попереднього аналізу конкурентоспроможності товару (таблиця 4.18). Концепція товару - письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 4.18 — Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Швидкість отримання результату	Швидке виявлення аномалій	Необхідно додатково підвищити швидкість виявлення аномалій та варіативність результатів для отримання переваги
2	Зручність застосування	Не потрібно мати глибоких знань, для того щоб проаналізувати дані	ПП сам може налаштовувати та використовувати продукт без допомоги розробників.
3	Практичність та точність результату	Користувач отримує точні (з малою похибкою розбіжності) результати.	Користувач на виході роботи ПП отримує модель та виявлені аномалії, котрі відповідають необхідним показникам достовірності та точності. Отримані дані можна використовувати для подальшого прогнозуваннч.

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 4.19).

1-й рівень При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Таблиця 4.19 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Генерація палітри кольорів зображення за допомогою інтелектуальних систем		
II. Товар реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Індивідуальний підхід. 2. Низька ціна. 3. Простота у використанні.	1.Нм 2.Нм 3.Нм	1.Технологічна 2.Економічна 3.Технологічна
	Якість: тестування сторонніми фірмами		
	Пакування: відсутнє		
	Марка: ModelAnomalis		

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне

визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (таблиця 4.20). Аналіз проведено експертним методом.

Таблиця 4.20 — Визначення меж встановлення ціни

№ п/п	Рівень цін на товаризамінники	Рівень цін на товарианалоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	1800$	2500$	У всіх трьох груп високий рівень доходів	500$--

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (таблиця 4.21).

Таблиця 4.21 — Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Канал нульового рівня	Продаж	0(напряму)	Власна

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 4.22).

Таблиця 4.22 — Концепція маркетингових комунікацій

№ п/ п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Встановлення програми у персональний комп’ютер і його використання	Інтернет	Низька ціна, простота використання, універсальність	Показати переваги рішення над конкурентами, виділити ключові особливості	Створення сайту продукту, розповсюдження інформації про продукт на спеціалізованих ресурсах.

Було визначено, що придбання продукту буде проводитись через мережу Інтернет або при безпосередньому спілкуванні із представниками компанії. Розповсюдження інформації про продукт буде проводитись виключно через Інтернет, адже аудиторія даного продукту активно користується всесвітньою мережею.

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

4.5. Висновки

В даному розділі проведено підготовчий аналіз для впровадження розробленої системи в якості стартап проекту. Досліджено аналогічні конкурентні системи, встановлено сильні та слабкі сторони системи в порівнянні з ними. Також було досліджено можливі шляхи розповсюдження продукту та його ймовірну аудиторію, рівень доходів та ймовірну ціну продукту, що розробляється.

Було проведено аналіз потенційних ризиків і можливостей, а також розраховані основні фінансово-економічні показники проекту. Отримані результати кажуть про те, що реалізація проекту є доцільною. Було визначено сильні сторони проекту: зручність у використанні, ціна, якість. Серед слабких варто виділити невелику кількість функціоналу. Варто відмітити можливість реклами продукту на спеціалізованих ресурсах із зазначенням сильних сторін проекту.

ВИСНОВКИ

Метою дослідження було проаналізувати дані та реалізувати кілька методів виявлення аномалій на основі механізмів штучного інтелекту та обрати найкращий з них для подальшого використання при вирішенні подібних задач.

У ході проведення дослідження були вивчені властивості досліджуваного часового ряду, а саме – даних з плантацій сонячних батарей у Індії, після чого було встановлено необхідність їх первинної обробки та вибору значущих параметрів для виявлення аномалій.

Після виконання низки маніпуляцій над вхідними даними були проведені експерименти з побудовою та тестування моделей машинного навчання з метою визначення найкращої з них за якістю виявлення аномалій.

Однією з головних задач, що було поставлено – була побудова моделей прогнозування на основі методів штучного інтелекту, а саме – LSTM з автокодувальником та ізоляційний ліс (Isolation Forest).

Як результат проведення експериментів було встановлено, що підхід, заснований на LSTM автокодувальнику є більш ефективним способом виявлення аномалій, ніж підхід Isolation Forest, можливо, з більшим набором даних Isolation Forest може перевершити автокодувальник, маючи швидшу та досить хорошу модель для виявлення аномалій.

Результат даної роботи можна застосувати при вирішенні подібних задач виявлення аномалій часових рядів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Sub biah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakan tan, Pranav Shyam, Girish Sastry, Amanda Askell, Sand hini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Advances in Neural Information Pro cessing Systems, pages 1877–1901, 2020. 1, 8;

2. Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to end object detection with transformers. In European Confer ence on Computer Vision, pages 213–229. Springer, 2020. 3;

3. Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Pi otr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint arXiv:2006.09882, 2020. 1, 2, 4, 5, 6, 11;

4. Mathilde Caron, Hugo Touvron, Ishan Misra, Herve J ´ egou, ´ Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerg ing properties in self-supervised vision transformers. arXiv preprint arXiv:2104.14294, 2021. 1, 2, 3, 4, 5, 6, 8, 11, 12, 13;

5. Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tian heng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155, 2019. 11;

6. Ting Chen, Simon Kornblith, Mohammad Norouzi, and Ge offrey Hinton. A simple framework for contrastive learning of visual representations. In ICML, 2020. 1, 4, 12;

7. Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297, 2020. 1, 2, 4, 5, 6;

8. Xinlei Chen and Kaiming He. Exploring simple siamese rep resentation learning. In Proceedings of the IEEE/CVF Con ference on Computer Vision and Pattern Recognition, pages 15750–15758, 2021. 1;

9. Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057, 2021. 1, 2, 3, 4, 5, 6, 12, 13;

10. MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https : / / github . com / open - mmlab/mmsegmentation, 2020. 11;

11. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009. 1, 2, 4, 5, 11, 12;

12. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018. 1;

13. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Syl vain Gelly, et al. An image is worth 16x16 words: Trans formers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 1, 2, 3, 4, 8;

14. Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. arXiv preprint arXiv:2106.09681, 2021. 11;

15. Priya Goyal, Piotr Dollar, Ross Girshick, Pieter Noord- ´ huis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He.

Accurate, large mini batch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677, 2017. 11;

16. Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Herve J ´ egou, and Matthijs Douze. ´ Levit: a vision transformer in convnet's clothing for faster inference. arXiv preprint arXiv:2104.01136, 2021. 4;

17. Jean-Bastien Grill, Florian Strub, Florent Altche, Corentin ´ Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Do ersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Moham mad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In NeurIPS, 2020. 1, 4;

18. Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual rep resentation learning. In CVPR, 2020. 1, 4;

19. Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Gir- ´ shick. Mask r-cnn. In ICCV, 2017. 5, 11;

20. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 1, 5;

21. Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk;

22. Abhishek Shah, LSTM Autoencoder for Anomaly Detection for ECG data. URL: https://medium.com/@jwbtmf/lstm-autoencoder-for-anomaly-detection-for-ecg-data-5c0b07d00e50 (дата звернення: 26.11.2022);

23. Sebastian Schmidl, Phillip Wenig, Thorsten Papenbrock. Anomaly Detection in Time Series: A Comprehensive Evaluation. URL: https://www.vldb.org/pvldb/vol15/p1779-wenig.pdf (дата звернення: 26.11.2022);

24. Anomaly detection using Isolation Forest – A Complete Guide. URL: https://www.analyticsvidhya.com/blog/2021/07/anomaly-detection-using-isolation-forest-a-complete-guide/ (дата звернення: 26.11.2022).

ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО КОДУ

Файл Anomalies_detection.ipynb

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go


import os
for dirname, _, filenames in os.walk('drive/MyDrive/Colab
Notebooks/data/'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

generation1 = pd.read_csv(os.path.join(dirname, filenames[0]))
weather1 = pd.read_csv(os.path.join(dirname, filenames[-1]))
generation1['DATE_TIME'] =
pd.to_datetime(generation1['DATE_TIME'], dayfirst=True)
weather1['DATE_TIME'] = pd.to_datetime(weather1['DATE_TIME'],
dayfirst=True)

inverters = list(generation1['SOURCE_KEY'].unique())
print(f"total number of inverters {len(inverters)}")

inv_1 = generation1[generation1['SOURCE_KEY']==inverters[0]]
mask = ((weather1['DATE_TIME'] >= min(inv_1["DATE_TIME"])) &
(weather1['DATE_TIME'] <= max(inv_1["DATE_TIME"])))
weather_filtered = weather1.loc[mask]
weather_filtered.shape

fig = go.Figure()
```

```
fig.add_trace(go.Scatter(x=inv_1["DATE_TIME"],
y=inv_1["AC_POWER"],
                    mode='lines',
                    name='AC Power'))

fig.add_trace(go.Scatter(x=weather_filtered["DATE_TIME"],
y=weather_filtered["IRRADIATION"],
                    mode='lines',
                    name='Irradiation',
                    yaxis='y2'))

fig.update_layout(title_text="Irradiation vs AC POWER",
                  yaxis1=dict(title="AC Power",
                              side='left'),
                  yaxis2=dict(title="Irradiation index",
                              side='right',
                              anchor="x",
                              overlaying="y"
                             ))

fig.show()

df = inv_1.merge(weather_filtered, on="DATE_TIME", how='left')
df = df[['DATE_TIME', 'AC_POWER', 'AMBIENT_TEMPERATURE',
'MODULE_TEMPERATURE', 'IRRADIATION']]
df

from sklearn.ensemble import IsolationForest

train_prp = .6
train = df.loc[:df.shape[0]*train_prp]
test = df.loc[df.shape[0]*train_prp:]

features = ['AC_POWER', "IRRADIATION"]
```

```
clf = IsolationForest(n_estimators=1000,
                      max_samples='auto',
                      contamination=.03,
                      max_features=2,
                      bootstrap=False,
                      n_jobs=-1,
                      random_state=42,
                      verbose=0)
clf.fit(train[features])
pred = clf.predict(test[features])

test['anomaly'] = pred
anomalies = test[test['anomaly'] == -1][['AC_POWER']]
anomalies = anomalies.rename(columns={'AC_POWER':'anomalies'})
test = test.merge(anomalies, left_index=True, right_index=True,
how='left')
test

pred = clf.predict(train[features])

train['anomaly'] = pred
anomalies = train[train['anomaly'] == -1][['AC_POWER']]
anomalies = anomalies.rename(columns={'AC_POWER':'anomalies'})
train = train.merge(anomalies, left_index=True,
right_index=True, how='left')
train

fig = go.Figure()

fig.add_trace(go.Scatter(x=test["DATE_TIME"],
y=test["AC_POWER"],
                    mode='lines',
                    name='AC Power'))
```

```
fig.add_trace(go.Scatter(x=test["DATE_TIME"],
y=test["anomalies"],
                    name='Anomaly',
                    mode='markers',
                    marker=dict(color="red",
                                size=11,
                                line=dict(color="red",
                                          width=2))))

fig.update_layout(title_text="Визначення аномалій за допомогою
Isolation Forest",
                  yaxis1=dict(title="AC Power"))

fig.show()

x_min, x_max = (train['AC_POWER'].min(),
train['AC_POWER'].max())
y_min, y_max = (train['IRRADIATION'].min(),
train['IRRADIATION'].max())

xrange = np.linspace(x_min - (x_min*.2), x_max + (x_max*.2),
1000)
yrange = np.linspace(y_min - (y_min*.2), y_max + (y_max*.2),
1000)
xx, yy = np.meshgrid(xrange, yrange)
Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

n_train =
train[['AC_POWER','IRRADIATION']].loc[train['anomaly']==1]
a_train =
train[['AC_POWER','IRRADIATION']].loc[train['anomaly']==-1]
n_test =
test[['AC_POWER','IRRADIATION']].loc[test['anomaly']==1]
```

```
a_test = test[['AC_POWER','IRRADIATION']].loc[test['anomaly']==-
1]

trace_specs = [
    [n_train, 'anomaly', 'Train', 'square', 'green'],
    [a_train, 'not anomaly', 'Train', 'square', 'red'],
    [n_test, 'anomaly', 'Test', 'circle', 'blue'],
    [a_test, 'not anomaly', 'Test', 'circle', 'orange']]

fig = go.Figure(data=[
    go.Scatter(
        x=data["AC_POWER"], y=data['IRRADIATION'],
        name=f'{split} Split, Label {label}',
        mode='markers', marker_symbol=marker,
        marker=dict(color=color)
    )
    for data, label, split, marker, color in trace_specs
])



fig.add_trace(
    go.Contour(
        x=xrange,
        y=yrange,
        z=Z,
        showscale=False,
        colorscale='RdBu',
        opacity=0.4,
        name='Score',
        hoverinfo='skip'
    )
)
fig.update_layout(title="Контур Isolation Forest",
                  yaxis=dict(title="Irradiation index"),
                  xaxis=dict(title="Power in kW"))
```

```
fig.show()

df = df[["DATE_TIME", "AC_POWER", "AMBIENT_TEMPERATURE",
"MODULE_TEMPERATURE", "IRRADIATION"]]
df_timestamp = df[["DATE_TIME"]]
df_ = df[["AC_POWER", "AMBIENT_TEMPERATURE",
"MODULE_TEMPERATURE", "IRRADIATION"]]

train_prp = .6
train = df_.loc[:df_.shape[0]*train_prp]
test = df_.loc[df_.shape[0]*train_prp:]

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train = scaler.fit_transform(train)
X_test = scaler.transform(test)
X_train = X_train.reshape(X_train.shape[0], 1, X_train.shape[1])
X_test = X_test.reshape(X_test.shape[0], 1, X_test.shape[1])
print(f"X_train shape: {X_train.shape}")
print(f"X_test shape: {X_test.shape}")

from tensorflow.keras.layers import Input, Dropout, Dense, LSTM,
TimeDistributed, RepeatVector
from tensorflow.keras.models import Model
from tensorflow.keras import regularizers

def autoencoder_model(X):
    inputs = Input(shape=(X.shape[1], X.shape[2]))
    L1 = LSTM(16, activation='relu', return_sequences=True,
kernel_regularizer=regularizers.l2(0.00))(inputs)
    L2 = LSTM(4, activation='relu', return_sequences=False)(L1)
    L3 = RepeatVector(X.shape[1])(L2)
    L4 = LSTM(4, activation='relu', return_sequences=True)(L3)
    L5 = LSTM(16, activation='relu', return_sequences=True)(L4)
```

```
    output = TimeDistributed(Dense(X.shape[2]))(L5)
    model = Model(inputs=inputs, outputs=output)
    return model

model = autoencoder_model(X_train)
model.compile(optimizer='adam', loss='mae')
model.summary()

epochs = 100
batch = 10
history = model.fit(X_train, X_train, epochs=epochs,
batch_size=batch, validation_split=.2, verbose=0).history

fig = go.Figure()

fig.add_trace(go.Scatter(x=[x for x in
range(len(history['loss']))], y=history['loss'],
                    mode='lines',
                    name='функція втрат'))

fig.add_trace(go.Scatter(x=[x for x in
range(len(history['val_loss']))], y=history['val_loss'],
                    mode='lines',
                    name='валідаційна функція втрат'))

fig.update_layout(title="Помилка автокодувальника з LSTM
протягом навчання",
                  yaxis=dict(title="Функція втрат"),
                  xaxis=dict(title="Епоха"))

fig.show()

X_pred = model.predict(X_train)
X_pred = X_pred.reshape(X_pred.shape[0], X_pred.shape[2])
X_pred = scaler.inverse_transform(X_pred)
```

```
X_pred = pd.DataFrame(X_pred, columns=train.columns)

scores = pd.DataFrame()
scores['AC_train'] = train['AC_POWER']
scores["AC_predicted"] = X_pred["AC_POWER"]
scores['loss_mae'] = (scores['AC_train']-
scores['AC_predicted']).abs()

fig = go.Figure(data=[go.Histogram(x=scores['loss_mae'])])
fig.update_layout(title="Розподіл помилок",
                  xaxis=dict(title="Різниця помилок між
прогнозованими та реальними даними [AC Power]"),
                  yaxis=dict(title="Кількість прикладів даних"))
fig.show()

X_pred = model.predict(X_test)
X_pred = X_pred.reshape(X_pred.shape[0], X_pred.shape[2])
X_pred = scaler.inverse_transform(X_pred)
X_pred = pd.DataFrame(X_pred, columns=train.columns)
X_pred.index = test.index

scores = X_pred
scores['datetime'] = df_timestamp.loc[1893:]
scores['real AC'] = test['AC_POWER']
scores["loss_mae"] = (scores['real AC'] -
scores['AC_POWER']).abs()
scores['Threshold'] = 200
scores['Anomaly'] = np.where(scores["loss_mae"] >
scores["Threshold"], 1, 0)

fig = go.Figure()
fig.add_trace(go.Scatter(x=scores['datetime'],
                         y=scores['loss_mae'],
                         name="Функція втрат"))
fig.add_trace(go.Scatter(x=scores['datetime'],
```

```
                         y=scores['Threshold'],
                         name="Поріг визначення аномалії"))

fig.update_layout(title="Функція втрат у часі та поріг
аномалій",
                  xaxis=dict(title="Дата і час"),
                  yaxis=dict(title="Функція втрат"))
fig.show()

anomalies = scores[scores['Anomaly'] == 1][['real AC']]
anomalies = anomalies.rename(columns={'real AC':'anomalies'})
scores = scores.merge(anomalies, left_index=True,
right_index=True, how='left')

fig = go.Figure()

fig.add_trace(go.Scatter(x=scores["datetime"], y=scores["real
AC"],
                   mode='lines',
                   name='AC Power'))

fig.add_trace(go.Scatter(x=scores["datetime"],
y=scores["anomalies"],
                   name='аномалія',
                   mode='markers',
                   marker=dict(color="red",
                               size=11,
                               line=dict(color="red",
                                         width=2))))

fig.update_layout(title_text="Аномалії, визначені LSTM
автокодувальником")

fig.show()
```