ВМіністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп’ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«___» _____ 2023 р

Пояснювальна записка

до кваліфікаційної роботи
магістра

на тему: «**МЕТОДИ ВИЯВЛЕННЯ ВИКИДІВ У ПРОБНИХ ВИБІРКАХ ПРИ УПРАВЛІННІ ПРОЦЕСАМИ В СИСТЕМАХ ЗА СТАНОМ**»

Захищено на засіданні
Атестаційної комісії № 42
протокол № __ від __.12.2023 р.
Оцінка _____ / _____
Голова Атестаційної комісії
_____ **СКОБ Ю. О.**.
(підпис) (прізвище та ініціали)

Виконав:
студент 2 курсу, групи КУ– 61
Галузь знань: 15 – Автоматизація та приладобудування
Спеціальність: 151 – «Автоматизація та комп'ютерно-інтегровані технології»
ЛИХАЧ Олег Юрійович

Керівник: к.т.н., доцент
Бикова Тетяна Володимирівна

Рецензент: в.о. завідувача кафедри теоретичної та прикладної інформатики
Євген Меняйлов Сергійович

Харків – 2023

АНОТАЦІЯ

Кваліфікаційна робота складається зі вступу, восьми розділів, висновків, переліку використаних джерел і трьох додатків. Загальний обсяг роботи становить 72 сторінки, з яких 51 сторінок основної частини з 35 рисунками та трьома таблицями, 4 сторінок списку використаних джерел із 39-х найменувань, 3 додатків на 9 сторінках.

У роботах, присвячених вирішенню завдань виявлення викидів, відсутнє визначення щодо найкращих серед існуючих математичних моделей або методів для виявлення аномалій та не враховується метрика для визначення точності цих моделей. Таким чином, виникає потреба у виборі метрик для оцінювання правильності виявлення викидів, найкращих математичних моделей, методів та засобів реалізації інформаційної технології при виявленні аномалій у часових рядах та вибірках даних в системах автоматизованого збору даних.

В ході виконання кваліфікаційної роботи були використані методи обробки та підготовки даних, математичні моделі і методи виявлення викидів. Програмне забезпечення розроблено за допомогою мови Python. Також були використані наступні бібліотеки: scikit-learn, NumPy, Pandas, Tensorflow і інші.

У ході виконання роботи отримано: огляд методів виявлення викидів, методика обробки та підготовки вибірки та створено новий метод виявлення викидів шляхом інтеграції методу Z-score в метод Isolation Forest.

Головним напрямком використання методу являються інформаційні система моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

Ключові слова: ВИЯВЛЕННЯ ВИКИДІВ, МАШИННЕ НАВЧАННЯ, ГЛИБОКЕ НАВЧАННЯ, МЕТРИКИ ОЦІНКИ ЯКОСТІ МОДЕЛЕЙ ТА МЕТОДІВ.

ABSTRACT

The qualification work consists of an introduction, eight chapters, conclusions, a list of references and three appendices. The total volume of the work is 72 pages, including 51 pages of the main part with 35 figures and three tables, 4 pages of the list of references from 39 titles, 3 appendices on 9 pages.

In the works devoted to solving the problems of anomaly detection, there is no definition of the best among the existing mathematical models or methods for detecting anomalies and no metrics for determining the accuracy of these models are taken into account. Thus, there is a need to select metrics for assessing the correctness of emissions detection, the best mathematical models, methods and means of implementing information technology in detecting anomalies in time series and data samples in automated data collection systems.

In the course of the qualification work, methods of data processing and preparation, mathematical models and methods of anomaly detection were used. The software was developed using the Python language. The following libraries were also used: scikit-learn, NumPy, Pandas, Tensorflow and others.

In the course of the work, the following results were obtained: an overview of anomaly detection methods, methods of processing and sample preparation, and a new method for detecting anomaly by integrating the Z-score method into the Isolation Forest method.

The main area of application of the method is the information system for monitoring the level of socio-economic development of countries in the context of their digital progress.

Keywords: EMISSIONS DETECTION, MACHINE LEARNING, DEEP LEARNING, METRICS FOR ASSESSING THE QUALITY OF MODELS AND METHODS.

ЗМІСТ

ВСТУП

В сучасному світі, де обробка даних та аналіз інформації стали життєво важливими для багатьох галузей, завдання виявлення має вирішальне значення. Фахівці можуть виявити аномалії в даних, які можуть вплинути на прийняття рішень у таких сферах, як фінанси, медицина, інженерія та екологія. Вони можуть зробити це за допомогою відповідних методів і технологій. Виявлення викидів допомагає виявляти аномальні явища вчасно, запобігати небажаним наслідкам і покращувати процеси на основі інформації, яку вони отримують.

Враховуючи велику кількість даних, виявлення викидів є життєво важливим для багатьох сфер діяльності та установ. Передбачення аномалій у великих обсягах даних може допомогти запобігти фінансовим збиткам, захистити від хакерських атак, покращити якість медичних діагнозів та багато іншого. Терабайти даних створюються щодня, і важливо мати надійні та ефективні методи виявлення аномалій, щоб знайти та проаналізувати потенційні проблеми в цих даних. Зростаюча доступність великих обсягів даних і прогрес комп’ютерних технологій призвели до того, що аналіз даних є необхідним для розуміння багатьох сфер життя. Виявлення викидів є важливою частиною такого аналізу, оскільки воно дозволяє зрозуміти, коли дані мають аномальну поведінку, яка може призвести до негативних наслідків.

Будемо розглядати в якості об’єкту дослідження управління процесів в системах за їх станом, наприклад, економічних систем, заснованих на даних моніторингу контрольованих змінних стану. Результати моніторингу – це вибірки даних та часові ряди. Вибірка даних представляє собою набір значень за певній проміжок часу, які можуть суттєво змінюватись в залежності від ситуації в системі. В той час як часові ряди являють собою сукупність виміряних значень змінних, одержуваних на певних інтервалах часу, що нерозривно примикають один до одного та протягом яких значення змінних істотно не змінюються. Часові ряди, будучи дискретною моделлю контролю стану динамічних систем, зазвичай містять параметричну невизначеність, є нестаціонарними і зашумленими.

При розв’язанні задачі виявлення викидів у вибірках даних та часових рядах потрібно попередньо обробити вхідні дані та видалити пропущені значення для того, щоб можна було використати методи виявлення викидів. Після того, як вхідні дані були оброблені та пропущені значення були видалені потрібно знайти викиди (також відомі, як аномалії) і видалити їх з наборів даних та часових рядів. Це дозволить підвищити рівень достовірності інформації та покращити управління процесів в системі.

Задача виявлення викидів в результаті її декомпозиції повинна бути представлена, як послідовність вирішення взаємопов’язаних задач таких як:

- моніторинг стану системи (вибір та вимірювання значень контрольованих змінних стану системи через певні проміжки часу);
- попередня обробка даних, яка призводить до приведення даних моніторингу до виду, придатного для виявлення викидів;
- знаходження аномальних значень у часових рядах та вибірках даних моніторингу системи.

Аналіз існуючих літературних джерел показує, що при розробці математичних моделей та методів вирішення завдань виявлення викидів у вибірках даних та часових рядах виникає низка проблем:

- невизначеність вхідних даних (обмежений обсяг вибірок, наявність пропущених значень, корельованість змінних станів);
- велика розмірність множини змінних стану;
- невизначеність у виборі форми приведення вхідних даних до нормального вигляду, придатного для моделей виявлення викидів;
- невизначеність у виборі критеріїв якості математичних моделей та методів;
- невизначеність у виборі правильних результатів рішень щодо аномальних значень.

Слід зазначити, що у роботах, присвячених вирішенню завдань щодо виявлення викидів, відсутнє визначення щодо найкращої серед існуючих

математичної моделі або методу для виявлення аномалій та не враховується метрика для визначення точності цих моделей.

Розроблене на сьогоднішній день інформаційне забезпечення не дозволяє з досить високим рівнем достовірності вирішувати завдання виявлення викидів у вибірках даних та часових рядах.

Таким чином, виникає потреба у виборі метрик для оцінювання правильності виявлення викидів, найкращих математичних моделей, методів та засобів реалізації інформаційної технології при виявленні аномалій у часових рядах та вибірках даних при управлінні процесів в системах за їх станом.

В кваліфікаційній роботі розглядається новий метод виявлення викидів на основі інформаційних систем моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

РОЗДІЛ 1.

АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ МАТЕМАТИЧНИХ МОДЕЛЕЙ, МЕТОДІВ І ЗАСОБІВ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ МЕТОДУ ВИЯВЛЕННЯ ВИКИДІВ В СИСТЕМАХ МОНІТОРИНГУ РІВНЯ СОЦІАЛЬНО-ЕКОНОМІЧНОГО РОЗВИТКУ КРАЇН

1.1 Аналітичний огляд існуючих методів виявлення викидів.

1.1.1 Метрики для оцінювання якості виявлення викидів.

Метрика в загальному сенсі машинного навчання – це еталон вимірювання якості математичного методу або моделі. Існує проблема у виборі єдиної метрики для визначення якості математичних методів, моделей виявлення викидів у пробних вибірках.

В даний час є велика кількість метрик, тому розглянемо популярні метрики, які підтвердили свою популярність у використанні за довгі роки у задачах виявлення викидів.

Точність (Accuracy) – це найбільш простий та інтуїтивно зрозумілий показник продуктивності класифікатора. Він розраховується по формулі 1.1, як відношення кількості правильно передбаченого класу до загальної кількості передбачень:

$$\text{Точність} = \frac{TP+TN}{T}, \qquad (1.1)$$

де TP – кількість значень правильної класифікації позитивного класу, TN – кількість значень правильної класифікації негативного класу, T – загальна кількість передбачень.

Коли ми стикаємося з проблемою дисбалансу класів, точність є неправильною метрикою для використання. Наприклад, нехай існує 2 класи: клас

А становить 99% набору даних, а клас В – це решта 1%. Якщо прогнозувати клас А кожного випадку, буде досягнута точність 99%. За метрикою точності можна вважати, що модель працює чудово, але насправді модель працює дуже погано.

Точність (Precision) і відклик (Recall). Загалом, існує компроміс між відкликом (відсоток дійсно позитивних випадків, які були класифіковані як такі) і точністю (відсоток позитивних класифікацій, які дійсно позитивні). Ці метрики можна розрахувати за формулами 1.2 – 1.11:

$$\text{Відклик} = \frac{TP}{TP+FN}, \tag{1.2}$$

де TP – кількість значень правильної класифікації позитивного класу, FN – кількість значень неправильної класифікації негативного класу.

$$\text{Точність} = \frac{TP}{TP+FP}, \tag{1.3}$$

де TP – кількість значень правильної класифікації позитивного класу, FP – кількість значень неправильної класифікації позитивного класу.

У ситуаціях, коли потрібно виявити екземпляри класу меншості, зазвичай більш важливою метрикою є відклик, ніж точність. Але, як сказано раніше, повинен бути компроміс між цими метриками.

Метрика F. Ця метрика використовується в тих випадках, коли потрібно досягнути високого значення точності (Precision) і відклику (Recall) та отримати лише один показник якості. F1 можна розрахувати за формулою:

$$F1 = \frac{2*precision*recall}{precision+recall}, \tag{1.4}$$

де precision – точність, recall – відклик.

Метрику F1 не можна коригувати в залежності від потреб класифікації, тому виник окремий випадок метрики, відомої як F-Beta, яка має, корегувальний

параметер β. Цей параметер надає змогу корегувати значення відклику та точності в такій залежності, що чим більше значення β, то тим більша залежність метрики від відклику, і тим менша від точності. Дана метрика розраховується за формулою:

$$F_\beta = (1 + \beta^2) * \frac{precision * recall}{\beta^2 * precision + recall}, \quad (1.5)$$

де precision – точність, recall – відклик, β – параметер корегування.

Метрика Kappa. Kappa або Cohen's-Kappa схожа на точність класифікації, за винятком того, що вона нормалізується на базовій лінії випадкових шансів у наборі даних. Метрика розраховується за формулою:

$$k = \frac{p_0 - p_e}{1 - p_e}, \quad (1.6)$$

де p_0 – це дотримана угода, p_e – це очікувана угода.

Ця метрика відображає наскільки краще працює класифікатор (p_0) порівнюючи з продуктивністю класифікатора, який просто пропонує випадкові значення згідно з частотою кожного класу (p_e).

Стандартизованого способу інтерпретації його значень не існує. Ландіс і Кох [25] пропонують спосіб характеристики цінностей. За їхньою схемою, значення менше 0 вказує на відсутність згоди; у межах 0–0.20 як незначн; при 0.21–0.40 як справедливу; при 0.41–0.60 як помірну; при 0.61–0.80 як суттєву; а при 0.81–1 як майже ідеальну згоду.

Метрика ROC-AUC. Крива ROC-AUC є вимірюванням продуктивності для проблем класифікації, виявлення викидів при різних порогових налаштуваннях. ROC – це крива ймовірності, а AUC – ступінь або міра відокремленості. Ця метрика відображає, наскільки математична модель здатна розрізняти класи. Чим вища міра AUC, тим краще модель правильно прогнозує класи.

Крива ROC зображена на рис. 1.1, де відображена залежність TPR (True Positive Rate) від FPR (False Positive Rate). По осі ординат відображена міра TPR,

по осі абсцис — FPR. TPR – це частка правильних прогнозів у прогнозах позитивного класу. Ця міра розраховується за формулою:

$$TPR = \frac{TP}{TP+FN}, \quad (1.7)$$

де TP – кількість значень правильної класифікації позитивного класу, FN – кількість значень неправильної класифікації негативного класу.

FPR – це частка неправильних прогнозів у прогнозах позитивного класу. Ця міра розраховується за формулою :

$$FPR = \frac{FP}{FP+TN}, \quad (1.8)$$

де FP – кількість значень неправильної класифікації позитивного класу, TN – кількість значень правильної класифікації негативного класу.

Площа під кривою ROC визначає значення метрики ROC-AUC. Для відмінної моделі значення цієї метрики буде дорівнювати одиниці. Діагональна пряма, яка відображена на рис. 1.1 вказує на модель, яка не має змоги відрізнити позитивний клас від негативного, тобто не має можливості розділення класів.

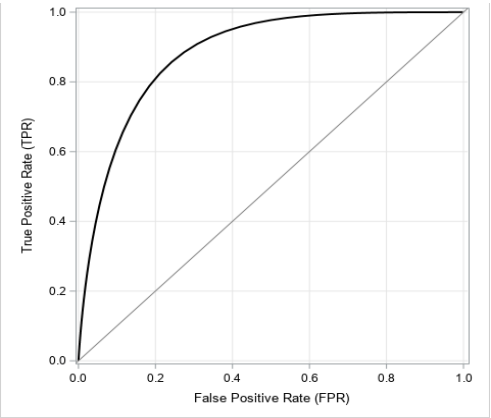

Рис. 1.1 – Крива ROC

Цю метрику слід використовувати, коли модель повинна працювати однаково добре, як на позитивному, так і на негативному класі.

Метрика PR-AUC. Ця метрика розраховується так само, як ROC-AUC. Відмінною рисою є те, що крива будується на основі точності (Precision) і відклику (Recall). Крива PR зображена на рис. 1.2, де відображена залежність точності від відклику. По осі ординат відображена міра точності, по осі абсцис — міра відкликуіПлоща під кривою PR визначає значення метрики PR-AUC. Для відмінної моделі значення цієї метрики буде дорівнювати одиниці.

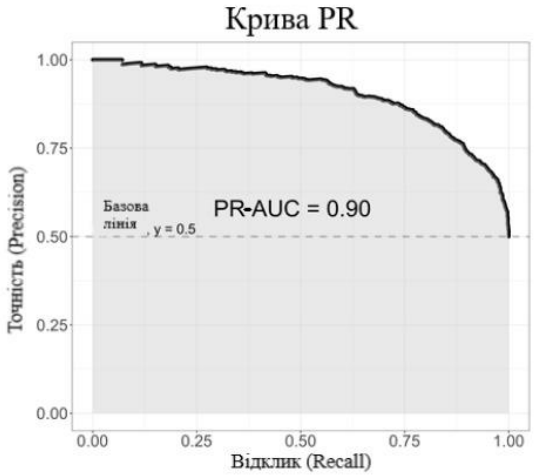

Рис. 1.2 – Крива PR

Але не всі значення від 0 до 1 є досяжними для PR-AUC. Залежно від даних, базовою лінією кривої PR є горизонтальна лінія, яка дорівнює значенню точності (Precision) — найменше значення точності. Коли поріг стає 0 (тобто крайня права точка на графіку), всі зразки класифікуються як позитивні. Тобто кількість зразків позитивного класу дорівнює кількості позитивних зразків. Таким чином, значення базової лінії зменшується, коли дані стають більш незбалансованими.

Цю метрику слід використовувати, коли модель повинна відмінно визначати або позитивні класи, або негативні. Тобто за допомогою PR-AUC можна сфокусуватися на правильному прогнозуванні одного із класів.

Метрика pAUC (Partial AUC). Часткова AUC була запропонована як альтернативний захід до стандартної AUC. При використанні часткової AUC враховується лише конкретна область простору ROC. Приклад кривої pAUC відображено на рис. 1.3, де pAUC розраховується для затіненої області.

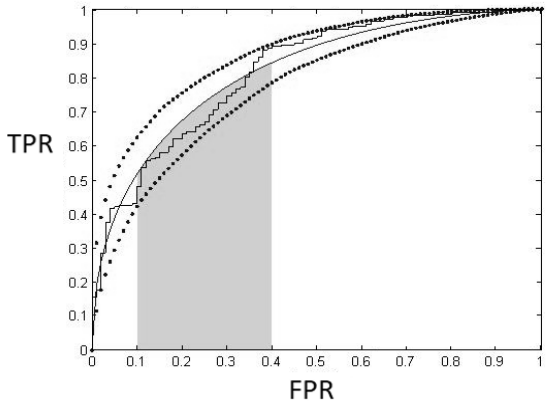

Рис. 1.3 – Крива pAUC

Метрика двосторонній pAUC (Partial AUC). На відміну від pAUC, замість обмеження лише частоти помилкових позитивних результатів (FPR), двосторонній pAUC фокусується на частковій площі під кривою з обмеженнями як по горизонталі, так і по вертикалі.

Приклад кривої двостороннього pAUC відображено на рис.1.4. Двосторонній pAUC позначає площу заштрихованої області A. Ця заштрихована область безпосередньо визначається явною верхньою межею FPR (p0 = 0,5) і нижньою межею TPR (q0 = 0,65). На відміну від цього, pAUC позначає площу обох регіонів A і B.

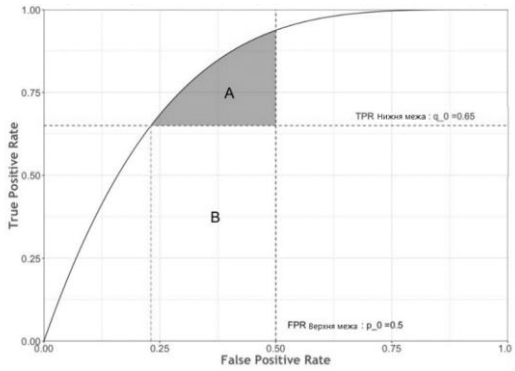

Рис. 1.4 – Крива двостороннього pAUC

Метрика LogLoss. Вона є однією з найважливіших класифікаційних метрик на основі ймовірностей. Ця метрика розраховується за формулою 1.9. LogLoss вказує на те, наскільки ймовірність прогнозу близька до відповідного

фактичного/істинного значення. Та чим більше прогнозована ймовірність відрізняється від фактичного значення, тим вище значення логарифмічних втрат.

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i)], \qquad (1.9)$$

де y_i – справжній клас, p_i – вірогідність того, що y_i вказує на позитивний клас, n – загальна кількість спостережень із вибірки.

До недоліків цієї метрики можна віднести те, що при незбалансованості класів, мажоритарний клас може домінувати над LogLoss.

Проаналізувавши усі метрики, які використовуються при оцінці якості методів (моделей) виявлення викидів, можна дійти висновку, що найкращою є метрика PR-AUC. Ця метрика була обрана найкращою, через те, що вона фокусується на малих позитивних класах, в нашому випадку викидах, та дозволяє об’єктивно оцінити якість математичних моделей та методів.

1.1.2 Математичні моделі і методи виявлення викидів.

Велика увага приділяється розгляду завдань теорії та практики управління процесів у динамічних системах, як науковцями в Україні, так і за її межами. На цей час опубліковано безліч робіт присвячених опису математичних моделей та методів виявлення викидів у процесах управління систем за станом [1-15].

Розглядаючи класифікацію методів (моделей) виявлення викидів в системах, реально визначити, що їх можна поділити на традиційні методи та методи глибокого навчання. Класифікація цих моделей та методів виявлення викидів представлена на рис. 1.5-1.6.

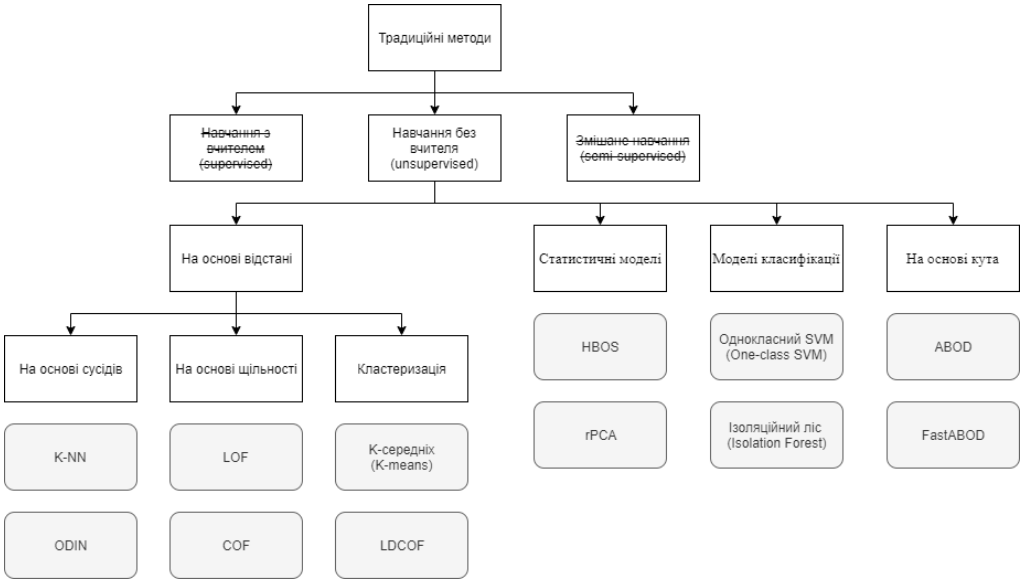

Рис. 1.5 – Класифікація моделей та методів виявлення викидів

Можна виділити вищу ланку ієрархії моделей та методів виявлення викидів – це традиційні моделі та методи глибокого навчання (рис. 1.6). Серед них можна виділити чотири основних типи моделей: моделі на основі відстані, статистичні моделі, моделі класифікації та моделі на основі кута. Аналізуючи типи моделей для глибокого навчання можна виділити три типи навчання: глибоке навчання для вилучення функцій; навчання особливостей представлення нормальності; та наскрізне навчання для визначення викидів.

До множини методів, моделей на основі відстані відносяться наступні: K-NN[16], ODIN [17], LOF [18] та K-means [19]. Серед статистичних моделей можна виділити: HBOS [20] та rPCA [21]. Прикладами моделей класифікації для виявлення викидів є: SVM [22] та Isolation Forest [23]. До методів, моделей побудованих на основі кута можна віднести ABOD [24] та FastABOD.

До множини методів, моделей на основі відстані відносяться наступні: K-NN[16], ODIN [17], LOF [18] та K-means [19]. Серед статистичних моделей можна виділити: HBOS [20] та rPCA [21]. Прикладами моделей класифікації для виявлення викидів є: SVM [22] та Isolation Forest [23]. До методів, моделей побудованих на основі кута, можна віднести ABOD [24] та FastABOD.

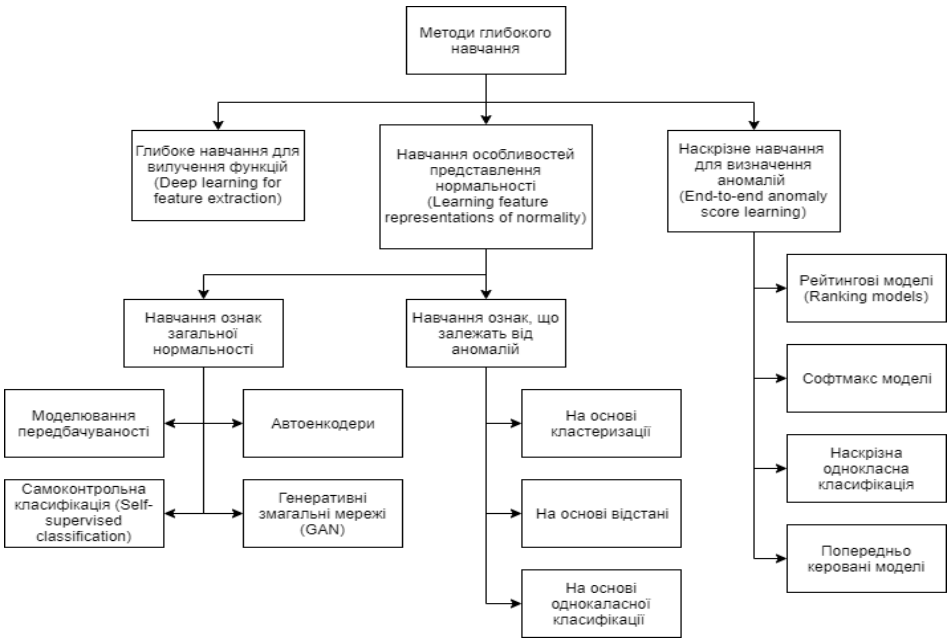

Рис. 1.6 – Класифікація методів глибокого навчання для виявлення викидів

1.1.3 Традиційні методи виявлення викидів

Серед традиційних методів можна виділити такі як:

1. Z-оцінка (стандартна оцінка спостереження) – це індикатор, що визначає положення вихідної оцінки з погляду її відстані від середнього значення при вимірі в одиницях стандартного відхилення, при умові гауссівського розподілу.

Метод дозволяє побачити, наскільки вище або нижче середнього знаходиться це значення на кривій розподілу (рис. 1.7).

Це робить z-оцінку параметричним методом. Інколи точки даних не описуються розподілом Гаусса. Ця проблема може бути вирішена шляхом застосування перетворень до даних, таких як масштабування.

Під час обчислення z-оцінки для кожної вибірки в наборі даних необхідно вказати поріг. Аналізуючи точки даних, які лежать за певним порогом, можна визначити, чи відносяться вони до аномальних.

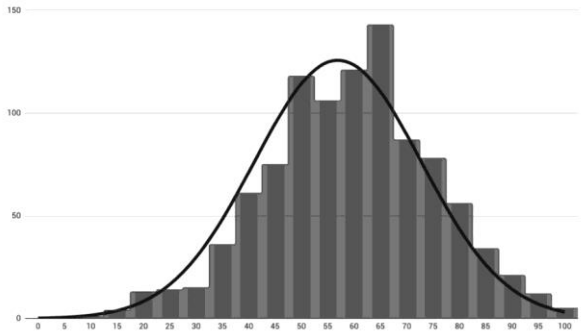

Рис. 1.7 – Приклад роботи методу Z-оцінка [26]

Z-оцінку для будь-якої точки даних можна розрахувати за формулою:

$$z = \frac{x-\mu}{\sigma}, \qquad (1.10)$$

де x – вхідний показник, μ - середнє значення набору даних, σ – стандартне відхилення для набору даних.

Даний метод ефективний та простий для виявлення викидів в наборі даних з параметричними розподілами в малорозмірному просторі об’єктів.

Визначимо переваги та недоліки методу Z-оцінки. До переваг можна віднести наступні:

- ефективний метод, щоб описати значення у просторі ознак за допомогою розподілу Гауса.
- проста реалізація.

Серед недоліків можна виділити наступні:

- метод ефективний у просторі об'єктів низької розмірності;
- якщо розподіли не можна вважати параметричними, метод працює не точно.

2. DBSCAN — це метод кластеризації на основі щільності. Цей метод зосереджений на пошуку сусідів за щільністю на n-вимірній сфері з радіусом ϵ (рис.1.8.). Кластер можна визначити як максимальний набір точок, пов’язаних із щільністю в просторі ознак.

Це ефективний метод для роботи з наборами даних середнього розміру. DBSCAN самостійно оцінює кількість кластерів; немає потреби вказувати кількість бажаних кластерів; це модель машинного навчання без вчителя.

Dbscan визначає такі класи точок як:

- основна точка: A є основною точкою, якщо її околиця (визначена ε) містить принаймні стільки ж, або більше точок, ніж параметр MinPts(мінімальна кількість спостережень у кластері);
- гранична точка: C — це гранична точка, яка лежить у кластері, і її околиці не містять більше точок, ніж MinPts, але вона все ще є досяжною для щільності іншими точками в кластері;
- викид: N — це точка викиду, яка не лежить у жодному кластері, і вона не пов’язана з будь-якою іншою точкою. Таким чином, ця точка матиме власний кластер [27].

Визначимо переваги та недоліки даного методу. До переваг можна віднести наступні:

- точно працює, якщо простір ознак пошуку викидів є багатовимірним;
- легка візуалізація результатів.

Серед недоліків можна виділити наступні:

- значення у просторі функцій необхідно відповідно масштабувати;
- це неконтрольована модель, і її необхідно повторно калібрувати щоразу, коли аналізується новий пакет даних.

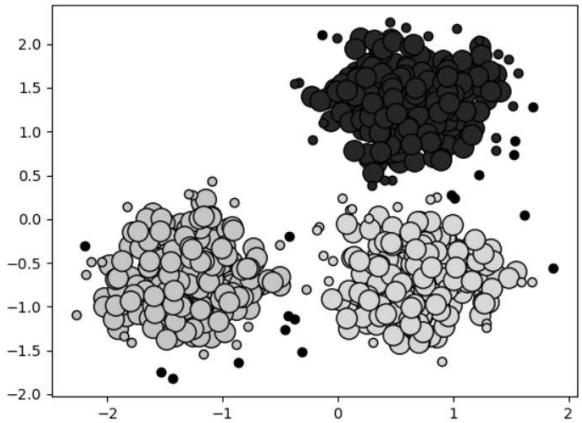

Рис. 1.8 – Приклад роботи методу DBSCAN[28]

3. Ізоляційний ліс – ефективний метод для виявлення викидів (аномальних значень) в наборах даних. Цей метод заснований на бінарних деревах рішень. Основний принцип ізоляційного лісу полягає в тому, що викиди є нечисленними і далекими від решти спостережень.

Щоб побудувати дерево, алгоритм випадковим чином вибирає об’єкт із простору ознак і випадкове розділене значення в діапазоні між максимумами та мінімумами. Це робиться для всіх спостережень у навчальному наборі. Для побудови лісу складається ансамбль дерев із усередненням усіх дерев у лісі.

Даний метод порівнює спостереження зі значенням розщеплення у вузлі; цей вузол матиме два дочірніх вузла, на яких буде зроблено ще одне випадкове порівняння. Кількість розщеплень, зроблених алгоритмом для екземпляра, називається довжиною шляху. Викиди матимуть коротшу довжину шляху, ніж решта спостережень.

Оцінку аномалії можна обчислити за формулою:

$$S(x,n) = 2^{-\frac{E(h(x))}{e(n)}}, \qquad (1.11)$$

де E(h(x)) – середня довжина шляху вибірки, C(n) –невдалий пошук довжини, N – кількість зовнішніх вузлів.

Розглянемо переваги та недоліки методу. До переваг можна віднести відсутність необхідності масштабувати значення у просторі функцій.

Серед недоліків можна виділити наступні:

- складна візуалізація результатів;
- при неправильній оптимізації, для великого набора даних, довгий час навчання.

4. Local Outlier Factor (LOF) — метод машинного навчання без вчителя для виявлення аномалії, який обчислює локальне відхилення щільності точки щодо її сусідів. Даний метод вважає викидами зразки, які мають значно нижчу щільність, ніж їхні сусіди.

Кількість сусідів, що розглядаються, зазвичай встановлюється:

- більше, ніж мінімальна кількість вибірок, яку повинен містити кластер, тому інші вибірки можуть бути локальними викидами щодо цього кластера;
- менше, ніж максимальна кількість близьких сусідів за вибірками, які потенційно можуть бути локальними викидами [29].

LOF дає кращі результати, аніж глобальний підхід до пошуку викидів. Оскільки граничного значення LOF немає, вибір точки, як викиду залежить від користувача.

Розглянемо переваги та недоліки методу. До переваг можна віднести те, що точка буде вважатися викидом, якщо вона знаходиться на невеликій відстані від надзвичайно щільного скупчення, глобальний підхід може не розглядати цей момент як вибій. Але LOF може ефективно ідентифікувати локальні викиди.

Серед недоліків можна виділити наступні:

- немає певного порогового значення, вище якого точка визначається як викид;
- ідентифікація викиду залежить від проблеми та користувача.

1.1.4 Методи глибокого навчання для виявлення викидів.

Серед методів глибокого навчання можна виділити наступні.

1. Автоенкодери - це нейронні мережі, призначені для вивчення низькорозмірного уявлення за деякими вхідними даними (рис. 1.9). Вони складаються з двох компонентів: кодувальника, який вчиться відображати вхідні дані в низькорозмірне уявлення (так зване вузьке місце); і декодера, який вчиться відображати це низькорозмірне уявлення назад у вхідні дані. Структуруючи проблему навчання таким чином, мережа кодувальника вивчає ефективну функцію стиснення, яка відображає вхідні дані в помітне уявлення нижчої розмірності, тому мережа декодера може успішно відновлювати вхідні дані.

Модель навчається шляхом мінімізації помилки відновлення, яка є різницею (середньоквадратичною помилкою) між вихідним введенням і відновленим висновком, створеним декодером [30].

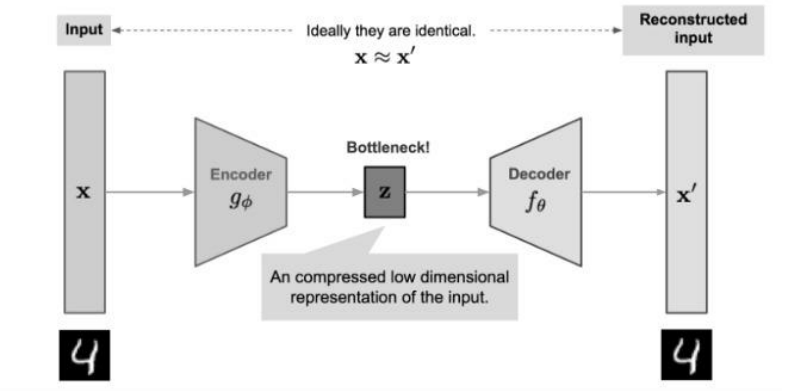

Рис. 1.9 – Архітектура AutoEncoder[31]

Застосування автоенкодера для виявлення аномалій слідує загальному принципу: спочатку потрібно змоделювати нормальну поведінку, а потім генерувати оцінку аномалії кожної нової вибірки даних. Щоб змоделювати нормальну поведінку, слід використовувати напівконтрольований підхід, тобто навчання моделі проходить на нормальних вибірках даних. Таким чином модель вивчає функцію відображення, яка успішно відновлює нормальні вибірки даних з дуже невеликою помилкою відновлення. Така поведінка відтворюється під час тестування, коли помилка відновлення мала для нормальних вибірок даних та велика для аномальних вибірок даних.

Щоб ідентифікувати аномалії, використовується обробка помилки відновлення як оцінка аномалії; після цього можливо вибрати зразки з помилками відновлення, що перевищують заданий поріг.

Розглянемо переваги та недоліки методу. До переваг можна віднести наступні:

- компактність та швидкість кодування;
- зменшення розмірності даних, що прискорює навчання моделі.

Серед недоліків можна виділити такі як:

- складна візуалізація результатів;
- складне навчання.

2. Варіаційний автоенкодер (VAE) – це розширення автоенкодера (рис. 1.10). Подібно до автоенкодера, він складається з кодувальника та мережевого компонента декодера, але він включає важливі зміни в структурі завдання навчання, щоб пристосуватися до варіаційного висновку.

На відміну від навчання зіставлення вхідних даних з фіксованим вектором вузького місця (точкова оцінка), VAE вивчає зіставлення вхідних даних з розподілом і вчиться відновлювати вихідні дані шляхом вибірки цього розподілу з використанням прихованого коду.

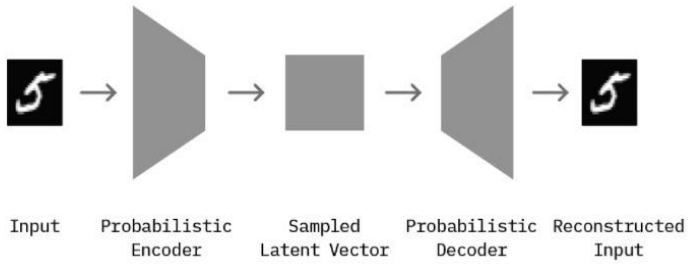

Рис. 1.10 – Модель варіаційного автоенкодера [31]

Модель VAE навчається шляхом мінімізації різниці між розрахунковим розподілом, створеним моделлю, та реальним розподілом даних. Ця різниця оцінюється за допомогою дивергенції Кульбака-Лейблера, яка кількісно визначає відстань між двома розподілами, вимірюючи, скільки інформації втрачається, коли один розподіл використовується для подання іншого.

Використання VAE для виявлення аномалій: подібно автоенкодеру, навчання VAE починається на нормальних вибірках даних. Під час тестування можна визначити оцінку аномалії двома способами. По-перше, вилучити зразки прихованого коду з кодера з урахуванням наших вхідних даних, відстежити відновлені значення з декодера та обчислити середню помилку відновлення. Аномалії позначаються з урахуванням порогу помилки відновлення.

В якості альтернативи потрібно вивести середнє значення та параметр дисперсії з декодера та обчислити ймовірність того, що нова точка даних належить розподілу нормальних даних, на якому була навчена модель. Якщо

точка даних знаходиться в області низької щільності (нижче деякого порога), це позначається як аномалія (рис. 1.11).

Розглянемо переваги та недоліки методу. До переваг можна віднести наступні:

- чіткий спосіб оцінки якості моделі (логарифмічна ймовірність);
- легка візуалізація.

Серед недоліків можна виділити такі як:

- неоптимальні фактори варіації;
- може мати градієнти високої дисперсії.

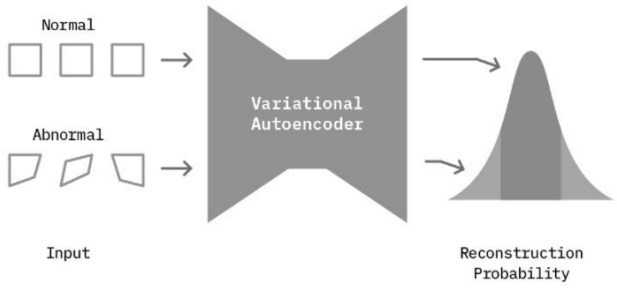

Рис.1.11 – Оцінка аномалій за допомогою VAE. [31]

3. Генеративні змагальні мережі (GAN) – це нейронні мережі, розроблені для вивчення генеративної моделі розподілу вхідних даних (рис. 1.12). У класичному формулюванні вони складаються з пари (зазвичай з прямим зв'язком) нейронних мереж, а саме з генератора G і дискримінатора D. Обидві мережі навчаються спільно з кінцевою метою визначити розподіл вихідних даних X.

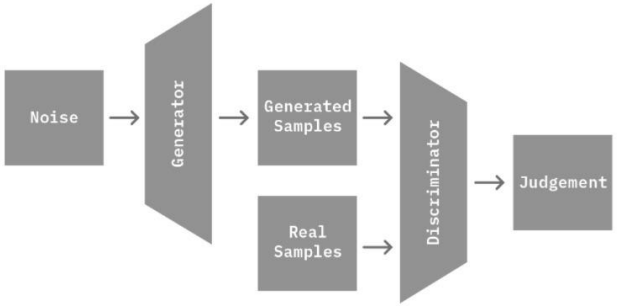

Рис. 1.12 – Класична модель GAN [31]

Щоб ідентифікувати аномалії, використовується напівкерований підхід, тобто потрібно навчати модель від послідовності до послідовності на нормальних даних. Моделі послідовність-послідовність (sequence-to-sequence) - це клас нейронних мереж, в основному призначений для вивчення зіставлень між даними, які найкраще представлені у вигляді послідовностей. Потім під час тестування можна порівняти різницю (середньоквадратичну помилку) між вихідною послідовністю, згенерованою моделлю, та її вхідними даними. Це значення використовується як показник аномалій.

Розглянемо переваги та недоліки методу. Недоліком генеративних змагальних мереж є складність оцінки моделі. До переваг можна віднести наступні:

- GAN генеруює дані, схожі вхідні;
- GAN деталізує дані та може легко інтерпретуватися у різні версії.

1.2 Аналітичний огляд існуючих програмних засобів для виявлення викидів.

Розглянемо існуючі програмні засоби для задач виявлення кикидів. Бібліотека scikit-learn в Python реалізує різні алгоритми, методи і математичні моделі машинного навчання, включаючи і виявлення викидів.

Scikit-learn – це бібліотека для машинного навчання в мові програмування Python, яка містить різні алгоритми та методи для виявлення викидів в даних.

Цей алгоритм передбачає, що звичайні дані надходять із відомого розподілу, такого як розподіл Гауса. Для виявлення викидів Scikit-learn надає об’єкт під назвою covariance.EllipticEnvelop .

Цей об’єкт адаптує надійну оцінку коваріації до даних і, таким чином, адаптує еліпс до центральних точок даних. Він ігнорує точки за межами центрального режиму.[32]

Перейдемо до PyOD (Python Outlier Detection) бібліотеки. PyOD працює як з Python 2, так і з Python 3, а також покладається на дуже часто використовувані бібліотеки NumPy, SciPi та scikit-learn для виконання своїх основних функцій.

PyOD має багато алгоритмів, таких як Isolation Forest, One-Class SVM, k-Nearest Neighbors, і AutoEncoder для виявлення викидів. PyOD надає доступ до своєї колекції алгоритмів виявлення викидів за допомогою серії простих у використанні уніфікованих API. Кожен API постачається з повною детальною документацією з інструкціями та афілійованими прикладами, щоб візуалізувати, як розпочати впровадження кожного різного типу алгоритму.[33]

TensorFlow та Keras дозволяють створювати нейронні мережі для виявлення викидів, включаючи Autoencoders і Variational Autoencoders (VAE). Ці інструменти можуть бути корисними для виявлення складних аномалій у великих обсягах даних.

XGBoost і LightGBM - це бібліотеки для градієнтного бустінгу, які також можна використовувати для виявлення викидів.

1.3 Постановка задачі дослідження

Об'єктом дослідження даної кваліфікаційної роботи є інформаційна система моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

Предметом дослідження даної кваліфікаційної роботи є методи і моделі машинного моделювання рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу на основі впровадження програмних засобів в системи моніторингу

Мета даної кваліфікаційної роботи – підвищення рівня якості виявлення викидів за допомогою вибору найефективнішої метрики для виявлення викидів та покращення попередньої обробки даних на основі критеріїв в інформаційних системах моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

Для досягнення мети потрібно виконати наступний перелік сформованих задач, таких як: визначення постановки задачі виявлення викидів; аналітичний огляд існуючих методів виявлення викидів; аналітичний огляд існуючих програмних засобів для виявлення викидів; формалізація уявлення вхідних

даних; підготовка та обробка вхідних даних для покращення ефективності методу; опис методу виявлення викидів; опис обчислювального алгоритму вирішення задачі; порівняння ефективності різних методів виявлення викидів; огляд та аналіз отриманих результатів.

Отже дана робота присвячена розробці ефективного методу виявлення викидів на основі інформаційних критеріїв в системах моніторингу використовуючи програмні засоби, які мають високі показники точності та можуть використовуватися для моніторингу систем різних галузей та напрямків.

Висновки до розділу 1

В підрозділі 1.1 були розглянуті метрики виявлення викидів шляхом аналізу науково-технічної літератури та інформації, знайденої в Інтернеті. Визначено, що найкращою є метрика PR-AUC. Ця метрика була обрана найкращою, через те що вона фокусується на малих позитивних класах - в нашому випадку викидах, та дозволяє об’єктивно оцінити якість математичних моделей та методів.

Програмні засоби для вирішення задач виявлення викидів, доступні для використання в прикладних проектах, розглядаються в підрозділі 1.2.

В підрозділі 1.3 була визначена постановка задачі дослідження. Були описані предмет, об’єкт, мета та задачі, які потрібно розробити для досягнення мети дослідження. Також була сформульована постановка задачі виявлення викидів.

РОЗДІЛ 2.

ОПИС ВХІДНИХ ДАНИХ ТА ЇХ ОБРОБКИ

2.1 Формалізація, підготовка та обробка вхідних даних

Попередня обробка даних — це підготовка необроблених даних для моделі машинного навчання. Тому попередня обробка даних є найважливішим кроком під час створення моделі машинного навчання.

У отриманих даних, завжди є шум і відсутні значення. Це трапляється через несподівані події, технічні проблеми чи низку інших перешкод. Алгоритми не можуть використовувати неповні та шумні дані, оскільки зазвичай вони не призначені для обробки відсутніх значень, а шум спричиняє порушення справжньої моделі вибірки. Попередня обробка даних спрямована на вирішення цих проблем шляхом ретельної обробки наявних даних.

У сфері штучного інтелекту та машинного навчання файли CSV зазвичай використовуються для зберігання наборів даних, які використовуватимуться для навчання моделей машинного навчання. Дані зазвичай організовані в рядки та стовпці, де кожен стовпець представляє змінну або функцію набору даних, а кожен рядок представляє екземпляр або приклад. Файли CSV можна відкривати та працювати з ними за допомогою різноманітних програм і мов програмування, що робить їх дуже популярними для обміну даними та спільної роботи в проектах штучного інтелекту та машинного навчання.

Етапи попередньої обробки даних можна спростити за допомогою інструментів і бібліотек, які полегшують керування процесом і його виконання. Без певних бібліотек розробка та оптимізація одного лінійного рішення може зайняти години кодування. Попередня обробка даних за допомогою Python: Python — це мова програмування, яка підтримує незліченні бібліотеки з відкритим кодом, які можуть обчислювати складні операції за допомогою одного рядка коду. Наприклад, для інтелектуального імпутування відсутніх значень потрібно лише використовувати пакет бібліотеки імпутацій scikit learn. Або щоб масштабувати набори даних, можна просто викликати функцію MinMaxScaler із

пакета бібліотеки попередньої обробки. У пакеті бібліотеки попередньої обробки є незліченна кількість функцій попередньої обробки даних.

Autumunge — чудовий інструмент, створений як платформа бібліотеки Python, який готує табличні дані для прямого застосування алгоритмів машинного навчання.

Попередня обробка даних за допомогою R: R — це структура, яка здебільшого використовується для дослідницьких/академічних цілей. Він має кілька пакетів, які схожі на бібліотеки Python і добре підтримують етапи попередньої обробки даних.

Попередня обробка даних за допомогою Weka: Weka — це програмне забезпечення, яке підтримує інтелектуальний аналіз даних і попередню обробку даних за допомогою вбудованих інструментів попередньої обробки даних і моделей машинного навчання для інтелектуального майнінгу.

Попередня обробка даних за допомогою RapidMiner: Подібно до Weka, RapidMiner — це програмне забезпечення з відкритим вихідним кодом, яке має різні ефективні інструменти для підтримки попередньої обробки даних.

Після того, як дані зібрано, їх потрібно вивчити або оцінити, щоб виявити ключові тенденції та невідповідності. Основними цілями оцінки якості даних є:

1. Отримати огляд даних: зрозуміти формати даних і загальну структуру, у якій дані зберігаються. Необхідно знайти такі властивості даних, як середнє значення, медіана, стандартні квантилі та стандартне відхилення. Ці деталі можуть допомогти виявити порушення в даних.

2. Визначити відсутні дані: відсутні дані є звичайним явищем у більшості реальних наборів даних. Це може порушити справжні шаблони даних і навіть призвести до більшої втрати даних, коли цілі рядки та стовпці видаляються через відсутність кількох клітинок у наборі даних.

3. Усунути невідповідності: як і відсутні значення, дані реального світу також мають численні невідповідності, як-от неправильне написання, неправильно заповнені стовпці та рядки (наприклад: зарплата, заповнена в стовпці статі), дубльовані дані та багато іншого. Іноді ці невідповідності можна

вирішити за допомогою автоматизації, але найчастіше вони потребують ручної перевірки.

Розглянемо кілька популярних методів попередньої обробки даних.

1. Обробка відсутніх значень

Нам потрібно обробити ці відсутні значення, щоб оптимально використовувати доступні дані. Розглянемо декілька способів:

- Видалити зразки з відсутніми значеннями: це корисно, коли велика кількість зразків і кількість відсутніх значень в одному рядку/зразку. Це не рекомендоване рішення для інших випадків, оскільки це призводить до великої втрати даних.
- Заміна відсутніх нулем: іноді ця техніка працює для базових наборів даних, оскільки дані, про які йдеться, припускають нуль як базове число, що означає відсутність значення. Однак у більшості випадків нуль сам по собі може позначати значення. Подібним чином, у більшості випадків, якщо відсутні значення заповнюються 0, це введе в оману модель. Нуль можна використовувати як заміну лише тоді, коли набір даних не залежить від його ефекту.
- Заміна відсутнього значення середнім значенням, медіаною або модою: ми можемо вирішити вищезазначену проблему, яка виникає внаслідок неправильного використання 0, використовуючи статистичні функції, такі як середнє, медіана або мода, як заміну відсутніх значень. Незважаючи на те, що вони також є припущеннями, ці значення мають більший сенс і є ближчими наближеннями в порівнянні з одним значенням, наприклад нулем.
- Інтерполяція відсутніх значень: інтерполяція допомагає генерувати значення в межах діапазону на основі заданого розміру кроку. Наприклад, якщо в стовпці між клітинками зі значеннями 0 і 10 є 9 пропущених значень, інтерполяція заповнить пропущені клітинки числами від 1 до 9. Зрозуміло, що набір даних потрібно відсортувати за більш надійною змінною.
- Екстраполяція відсутніх значень: екстраполяція допомагає заповнити значення, які виходять за межі заданого діапазону, як-от екстремальні

значення функції. Екстраполяція використовує іншу змінну (зазвичай цільову змінну), щоб порівняти відповідну змінну та заповнити її керованим посиланням.

- Створення моделі з іншими значеннями для передбачення відсутніх значень: на сьогоднішній день це найбільш інтуїтивно зрозумілий з усіх методів, які описано вище. Тут алгоритм вивчає всі змінні, крім фактичної цільової змінної (оскільки це призведе до витоку даних). Цільова змінна для цього алгоритму стає функцією з відсутніми значеннями. Модель, якщо вона добре навчена, може передбачити відсутні точки та забезпечити найближче наближення.

2. Масштабування.

Різні стовпці можуть бути присутніми в різних діапазонах. Наприклад, може бути один стовпець з одиницею відстані, а інший з одиницею валюти. Ці два стовпці матимуть різко різні діапазони, через що будь-якій моделі машинного навчання буде важко досягти оптимального стану обчислень.

У більш технічних термінах, якщо розглядати використання градієнтного спуску, алгоритму градієнтного спуску знадобиться більше часу, щоб зблизитися, оскільки він має обробляти різні діапазони, які знаходяться далеко один від одного.

Деякі популярні методи масштабування:

- Мінімально-максимальний масштабувальник: мінімально-максимальний масштабувальник зменшує значення функції між будь-яким діапазоном вибору. Наприклад, від 0 до 5.
- Стандартний масштабувальник: стандартний масштабувальник припускає, що змінна має нормальний розподіл, а потім зменшує її таким чином, щоб стандартне відхилення дорівнювало 1, а розподіл було зосереджено на 0.
- Надійний масштабувальник: надійний масштабувальник працює найкраще, коли в наборі даних є викиди. Він масштабує дані щодо міжквартильного діапазону після видалення медіани.
- Max-Abs Scaler: подібний до min-max масштабувальника, але замість заданого діапазону функція масштабується до максимального абсолютного

значення. Розрідженість даних зберігається, оскільки вони не центруються.

3. Кореляція.

Кореляція — це метод однофакторного аналізу. Він виявляє лінійні залежності між двома змінними. Одним з головних недоліків кореляції є те, що вона не може належним чином охопити нелінійні зв’язки, навіть сильні. Зображення нижче демонструють переваги та недоліки кореляції:

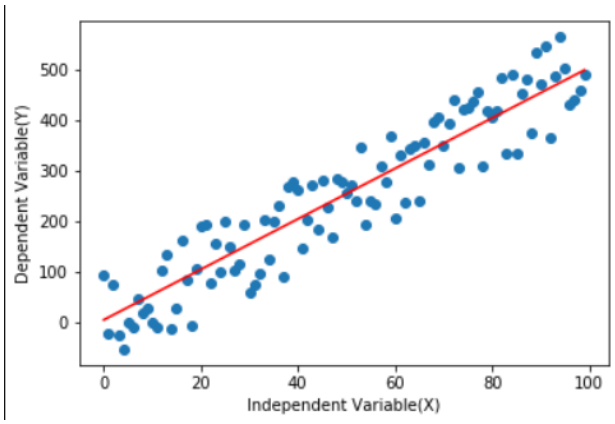

Рис. 2.1 – Лінійний зв’язок

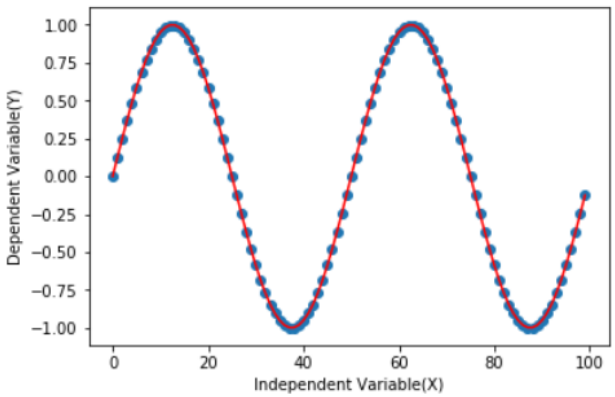

Рис. 2.2 – Синусне співвідношення

4. Зменшення розмірності

Для представлення даних вибірки зазвичай використовуються методи зменшення розмірності. Ці методи зазвичай використовуються для навчання математичних моделей і методів шляхом зменшення розмірності даних. PCA (Principal Component Analysis), T-SNE (t-distributed Stochastic Neighbor

Embedding) і SVM (Support Vector Machines) є найбільш відомими традиційними методами. Автоенкодери є популярним методом глибокого навчання для зменшення розмірності.[34]

Використовуючи вищенаведені пункти проведено попередню обробку даних.

Для вибірки рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу розглянемо процес обробки даних. Першим кроком було зроблено завантаження даних з файлу csv, перевірка їх правильності та видалення непотрібних змінних стану. Після цього етапу вибірка була отримана для подальшої обробки. Рис. 2.3 показує вибірку.

Country Name	EGI_2020	EPI_2020	OSI_2020	HCI_2020	TII_2020	EGI_2018	EPI_2018	OSI_2018	HCI_2018	TII_2018	...	ICT_16	ICT_15	ICT_13	ICT_12	SPI_20	SPI_18	SPI_17	SPI_
Albania	73.99	84.52	84.12	80.01	57.85	65.19	75.84	73.61	78.77	43.18	...	4.90	4.73	4.62	4.42	75.41	71.77	71.65	70.
Algeria	51.73	15.48	27.65	69.66	57.87	42.27	20.22	21.53	66.40	38.89	...	4.32	3.71	4.46	4.22	69.92	66.83	66.83	65.
Argentina	82.79	85.71	84.71	91.00	72.65	73.35	62.36	75.00	85.79	59.27	...	6.68	6.40	6.62	6.38	80.66	74.98	74.84	74.
Armenia	71.36	75.00	70.00	78.72	65.36	59.44	56.74	56.25	75.47	46.60	...	5.56	5.32	5.64	5.55	76.46	70.87	70.53	69.
Australia	94.32	96.43	94.71	100.00	88.25	90.53	98.31	97.22	100.00	74.36	...	8.08	8.29	8.23	8.14	91.29	88.32	88.23	87.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
United States of America	92.97	100.00	94.71	92.39	91.82	87.69	98.31	98.61	88.83	75.64	...	8.13	8.19	7.78	7.75	85.71	84.78	85.27	86.
Uruguay	85.00	85.71	84.12	85.14	85.74	78.58	91.57	88.89	77.19	69.67	...	6.75	6.70	7.05	6.80	82.99	79.40	NaN	N.
Viet Nam	66.67	70.24	65.29	67.79	66.94	59.31	69.10	73.61	65.43	38.90	...	4.18	4.28	4.48	4.39	68.85	NaN	NaN	N.
Zambia	42.42	30.95	25.88	67.45	33.94	41.11	39.89	47.92	56.89	18.53	...	2.19	2.04	2.68	2.63	55.34	NaN	NaN	N.
Zimbabwe	50.19	45.24	52.35	61.35	36.88	36.92	27.53	32.64	56.68	21.44	...	2.85	2.90	3.12	2.99	52.26	45.26	43.76	42.

Рис. 2.3 – Завантажена вибірка

Так як цільова змінна категоріальна і представлена рядком, щоб перевести цю змінну у числовий формат, потрібно буде виконати кодування. Кодування категоріальних змінних було виконано за допомогою спеціальної функції бібліотеки scikit-learn. Таким чином, цільова змінна була переведена в числовий формат, який містив значення від 0 до 3. Наступним кроком було визначити, як змінні стану корелюють один з одним. Було створено дві матриці кореляції: загальна та зі збільшенням змінних стану, які були найбільш кореляційними. Рис. 2.4 і 2.5 показують ці два приклади.

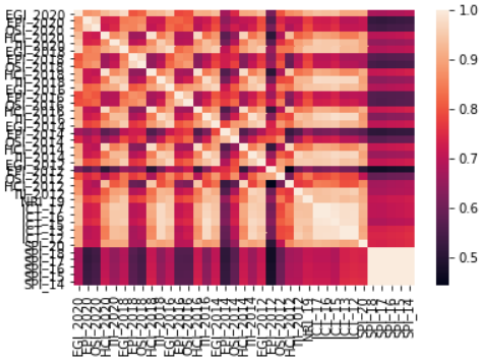

Рис. 2.4 – Загальна кореляція змінних стану

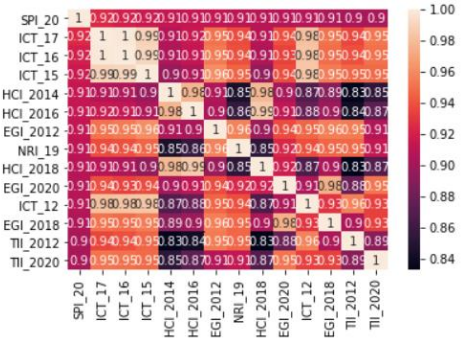

Рис. 2.5 – Кореляція змінних стану зі збільшенням найбільш корельованих

Легко помітити, що всі змінні стану демонструють сильну взаємозалежність і ідеально підходять для використання як вхідні дані для методу виявлення викидів. Згідно з рисунком 2.1 відзначена присутність значень "NaN" у рядках, що свідчить про пропущені дані у вибірці. З цієї причини було вирішено створити таблицю, яка відображає змінні стану, в яких виявлено пропуски, та їх відсоток від загальної кількості значень. Відомості про цю таблицю можна знайти на рисунку 2.6.

	Total	Percent
SPI_14	13	11.304348
SPI_15	13	11.304348
SPI_16	13	11.304348
SPI_17	13	11.304348
SPI_18	8	6.956522
ICT_12	1	0.869565
ICT_13	1	0.869565

Рис. 2.6 – Таблиця пропущених значень

Виникла гіпотеза, що пропущені значення для змінних стану ICT_12 та ICT_13 розташовані у одному рядку. Після перевірки цієї гіпотези підтверджено,

що обидва пропущених значення дійсно мали місце в одному рядку. Тому було вирішено використовувати методи машинного навчання для їхнього заповнення.

Для цього використовувалися методи випадкового лісу та лінійної регресії. Під час тренування моделей були виключені змінні стану з пропущеними значеннями, за винятком ICT_12 та ICT_13. Ці змінні слугували цільовими, і був вилучений рядок із пропущеними значеннями. Обидві моделі були навчені на цих даних, а їхня ефективність оцінювалася за допомогою середньоквадратичного відхилення (mean squared error).

Результати показали, що модель лінійної регресії виявилася більш ефективною, тому її було обрано для заповнення пропущених значень. Після відновлення пропущених даних для змінних стану ICT_12 та ICT_13, інші змінні стану з пропущеними значеннями були виключені з вибірки, що призвело до отримання вибірки без пропущених значень. Наступним етапом було проведено перевірку розподілів змінних стану та їх трансформацію для наближення до нормального розподілу.

Після цього переходимо до аналізу розподілів змінних стану та проведено їх трансформацію з метою нормалізації. Для оцінки нормальності розподілу кожної змінної стану використовувалася функція для обчислення ступеня скошеності (перекосу), і були побудовані графіки "ящик з вусами". Графіки "ящик з вусами" для вихідних даних до обробки наведено на Рисунку 2.7.

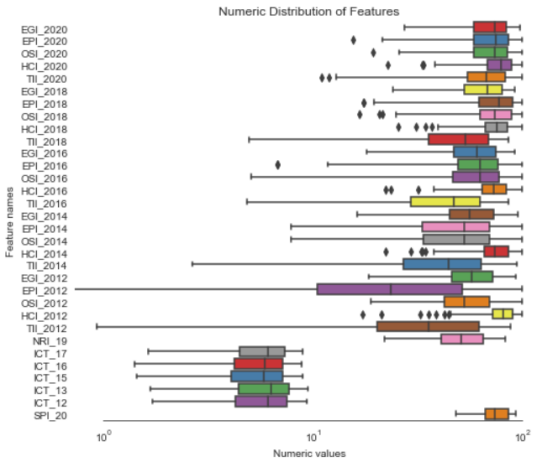

Рис. 2.7 – Ящики з вусами для змінних стану

Складність більше 0.5 була визначена для змінної стану EPI_2012, і застосовано трансформацію Бокса-Кокса, яка змінила перекіс на -0.05, що еквівалентно майже нормальному розподілу. Після цього етапу були побудовані гістограми для всіх змінних стану, і їх можна побачити на рисунку 2.8.

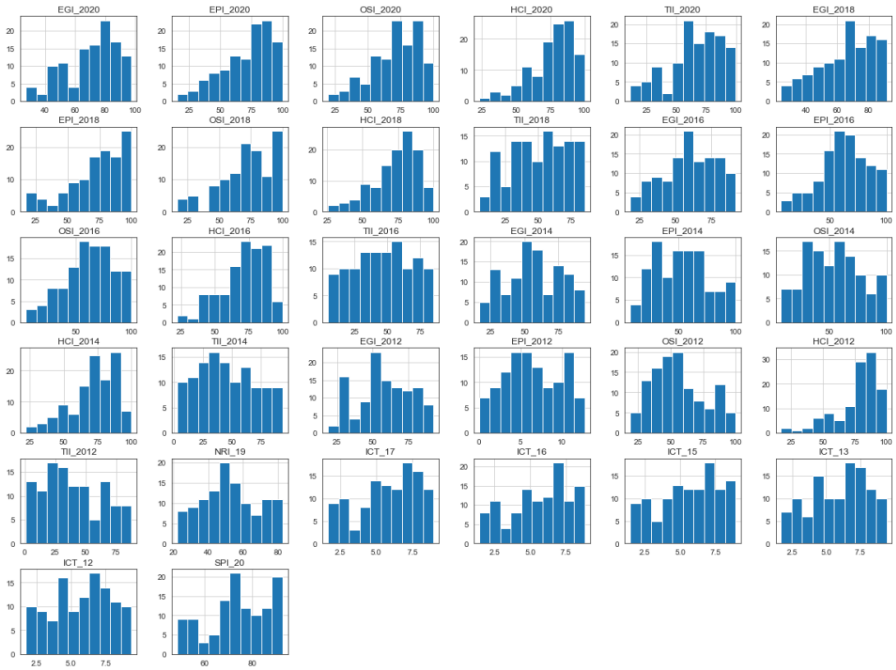

Рис. 2.8 – Гістограми змінних стану

На гістограмах видно, що практично всі дані мають нормальний розподіл, наскільки це можливо після використання трансформацій. Наступним кроком є стандартизація та нормалізація вхідних даних. Для візуалізації результатів стандартизації та нормалізації в цьому дослідженні будуть використані два методи зменшення розмірності: PCA (Principal Component Analysis) та T-SNE (t-distributed Stochastic Neighbor Embedding).

Зменшення розмірності відбувається до двох вимірів, щоб можна було відобразити результати на двовимірному графіку. Перед переходом до наступного етапу, пропонується докладніше розглянути ці методи зменшення розмірності:

- PCA (Principal Component Analysis), також відомий як метод головних компонентів, є основою багатоваріантного аналізу даних на основі

проекційних методів. Одним із основних застосувань PCA є представлення багатовимірної таблиці даних як меншого набору змінних (сумарних індексів) для аналізу тенденцій, стрибків, кластерів та викидів.

Він допомагає виявити зв'язки між спостереженнями та змінними, а також між самими змінними. PCA знаходить лінії, площини та гіперплощини в просторі з K вимірами, які найкраще апроксимують дані у випадку найменших квадратів. Це допомагає максимізувати дисперсію координат на цих просторових областях.

PCA є дуже гнучким інструментом, який дозволяє аналізувати набори даних з різними властивостями, такими як мультиколінеарність, відсутність значення, категоріальні дані та неточні вимірювання. Його мета – виділити важливу інформацію з даних та виразити її у вигляді набору головних компонентів.

Зі статистичної точки зору PCA знаходить лінії, площини та гіперплощини, які найкраще апроксимують дані в K-вимірному просторі з найменшими квадратами для набору точок даних, і робить дисперсію координат на цих просторових областях якомога більшою.

- T-SNE (t-Distributed Stochastic Neighbor Embedding) – це нелінійний метод навчання без вчителя, який переважно використовується для аналізу і візуалізації даних у великорозмірному просторі. T-SNE допомагає вивчити, як дані розташовані у великорозмірному просторі. Алгоритм T-SNE обчислює подібність між парами прикладів у великорозмірному просторі та у просторі з меншою розмірністю.

Потім метод намагається оптимізувати ці дві міри подібності, використовуючи функцію втрат. T-SNE може бути використаний для обробки великих наборів даних з високою розмірністю; і вихідні дані з меншою розмірністю можуть стати вхідними для інших моделей класифікації. Крім того, T-SNE може бути корисним для вивчення, дослідження або оцінки сегментації даних.

Опісля розгляду методів зменшення розмірності, варто переходити до наступного кроку – стандартизації даних. У рамках цієї роботи було використано три різних методи стандартизації даних, застосовуючи бібліотеку scikit-learn:

1. Стандартизація з використанням надійного скалера (Robust Scaler). Цей метод дозволяє масштабувати дані з урахуванням їх стійкості до викидів. Скалер використовує медіану та квантильний діапазон (за замовчуванням IQR: міжквартильний діапазон) для нормалізації даних. IQR представляє різницю між 1-м квартилем (25-й квантиль) та 3-м квартилем (75-й квантиль).

На рисунку 2.9 наведено короткі описи статистичних характеристик даних, такі як середнє значення, дисперсія, мінімальні та максимальні значення, а також квантилі.

	EGI_2020	EPI_2020	OSI_2020	HCI_2020	TII_2020	EGI_2018	EPI_2018	OSI_2018	HCI_2018	TII_2018
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	-1.642355	-1.074476	-0.821370	-0.452619	-2.561217	-1.823519	-4.634957	-0.697280	-0.699022	-1.448592
std	8.522131	5.966719	4.950490	2.611162	39.326203	10.682450	21.626518	8.492294	5.129921	11.198990
min	-23.024463	-17.549757	-14.675018	-9.282658	-100.214823	-27.143563	-59.550000	-24.013158	-16.257528	-25.341712
25%	-7.637756	-4.915229	-3.988720	-1.795012	-22.153598	-9.497776	-15.170000	-4.539895	-2.881972	-9.350615
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	4.788732	3.157895	3.110171	1.679767	27.756892	7.199778	13.200000	6.296390	3.094724	8.208955
max	11.658018	7.371370	7.177527	3.528670	58.807734	14.405733	23.030000	11.129386	8.100943	16.894475

Рис. 2.9 – Короткі характеристики вибірки після обробки за допомогою надійного скалеру

Як було зазначено раніше, медіана видалена, і це відображено на Рисунку 2.6 у рядку "50%". Крім того, для кожної змінної стану побудовано графік "ящик з вусами".

Ці графіки наведено на Рисунку 2.10. З метою більш повного розуміння вигляду змінних стану після трансформації використовувалися методи зменшення розмірності.

Додатково були побудовані графіки, на яких представлена вибірка у двовимірному просторі.

Графіки, отримані за допомогою методів PCA (Principal Component Analysis) та T-SNE (t-Distributed Stochastic Neighbor Embedding), представлені на Рисунках 2.11 і 2.12 відповідно.

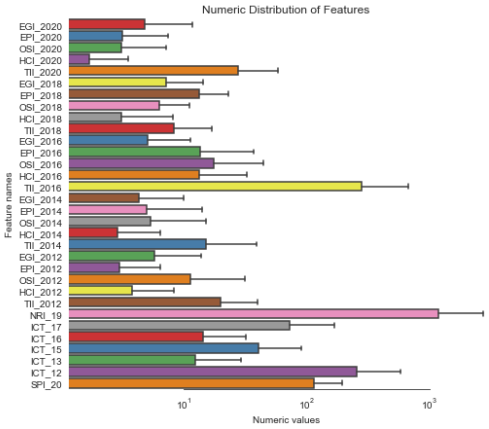

Рис. 2.10 – Ящики з вусами після використання надійного скалеру

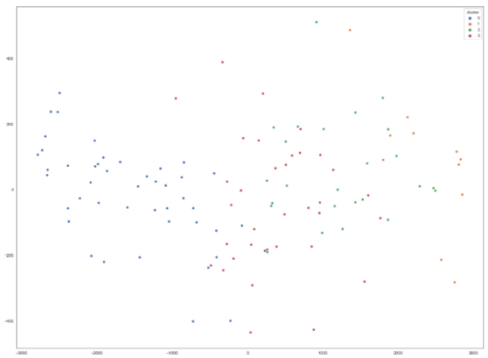

Рис. 2.11 – Відображення вибірки за допомогою PCA

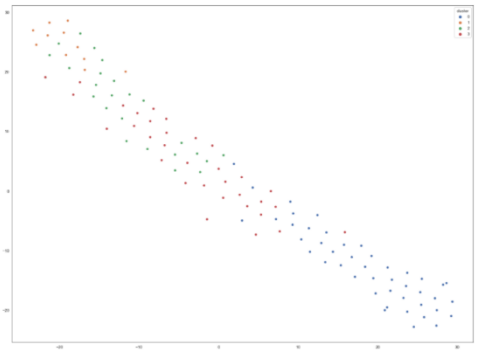

Рис. 2.12 – Відображення вибірки за допомогою T-SNE

На даних графіках різними кольорами точок відображена приналежність точки до певного типу кластера.

2. За допомогою стандартного скалера (Standard Scaler) виконується процес стандартизації ознак, що включає усереднення та масштабування даних

для отримання одиниці дисперсії. Формула стандартизації виглядає так:

$$z = \frac{x-u}{s}, (2.1)$$

де: u – середнє значення навчальної вибірки; s – стандартне відхилення навчальної вибірки. На рисунках 2.13–2.14 надана інформація про стандартизовані дані, графіки ящиків з вусами та двовимірні графіки стандартизованої вибірки, які були створені за допомогою попереднього методу стандартизації.

	EGI_2020	EPI_2020	OSI_2020	HCI_2020	TII_2020	EGI_2018	EPI_2018
count	1.150000e+02	1.150000e+02	1.150000e+02	1.150000e+02	1.150000e+02	1.150000e+02	1.150000e+02
mean	7.839140e-16	-3.050700e-16	-4.827057e-16	-2.187622e-15	6.101400e-16	2.239754e-16	-9.654113e-17
std	1.004376e+00	1.004376e+00	1.004376e+00	1.004376e+00	1.004376e+00	1.004376e+00	1.004376e+00
min	-2.519990e+00	-2.773280e+00	-2.810687e+00	-3.396450e+00	-2.494036e+00	-2.380620e+00	-2.550358e+00
25%	-7.065886e-01	-6.465131e-01	-6.426054e-01	-5.163477e-01	-5.003820e-01	-7.215426e-01	-4.892673e-01
50%	1.935598e-01	1.808664e-01	1.666430e-01	1.740987e-01	6.541251e-02	1.714494e-01	2.152561e-01
75%	7.579363e-01	7.124341e-01	7.976478e-01	8.202162e-01	7.743131e-01	8.483809e-01	8.282891e-01
max	1.567517e+00	1.421687e+00	1.622850e+00	1.531392e+00	1.567340e+00	1.525893e+00	1.284813e+00

Рис. 2.13 – Короткі характеристики вибірки після використання стандартного скалеру

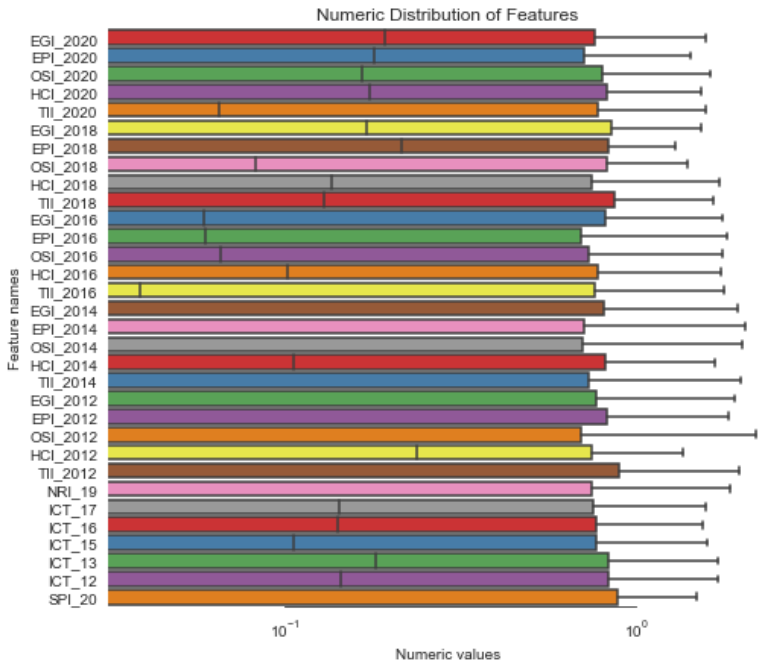

Рис. 2.14 – Ящики з вусами після використання стандартного скалеру

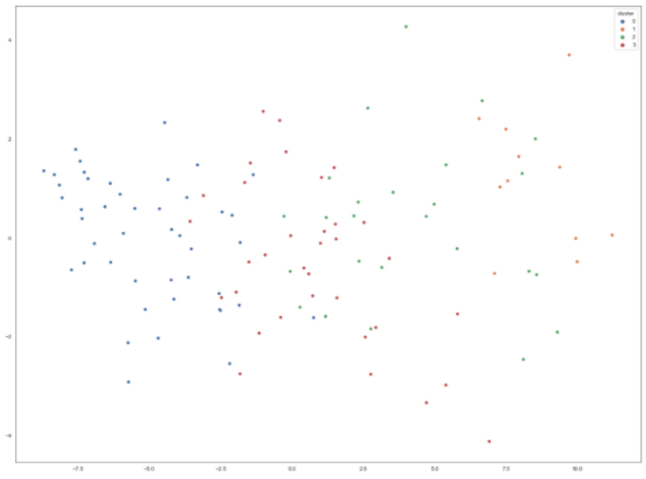

Рис. 2.15 – Відображення вибірки за допомогою PCA

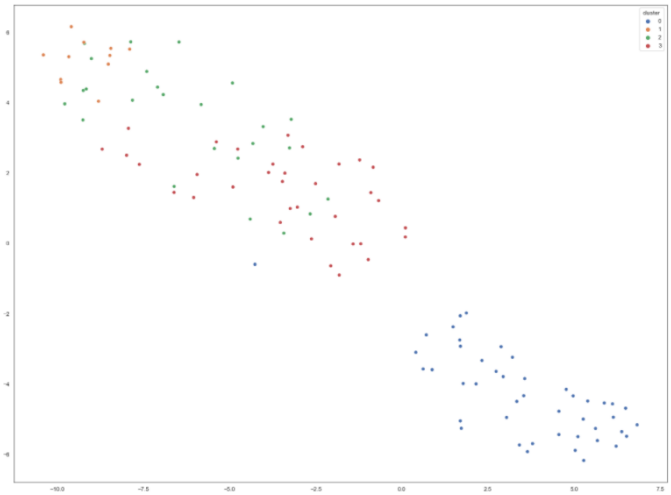

Рис. 2.16 – Відображення вибірки за допомогою T-SNE

3. Використання скалера мінімум-максимум (MinMaxScaler) дозволяє масштабувати та зсувати кожну ознаку незалежно так, щоб вона знаходилася в заданому діапазоні на навчальному наборі, наприклад, від 0 до 1, або від -3 до 3. Після порівняння обох діапазонів виявлено, що масштабування до діапазону від 0 до 1 показало кращі результати, тому було представлено результати саме для цього діапазону. Додатково, наведемо короткі статистичні описи даних, графіки ящиків з вусами та двовимірні графіки масштабованого набору даних.

	EGI_2020	EPI_2020	OSI_2020	HCI_2020	TII_2020	EGI_2018	EPI_2018	OSI_2018	HCI_2018
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	0.616510	0.661097	0.633960	0.689237	0.614087	0.609398	0.664992	0.663466	0.638731
std	0.245719	0.239424	0.226541	0.203817	0.247300	0.257103	0.261886	0.241653	0.210601
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.443645	0.506981	0.489018	0.584455	0.490881	0.424695	0.537418	0.554122	0.549113
50%	0.663864	0.704212	0.671547	0.724566	0.630193	0.653286	0.721119	0.683307	0.667428
75%	0.801938	0.830928	0.813873	0.855682	0.804739	0.826569	0.880964	0.862474	0.794477
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Рис. 2.12 – Короткі характеристики вибірки після використання мінімум-максимум скалеру

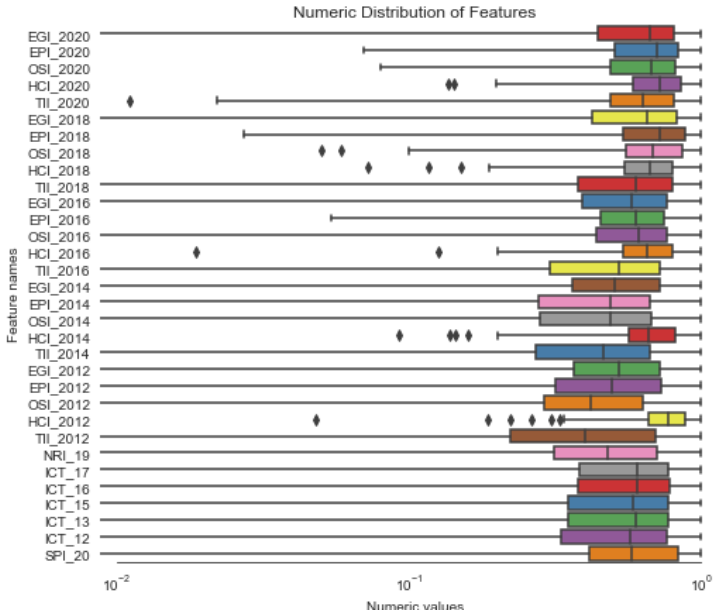

Рис. 2.13 – Ящики з вусами після використання мінімум-максимум скалеру

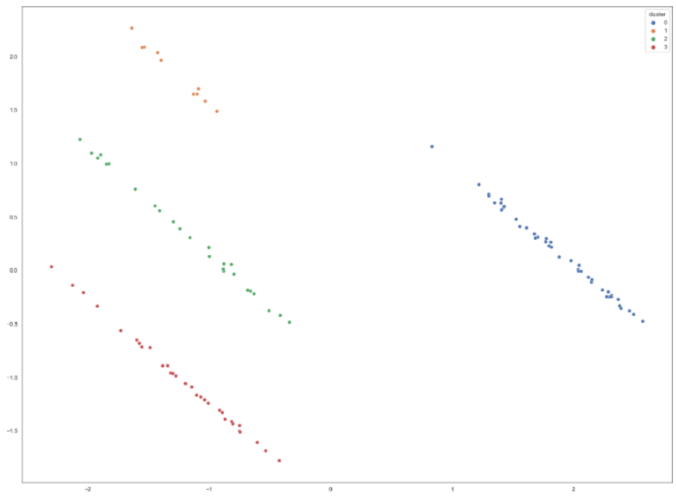

Рис. 2.17 – Відображення вибірки за допомогою PCA

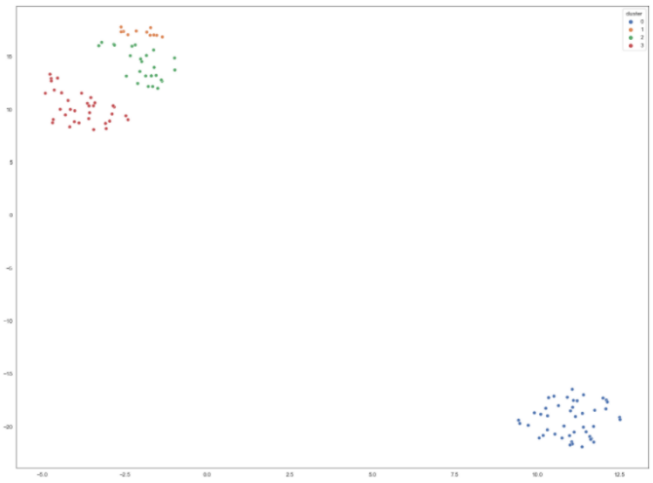

Рис. 2.18 – Відображення вибірки за допомогою T-SNE

Після стандартизації вибірки отримано чотири різних версії:

- Вибірка, яка була стандартизована за допомогою надійного скалера.
- Вибірка, яка була стандартизована за допомогою стандартного скалера.
- Вибірка, яка була стандартизована за допомогою мінімум-максимум скалера в діапазоні від -3 до 3.
- Вибірка, яка була стандартизована за допомогою мінімум-максимум скалера в діапазоні від 0 до 1.

2.2 Огляд результатів та ефективності існуючих методів виявлення викидів

В якості наборів даних були обрані різні вибірки, які використовуються для оцінки якості методів виявлення викидів, наприклад ForestCover, Satellite та інші. Також було створено власний набір даних, який буде позначатися, як «Economical (наш)». Цей набір даних відображає соціально-економічний розвиток країн за 2012-2020 роки. Він представляє 115 альтернатив з 32 змінними стану. Набір даних відображено на рис. 2.14.

Country Name	EGI_2020	EPI_2020	OSI_2020	HCI_2020	TII_2020	EGI_2018	EPI_2018	OSI_2018	HCI_2018	TII_2018	...	ICT_16	ICT_15	ICT_13	ICT_12	SPI_20	SPI_18	SPI_17	SPI_
Albania	73.99	84.52	84.12	80.01	57.85	65.19	75.84	73.61	78.77	43.18	...	4.90	4.73	4.62	4.42	75.41	71.77	71.65	70.
Algeria	51.73	15.48	27.65	69.66	57.87	42.27	20.22	21.53	66.40	38.89	...	4.32	3.71	4.46	4.22	69.92	66.83	66.83	65.
Argentina	82.79	85.71	84.71	91.00	72.65	73.35	62.36	75.00	85.79	59.27	...	6.68	6.40	6.62	6.38	80.66	74.98	74.84	74.
Armenia	71.36	75.00	70.00	78.72	65.36	59.44	56.74	56.25	75.47	46.60	...	5.56	5.32	5.64	5.55	76.46	70.87	70.53	69.
Australia	94.32	96.43	94.71	100.00	88.25	90.53	98.31	97.22	100.00	74.36	...	8.08	8.29	8.23	8.14	91.29	88.32	88.23	87.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
United States of America	92.97	100.00	94.71	92.39	91.82	87.69	98.31	98.61	88.83	75.64	...	8.13	8.19	7.78	7.75	85.71	84.78	85.27	86.
Uruguay	85.00	85.71	84.12	85.14	85.74	78.58	91.57	88.89	77.19	69.67	...	6.75	6.70	7.05	6.80	82.99	79.40	NaN	N.
Viet Nam	66.67	70.24	65.29	67.79	66.94	59.31	69.10	73.61	65.43	38.90	...	4.18	4.28	4.48	4.39	68.85	NaN	NaN	N.
Zambia	42.42	30.95	25.88	67.45	33.94	41.11	39.89	47.92	56.89	18.53	...	2.19	2.04	2.68	2.63	55.34	NaN	NaN	N.
Zimbabwe	50.19	45.24	52.35	61.35	36.88	36.92	27.53	32.64	56.68	21.44	...	2.85	2.90	3.12	2.99	52.26	45.26	43.76	42.

Рис. 2.14 – Набір даних «Economical»

Було проведено дослідження щодо ефективності традиційних математичних моделей та методів для виявлення викидів. В якості показника ефективності було обрано значення метрики PR-AUC.

Для тестування та оцінки якості було розроблено програмне забезпечення, яке дозволяє:

- виконувати тренування математичних моделей та методів;
- розраховувати показник ефективності.

Програмне забезпечення розроблено за допомогою мови Python. Також були використані наступні бібліотеки: scikit-learn, NumPy, Pandas і інші.

В якості наборів даних були використані різні вибірки, які були створені саме для оцінки якості математичних моделей та методів виявлення викидів (аномальних значень). Перед використанням, ці набори даних були попередньо чисельно оброблені. Кожен набір даних не має пропущених значень та складається тільки з числових даних. Тобто, набори даних були приведені до виду, який підтримує кожна розроблена математична модель та метод.

Дослідження якості для кожного набору даних проводилося таким чином.

1. Набір даних завантажувався до програмного забезпечення.
2. Створювалася математична модель або метод для дослідження.
3. Виконувалося тренування математичної моделі або методу з використанням крос перевірки (cross validation).

4. Розраховувалися значення метрики PR-AUC та обчислювалося середнє значення цієї метрики.

5. Виконувалося налаштування параметрів для математичної моделі або методу.

6. Виконувався пункт 3 для математичної моделі або методу з налаштованими параметрами.

7. Виконувався пункт 4 для математичної моделі або методу з налаштованими параметрами.

Дослідження якості проводилося таким чином, що набори даних подавалися до програмного забезпечення, яке в свою чергу виконувало усі необхідні дії з ними. В результаті роботи програми видавалися результати виявлення викидів та значення метрики PR-AUC.

Результати оцінки якості виявлення викидів для різних наборів даних приведені в таблиці 1.

Таблиця 2.1

Результати тестування традиційних методів для виявлення викидів

Набір даних	PR-AUC			
	Ізоляційний ліс	DBSCAN	LOF	Z-оцінка
Http (KDDCUP99)	**0.90**	0.60	0.45	0.50
ForestCover	**0.85**	0.83	0.72	0.60
Mulcross	**0.89**	0.33	0.37	0.71
Smtp (KDDCUP99)	**0.83**	0.80	0.65	0.60
Shuttle	**0.81**	0.60	0.55	0.67
Mammography	**0.86**	0.77	0.67	0.65
Satellite	**0.82**	0.68	0.72	0.60
Pima	**0.71**	0.65	0.52	0.70
Breastw	0.67	**0.71**	0.49	0.85
Arrhythmia	**0.87**	0.86	0.37	0.61
Ionosphere	**0.80**	0.78	0.73	0.73

Economical (наш)	**0.94**	0.73	0.65	0.85

Також було проведено дослідження щодо ефективності математичних моделей та методів глибокого навчання для виявлення викидів. Це дослідження є подібним до того, що проводилося для оцінки якості традиційних методів виявлення викидів. Відмінність полягає тільки у різних математичних моделях і методах, шо оцінюються. Тому було розроблено програмне забезпечення реалізації автоенкодера, варіаційного автоенкодера та генеративних змагальних мереж. Головною бібліотекою для їх реалізації була Tensorflow, яка є комплексною платформою з відкритим вхідним кодом для машинного навчання.

Результати дослідження оцінки якості виявлення викидів (аномальних значень) для визначених наборів даних наведені у таблиці 2.2.

Таблиця 2.2

Результати тестування методів глибокого навчання для виявлення викидів

Набір даних	PR-AUC		
	Автоенкодер	Варіаційний автоенкодер	GAN
Http (KDDCUP99)	0.90	**0.97**	0.95
ForestCover	0.88	0.93	**0.97**
Mulcross	0.92	**0.95**	0.76
Smtp (KDDCUP99)	0.89	**0.91**	0.90
Shuttle	0.93	**0.95**	0.94
Mammography	0.87	0.89	**0.90**
Satellite	0.85	0.87	**0.88**
Pima	0.81	0.84	**0.86**
Breastw	0.75	0.77	**0.79**
Arrhythmia	0.89	0.90	**0.93**
Ionosphere	0.85	0.86	**0.90**
Economical (наш)	0.88	**0.90**	0.86

Вибираючи найкращі методи серед традиційних та глибокого навчання, не важко побачити, що кращими є ізоляційний ліс та метод, що базується на генеративних змагальних мережах (GAN). [35-36]

Висновки до розділу 2

В підрозділі 2.1 було описано уявлення вхідних даних та їх обробка.

В підрозділі 2.2 було порівняно існуючі методи для виявлення викидів та проведено оцінку ефективності даних методів. Були використані наступні методи обробки даних: вилучення зайвих змінних стану, перетворення категоріальних змінних стану, стандартизація даних та перетворення змінних стану для наближення до нормального розподілу.

РОЗДІЛ 3.

РОЗРОБКА ЗАСОБІВ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ВИЯВЛЕННЯ ВИКИДІВ

3.1 Інтеграція методу виявлення викидів Z-score в метод Isolation forest.

Нижче буде розглянуто базовий алгоритм Isolation Forest.

Ізоляційний ліс створює ансамбль бінарних дерев для заданого набору даних. Аномалії, через свою природу мають найкоротший шлях у деревах, ніж звичайні екземпляри. Isolation Forest швидко конвергує з дуже невеликою кількістю дерев, а підвибірка дає змогу досягти хороших результатів, залишаючись ефективними з точки зору обчислень.

Загальна стратегія буде наступною. Спочатку кодується дерево, потім створюється ліс дерев (ансамбування) і, нарешті, вимірюється, наскільки далеко заходить певний екземпляр у кожному дереві, і визначається, чи є він викидом чи ні.

Оцінку аномалії можна обчислити за формулою:

$$S(x,n) = 2^{-\frac{E(h(x))}{e(n)}}, \qquad (2.2)$$

де E(h(x)) – середня довжина шляху вибірки, C(n) –невдалий пошук довжини, N – кількість зовнішніх вузлів.

Використовуючи цей метод, можна отримати математичне визначення того, що таке викид. [37]

Надалі буде розглянуто базовий алгоритм методу Z-score.

Z-score — це статистичний показник, який описує зв’язок значення із середнім значенням групи значень. Z-score вимірюється в термінах стандартних відхилень від середнього. Якщо Z-score дорівнює 0, то це означає, що оцінка точки даних ідентична середньому балу. Z-score 1,0 означатиме значення, яке є одним стандартним відхиленням від середнього.

Z-score може бути позитивними або негативними, причому позитивне значення вказує на те, що оцінка вище середнього, а негативна оцінка вказує на те, що вона нижче середнього.

Z-score для будь-якої точки даних можна розрахувати за формулою:

$$z = \frac{x-\mu}{\sigma}, \qquad (2.3)$$

де x – вхідний показник, μ — значення набору даних, σ – стандартне відхилення для набору даних.

У більшості великих наборів даних (за умови нормального розподілу даних) 99,7% значень знаходяться в межах від -3 до 3 стандартних відхилень, 95% від -2 до 2 стандартних відхилень і 68% від -1 до 1 стандартних відхилень.

Стандартне відхилення вказує на ступінь мінливості (або дисперсії) в межах даного набору даних. Наприклад, якщо вибірка даних із нормальним розподілом мала стандартне відхилення 3,1, а інша — 6,3, модель зі стандартним відхиленням (SD) 6,3 є більш розсіяною та матиме на графіку нижчий пік, ніж вибірка з SD 3.1. Крива розподілу має негативні та позитивні сторони, тому існують позитивні та негативні стандартні відхилення та z-показники. Однак це не має ніякого відношення до самого значення, окрім вказівки, на якій стороні середнього воно знаходиться. Від'ємне значення означає, що воно знаходиться ліворуч від середнього, а додатнє значення вказує, що воно знаходиться праворуч. Z-показник показує кількість стандартних відхилень даної точки даних від середнього значення. Отже стандартне відхилення має бути обчислене спочатку, оскільки z-показник використовує його для передачі мінливості точки даних. [38]

Для виявлення викидів у даних буде ефективним поєднання методів Isolation Forest та Z-score. Цей підхід можна математично описати таким чином.

Для виявлення аномалій у даних спочатку використовується метод ізольованого лісу. Ідея залишається без змін: розбиваємо простір даних на підпростори шляхом вибору випадкових поділів, і обчислюємо глибину

окремого прикладу. Приклади, які проходять через коротші шляхи, вважаються аномальними.

Після цього застосовуємо Z-score для виявлення подальших аномалій. Для цього для кожної змінної в наших даних ми обчислюємо середнє значення (μ) і стандартне відхилення (σ).

Дані, які мають високий показник Z-score, вважаються аномальними. Таким чином, ми враховуємо структурні дані Ізоляційного лісу та інформацію про відстань від середнього значення Z-оцінки.

Для цього ми можемо використати наступну формулу:

$$Score_i = IF_{Score_i} * \frac{1}{1+e^{-z}} \quad (2.4),$$

де IF_{Score_i} – це глибина даних i за методом Isolation Forest

Ця формула використовує скалярний добуток IF Score та зворотній сигмоїдальної функції Z-score для кожного прикладу.

Підвищення точності виявлення аномалій в даних може бути досягнуто за допомогою цього методу, який поєднує два різні підходи до виявлення викидів: один, заснований на структурі даних, а інший, заснований на статистичних характеристиках змінних.

3.2. Розробка програмного забезпечення для методу виявлення викидів.

Python — це високорівнева, інтерпретована та універсальна динамічна мова програмування, яка зосереджується на зручності читання коду. Зазвичай вона має невеликі програми порівняно з Java і C. Вона була заснована у 1991 році розробником Гвідо Ван Россумом. Python входить до числа найбільш популярних і швидкозростаючих мов у світі. Python — потужна, гнучка та проста у використанні мова. Крім того, спільнота Python дуже активна. Її використовують у багатьох організаціях, оскільки вона підтримує кілька парадигм програмування. Вона також виконує автоматичне керування пам’яттю.[39]

Існує велика кількість бібліотек на базі мови Python, які можна легко використовувати для створення алгоритмів для машинного навчання та математичних методів. NumPy, SciPy, scikit-learn, Pandas є прикладами таких бібліотек. Крім того, існують дуже зручні бібліотеки, які можна використовувати для візуалізації результатів досліджень, такі як matplotlib, seaborn і plotly.

Розглянемо детальніше використані бібліотеки під час розроблення програмної реалізації:

NumPy - це основна бібліотека для наукових обчислень в Python. Вона надає потужні масиви та функції високого рівня для математичних операцій над ними.

Основні функції:

1) NumPy впроваджує N-вимірні масиви, які дозволяють ефективно виконувати операції над великими наборами даних.

2) Векторизація дозволяє виконувати операції над масивами елементів за один раз, що полегшує обробку даних.

Pandas - це бібліотека для обробки та аналізу даних, що надає структури даних, такі як DataFrame та Series, для зручного представлення та операцій над даними.

Основні функції:

1) Таблична структура для представлення та обробки даних, подібна до SQL-таблиць.

2) Можливості для фільтрації, групування, об'єднання даних, а також вставка та видалення рядків і стовпців.

3) Підтримка різноманітних форматів даних, таких як CSV, Excel, SQL, HDF5.

Scikit-learn - це бібліотека для машинного навчання, яка пропонує зручний інтерфейс для використання різноманітних алгоритмів класифікації, регресії, кластеризації та інших.

Основні функції:

1) Реалізація різноманітних алгоритмів машинного навчання, таких як

Support Vector Machines, Random Forests, k-Means тощо.

2) Підтримка для попередньої обробки даних, валідації моделей, підбору параметрів.

3) Легка інтеграція з іншими бібліотеками Python, такими як NumPy та Pandas.

Matplotlib - це потужна бібліотека для візуалізації даних в Python.

Seaborn - це вищий рівень абстракції над Matplotlib, який спрощує створення стильних графіків.

Основні функції:

1) Можливості побудови різних типів графіків, включаючи лінійні, кругові, стовпчасті тощо.

2) Підтримка анімацій для вивчення змін даних з часом.

3) Великі можливості налаштування вигляду графіків для досягнення необхідного стилю.

Дане програмне забезпечення має на меті створити інструмент, який може виявити незвичайні або аномальні значення в наборі даних. Цей проєкт покликаний допомогти аналітикам і дослідникам знаходити аномалії.

Опис алгоритму виявлення аномалій:

1. Підготовка даних.

Починати варто зі збору та обробки інформації. Дані повинні бути у форматі, придатному для аналізу, з числовими стовпцями, які містять значення, які ми хочемо проаналізувати.

2. Нормалізація даних (Z-score).

Цей крок допомагає перетворити значення в кожному стовпці таким чином, щоб вони мали стандартне відхилення 1 і середнє значення 0. Це полегшує порівняння даних і виявлення аномалій.

3. Використання isolation Forest.

Алгоритм використовує дерева рішень, щоб відрізнити нормальні значення від аномальних. Він працює так: випадковий стовпець і поріг вибираються. Дані поділяються на дві категорії: значення з меншим порогом і значення з більшим

порогом. Доки аномальні показники не будуть відокремлені в окремому піддереві, процес повторюється.

4. Визначення аномалій.

Після створення ізоляційного лісу можна знайти аномалії. Точки, які знаходяться в одному піддереві або потрапляють у кілька дерев, зазвичай вважаються аномаліями.

5. Отримання результатів.

Кожен запис даних може містити результати виявлення аномалій у вигляді індикатора, де 1 означає аномалію, а 0 — нормальні дані. Крім того, для покращення розуміння та аналізу результати можна представити у вигляді графіків.

6. Оцінка результатів

У кінці треба оцінити точність методу за допомогою метрики PR-AUC.

Візуалізацію отриманих результатів зображено на рисунку 3.1.

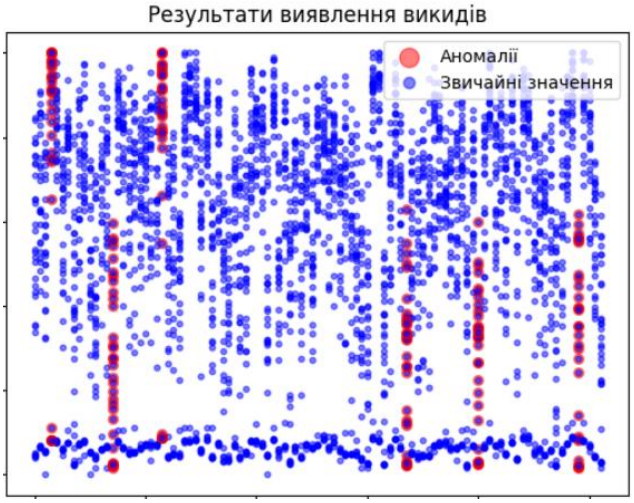

Рис. 3.1 – візуалізація отриманих результатів

3.3 Порівняння ефективності нового методу з методами Isolation Forest та Z-score.

Спочатку необхідно обрати метрики для оцінки якості методу виявлення викидів. Відповідно до підрозділу 1.1.1 найкращою у нашому випадку є метрика ROC-AUC. Ця метрика була обрана найкращою, через те що вона фокусується на малих позитивних класах – в нашому випадку викидах, та дозволяє

об’єктивно оцінити якість математичних моделей та методів. Також точність буде оцінено за допомогою метрики accurancy.

Для порівняння з новим методом виявлення викидів буде використано класичні методи виявлення викидів, такі як Isolation Forest та Z-score.

Надалі буде використано підготовлену вибірку в якості вхідних даних.

Результати тестувань наведено в таблиці 3.1.

Таблиця 3.1

Результати порівняння різних методів виявлення викидів

Метод виявлення викидів	*Точність*	*PR-AUC*
Isolation forest	0.923	0.94
Z-score	0.841	0.85
Новий метод	**0.957**	**0.974**

В таблиці 3.1 напівжирним виділено результати, які виявились найкращими. Не важко побачити, що новий метод виявлення викидів перевершує інші методи та є ефективним для задач моніторингу і оцінки рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

На рисунку 3.2 зображено виявлені аномалії шляхом використання методу Isolation Forest, без інтеграції методу Z-оцінка.

Рис. 3.2 – результати роботи методу Isolation Forest

Висновки до розділу 3

Підрозділ 3.1 містить опис мови програмування, на якій базується програмне забезпечення, а також пояснення. Окрім опису, у цій частині були розглянуті процеси розробки бібліотек і програмного забезпечення, які використовувалися. Наведено приклад роботи програмного забезпечення.

Підрозділ 3.2 містить порівняння нового методу з іншими методами виявлення викидів та доведено, що новий метод значно перевищує показники якості інших методів виявлення викидів.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи досягнуті такі результати:

- Була визначена постановка завдання цієї роботи, включаючи її мету, об'єкт і предмет дослідження.
- Була сформульована задача виявлення викидів даних.
- Проведено аналіз існуючих методів виявлення викидів і вивчено метрики для оцінювання якості методу виявлення викидів.
- Проведено огляд наявних програмних засобів для виявлення викидів даних.
- Описано методи обробки та підготовки вхідних даних для поліпшення ефективності методу виявлення викидів.
- Розроблено новий метод виявлення викидів, шляхом інтеграції методу Z-score в метод Isolation Forest.
- Розроблено обчислювальний алгоритм для нового методу виявлення викидів.
- Проведено порівняльний аналіз ефективності різних методів виявлення викидів та надано огляд та аналіз отриманих результатів.

Основним результатом цієї кваліфікаційної роботи стало розроблення нового ефективного методу виявлення викидів , спрямованого на використання в системах моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

Новий метод виявився особливо корисним у системах моніторингу рівня соціально-економічного розвитку країн з огляду на цифровий прогрес. Було досягнуто високої точності та значення PR-AUC, що свідчить про ефективність методу, навіть при роботі з несбалансованими економічними даними.

У майбутньому планується подальше вдосконалення методу, зокрема застосування глибокого навчання для зменшення розмірності вхідних даних. Також буде вивчено можливість адаптації методу до різних типів вхідних даних з метою зробити його універсальним і більш ефективним.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шкодирєв В.П., Ягафаров К.І., Баштовенко В.А., Ільїна Є.Е.. Огляд методів виявлення аномалій в потоках даних. URL: http://ceur-ws.org/Vol-1864/paper_33.pdf (Дата звернення: 20.09.2023).

2. Ломоносова М.В. Виявлення аномалій у роботі механізмів методами машинного навчання. URL: http://ceur-ws.org/Vol-2022/paper59.pdf (Дата звернення: 20.09.2023).

3. Chalapathy R., Chawla S. Deep Learning for Anomaly Detection: A Survey. URL: https://arxiv.org/abs/1901.03407 (Дата звернення: 20.09.2023)

4. Thudumu Srikanth, Branch Philip, Jin Jiong & Singh Jugdutt (Jack). A comprehensive survey of anomaly detection techniques for high dimensional big data.

5. Deep Learning for Anomaly Detection: A Review: ACM Computing Surveys: Vol 54, No 2. URL: https://dl.acm.org/doi/10.1145/3439950 (Дата звернення: 20.09.2023).

6. Muruti G., Rahim F., bin Ibrahim Z. A Survey on Anomalies Detection Techniques and Measurement Methods // 2018 IEEE Conference on Application, Information and Network Security (AINS). 2018.

7. Agrawal Shikha, Agrawal Jitendra. Survey on Anomaly Detection using Data Mining Techniques.

8. Pang G. та ін. Deep Learning for Anomaly Detection // ACM Computing Surveys. 2021. Т. 54. № 2. С. 1-38.

9. Nassif A. та ін. Machine Learning for Anomaly Detection: A Systematic Review // IEEE Access. 2021. Т. 9. С. 78658-78700.

10. Golan Izhak, El-Yaniv Ran. Deep Anomaly Detection Using Geometric Transformations. URL: https://proceedings.neurips.cc/paper/2018/file/5e62d03aec0d17facfc5355dd90d441c-Paper.pdf (Дата звернення: 21.09.2023).

11. Braei Mohammad, Wagner Sebastian. Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art. URL: https://www.semanticscholar.org/paper/Anomaly-Detection-in-Univariate-Time-

series%3A-A-on-Braei-Wagner/cf45bce52cca1f6e450ddaa1d19fe6e30661dffb (Дата звернення: 21.09.2023).

12. Rehman ur Atiq & Belhaouari Samir Brahim. Unsupervised outlier detection in multidimensional data

13. Hodge Victoria J. and Austin Jim. A Survey of Outlier Detection Methodologies. URL: https://core.ac.uk/download/pdf/58585.pdf (Дата звернення: 21.09.2023).

14. Singh Karanjit and Dr. Upadhyaya Shuchita. Outlier Detection: Applications And Techniques. URL: https://www.researchgate.net/publication/267964435_Outlier_Detection_Applications_And_Techniques (Дата звернення: 21.09.2023).

15. Wang S. та ін. Effective End-to-end Unsupervised Outlier Detection via Inlier Priority of Discriminative Network. URL: https://proceedings.neurips.cc/paper/2019/hash/6c4bb406b3e7cd5447f7a76fd7008806-Abstract.html (Дата звернення: 22.09.2023).

16. Singh Karanjit and Dr. Upadhyaya Shuchita. Outlier Detection: Applications And Techniques. URL: https://www.researchgate.net/publication/228686398_k-Nearest_neighbour_classifiers (Дата звернення: 23.09.2023).

17. Hsu Yen-Chang, Shen Yilin, Jin Hongxia, Kira Zsolt. Generalized ODIN: Detectiong Out-of-distribution Image without Learning from Out-jf-distribution Data. URL: https://openaccess.thecvf.com/content_CVPR_2020/papers/Hsu_Generalized_ODIN_Detecting_Out-of-Distribution_Image_Without_Learning_From_Out-of-Distribution_Data_CVPR_2020_paper.pdf (Дата звернення: 23.09.2023).

18. Breunig M. та ін. LOF // Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00. 2000. URL: https://www.researchgate.net/publication/221214719_LOF_Identifying_Density-Based_Local_Outliers (Дата звернення: 23.09.2023).

19. Na S., Xumin L., Yong G. Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm // 2010 Third International Symposium on Intelligent Information Technology and Security Informatics. 2010.

20. Goldstein Markus, Dengel Andreas. Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm. URL: https://www.researchgate.net/publication/231614824_Histogrambased_Outlier_Score_HBOS_A_fast_Unsupervised_Anomaly_Detection_Algorithm (Дата звернення: 15. 11. 2021).

21. Warp-core. URL: https://workday.github.io/warp-core/contents/anomaly_detection/ (Дата звернення: 23.09.2023).

22. Support Vector Machines: Theory and Applications. URL: https://www.researchgate.net/publication/221621494_Support_Vector_Machines_Theory_and_Applications (Дата звернення: 23.09.2023).

23. Liu Fei Tony, Ting Kai Ming. Gippsland School of Information Technology Monash University, Victoria, Australia. Isolation Forest. URL: https://www.researchgate.net/publication/224384174_Isolation_Forest (Дата звернення: 23.09.2023).

24. Wang Xuehui, Zhang Yong, Liu Hao, Wang Yang, Wang Lichun, and Yin Baocai. An Improved Robust Principal Component Analysis Model for Anomalies Detection of Subway Passenger Flow.

25. JR L., GG K. The measurement of observer agreement for categorical data. URL: https://pubmed.ncbi.nlm.nih.gov/843571/ (Дата звернення: 23.09.2023).

26. Study Finance. URL: https://studyfinance.com/static/media/z-score.png (Дата звернення: 23.09.2023).

27. A Brief Overview of Outlier Detection Techniques .URL: https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561 (Дата звернення: 23.09.2023).

28. Demo of DBSCAN clustering algorithm. URL: https://scikit-learn.org/stable/_images/sphx_glr_plot_dbscan_001.png (Дата звернення: 23.09.2023).

29. Outlier detection with Local Outlier Factor (LOF). URL: https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_outlier_detection.html#:~:text=The%20Local%20Outlier%20Factor%20(LOF,lower%20density%20than%20their%20neighbors. (Дата звернення: 23.09.2023).

30. Anomaly Detection using Autoencoders. URL: https://towardsdatascience.com/anomaly-detection-using-autoencoders-5b032178a1ea (Дата звернення: 23.09.2023).

31. Cloudera Fast Forward. Deep Learning for Anomaly Detection. URL: https://ff12.fastforwardlabs.com/ (Дата звернення: 23.09.2023)

32. Scikit Learn – Anomaly Detection. URL: https://www.tutorialspoint.com/scikit_learn/scikit_learn_anomaly_detection.htm (Дата звернення: 01.10.2023).

33. Anomaly Detection with PyOD. URL: https://towardsdatascience.com/anamoly-detection-with-pyod-fea90f0b4b42 (Дата звернення: 01.10.2023)

34. A Comprehensive Guide to Data Preprocessinghttps://neptune.ai/blog/data-preprocessing-guide (Дата звернення: 01.10.2023).

35. Шевченко Д.О, Лихач О.Ю, Угрюмов М.Л. Методи виявлення викидів в системах автоматизованого збору даних. Праці 8-ої Міжнародній конференції «Комп'ютерне моделювання в наукоємних технологіях (КМНТ-2022)

36. Лихач О. Ю., Угрюмов М. Л., Шевченко Д. О., Шматков С. І. Методи виявлення викидів в пробних вибірках при управлінні процесами в системах за станом. Вісник Харківського національного університету імені В. Н. Каразіна серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління». 2022. № 53. С. 21-40.

37. Isolation Forest from Scratch. URL: https://towardsdatascience.com/isolation-forest-from-scratch-e7e5978e6f4c (Дата звернення: 05.10.2023).

38. How to Calculate Z-Score and Its Meaning. URL: https://www.investopedia.com/terms/z/zscore.asp (Дата звернення: 05.10.2023).

39. Python Language advantages and applications URL: https://www.geeksforgeeks.org/python-language-advantages-applications/ (Дата звернення: 05.10.2023).

ДОДАТКИ

Додаток А

Завдання на кваліфікаційну роботу

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Магістр**
Галузь знань: **15 – Автоматизація та приладобудування**
Спеціальність: **151 – «Автоматизація та комп'ютерно-інтегровані технології»**

ЗАТВЕРДЖУЮ
Завідувач кафедри теоретичної та прикладної системотехніки
д.т.н., проф. Шматков С. І.

«08 » грудня 2022 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Лихач Олег Юрійович
(прізвище, ім'я, по батькові студента)

1. Тема роботи «Метод виявлення викидів в пробних вибірках при управлінні процесами в системах за станом»
керівник роботи Бикова Тетяна Володимирівна, к.т.н. доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від «10» листопада 2023 року № 4101-5/3197

2. Строк подання студентом роботи 28 .11.2023
3. Перелік питань, які потрібно розробити:

1. Визначення постановки задачі виявлення викидів.
2. Аналітичний огляд існуючих методів виявлення викидів.
3. Огляд існуючих програмних засобів для виявлення викидів.
4. Формалізація і уявлення вхідних даних.
5. Підготовка та обробка вхідних даних для покращення ефективності методу.
5. Опис методу виявлення викидів.
6. Опис обчислювального алгоритму вирішення задачі.
7. Опис обраного програмного забезпечення для вирішення задачі.
8. Порівняння ефективності різних методів виявлення викидів.
9. Огляд та аналіз отриманих результатів.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1.	Визначення постановки задачі методу виявлення викидів.	08.12.2022-09.01.2023
2.	Аналітичний огляд існуючих методів виявлення викидів.	09.01.2023-09.02.2023
3.	Огляд існуючих програмних засобів для виявлення викидів.	09.02.2022-11.03.2023
4.	Формалізація і уявлення вхідних даних.	11.03.2023-11.04.2023
5.	Підготовка та обробка вхідних даних для ефективності методу.	11.04.2023-13.05.2023
6.	Опис методу виявлення викидів даних.	11.04.2023-13.05.2023
7.	Опис обчислювального алгоритму вирішення задачі.	13.05.2023-25.06.2023
8.	Опис обраного програмного забезпечення для вирішення задачі.	25.06.2023-29.07.2023
9.	Порівняння ефективності різних методів виявлення викидів.	29.07.2023-29.09.2023
10.	Огляд та аналіз отриманих результатів.	29.09.2023-03.10.2023
11.	Розробка пояснювальної записки кваліфікаційної роботи	03.10.2023-15.10.2023
12.	Представлення кваліфікаційної робти керівнику та рецензенту.	15.10.2023-01.11.2023
13.	Підготовка до захисту кваліфікаційної роботи.	01.11.2023-15.11.2023

5. Дата видачі завдання 08.12.2022

Студент Лихач О.Ю.
ініціали, прізвище _____ підпис

Керівник роботи Бикова Т.В.
ініціали, прізвище _____ підпис

Додаток Б

Затверджую

_____

«_____»_____2020 р.

Технічне завдання

на розробку програмного виробу

«Програма виявлення викидів в пробних вибірках при управлінні процесами в системах за станом»

Назва розділу	Назва і зміст підрозділу
1. Введення	1.1. Назва виробу: метод виявлення викидів у пробних вибірках при управлінні процесами за станом 1.2. Галузь застосування: економіка.
2. Підстава для розробки	2.1. Навчальний план за спеціальністю 151 – «Автоматизація та комп’ютерно-інтегровані технології» 2.2. 2.2. Завдання на кваліфікаційну роботу магістра затверджено наказом ректора № 4101-5/3197 від 10.11.2023.
3. Призначення розробки	3.1. Мета розробки програмного виробу: можливість з достатньо високим рівнем достовірності вирішувати завдання виявлення викидів у вибірках даних та часових рядах. 3.2. Призначення виробу: використання виробу для виявлення: помилок даних, неправильно класифікованих об'єктів.
4. Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: можливість виявляти викиди із вхідної вибірки на основі інформаційних критеріїв. 4.2. Вимоги до надійності: забезпечення працездатності методу виявлення викидів при подачі вхідних даних неправильного формату. 4.3. Вимоги до умов експлуатації: встановлення мови програмування Python та необхідних бібліотек для роботи методу виявлення викидів. 4.4. Вимоги до складу і параметрів технічних засобів: комп’ютер або ноутбук із 4 ГБ оперативної пам’яті та процесором не нижче Intel Core i3 9-го покоління . 4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 8/10, підтримка Anaconda, Python 3. 4.6. Вимоги до маркування та упаковки: відсутні. 4.7. Вимоги до транспортування і зберігання: відсутні. 4.8. Спеціальні вимоги: відсутні.

<table>
<tr><td>5. Вимоги до технічної документації</td><td colspan="2">Документацією до виробу «Метод виявлення викидів у пробних вибірках при управлінні процесами за станом» вважати:
1) Технічне завдання на розробку виробу (представити як Додаток до пояснювальної записки до кваліфікаційної роботи).
2) Програму і методику випробувань розробленого виробу (представити як Додаток до пояснювальної записки до кваліфікаційної роботи).
3) Опис виробу.
4) Фрагменти тексту програми (записати на диск з пояснювальною запискою до кваліфікаційної роботи).</td></tr>
<tr><td>6. Техніко-економічні показники</td><td colspan="2">Орієнтовна оцінка ефективності: точність (accuracy) повинна бути вище 94.5%, метрика PR-AUC повинна бути вище 0.95.</td></tr>
<tr><td>7. Стадії і етапи розробки</td><td>Дата</td><td>Назва етапу</td></tr>
<tr><td></td><td>08.12.2022-09.01.2023

09.01.2023-09.02.2023

09.02.2022-11.03.2023

11.03.2023-11.04.2023

11.04.2023-13.05.2023

11.04.2023-13.05.2023

13.05.2023-25.06.2023

25.06.2023-29.07.2023

29.07.2023-29.09.2023

29.09.2023-03.10.2023

03.10.2023-15.10.2023

15.10.2023-01.11.2023</td><td>1. Визначення постановки задачі методу виявлення викидів.
2. Аналітичний огляд існуючих методів виявлення викидів.
3. Огляд існуючих програмних засобів для виявлення викидів.
4. Формалізація і уявлення вхідних даних.
5. Підготовка та обробка вхідних даних для ефективності методу.
6. Опис методу виявлення викидів даних.
7. Опис обчислювального алгоритму вирішення задачі.
8. Опис обраного програмного забезпечення для вирішення задачі.
9. Порівняння ефективності різних методів виявлення викидів.
10. Огляд та аналіз отриманих результатів.
11. Розробка пояснювальної записки кваліфікаційної</td></tr>
</table>

	01.11.2023-15.11.2023	роботи 12. Представлення кваліфікаційної роботи керівнику та рецензенту. 13. Підготовка до захисту кваліфікаційної роботи.
8. Порядок контролю і приймання	Загальні вимоги до приймання розроблюваного виробу: 1) Перевірка ходу розробки програмного виробу керівнику робіт виконувати 1 раз в 3 тижні. 2) Випробування виробу відповідно до Програми і методики випробувань провести на базі комп'ютерного класу або на базі приватного приміщення. 3) Захист розробленого виробу провести на засіданні атестаційної комісії. 4) Пояснювальну записку представити на паперових носіях в одному примірнику та в електронному вигляді.	

Виконавець:
студент групи КУ-61

Лихач О.Ю. _____

Замовник:
к.т.н., доцент кафедри теоретичної та прикладної системотехніки

Бикова Т.В. _____

Додаток В

Програма і методика випробувань програмного виробу

«Програма виявлення викидів в пробних вибірках при управлінні процесами в системах за станом»

1 Об'єкт випробувань

1.1 Найменування випробуваного програмного виробу: метод виявлення викидів в пробних вибірках при управлінні системами за станом

1.2 Область його застосування: економіка.

2. Мета випробувань

Перевірка працездатності виробу та правильності роботи методу виявлення викидів. Підтвердження характеристик розробленого виробу вимогам, які сформульовані в ТЗ.

3. Загальні положення

3.1 Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії та перевірку даного виробу.

3.2 Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу або будь-якого приміщення в період роботи атестаційної комісії.

3.3 Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї Програми і методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до виробу

4.1. Вимоги до функціональних характеристик: можливість виявляти викиди (аномальні значення) в пробних вибірках.

4.2. Вимоги до надійності: забезпечення працездатності методу виявлення викидів при подачі вхідних даних неправильного формату.

4.3. Вимоги до умов експлуатації: встановлення мови програмування Python та необхідних бібліотек для роботи методу виявлення викидів.

4.4. Вимоги до складу і параметрів технічних засобів: комп'ютер або ноутбук із 4 ГБ оперативної пам'яті та процесором не нижче Intel Core i3 9-го покоління .

4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 8/10, підтримка Anaconda, Python 3.

4.6. Вимоги до маркування та упаковки: жорстка упаковка з пластмаси, маркування на українській та англійській мовах.

4.7. Вимоги до транспортування і зберігання: транспортування в упаковці будь-яким способом, температура транспортування/зберігання +5-+20 C^o.

4.8. Спеціальні вимоги: відсутні.

5. Вимоги до документації

Склад програмної документації, що подається на випробування, включає:

1) Технічне завдання на розробку виробу (представити як Додаток до пояснювальної записки до кваліфікаційної роботи).

2) Програму і методику випробувань розробленого виробу (представити як Додаток до пояснювальної записки до кваліфікаційної роботи).

3) Опис виробу.

4) Фрагменти тексту програми (записати на диск з пояснювальною запискою до кваліфікаційної роботи).

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Засоби випробувань представлено на ПК на яких встановлено наступні програмні засоби: Anaconda, Python 3.8. В програмному засобі Anaconda

створено середовище, де встановлені усі необхідні бібліотеки для запуску, такі як numpy, scikit-learn, pandas, matplotlib, seaborn, scipy.

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- ознайомчий (1-й етап);

- власне випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1) Перевірку комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.

2) Виконання тесту:

2.1) Запустити термінал або команду строку.

2.2) Проглянути написаний код та перевірити на наявність помилок.

2.3) Перевірити наявність встановлених бібліотек шляхом запуску програми на даних, які згенеровані випадковим чином

3) Якість програмної документації перевіряється на відповідність до стандартів ISO/IEC 26514:2008.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

1) Перевірку відповідності технічних характеристик програми вимогам технічного завдання.

2) Перевірку ступеня виконання функціональних вимог до програми.

3) Методику проведення перевірок:

3.1) Запустити термінал або команду строку.

3.2) Перевірити наявність встановлених бібліотек шляхом запуску програми на даних, які згенеровані випадковим чином. При відсутності бібліотек, встановити їх за допомогою Anaconda Navigator.

3.3) Порядок проведення випробувань:

- Запустити програму за допомогою терміналу з використанням прапорів, які вказують на шлях до вхідних даних та параметри методу виявлення викидів.

- Перевірити працездатність розробленого програмного продукту.

- Перевірити правильність виконання виявлення викидів.

- Перевірити правильність візуалізації результатів.

-Перевірити відповідність показників якості методу виявлення викидів.

4) Якщо перевірки на першому та другому етапах виконано успішно, то виріб вважається таким, що пройшов випробування.

Виконавець:

студент групи КУ-61 Лихач О.Ю. _____

Додаток Г

Лістинг Г.1 – Приклад використання нового методу

```
import numpy as np
import pandas as pd
from sklearn.ensemble import IsolationForest
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import drive
drive.mount('/content/drive')
dataset_path = '/content/drive/MyDrive/Colab
Notebooks/original.csv'
data = pd.read_csv(dataset_path)
data = data.dropna()
data.info()
x=data.drop(['cluster','Country Name'],axis=1)
scaler = StandardScaler()
x_scaled = scaler.fit_transform(x)
model = IsolationForest(contamination=0.05)
model.fit(x_scaled)
anomalies = model.predict(x_scaled)
anomaly_indices = np.where(anomalies == -1)
print("Індекси аномалій:", anomaly_indices)
anomalous_data = data[anomalies == -1]
print("Аномальні значення:")
print(anomalous_data)
plt.figure(figsize=(20, 8))
for current_column in x.columns:
    column_data = data[current_column]
    column_anomalies = column_data[anomalies == -1]

      plt.scatter(data.index[anomalies == -1], column_anomalies,
c='r', s=30, alpha=0.5)
    plt.scatter(data.index, column_data, c='b', s=10, alpha=0.5)
```

```
plt.legend(['Аномалії', 'Звичайні значення'], loc='upper right',
markerscale=2)
plt.title("Результати виявлення викидів")
plt.xlabel("Індекси зразків")
plt.ylabel("Значення")
plt.show()
```