https://doi.org/10.31891/2307-5732-2025-351-45
УДК 004.85

ПАСТУХ ОЛЕГ
Тернопільський національний технічний університет імені Івана Пулюя
https://orcid.org/0000-0002-0080-7053
e-mail: oleg.pastuh@gmail.com
ХОМИШИН ВІКТОР
Тернопільський національний технічний університет імені Івана Пулюя
https://orcid.org/0000-0003-4369-501X
e-mail: homyshyn@gmail.com

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДІВ КЛАСТЕРНОГО АНАЛІЗУ ВИЯВЛЕННЯ ВИКИДІВ У СФЕРІ НЕРУХОМОСТІ

У роботі проведено комплексну оцінку існуючих методів кластерного аналізу виявлення викидів на реальному наборі даних. Досліджено роботу 23 алгоритмів, які побудовані на основі 4-х типів моделей: ймовірнісних, лінійних, моделей на основі близькості та на основі графів. Набір даних підготовлено на базі розробленого програмного забезпечення для агенцій нерухомості. Дослідження охоплювало ринок нерухомості м. Тернополя, зокрема продаж квартир та кімнат. Підготовлений набір даних містить 760 об'єктів нерухомості з 12 ознаками. Для кожного об'єкту нерухомості на підставі його характеристик експертом поетапно проставлялася мітка аномальності. Це дозволило сформувати набори даних з поміткою аномальності у 10, 15, 20 та 25 % об'єктів. Тестування алгоритмів проводилося з використанням двох способів кодування категоріальних ознак – кодування міток та однократне кодування. Стандартизація набору даних проведена з використанням масштабувальника RobustScaler, який є стійким до викидів. Результати роботи оцінювалися за трьома показниками: AUC-ROC, Precision @ Rank n та час виконання алгоритму. Вони дозволили оцінити точність та ефективність використаних алгоритмів та визначити їхню придатність для реальних задач виявлення аномалій у даних про нерухомість. Візуалізація результатів роботи алгоритмів проведена з використанням t-SNE методу зменшення розмірності й дала можливість оцінити, наскільки добре кожна модель кластеризує нормальні та аномальні об'єкти. Також у роботі більш детально досліджено недетерміновані алгоритми виявлення аномалій на предмет стабільності результатів, подано діаграми розмаху їх метрик та описано можливість їх практичного використання. Загалом, дане дослідження охоплює сучасні моделі та алгоритми виявлення аномалій, етапи обробки та аналізу інформації, технології візуалізації даних, сприяє вдосконаленню методологій, заснованих на машинному навчанні у сфері нерухомості, підтримці прийняття обґрунтованих рішень при аналізі ринку нерухомості та надання цінних рекомендацій зацікавленим сторонам.

Ключові слова: виявлення аномалій, виявлення викидів, нерухомість, навчання без вчителя, аналіз даних.

PASTUKH OLEH, KHOMYSHYN VIKTOR
Ternopil Ivan Puluj National Technical University

EFFICIENCY RESEARCH OF CLUSTER ANALYSIS METHODS FOR DETECTING OUTLIERS IN REAL ESTATE MARKET

The work conducted a comprehensive assessment of existing uncontrolled methods for detecting outliers on a real dataset. The work of 23 algorithms built based on 4 types of models was studied: probabilistic, linear, proximity-based and graph-based models. The dataset was prepared based on developed software for real estate agencies. The study covered the real estate market of Ternopil, in particular the sale of apartments and rooms. The prepared dataset contains 760 real estate objects with 12 features. For each real estate object, based on its characteristics, an anomaly label was gradually applied by the expert. This allowed the formation of datasets with an anomaly label in 10, 15, 20 and 25% of the objects. The testing of the algorithms was carried out using two methods of encoding categorical features: Label Encoding and One-Hot Encoding. Standardization of the dataset was carried out using the RobustScaler scaler, which is resistant to outliers. The results of the work were evaluated by three indicators: AUC-ROC, Precision @ Rank n and algorithm execution time. They allowed us to assess the accuracy and efficiency of the algorithms used and determine their suitability for real-world problems of anomaly detection in real estate data. The visualization of the results of the algorithms was carried out using the t-SNE dimensionality reduction method and made it possible to assess how well each model clusters normal and anomalous objects. The work also examines in more detail the stability of the results of nondeterministic anomaly detection algorithms, presents diagrams of the range of their metrics, and describes the possibility of their practical use. In general, this study covers modern models and algorithms for anomaly detection, stages of information processing and analysis, data visualization technologies, contributes to the improvement of methodologies based on machine learning in the real estate sector, supports informed decision-making in the analysis of the real estate market and provides valuable recommendations to stakeholders.

Keywords: anomaly detection, outlier detection, real estate, unsupervised learning, data mining.

Стаття надійшла до редакції / Received 05.03.2025
Прийнята до друку / Accepted 26.03.2025

Постановка проблеми

Виявлення викидів (аномалій) відноситься до процесу знаходження елементів, які є незвичайними. Для табличних даних це зазвичай означає ідентифікацію незвичайних рядків у таблиці; для зображень – незвичайних зображень; для текстових даних – незвичайних документів, й так само для інших типів даних. Конкретні визначення "нормального" та "незвичайного" можуть варіюватися, але на фундаментальному рівні виявлення аномалій працює на припущенні, що більшість елементів у наборі даних можна вважати нормальними, тоді як ті, що значно відрізняються від більшості, можуть

вважатися незвичайними або аномальними [1]. Хоча виявлення аномалій часто не привертає такої ж уваги, як інші сфери машинного навчання, такі як прогнозування чи навчання з підкріпленням, генеративний штучний інтелект, воно займає не менш важливе місце.

Викид – це термін, який складно чітко описати. Проте у цьому є певна перевага, яка дозволяє дослідникам підходити до виявлення викидів з різних позицій, хоча це також несе за собою ряд невизначеностей. Частково це пов'язано з тим, що контекст має велике значення. Те, що є шумом в одному випадку, є сигналом в іншому.

У роботі [2] зібрано більше 10 формулювань терміну «викид», описано проблеми методів ідентифікації викидів, запропоновано схему, яка дозволяє аналізувати в загальному вигляді помилки, що можуть вплинути на експериментальні спостереження. Це дозволило побудувати загальну структуру, в якій можна проводити якісний аналіз даних, що при застосуванні до аналізу викидів дає певні переваги, порівняно з класичним підходом.

Дослідження [3] подає наступне формулювання терміну «викид» – це спостереження (або підмножина спостережень), яке не сумісне (не відповідає) решті спостереженням з цього набору даних. Це значно розширило концепцію викидів, додало ідею колективних викидів: оскільки окремі елементи (наприклад, одиничні покупки за кредитною карткою) є нормальними, набір із кількох елементів (наприклад, серія покупок за цією ж карткою впродовж невеликого проміжку часу) може бути викидом, навіть якщо кожен елемент у наборі типовий.

Хоча терміни "викид" (outlier) і "аномалія" (anomaly) часто використовуються взаємозамінно, вони мають певні відмінності залежно від контексту.

Схожість даних термінів у наступному:

– як викид, так і аномалія вказують на точки даних, які не відповідають загальним патернам або очікуванням;

– обидва можуть бути ознаками помилок у даних або інших факторів й потребують додаткового аналізу.

Відмінності полягають у наступному:

1. Контекст:

– викид зазвичай розглядається як одинична точка, яка значно відрізняється від решти даних. Викиди часто бувають випадковими й не завжди вказують на порушення (наприклад, людська помилка при введенні значення);

– аномалія може охоплювати не лише окремі точки, але й складніші патерни, наприклад, тривалі відхилення в часових рядах чи зміни тренду. Аномалія більше фокусується на відхиленні від очікуваної поведінки.

2. Інтерпретація:

– викид часто розглядається як статистична аномалія, яка не обов'язково пов'язана з реальними подіями;

– аномалії зазвичай мають причинно-наслідковий характер, наприклад, кібератака, збій у системі, тощо.

3. Методи виявлення:

– викиди, зазвичай, визначаються через статистичні методи (наприклад, відхилення на 3σ від середнього);

– аномалії часто вимагають більш складного аналізу, наприклад, алгоритмів машинного навчання, часових рядів чи нейронних мереж.

Підсумовуючи подане, окреслимо викид як специфічний підтип аномалій, але не всі аномалії є викидами. Вибір терміну залежить від конкретного завдання та контексту аналізу.

Виявлення аномалій використовується у багатьох сферах людського життя, зокрема [1]:

а) в наукових дослідженнях – для аналізу результатів спостережень й пошуку явищ, які виходять за межі нормальних;

б) у фінансовому секторі – для виявлення підозрілих транзакцій, шахрайства, відмивання грошей;

в) в кібербезпеці – для ідентифікації нетипової мережевої активності;

г) на транспорті – для аналізу дорожнього руху, виявлення заторів чи аварій;

д) у соціальних медіа – для виявлення ботів, ідентифікації фейкових новин, аналізу впливу контенту;

е) на виробництві – для моніторингу обладнання та виявлення збоїв чи несправностей;

ж) в медицині – для діагностики захворювань на базі нетипових результатів аналізів;

з) в бізнесі – маркетинг (виявлення змін у поведінці клієнтів), логістика (аналіз відхилень у термінах доставки чи витратах на транспортування), управління товарами (виявлення аномалій у закупівлях), управління персоналом (аналіз продуктивності співробітників), бізнес-процеси (виявлення надмірних витрат), ціноутворення (виявлення нетипових цін на товари, об'єкти чи послуги).

Нерухомість є одним із найдорожчих товарів на сучасному ринку. Цей вид активів має значну цінність не лише у фінансовому, а й у соціальному та економічному контексті. Динаміка ринку нерухомості відображає загальні тенденції у фінансовій системі країни, рівень добробуту населення, інвестиційний клімат та темпи економічного розвитку. Ціни на нерухомість можуть значно коливатися

під впливом різних факторів, таких як економічні умови, політична стабільність, попит та пропозиція, а також інші чинники. В окремому місті чи населеному пункті ціна об’єкта нерухомості залежить від низки локальних факторів: престижності району, відстані від центру, ступеня розвитку інфраструктури, транспортного сполучення, благоустрою, привабливості, тощо. Безпосередній вплив на ціну мають стан та характеристики самого об’єкта та будівлі. [4].

Між учасниками угоди купівлі-продажу майже завжди існує дискусія: продавець бажає продати об’єкт якомога дорожче, а покупець – навпаки, купити дешевше. Щодо оренди, то орендодавець бажає, щоб його майно коштувало дорожче, а орендатор вважає, що воно коштує дешевше, і тому плата за його оренду повинна бути нижчою. Вирішити цей конфлікт інтересів має незалежний оцінювач, який повинен компетентно оцінити об’єкт нерухомості, аргументовано переконати учасників у тому, що розрахована ним сума і є та об’єктивна вартість, яка відображає цінність об’єкта на ринку із заданими характеристиками, в даному місці та в даний момент часу [5].

Одним з інструментів, який дозволяє більш точно аналізувати ринок на етапі оцінки об’єктів, є виявлення аномалій у цінах на нерухомість. Цей метод допомагає ідентифікувати незвичайні та потенційно проблемні зміни цін, що можуть сигналізувати про порушення у ринкових умовах. Загалом, виявлення аномалій у сфері нерухомості може мати доволі широке застосування (табл. 1).

Таблиця 1

Напрямки та результати використання методів виявлення аномалій у сфері нерухомості

Група	Напрямок	Результат
Технології та автоматизація	Аналіз даних	Виявлення аномалій у цінах або характеристиках на нерухомість
	Розробка інноваційних платформ	Створення додатків та веб-сервісів для аналізу аномалій у реальному часі
	Автоматизація оцінок	Автоматичне визначення ринкової вартості об'єкта на основі даних, з яких вилучено аномальні об’єкти
	Оптимізація пошуку нерухомості	Автоматизація підбору об'єктів за критеріями клієнта, зокрема, із низькими цінами
	Інтеграція технологій	Розробка мультиагентної системи оцінки нерухомості на базі сучасних інноваційних алгоритмів машинного навчання
Аналіз ринку	Виявлення недооцінених об'єктів	Ідентифікація об'єктів із цінами нижче ринкових через неправильну оцінку, поганий стан чи відсутність попиту
	Виявлення переоцінених об'єктів	Ідентифікація об'єктів, які пропонуються за цінами, значно вищими середньо-ринкових
	Виявлення підозрілих об'єктів	Ідентифікація об'єктів з неправдивою інформацією про площу чи характеристики
	Виявлення незвичних тенденцій	Пошук нових трендів в цінах чи попиті на нерухомість у певному районі
	Виявлення перспективних районів	Аналіз районів із швидким розвитком інфраструктури, зростанням попиту чи появою нових об'єктів
	Аналіз конкурентів	Пошук унікальних характеристик об'єктів конкурентів, що допомагає краще позиціонувати свої пропозиції
Інвестиції	Збільшення прибутковості	Пошук об'єктів з потенціалом для перепродажу, здачі в оренду чи комерційного використання
	Реновація та розвиток	Пошук об'єктів, що потребують реконструкції, але мають високий потенціал після ремонту
	Виявлення нерозкритого потенціалу	Пошук об'єктів, що використовують підвали або горища, вартість площ яких не була додана у ціну об'єкта
	Прогнозування ризиків	Оцінка майбутніх ризиків, таких як падіння цін через перевищення пропозиції чи зниження попиту в районі

На практиці для виявлення аномалій використовують алгоритми, відомі як детектори, які спеціально розроблені для виявлення викидів. Частина з них є простими методами, наприклад, статистичні, ймовірнісні, лінійні моделі, а інші – більш складні, що будуються на поведінковому аналізі, часових послідовностях, графах, нейронних мережах, тощо [6]. За останні 10 років кількість алгоритмів виявлення аномалій помітно зросла. Переважно вони доступні в бібліотеках Python, зокрема PyOD [7] та Scikit-learn [8].

Надалі у роботі ми розглянемо більше 20 алгоритмів виявлення аномалій, включаючи найновіші.

Більшість із них є простими, індивідуальними, легко та інтуїтивно зрозумілими. Кожен алгоритм ідентифікує різні, а іноді дуже різні елементи як викиди, кожен алгоритм має свою власну унікальну спрямованість та підхід й може бути цінним у певних контекстах, але жоден із них не є остаточним.

Аналіз досліджень та публікацій

Підвищення продуктивності й точності прогнозування, що є критично важливим при оцінці нерухомості, безпосередньо залежить від якості даних. Важливими факторами також є розмір набору даних, повнота та репрезентативність. Загалом, дослідники використовують різноманітні моделі та алгоритми в задачах виявлення викидів під час оцінки нерухомості за допомогою машинного навчання. Переважно досліджується саме житлова нерухомість.

У роботі [9] протестовано набір даних, що містить близько 70 тис. об'єктів нерухомості з 283 ознаками, 4 методи виявлення викидів з 3 різними підходами машинного навчання. Результати дослідження вказують, що при відповідному підході й процесі виявлення викидів якість даних можна підвищити, а отже, й ефективність прогнозування також. Крім того, це дослідження також показує, що використання кількох методів виявлення викидів, а не одного, може дати кращі результати. При цьому ефективність методів виявлення викидів може різнитися залежно від набору даних й моделі прогнозування, що використовується. З цієї причини необхідність збільшення кількості процесів, що відводяться на попередню обробку даних, та використання альтернативних моделей є необхідністю для цієї області.

У роботі [10] проведено дослідження різних методів виявлення викидів з використанням набору даних об'єктів нерухомості у провінції Стамбул (Туреччина). Моделі були протестовані на позначеному наборі даних. Найкращі результати отримані за допомогою однокласового методу опорних векторів (OCSVM) та середнього *k*-найближчого сусіда (AveKNN). Після усунення виявлених відхилень створена загальна система формування діапазонів цін на квадратний метр вартості нерухомості, яка буде використовуватися у різних сферах, таких як інвестиції, податковий аудит, виявлення неправдивих оголошень на сайтах нерухомості.

Для підвищення точності прогнозування цін на нерухомість запропоновано гібридний підхід, який поєднує виявлення викидів, вибір ознак та методи кластеризації [11]. Цей підхід інтегрує Support Vector Regressor (SVR) та LightGBM, використовуючи їхні сильні сторони. Ефективність гібридного підходу оцінювалася за допомогою таких метрик, як середня абсолютна похибка (MAE), середньоквадратична похибка (RMSE) та середня абсолютна відсоткова похибка (MAPE). Результати дослідження показують, що гібридний підхід перевершує інші традиційні експерименти з нормальним прогнозуванням, вибором ознак та виявленням викидів, досягаючи нижчих значень MAE, RMSE та MAPE, що вказує на підвищену точність прогнозування цін на нерухомість.

Окремі дослідження для аналізу даних нерухомості використовують статистичні методи. Зокрема, у роботі [12] порівнюється механізм двох окремо побудованих статистичних методів виявлення викидів в аналізі ринку нерухомості. Для цього автори використали статистичні інструменти, які називаються методом Баарда та аналізом залишків моделей. Згодом відбулося порівняння отриманих результатів оцінки параметрів аналізованих моделей та показників їх якості до та після видалення викидів. Отримані результати свідчать про те, що алгоритми обраних статистичних методів, виявляючи викиди, дозволяють усунути меншу їх кількість, при цьому отримати покращення параметрів функціональної моделі та її адаптацію до аналізованого набору даних.

Наявність навіть одного викиду в вибірковій оцінці може мати відчутні наслідки для моделей регресії, отриманих за допомогою методу найменших квадратів. У дослідженні [13] виявлення викидів проводилося за допомогою стійкої регресії, яка використовує метод найменшої медіани квадратів залишків (LMS). За допомогою спеціального програмного забезпечення проведено розрахунки на вибірці проданих будинків в одному з районів міста Барі, Італія. Експеримент показав, що регресійна модель, яка спочатку мала бути б відхилена, натомість показала чудову продуктивність після того, як із вибірки було видалено всі викиди, визначені LMS.

У роботі [14] описано дослідження, яке охоплювало райони з різними рівнями соціально-економічного розвитку у провінціях Стамбул та Коджаелі в Туреччині. Підготовлені дані включали близько 200 тис. оцінок нерухомості за 121 критерієм. Спершу набори даних були оптимізовані за допомогою техніки Boxplot щодо викидів на основі набору даних, а також методів кластерного аналізу та аналізу викидів на основі розташування. Далі за допомогою методу кореляції Пірсона шляхом аналізу локального зв'язку між вартістю нерухомості та критеріями були визначені 22 критерії, що впливають на вартість. Згодом на основі результатів аналізу просторово обмеженого багатовимірного кластерного аналізу (SCMCA) було виявлено п'ять різних географічних кластерів зі схожими характеристиками значень соціально-економічного розвитку. Результати дослідження показали, що моделі прогнозування вартості, побудовані на географічних кластерах, виявилися більш успішними, ніж модель всієї досліджуваної території.

Метою дослідження [15] була перевірка корисності методів виявлення викидів у контексті покращення результатів класифікації об'єктів нерухомості по групах, які схильні до збільшення або зменшення податкового навантаження. Об'єктом дослідження стали 524 земельні ділянки у Польщі. Спочатку автор провів оцінку моделі логістичної регресії всього набору аналізованих об'єктів нерухомості. Потім було застосовано чотири методи виявлення викидів (kNN, IForest, LOF та PCA) і

оцінка моделі була повторена без спостережень, тобто без об'єктів нерухомості, що помічені як аномальні. Дана процедура показала, що всі застосовані методи виявлення викидів покращили точність класифікації об'єктів нерухомості.

Поданий вище аналіз публікацій за напрямком нашого дослідження охоплював роботи, в яких детектування аномалій проводилося на наборах даних, пов'язаних з нерухомістю. Проте існує ряд праць, де проводиться порівняння й оцінка роботи алгоритмів виявлення аномалій на наборах даних із різних областей, включаючи синтетичні набори даних. Розглянемо кілька із них.

У роботі [16] проведено порівняння середньої точності, надійності, часу виконання обчислень та використання пам'яті 14 алгоритмів виявлення викидів без вчителя на синтетичних і реальних наборах даних з різних областей, включаючи виявлення шахрайства, виявлення вторгнень, медична діагностика та очищення даних. Вибрані методи реалізовані на мовах Python, R і Matlab та піддавалися ретельним тестам на масштабованість. Дослідження показало, що алгоритм IForest є чудовим методом для ефективного виявлення викидів, демонструючи при цьому відмінну масштабованість на великих наборах даних разом із прийнятним використанням пам'яті для наборів даних до одного мільйона зразків. Хороші результати у тестах продемонстрував алгоритм OCSVM, але він не підходить для великих наборів даних. Алгоритм SOD показав хорошу продуктивність виявлення викидів й ефективно обробляв масиви даних великої розмірності, проте ціною поганої масштабованості. Алгоритми LOF, ABOD, GWR, KL і LSA мали найнижчу продуктивність, при цьому три перших методи показали ще й погану масштабованість. Автори роботи зазначають, що хоча охоплення їхнього дослідження є достатнім для вирішення більшості проблем виявлення викидів, для обмежених середовищ можна вибирати спеціальні алгоритми.

Проблему порівняльного аналізу методів виявлення викидів без вчителя можна вирішити за допомогою повністю синтетичних даних, де викиди є характерними відхиленнями від звичайних випадків. У роботі [17] пропонується загальний процес генерації наборів даних для такого порівняльного аналізу. Основна ідея цього дослідження полягає в тому, щоб реконструювати звичайні екземпляри на основі існуючих реальних порівняльних даних, одночасно генеруючи викиди, щоб вони демонстрували глибокі інформативні характеристики. Тест із використанням сучасних методів виявлення викидів підтвердив практичність отриманих результатів.

Загалом, описані тести мають такі обмеження:

– дослідження сконцентровані на методах виявлення аномалій без вчителя;

– аналіз продуктивності алгоритмів щодо типів аномалій (локальні, глобальні) є неповним;

– відсутні дані щодо надійності моделей при зашумленості даних чи невідповідних ознаках;

– відсутність охоплення більш складних, великих, структурованих або неструктурованих наборів даних, які сьогодні привертають широку увагу та використовуються в задачах комп'ютерного зору (CV, Computer Vision) та обробки природної мови (NLP, Natural Language Processing).

У роботі [18] проведено найбільш повне тестування та оцінка продуктивності 30 алгоритмів виявлення аномалій на 57 контрольних наборах даних, з яких 47 є існуючими, а 10 – створено авторами. Автори дослідження сконцентрували увагу на трьох критичних аспектах порівняння: доступність навчання, типи аномалій, стійкість алгоритму в умовах шуму й пошкодження даних.

Дане дослідження вказує на наступні важливі моменти:

– жоден із протестованих алгоритмів без вчителя статистично не кращий за інші, що підкреслює важливість вибору алгоритму;

– при всього лише 1 % помічених аномалій більшість методів з частковим залученням вчителя можуть перевершити кращий метод без вчителя, що підтверджує важливість навчання;

– найкращі методи без вчителя для певних типів аномалій навіть кращі за методи з вчителем чи частковим залученням вчителя, що свідчить про необхідність розуміння характеристик даних;

– методи з частковим залученням вчителя демонструють хорошу надійність при дослідженні зашумлених та пошкоджених даних, ймовірно, завдяки їх ефективності у використанні міток та виборі ознак.

Вихідний код тесту ADBench відкритий та доступний за посиланням [19] з метою порівняльного аналізу нових пропонованих методів.

З огляду на зазначене, доцільність проведення досліджень у напрямку виявлення викидів під час оцінки нерухомості за допомогою машинного навчання є обґрунтованою й має на меті вибір оптимальних алгоритмів для сфери нерухомості, оптимізацію існуючих методів, підвищення їх точності, використання нових моделей та алгоритмів, розвиток масштабованих методів.

Виклад основного матеріалу

1. Збір даних

В рамках цього дослідження для аналізу інформації по нерухомості обрано місто Тернопіль – обласний центр на заході України з населенням близько 225 тис. чол. Для отримання даних та підтримки актуальності набору даних авторами було розроблено програмне забезпечення у вигляді додатку Windows (рис. 1), бази даних Microsoft SQL Server та веб-сайту [20]. Даний програмний комплекс створений більше 10 років тому, пройшов успішну апробацію ріелторами, врахувавши їх побажання та специфіку роботи та успішно використовується рядом агенцій нерухомості м. Тернополя.

Описане програмне забезпечення дозволяє автоматизувати роботу агентств нерухомості й містить функціонал, який шляхом парсингу Інтернет сторінок проводить збір даних по об’єктах нерухомості із сайтів DIM.RIA [21] та OLX.ua [22].

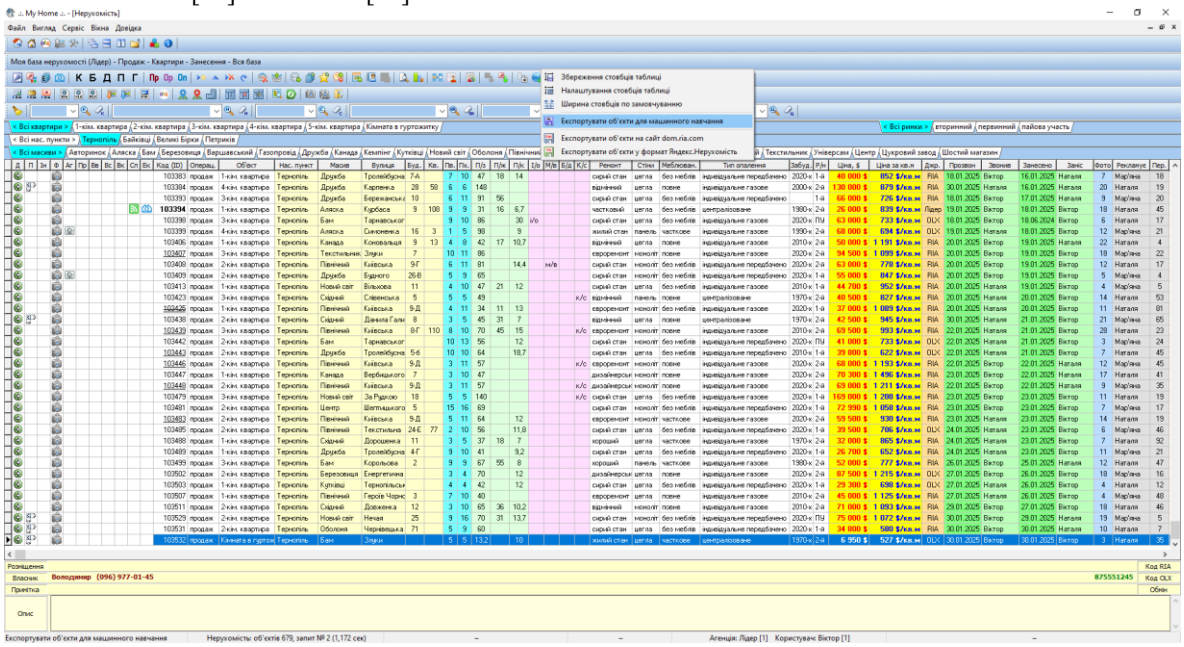


Рис. 1. Загальний вигляд головного вікна розробленої програми

Загалом, для дослідження використано дані, які знаходилися у базі станом на початок 2025 року. Набір даних містить опис 760 пропозицій продажу квартир та кімнат в гуртожитках у м. Тернополі. Актуальність кожного об'єкта а також відсутні в оголошенні (джерелі інформації) характеристики уточнювалися у власника. У базу даних не додавали об’єкти, які рекламують агентства нерухомості, адже у ціну таких об’єктів можуть бути включені комісійні агенції, що може спотворювати точність моделей. Для формування набору даних, який в подальшому буде використовуватися для машинного навчання, програмне забезпечення було доповнено функцією експорту необхідних полів у формат CSV.

2. Опис набору даних

Сучасні портали нерухомості та CRM системи для рієлторської діяльності при описі об’єктів нерухомості пропонують вводити до 200 його описових тегів, з них обов'язкових – до 15 [23]. Розроблене нами програмне забезпечення при описі квартир та кімнат містить близько 50 параметрів. Для аналізу інформації методами машинного навчання з програми ми експортуємо 14 стовбців (табл. 2.). Перший стовбець – це ID об'єкта (унікальний ідентифікатор), який в подальшому ми використовуємо лише в допоміжних цілях, оскільки він не має інформаційної цінності. Наступні 12 стовбців – це характеристики (ознаки), що описують об’єкт нерухомості. Останній стовбець – мітка аномальності (1 – аномальний об’єкт, 0 – нормальний об’єкт).

Таблиця 2

Опис набору даних

ID поля	Позначення поля	Тип поля	Опис
0	id	ціле число / неперервне	ID об'єкта у базі
1	realty_type	строка / дискретне	Тип нерухомості
2	district	строка / дискретне	Район міста (житловий масив)
3	total_area	ціле число / неперервне	Загальна площа, м^2
4	floor	ціле число / неперервне	Поверх, на якому розташована нерухомість
5	floors	ціле число / неперервне	Загальна кількість поверхів у будинку
6	repair_state	строка / дискретне	Стан ремонту
7	wall_material	строка / дискретне	Матеріал зовнішніх стін
8	furniture	строка / дискретне	Меблювання
9	heating	строка / дискретне	Тип опалення
10	build_year	строка / дискретне	Роки забудови
11	market	строка / дискретне	Ринок нерухомості
12	price	ціле число / неперервне	Ціна об'єкта, $
13	label	бінерне / дискретне	Мітка аномальності об’єкта

Мітка аномальності кожного об'єкту нерухомості у наборі даних оцінювалася експертом на підставі аналізу розташування об'єкта, його характеристик та вартості на етапі додавання об’єкта у базу або зміни його ціни (рідше – стану). Для цього було створено набір правил, які дозволяли виявляти аномальні значення, що суттєво відрізняються від середніх показників по регіону. Наприклад, об'єкти з надто низькою або високою вартістю у порівнянні з іншими об'єктами аналогічного типу та розташування вважалися потенційними аномаліями. При аналізі враховувалося, що ціна 1 кв.м. житла зазвичай є вищою для квартир невеликої площі, переважно однокімнатних. Окрім того, враховувалися нетипові поєднання характеристик, такі як висока вартість квартир у будинках старішої забудови, переважно 80-х років і раніше. У розпорядженні експерта по кожному об’єкту були його фотографії, в середньому від 5 до 20 шт., в окремих випадках – копія техпаспорту, за якими перевірялася введена у базу інформацію. Поточний стан бази даних доступний за посиланням [24]. Загалом, база даних була експортована в CSV формат 4 рази – з поміткою аномальності у 10, 15, 20 та 25 % об’єктів.

3. Процедура дослідження

Тестування ефективності методів виявлення викидів проводилося на мові програмування Python із використанням інструменту Spyder IDE, що входить в пакет програмних продуктів Anaconda. Даний програмний продукт спеціалізується на аналізі даних, машинному навчанні та наукових обчисленнях, що робить його зручним середовищем для тестування алгоритмів виявлення викидів. Spyder IDE надає широкі можливості для роботи з кодом, включаючи інтегрований редактор, налагоджувач та засоби візуалізації даних. Обробку даних виконано із використанням бібліотек pandas, numpy, sklearn та pyod, візуалізацію – за допомогою бібліотеки matplotlib.

Кожен експортований CSV файл з даними після завантаження в програму на Python та перед обробкою алгоритмами виявлення аномалій проходив попередню обробку, яка включала:

– виокремлення в окремі одномірні масиви значення полів «id» та «label»;

– видалення повторюваних рядків;

– видалення стовбців з відсутніми даними;

– кодування категоріальних ознак;

– масштабування даних.

Розглянемо детальніше особливості реалізації двох останніх кроків.

1. Кодування категоріальних ознак – це процедура, яка виконує перетворення категоріальних даних у числові значення за певними правилами. Під категоріальними даними розуміють такі поля набору даних, які можуть приймати одне значення із кількох можливих, тобто відносять результат спостереження до певної категорії на основі якісних атрибутів. У нашому наборі даних всі це дискретні поля (табл. 2). Більша частина класичних моделей машинного навчання використовує у своїй роботі саме числові ознаки, тому виникає необхідність правильного відображення категоріальних значень у числовому форматі. Варто зазначити, що не всі категоріальні дані одинакові. Умовно їх можна поділити на номінальні та порядкові. Номінальні дані є чисто описовими, без будь-якого порядку. У нашому наборі даних стовбець «Матеріал зовнішніх стін» містить такі значення: «цегла», «панель», «камінь» та ін. Одне значення не «більше» іншого, вони просто різні. З іншого боку, порядкові дані мають властивий їм порядок. Стовбець «Роки забудови» містить наступні значення: «1950-х років», «1960-х років», «1970-х років» і т.д. Хоча ми все ще маємо справу з категоріями, тут є природна прогресія. Ця відмінність відіграє важливу роль при виборі того чи іншого алгоритму, оскільки неправильне кодування може приводити до невірної інтерпретації даних алгоритмом.

Для кодування категоріальних ознак в машинному навчанні застосовують різноманітні методи [25]. У нашому досліджені використано два найпопулярніші із них, а саме:

а) кодування міток (Label Encoding) – кожній категорії в категоріальній змінній присвоюється унікальне ціле число. Кодувальник сортує по алфавіту унікальні значення стовбця а потім присвоює їм порядковий номер. Перше унікальне значення кодується нулем, друге одиницею, і так далі, останнє кодується числом, що дорівнює кількості унікальних значень мінус одиниця. Перевагою методу є економія пам’яті внаслідок заміни категоріальних значень цілими числами. Недоліком методу є введення порядкового зв'язку між рівнями, який, ймовірно, відсутній у даних. Це може спричинити проблеми в лінійних моделях чи моделях глибокого навчання через хибну інтерпретацію чисел.

б) однократне кодування (One-Hot Encoding, кодування з одним активним станом) – ґрунтується на створенні бінарних ознак, що показують приналежність до унікального значення, тобто кожна категорія в категоріальній змінній представлена у вигляді двійкового вектора, де лише один елемент має значення 1, а решта – значення 0. Таке подання даних гарантує, що категоріальна змінна не накладає жодного порядкового зв’язку між категорії, оскільки наявність 1 у певній позиції вказує на приналежність до цієї категорії. Недоліком методу є збільшення розмірності даних при великій кількості категорій та розрідження матриць, якщо в наборі даних є рідкісні категорії.

Результати оцінки ефективності 14 методів кодування категоріальних даних, виконані у роботі [25], вказують, що правильний вибір методів категоріального кодування даних вважається одним із найважливіших етапів у проектуванні моделі машинного навчання, а найкращий метод слід визначати окремо для кожного дослідження.

2. Стандартизація (масштабування) набору даних є звичайною ланкою у попередній обробці

даних для багатьох алгоритмів машинного навчання. Це пов'язано із тим, що окремі ознаки, у нашому випадку це «total_area» (загальна площа) та «price» (ціна об'єкта), мають дуже різні масштаби та містять дуже великі значення у порівнянні з іншими ознаками. Ці дві характеристики призводять до труднощів у візуалізації даних і, що важливіше, вони можуть погіршити прогностичну продуктивність багатьох алгоритмів машинного навчання. Немасштабовані дані також можуть уповільнити або навіть запобігти збіжності деяких алгоритмів на основі градієнту. Зазвичай стандартизація робиться шляхом виділення середнього значення та масштабування до одиничної дисперсії. Однак викиди часто можуть негативно впливати на середнє значення або дисперсію вибірки. У таких випадках використання медіани та інтерквартильного діапазону часто дає кращі результати. За наявності у наборі даних викидів хорошим варіантом є використання RobustScaler, який реалізований у нашому дослідженні [26]. Він проводить масштабування ознак з використанням статистичних методів, стійких до викидів. Цей масштабувальник видаляє медіану та масштабує дані відповідно до квантильного діапазону. За замовчуванням міжквартильний розмах IQR – це діапазон між 1-м квартилем (25 квантиль) та 3-м квартилем (75 квантиль). Центрування та масштабування відбуваються незалежно для кожної ознаки шляхом обчислення відповідних статистичних даних на зразках у навчальному наборі. Потім медіана та інтерквартильний діапазон зберігаються для подальшого використання в даних за допомогою методу перетворення. Порівняння впливу різних масштабувальників на дані з викидами описано у публікації [27].

Дослідження методів виявлення викидів проводилося з використанням бібліотеки PyOD (Python Outlier Detection). Бібліотека створена в 2017 році й стала популярним пакетом Python з відкритим вихідним кодом для виконання масштабованого виявлення викидів у багатовимірних даних. Вона включає понад 50 алгоритмів – від класичних до найсучасніших. Бібліотека PyOD успішно використовується як в академічних дослідженнях, так і в комерційних продуктах. Вона проста у використанні та має добре уніфіковану документацію з прикладами [7].

Для тестування й аналізу обрано 23 алгоритми, що побудовані на основі 4-х типів моделей:

а) ймовірнісні моделі (probabilistic) – передбачають, що нормальні дані мають вищу ймовірність за певним ймовірнісним розподілом або моделлю, тоді як аномальні точки мають значно меншу ймовірність;

б) моделі на основі близькості (proximity-based) – передбачають, що нормальні точки зазвичай мають багато сусідів поблизу, тоді як аномальні точки або викиди віддалені від інших точок й сусідів у них мало;

в) лінійні моделі (linear) – передбачають побудову лінійної моделі, що описує зв'язок між вхідними та вихідними даними, а потім кожна точка даних оцінюється, наскільки добре вона підходить до моделі. Точки, які мають великі відхилення (різницю між реальним значенням та передбаченим моделлю), вважаються аномаліями;

г) моделі на основі графів (graph-based) – передбачають побудову графу для моделювання взаємозв'язків між точками даних, де точки даних розглядаються як вершини графа, а подібності або відстані між ними визначають ребра. Аномалії ідентифікуються як точки, які демонструють незвичні зв'язки або структури у графі.

Повний перелік досліджуваних алгоритмів та їх характеристик подано в табл. 3.

Таблиця 3

Досліджувані алгоритми виявлення аномалій

Познач.	**Розшифрування**	**Опис**	**Рік**	**Багато-ядерність**	**Детермі-нованість**	**Джерело**
Ймовірнісні моделі						
ECOD	Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions	Метод виявлення викидів без вчителя із використанням емпіричних кумулятивних функцій розподілу	2022	Так	Так	[29]
COPOD	Copula-Based Outlier Detection	Детектор викидів на основі копули	2020	Так	Так	[30]
Sampling	Rapid distance-based outlier detection via sampling	Метод швидкого виявлення викидів на основі відстані за допомогою вибірки	2013	Ні	Ні	[31]
SOS	Stochastic Outlier Selection	Стохастичний відбір викидів	2012	Ні	Так	[32]
FastABOD	Fast Angle-Based Outlier Detection using approximation	Швидке виявлення викидів на основі кута з використанням апроксимації	2008	Ні	Так	[33]

Познач.	Розшифрування	Опис	Рік	Багато-ядерність	Детермі-нованість	Джерело
KDE	Outlier Detection with Kernel Density Functions	Виявлення викидів за допомогою функцій густини ядра	2007	Ні	Так	[34]
QMCD	Quasi-Monte Carlo Discrepancy outlier detection	Виявлення викидів з використанням дискретності на базі методу квазі Монте Карло	2001	Ні	Так	[35]
GMM	Gaussian Mixture Modeling for Outlier Analysis	Модель Гаусової суміші для виявлення викидів	1992	Ні	Так	[36]
Моделі на основі близькості						
ROD	Rotation-based Outlier Detection	Виявлення викидів на основі обертання	2020	Ні	Так	[37]
HBOS	Histogram-based Outlier Score	Оцінка викидів на основі гістограми	2012	Ні	Так	[38]
SOD	Subspace Outlier Detection	Підпросторове виявлення викидів	2009	Ні	Так	[39]
CBLOF	Clustering-Based Local Outlier Factor	Фактор локального викиду на основі кластеризації	2003	Так	Ні	[40]
COF	Connectivity-Based Outlier Factor	Фактор викиду, побудований на концепції локальної "зв'язності" точок у просторі даних	2002	Ні	Так	[41]
AvgKNN	Average kNN	Різновид kNN алгоритму, який використовує середнє значення відстані до всіх *k*-сусідів для оцінки викиду	2002	Так	Так	[42]
MedKNN	Median kNN	Різновид kNN алгоритму, який використовує медіанне значення відстані до всіх *k*-сусідів для оцінки викиду	2002	Так	Так	[42]
kNN	k Nearest Neighbors	Метод *k*-найближчих сусідів, використовує відстань до *k*-го сусіда для оцінки викиду	2000	Так	Так	[43]
LOF	Local Outlier Factor	Локальний фактор викиду	2000	Так	Так	[44]
Лінійні моделі						
KPCA	Kernel Principal Component Analysis	Метод аналізу головних компонент ядра (різновид PCA)	2007	Так	Так	[45]
PCA	Principal Component Analysis	Метод аналізу головних компонент	2003	Ні	Так	[46]
OCSVM	One-Class Support Vector Machines	Однокласовий метод опорних векторів	2001	Ні	Так	[47]
LMDD	Deviation-based Outlier Detection	Виявлення викидів на основі відхилень	1996	Ні	Ні	[48]
CD	Use Cook’s distance for outlier detection	Виявлення викидів на основі відстаней Кука	1977	Ні	Так	[49]
Моделі на основі графів						
LUNAR	Unifying Local Outlier Detection Methods via Graph Neural Networks	Уніфікований метод виявлення локальних викидів за допомогою графових нейронних мереж	2022	Ні	Ні	[50]

Джерело: сформовано й доповнено на основі [7] та [28].

Звернемо увагу на дві важливі характеристики усіх алгоритмів:

1. Багатоядерність алгоритму передбачає, що при його ініціалізації є можливість задати параметр *n_jobs*. Цей параметр вказує на кількість ядер процесора, які будуть використовуватися паралельно як для операцій навчання (*fit*), так і для прогнозування (*predict*). По замовчуванню, включаючи наше дослідження, використовується лише одне ядро, тобто *n_jobs* = 1. Якщо задати *n_jobs* = -1 – до обробки будуть задіяні всі доступні ядра процесора, що суттєво прискорить роботу алгоритмів на багатоядерних процесорах та зменшить час виконання складних алгоритмів, особливо при роботі з великими наборами даних.

2. Детермінованість алгоритму означає, що при однакових вхідних даних алгоритм завжди видає однаковий результат. У контексті виявлення аномалій детермінованість може впливати на відтворюваність та стабільність роботи моделі.

Для кожного із двох кодувальників категоріальних ознак по порядку оброблялися набори даних з поміткою аномальності у 10, 15, 20 та 25 % об'єктів. Відповідно для всіх алгоритмів встановлювався параметр забруднення *contamination*, який задає частку викидів у наборі даних, тобто 0,10, 0,15, 0,20 та 0,25. Тестування роботи алгоритмів на кожному наборі повторювалося 5 разів. Середнє значення по 5 випробуваннях вважалося кінцевим результатом експерименту. Для всіх досліджуваних алгоритмів використовувалися їх стандартні налаштування гіперпараметрів для справедливого порівняння.

Алгоритм PCA при однократному кодуванні категоріальних ознак не працює з набором даних, видаючи помилки. Причиною цього може бути те, що перетворення категоріальних ознак на бінарні вектори приводить до значного збільшення розмірності даних (набір даних стає розрідженим). Дослідження роботи алгоритму PCA на предмет появи помилок не дало ніякого обґрунтованого результату. Тому дані по алгоритму PCA при однократному кодуванні в таблицях результатів не приводяться. Таким чином, ми виконали 45 різноманітних комбінацій алгоритмів та кодувальників: 23 алгоритми у поєднанні з Label Encoder та 22 алгоритми – з One-Hot Encoder.

Розмір набору даних при різних способах кодування категоріальних ознак наступний:

а) кодувальник Label Encoding – набір даних із 760 спостережень та 12 ознак;

б) кодувальник One-Hot Encoding – набір даних із 760 спостережень та 67 ознак.

Тобто, однократне кодування набору даних збільшило кількість ознак (розмірність) у 5,6 рази.

Експерименти проводилися на ПК з операційною системою Windows 10 Pro x64, процесором Intel(R) Core(TM) i3-10105F 3,70 ГГц та об'ємом оперативної пам'яті 32 Гб.

Встановлення міток аномальності в наборах даних надало можливість провести порівняльний аналіз результатів автоматизованих методів виявлення викидів з експертними оцінками. Це дозволило оцінити точність та ефективність використаних алгоритмів та визначити їхню придатність для реальних задач виявлення аномалій у даних про нерухомість.

Результати роботи оцінювалися за трьома показниками й представлені у відповідних таблицях:

а) AUC-ROC (Area Under the Curve – Receiver Operating Characteristic) – площа під кривою робочої характеристики приймача (табл. 4);

б) P@n (Precision @ Rank n) – точність (частка) серед перших *n* передбачених результатів (табл. 5);

в) час виконання алгоритму (табл. 6).

Таблиця 4

Порівняння результатів роботи алгоритмів за параметром AUC-ROC

Алгоритм	**Кодувальник категоріальних ознак**									
	Label Encoder					**One-Hot Encoder**				
	Відсоток викидів у наборі даних					Відсоток викидів у наборі даних				
	10 %	15 %	20 %	25 %	Сер.зн.	10 %	15 %	20 %	25 %	Сер.зн.
AvgKNN	0,846	0,813	0,765	0,747	**0,793**	0,833	0,801	0,766	0,746	0,787
CBLOF	0,839	0,774	0,722	0,705	**0,760**	0,780	0,774	0,691	0,685	0,733
CD	0,717	0,712	0,679	0,658	0,692	0,683	0,675	0,650	0,638	0,662
COF	0,743	0,763	0,715	0,695	0,729	0,671	0,711	0,710	0,702	0,699
COPOD	0,790	0,760	0,708	0,650	0,727	0,747	0,728	0,685	0,637	0,699
ECOD	0,752	0,724	0,665	0,603	0,686	0,722	0,702	0,656	0,610	0,673
FastABOD	0,851	0,804	0,767	0,753	**0,794**	0,817	0,786	0,752	0,733	0,772
GMM	0,759	0,745	0,698	0,674	0,719	0,679	0,661	0,640	0,633	0,653
HBOS	0,709	0,686	0,650	0,623	0,667	0,693	0,680	0,643	0,603	0,655
KDE	0,815	0,755	0,702	0,668	0,735	0,849	0,804	0,745	0,715	**0,778**
kNN	0,833	0,801	0,746	0,729	0,777	0,841	0,807	0,766	0,746	**0,790**
KPCA	0,534	0,507	0,513	0,511	0,516	0,462	0,453	0,485	0,507	0,477
LMDD	0,768	0,688	0,629	0,612	0,674	0,819	0,764	0,706	0,664	**0,738**
LOF	0,788	0,798	0,721	0,690	**0,749**	0,749	0,759	0,720	0,700	0,732
LUNAR	0,709	0,709	0,675	0,613	0,677	0,637	0,603	0,589	0,662	0,623

Алгоритм	Кодувальник категоріальних ознак									
	Label Encoder					One-Hot Encoder				
	Відсоток викидів у наборі даних					Відсоток викидів у наборі даних				
	10 %	15 %	20 %	25 %	Сер.зн.	10 %	15 %	20 %	25 %	Сер.зн.
MedKNN	0,841	0,809	0,764	0,747	0,790	0,836	0,807	0,769	0,751	**0,791**
OCSVM	0,802	0,765	0,703	0,642	0,728	0,828	0,793	0,708	0,647	**0,744**
PCA	0,759	0,737	0,670	0,607	0,693	–	–	–	–	–
QMCD	0,755	0,727	0,674	0,631	0,697	0,882	0,831	0,734	0,676	**0,781**
ROD	0,761	0,710	0,648	0,594	0,678	0,672	0,654	0,607	0,566	0,625
Sampling	0,762	0,701	0,644	0,643	0,688	0,827	0,721	0,674	0,663	**0,721**
SOD	0,724	0,707	0,689	0,669	0,697	0,538	0,537	0,564	0,561	0,550
SOS	0,657	0,646	0,644	0,644	0,648	0,589	0,574	0,584	0,583	0,583

Таблиця 5

Порівняння результатів роботи алгоритмів за параметром Precision @ Rank n

Алгоритм	Кодувальник категоріальних ознак									
	Label Encoder					One-Hot Encoder				
	Відсоток викидів у наборі даних					Відсоток викидів у наборі даних				
	10 %	15 %	20 %	25 %	Сер.зн.	10 %	15 %	20 %	25 %	Сер.зн.
AvgKNN	0,434	0,412	0,434	0,474	**0,439**	0,461	0,439	0,467	0,500	0,467
CBLOF	0,500	0,425	0,409	0,434	**0,442**	0,412	0,425	0,384	0,414	0,409
CD	0,250	0,298	0,309	0,379	0,309	0,171	0,228	0,276	0,347	0,256
COF	0,276	0,377	0,355	0,395	0,351	0,197	0,316	0,375	0,411	0,325
COPOD	0,329	0,368	0,362	0,390	0,362	0,237	0,290	0,322	0,390	0,310
ECOD	0,263	0,298	0,329	0,363	0,313	0,237	0,237	0,296	0,337	0,277
FastABOD	0,461	0,404	0,447	0,490	**0,451**	0,329	0,404	0,428	0,463	0,406
GMM	0,342	0,360	0,362	0,400	0,366	0,184	0,219	0,303	0,358	0,266
HBOS	0,250	0,281	0,349	0,368	0,312	0,158	0,246	0,283	0,337	0,256
KDE	0,447	0,377	0,395	0,400	0,405	0,474	0,465	0,447	0,490	**0,469**
kNN	0,447	0,412	0,408	0,479	0,437	0,487	0,447	0,447	0,505	**0,472**
KPCA	0,224	0,186	0,235	0,259	0,226	0,105	0,159	0,194	0,259	0,179
LMDD	0,345	0,332	0,362	0,376	0,354	0,474	0,512	0,484	0,483	**0,488**
LOF	0,355	0,430	0,415	0,437	**0,409**	0,303	0,412	0,428	0,437	0,395
LUNAR	0,271	0,342	0,358	0,357	0,332	0,258	0,247	0,283	0,355	0,286
MedKNN	0,421	0,386	0,424	0,492	0,431	0,447	0,474	0,480	0,500	**0,475**
OCSVM	0,434	0,412	0,415	0,390	0,413	0,421	0,447	0,461	0,411	**0,435**
PCA	0,342	0,325	0,349	0,358	0,344	–	–	–	–	–
QMCD	0,303	0,325	0,342	0,368	0,335	0,500	0,500	0,507	0,458	**0,491**
ROD	0,263	0,333	0,349	0,342	0,322	0,171	0,246	0,257	0,295	0,242
Sampling	0,392	0,323	0,320	0,388	0,356	0,442	0,384	0,374	0,400	**0,400**
SOD	0,329	0,360	0,421	0,437	0,387	0,184	0,234	0,303	0,365	0,272
SOS	0,224	0,281	0,342	0,390	0,309	0,158	0,202	0,263	0,321	0,236

Таблиця 6

Середній час виконання алгоритмів, мс

Алгоритм	Кодувальник категоріальних ознак		Алгоритм	Кодувальник категоріальних ознак		Алгоритм	Кодувальник категоріальних ознак	
	Label Encoder	One-Hot Encoder		Label Encoder	One-Hot Encoder		Label Encoder	One-Hot Encoder
AvgKNN	8,9	7,4	HBOS	3,7	13,1	OCSVM	41,9	47,3
CBLOF	4,9	7,5	KDE	44,6	103,1	PCA	2,4	–
CD	15,2	348,7	kNN	8,6	8,0	QMCD	3,9	12,4
COF	183,7	350,1	KPCA	292,3	287,7	ROD	2 410,0	153 721,8
COPOD	2,8	7,7	LMDD	1 291,1	6 403,7	Sampling	1,3	2,5
ECOD	3,3	8,3	LOF	11,2	7,2	SOD	168,3	217,6
FastABOD	95,8	116,9	LUNAR	3 762,1	3 499,0	SOS	360,5	348,0
GMM	3,8	7,6	MedKNN	8,8	8,4	–	–	–

4. Аналіз отриманих результатів

Для оцінки якості бінарних класифікаційних моделей, зокрема в задачах пошуку аномалій, використовується метрика AUC-ROC. Вона відображає здатність моделі розділяти дані на два класи (нормальні та аномальні дані) за різних порогів класифікації. AUC-ROC – це площа під кривою ROC, тобто під графіком, що показує залежність істинно-позитивного результату TPR (True Positive Rate) від помилково-позитивного результату FPR (False Positive Rate). Ці показники обчислюються за формулами [51]:

$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN},$$

де *TP* (True Positives) – кількість аномалій, які модель правильно класифікувала як аномалії; *FN* (False Negatives) – кількість аномалій, які модель помилково класифікувала як нормальні; *FP* (False Positives) – кількість нормальних даних, які модель помилково класифікувала як аномалії; *TN* (True Negatives) – кількість нормальних даних, які модель правильно класифікувала як нормальні.

Крива ROC будується на основі змінного порогу класифікації, де кожен поріг дає нове значення TPR і FPR.

Оцінка якості моделей за показником AUC-ROC є наступною [52]:

а) AUC-ROC ≥ 0,9 – відмінна якість;

б) 0,8 ≤ AUC-ROC < 0,9 – хороша якість;

в) 0,7 ≤ AUC-ROC < 0,8 – прийнятна (задовільна) якість;

г) 0,5< AUC-ROC < 0,7 – низька якість;

д) AUC-ROC = 0,5 – відповідає випадковому вгадуванню.

З таблиць 4 та 5 виберемо алгоритми та кодувальники, для яких AUC-ROC ≥ 0,7 та P@n ≥ 0.4. Доповнимо результати середнім часом виконання алгоритмів з таблиці 6, просортуємо вибірку в порядку спадання AUC-ROC та зведемо ці дані в результуючу таблицю 7.

Таблиця 7

Рейтинг алгоритмів виявлення аномалій, що показали найкращі результати

Алгоритм	Детермі-нованість	Кодувальник	Середнє значення AUC-ROC	Середнє значення Precision @ Rank n	Середній час виконання, мс
FastABOD	Так	Label Encoder	0,794	0,451	95,8
AvgKNN	Так	Label Encoder	0,793	0,439	8,9
MedKNN	Так	One-Hot Encoder	0,791	0,475	8,4
kNN	Так	One-Hot Encoder	0,790	0,472	8,0
QMCD	Так	One-Hot Encoder	0,781	0,491	12,4
KDE	Так	One-Hot Encoder	0,778	0,469	103,1
CBLOF	Ні	Label Encoder	0,760	0,442	4,9
LOF	Так	Label Encoder	0,749	0,409	11,2
OCSVM	Так	One-Hot Encoder	0,744	0,435	47,3
LMDD	Ні	One-Hot Encoder	0,738	0,488	6 403,7
Sampling	Ні	One-Hot Encoder	0,721	0,400	2,5

Із 45 протестованих комбінацій алгоритм-кодувальник лідерами тестування є три алгоритми: FastABOD, група kNN (включаючи AvgKNN, MedKNN) та QMCD. Розглянемо особливості кожного із них:

а) FastABOD – найкраща AUC-ROC при низькій швидкодії та точності (P@n);

б) група алгоритмів kNN– дещо нижча AUC-ROC, вища точність у порівнянні з FastABOD та найкраща швидкодія, яка практично не залежить від типу кодувальника (див. табл. 6);

в) QMCD – незначно поступається попередникам у AUC-ROC, проте забезпечує найвищу точність й оптимальну швидкодію.

Решту алгоритмів з табл. 7 поступаються показниками відносно трьох попередніх. Виключенням є алгоритм LMDD, у якого теж висока точність P@n, проте цей алгоритм дуже повільний.

Варто зауважити, що всі представлені в таблиці 7 алгоритми при частці викидів 10 % (значення по замовчуванню для всіх алгоритмів) й використанні оптимального для алгоритму кодувальника забезпечують AUC-ROC на рівні 0,819 – 0,882 (крім LOF – 0,788), що вважається хорошим результатом.

Для візуальної оцінки якості роботи та порівняння різних алгоритмів виявлення аномалій в багатовимірних даних використовуються методи зменшення розмірності, які дають можливість відобразити такі дані на 2D або 3D площині. Для цього найчастіше застосовують декілька популярних методів: PCA (Principal Component Analysis) [53], t-SNE (t-distributed Stochastic Neighbor Embedding) [54] та UMAP (Uniform Manifold Approximation and Projection) [55].

Метод головних компонент (PCA) дозволяє лінійно зменшити розмірність даних, зберігаючи якомога більше дисперсії, що дає змогу отримати загальне уявлення про структуру даних. t-SNE (T-

розподілене вкладення стохастичної близькості), у свою чергу, ефективно розділяє кластери, відображаючи локальну структуру даних, хоча може втрачати глобальні зв'язки. UMAP (однорідна апроксимація та проекція різноманіття) є більш швидкою та масштабованою альтернативою t-SNE, зберігаючи як локальну, так і глобальну структуру даних.

Візуалізація багатовимірних даних у 2D або 3D просторі допомагає оцінити, наскільки добре алгоритм виявлення аномалій відокремлює нормальні та аномальні точки. Якщо після зменшення розмірності аномальні точки утворюють окремі скупчення або виходять за межі основної маси даних, тобто модель кластеризує нормальні та аномальні точки, це може свідчити про ефективність алгоритму.

У нашому дослідженні використано t-SNE метод зменшення розмірності. Задля забезпечення відтворювальності результатів при ініціалізації t-SNE вказувався параметр *random_state*, який задає початкове значення для генератора випадкових чисел. Якщо цього не робити, розташування множини точок на графіках буде різним при кожній ініціалізації алгоритму. На рис. 2 показано візуалізацію роботи алгоритмів виявлення аномалій, що працюють у парі з Label Encoder, на рис. 3 – з One-Hot Encoder. Частка викидів у наборі даних складала 10 %, що становить 76 аномалій. При цьому окремі алгоритми могли виявляти як більшу їх кількість, так і меншу. Синіми точками позначено нормальні об'єкти, червоними – викиди.

Аналіз графіків вказує на наступне:

1. Спосіб кодування категоріальних ознак впливає на результати візуалізації. При кодуванні Label Encoder графіки мають більшу кількість віддалених один від одного кластерів з високою щільністю точок а викиди знаходяться переважно на околицях цих кластерів. У випадку кодування One-Hot Encoder щільність точок значно нижча, всі об'єкти згруповані у дві великі області, викиди концентруються у двох-трьох місцях також на околицях областей.

2. Візуалізація дозволяє проводити швидку попередню оцінку якості роботи алгоритмів. Для прикладу, алгоритм QMCD з високими метриками (AUC-ROC = 0,882, P@n = 0,500) має чітко окреслені групи аномальних даних. Для порівняння, на рисунку 3ж подано результати роботи алгоритму KPCA в парі з One-Hot Encoder, який в експериментах для набору даних з 10 % аномальності показав найгірший результат: AUC-ROC = 0,462, P@n = 0,105. Візуальне порівняння його з іншими графіками підтверджує, що для даної задачі алгоритм KPCA не підходить.

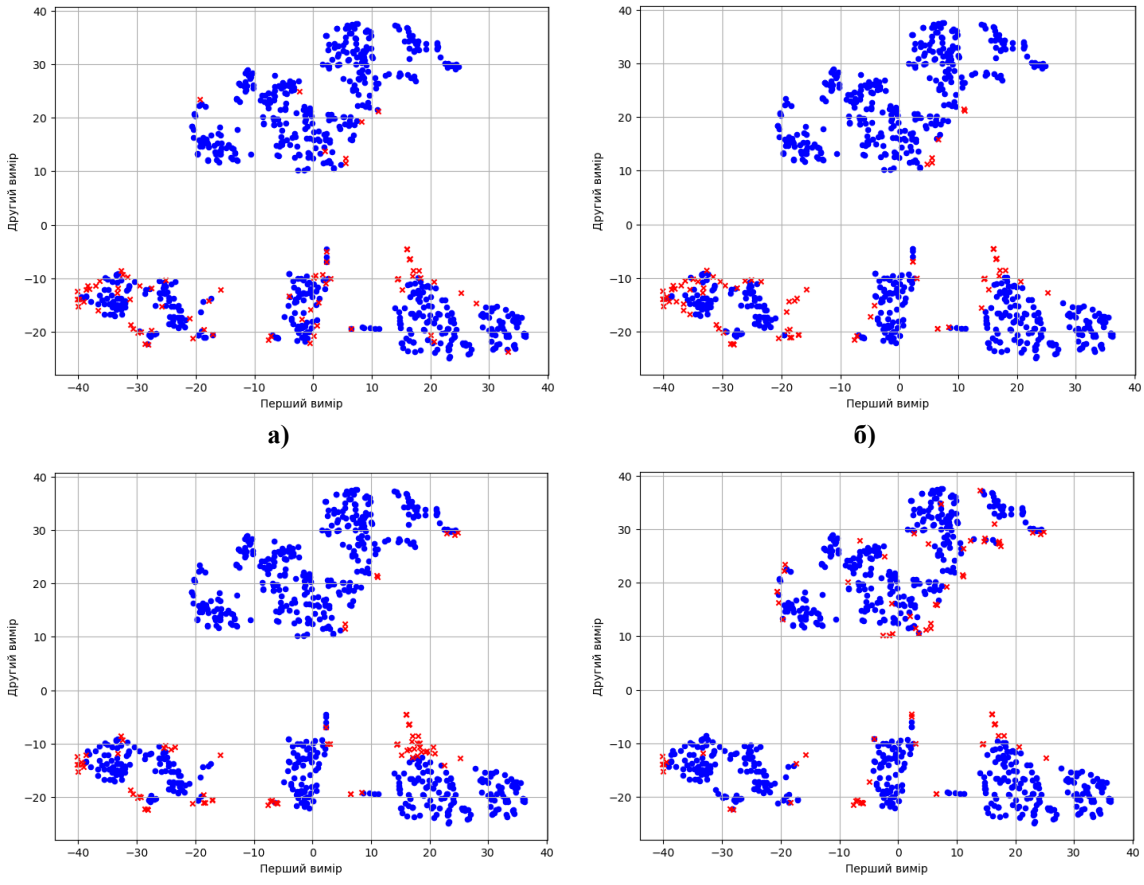


Рис. 2. Візуалізація результатів роботи алгоритмів, що працюють у парі з Label Encoder: а) – FastABOD; б) – AvgKNN; в) – CBLOF; г) – LOF

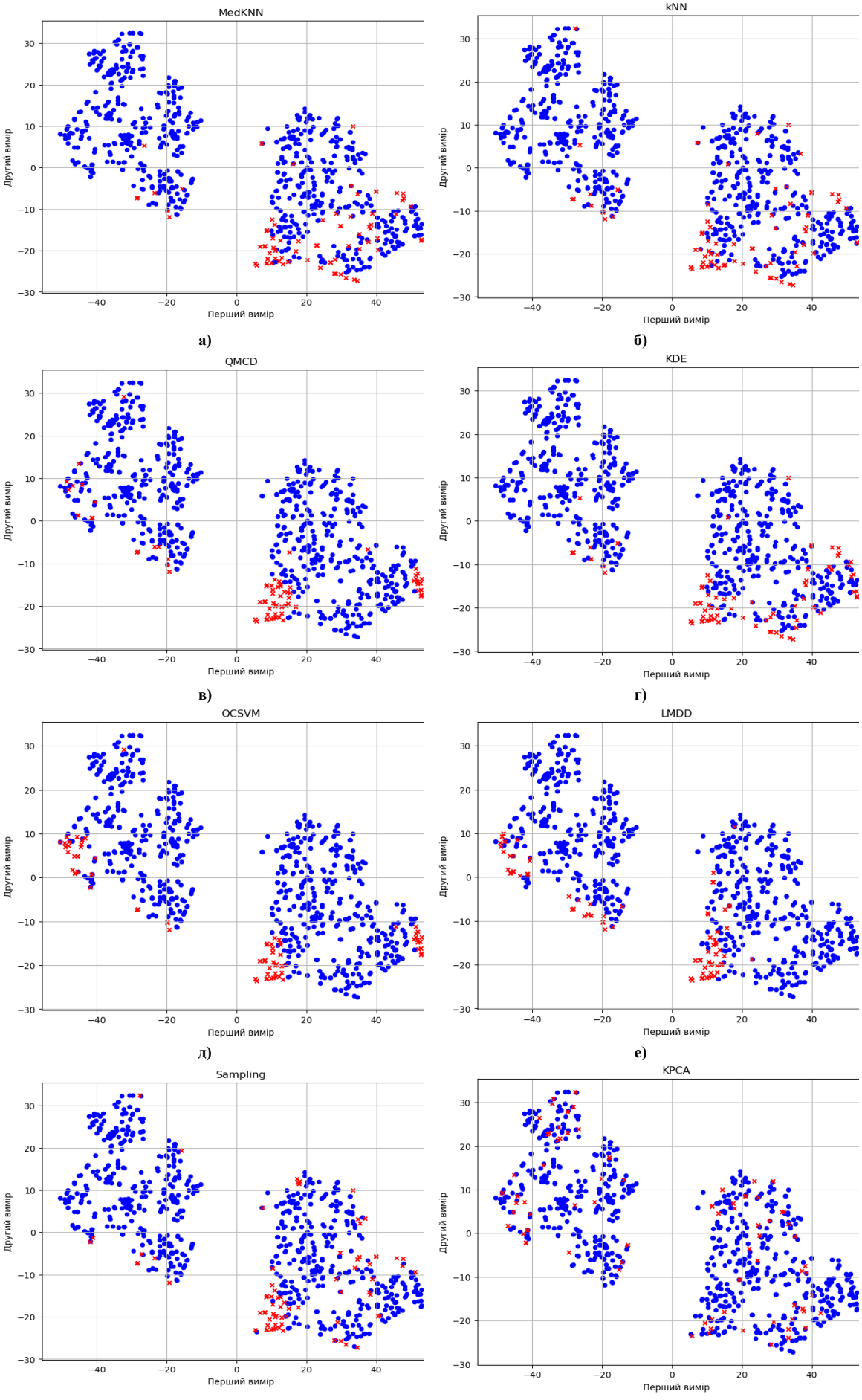
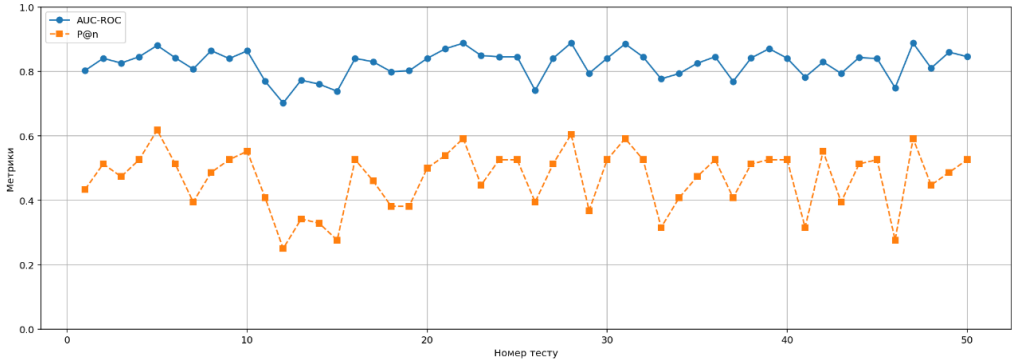
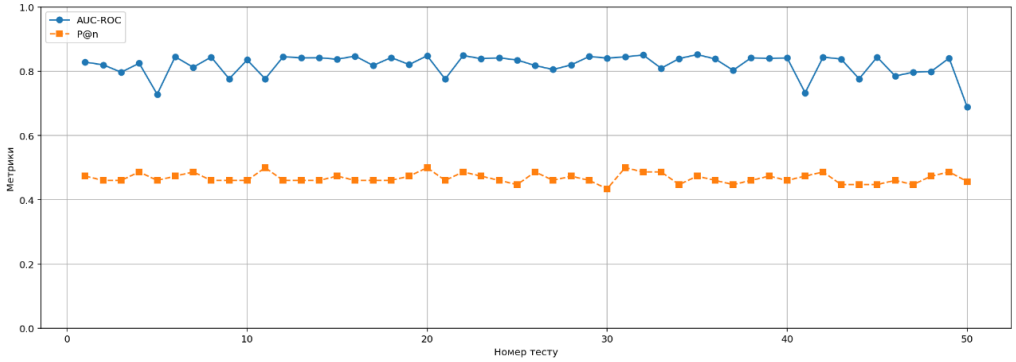


Рис. 3. Візуалізація результатів роботи алгоритмів, що працюють у парі з One-Hot Encoder: а) – MedKNN; б) – kNN; в) – QMCD; г) – KDE; д) – OCSVM; е) – LMDD; є) – Sampling; ж) – KPCA

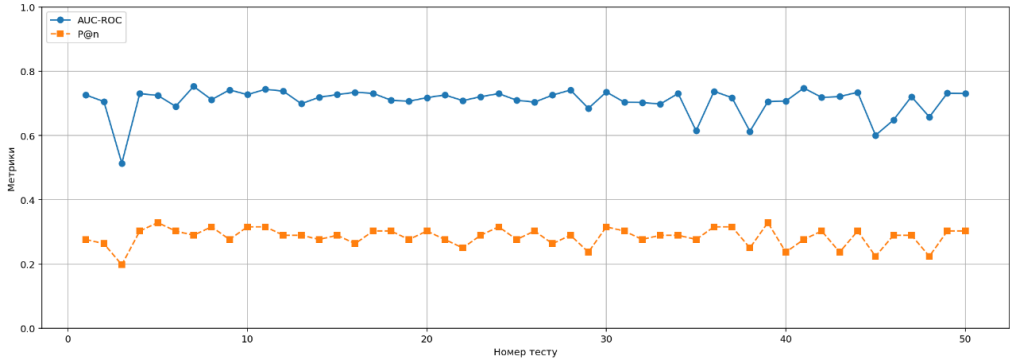
Тестування роботи алгоритмів виявлення аномалій, результати якого подані в таблицях 4 та 5 показало, що недетерміновані алгоритми CBLOF, LMDD, LUNAR та Sampling можуть давати доволі різні результати точності. Було вирішено дослідити ці алгоритми більш детальніше й провести за однакових умов та вхідних даних послідовно 50 тестувань кожного алгоритму. При цьому фіксувалися параметри AUC-ROC та P@n (рис. 4). Узагальнені результати даного етапу роботи подано на діаграмах розмаху метрик (рис. 5). Вони свідчать, що алгоритми LMDD та LUNAR мають незначні коливання метрик та є більш стабільними у своїх прогнозах, тоді як для алгоритмів CBLOF та Sampling ці дані доволі мінливі. Особливо це характерно для алгоритму Sampling, в якого параметр P@n для окремих тестувань може відрізнятися у більш ніж 4 рази. І хоча цей алгоритм був обраний нами як один з найкращих (табл. 7), практичне використання його у дослідженнях є недоцільним.



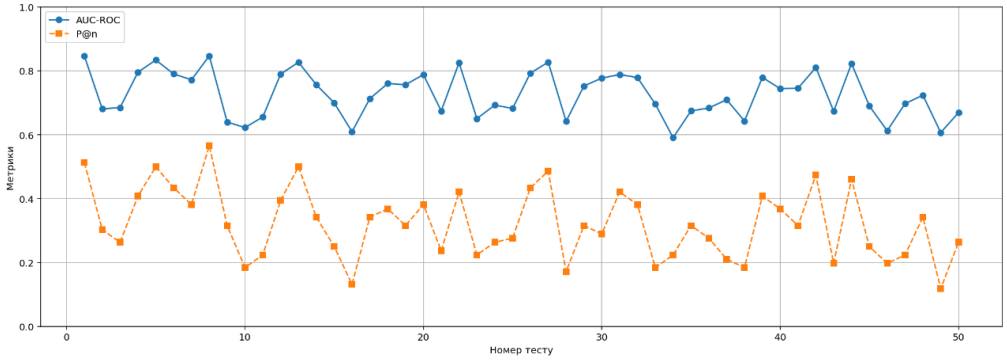
а)



б)

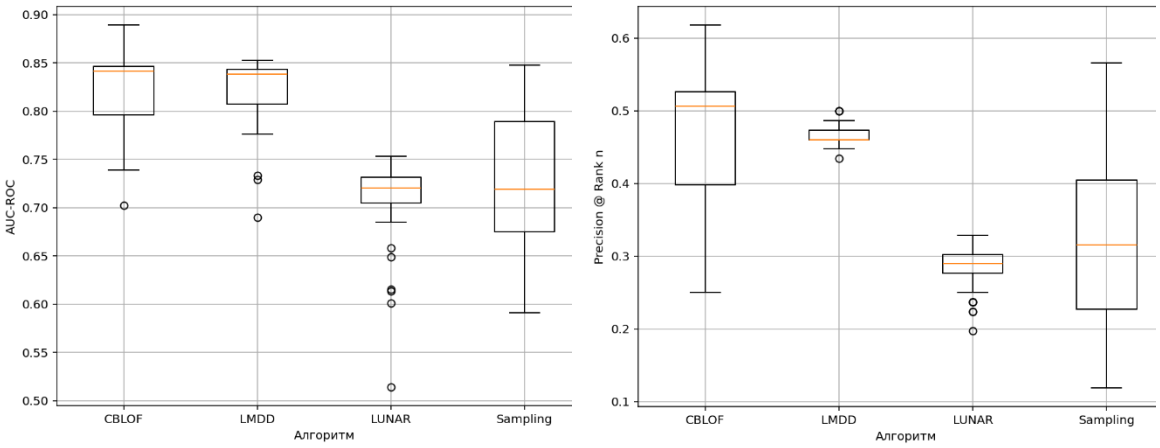


в)



г)

Рис. 4. Тестування роботи недетермінованих алгоритмів:
а) – CBLOF (Label Encoder); б) – LMDD (One-Hot Encoder); в) – LUNAR (Label Encoder); г) – Sampling (One-Hot Encoder)



а) **б)**

Рис. 5. Діаграми розмаху метрик недетермінованих алгоритмів
а) – параметр AUC-ROC; б) – параметр Precision @ Rank n

Висновки

1. Детальний огляд публікацій за тематикою дослідження показав, що ефективність методів виявлення викидів може різнитися залежно від набору даних й моделі прогнозування, що використовується. Окрім того, правильний вибір методів категоріального кодування даних вважається одним із найважливіших етапів у проектуванні моделі машинного навчання, а найкращий метод слід визначати окремо для кожного дослідження.

2. Проведено порівняльний аналіз методів кластерного аналізу виявлення викидів, який дозволив отримати перелік алгоритмів, що показують найкращі середні результати за точністю та швидкістю. Відмінну якість показали алгоритми FastABOD, група kNN (включаючи AvgKNN, MedKNN) та QMCD. Хороший результат забезпечують алгоритми KDE, CBLOF, LOF, OCSVM та LMDD. Ефективність роботи підтверджується візуалізацію результатів роботи алгоритмів з використанням t-SNE методу зменшення розмірності.

3. Встановлено, що кожен алгоритм забезпечує найкращий результат при роботі в парі з певним кодувальником категоріальних ознак. Для моделей AvgKNN, CBLOF, FastABOD та LOF це кодувальник LabelEncoder, для KDE, kNN, MedKNN, LMDD, OCSVM, QMCD – кодувальник OneHotEncoder.

4. Дослідження часу виконання роботи усіх алгоритмів показало, що найповільнішими є алгоритми LMDD, LUNAR та ROD. Середню швидкодію дають алгоритми COF, FastABOD, KPCA, SOD, SOS. Збільшення розмірності матриці набору даних у 5,6 разів при переході з Label Encoder до OneHot Encoder для алгоритму ROD привело до збільшення часу обробки у 64 рази, для CD – у 23 рази, для LMDD – у 5 разів, для решти алгоритмів – не більше ніж у 3,5 рази або час взагалі не змінювався. Цей фактор слід враховувати, якщо комп'ютерна система, у якій планується робота алгоритму, повинна забезпечити максимальну продуктивність.

5. Дослідження роботи недетермійованих алгоритмів показало, алгоритми LMDD та LUNAR мають незначні коливання метрик та є більш стабільними у своїх прогнозах, тоді як для алгоритмів CBLOF та Sampling ці дані доволі мінливі. За таким умов використання алгоритму Sampling з низьким параметром P@n є недоцільним.

6. Подальші дослідження за тематикою даної роботи можуть бути спрямовані на аналіз ефективності ансамблевих методів виявлення аномалій, використання нейронних мереж та комбінації детекторів викидів а також на оптимізацію гіперпараметрів алгоритмів, які у цьому дослідженні показали найкращі результати.

Література

1. Kennedy, B. (2024). *Outlier Detection in Python*. Manning.

2. Muñoz-Garcia, J., Moreno-Rebollo, J. L., Pascual-Acosta, A., & Munoz-Garcia, J. (1990). Outliers: A Formal Approach. *International Statistical Review / Revue Internationale de Statistique*, *58*(3), 215. https://doi.org/10.2307/1403805.

3. Barnett, V., & Lewis, T. (1994). *Outliers in statistical data* (3rd edition). John Wiley & Sons, Chichester.

4. Кучеренко, В., Квач, Я., Сментина, Н., Улибіна, В., & Андрейченко, А. (2013). *Оцінка бізнесу та нерухомості: навч. пос.* (2-ге вид.). Асторопринт.

5. Кучеренко, В., Заєць, М., Захарченко, О., Сментина, Н., & Улибіна, В. (2013). *Оцінка та управління нерухомістю: навч. пос.* ТОВ «Лерадрук».

6. Aggarwal, C. (2017). *Outlier Analysis. Second Edition.* Springer.

7. *Pyod 2.0.3 documentation*. (б. д.). pyod 2.0.3 documentation. https://pyod.readthedocs.io/en/latest/.

8. *Scikit-learn: machine learning in Python — scikit-learn 1.6.1 documentation*. (б. д.). scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation. https://scikit-learn.org/stable/.

9. Çılgın, C., Gökşen, Y., & Gökçen, H. (2023). The Effect of Outlier Detection Methods in Real Estate Valuation with Machine Learning. *İzmir Sosyal Bilimler Dergisi*, 5(1), 9–20. https://doi.org/10.47899/ijss.1270433.

10. Çetiner, M., Dinçsoy, Ö., & Toraman, T. (2020). Outlier Detection for Analysis of Real Estate Price. *In 28th Signal Processing and Communications Applications Conference (SIU)* (pp. 1–4). https://doi.org/10.1109/SIU49456.2020.9302110.

11. Abut, F., Arlı, H., Akay, M., & Adıgüzel, Y. (2023). A New Hybrid Approach for Real Estate Price Prediction Using Outlier Detection, Feature Selection, and Clustering Techniques. *In 8th International Conference on Computer Science and Engineering (UBMK)* (pp. 1–6). https://doi.org/10.1109/UBMK59864.2023.10286673.

12. Śpiewak, B., & Barańska, A. (2020). Chosen statistical methods for the detection of outliers in real estate market analysis. *Acta Scientiarum Polonorum Administratio Locorum*, *19*(2), 109–118. https://doi.org/10.31648/aspal.4608.

13. Morano, P., De Mare, G., & Tajani, F. (2013). LMS for Outliers Detection in the Analysis of a Real Estate Segment of Bari. *Computational Science and Its Applications (ICCSA 2013). Lecture Notes in Computer Science, 7974*. (pp. 457–472). https://doi.org/10.1007/978-3-642-39649-6_33.

14. Sisman, S., & Aydinoglu, A. (2022). Improving performance of mass real estate valuation through application of the dataset optimization and Spatially Constrained Multivariate Clustering Analysis. *Land Use Policy*, *119*, 106–167. https://doi.org/10.1016/j.landusepol.2022.106167.

15. Gnat, S. (2020). Testing the effectiveness of outlier detecting methods in property classification. *Real Estate Management and Valuation, 28(4)*, 81–92. https://doi.org/10.1515/remav-2020-0033.

16. Domingues, R., Filippone, M., Michiardi, P., & Zouaoui, J. (2018). A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition, 74*, 406–421. https://doi.org/10.1016/j.patcog.2017.09.037.

17. Steinbuss, G., & Böhm, K. (2021). Benchmarking Unsupervised Outlier Detection with Realistic Synthetic Data. *ACM Transactions on Knowledge Discovery from Data (TKDD), 15(4)*, 1–20. https://doi.org/10.1145/3441453.

18. Han, S., Hu, X., Huang, H., Jiang, M., & Zhao, Y. (2022). ADBench: Anomaly Detection Benchmark. *In 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*. https://doi.org/10.48550/arXiv.2206.09426.

19. *GitHub - Minqi824/ADBench: Official Implement of "ADBench: Anomaly Detection Benchmark", NeurIPS 2022.* GitHub. https://github.com/Minqi824/ADBench.

20. *Програмне забезпечення для агенцій нерухомості «MyHome»*. Агенція нерухомості «Лідер» м. Тернопіль. https://lider.org.ua/myhome.aspx.

21. *DIM.RIA™ — вся нерухомість України. Продаж і оренда будь-якої нерухомості*. DOM.RIA.com. https://dom.ria.com/uk/.

22. *Оголошення OLX.ua: сервіс оголошень України*. Купівля/продаж б/у та нових товарів, різноманітні послуги на сайті OLX.ua. https://www.olx.ua/uk/.

23. *DIM.RIA - XML-формат*. DOM.RIA.com. https://dim.ria.com/uk/xmldoc/.

24. *Продаж квартир та кімнат в Тернополі та Тернопільському районі*. Агенція нерухомості «Лідер» м. Тернопіль. https://lider.org.ua/base.aspx?t=kva.

25. Bolikulov, F., Nasimov, R., Rashidov, A., Akhmedov, F., & Cho, Y.-I. (2024). Effective Methods of Categorical Data Encoding for Artificial Intelligence Algorithms. *Mathematics*, *12*(16), 2553. https://doi.org/10.3390/math12162553.

26. *RobustScaler. scikit-learn.* https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing. RobustScaler.html.

27. *Compare the effect of different scalers on data with outliers*. scikit-learn. https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html.

28. Zhao, H., Nasrullah, Z., & Li, Z. (2019). PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research*, *20*, 1–7. https://doi.org/10.48550/arXiv.1901.01588.

29. Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C., & Chen, G. (2023). ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions. *IEEE Transactions on Knowledge and Data Engineering, 35(12), 12181–12193.* https://doi.org/10.48550/arXiv.2201.00382.

30. Li, Z., Zhao, Y., Botta, N., Ionescu, C., & Hu, X. (2020). COPOD: Copula-Based Outlier Detection. In *2020 IEEE International Conference on Data Mining (ICDM)* (pp. 1118–1123). https://doi.org/10.48550/arXiv.2009.09463.

31. Sugiyama, M., & Borgwardt, K. (2013). Rapid Distance-Based Outlier Detection via Sampling. In *Advances in Neural Information Processing Systems (NIPS 2013)*. Curran Associates, Inc.

32. Janssens, J., Huszár, F., Postma, E., & Herik, H. (2012). Stochastic outlier selection. *Technical*

report TiCC TR 2012-001. Tilburg Center for Cognition and Communication, The Netherlands.

33. Kriegel, H., Schubert, M., & Zimek, A. (2008). Angle-based outlier detection in high-dimensional data. *In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 444–452). Association for Computing Machinery. https://doi.org/10.1145/1401890.1401946.

34. Latecki, L., Lazarevic, A., Pokrajac, D. (2007). Outlier Detection with Kernel Density Functions. *In: Perner, P. (eds) Machine Learning and Data Mining in Pattern Recognition. MLDM 2007. Lecture Notes in Computer Science, 4571*, 61–75. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-73499-4_6.

35. Fang, K-T., & Ma, C-X. (2001). Wrap-Around L2-Discrepancy of Random Sampling, Latin Hypercube and Uniform Designs. *Journal of Complexity, 17(4), 608–624*. https://doi.org/10.1006/jcom.2001.0589.

36. Yang, X., Latecki, L., Pokrajac, D. (2009). Outlier Detection with Globally Optimal Exemplar-Based GMM. *Proceedings of the SIAM International Conference on Data Mining (SDM 2009)* (pp. 145–154). https://doi.org/10.1137/1.9781611972795.13.

37. Almardeny, Y., Boujnah, N., & Cleary, F. (2022). A Novel Outlier Detection Method for Multivariate Data. *IEEE Transactions on Knowledge and Data Engineering, 34(9), 4052–4062*. https://doi.org/10.1109/TKDE.2020.3036524.

38. Goldstein, M., & Dengel, A. (2012). Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm. *In 35th German Conference on Artificial Intelligence* (pp. 59–63).

39. Kriegel, HP., Kröger, P., Schubert, E., Zimek, A. (2009). Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data. *In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, TB. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2009. Lecture Notes in Computer Science, 5476*, 831–838. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01307-2_86.

40. He, Z., Xu, X., & Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters, 24(9), 1641–1650.* https://doi.org/10.1016/S0167-8655(03)00003-5.

41. Tang, J., Chen, Z., Fu, A.Wc., Cheung, D.W. (2002). Enhancing Effectiveness of Outlier Detections for Low Density Patterns. *In: Chen, MS., Yu, P.S., Liu, B. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2002. Lecture Notes in Computer Science, 2336*, 535–548. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-47887-6_53.

42. Angiulli, F., Pizzuti, C. (2002). Fast Outlier Detection in High Dimensional Spaces. *In: Elomaa, T., Mannila, H., Toivonen, H. (eds) Principles of Data Mining and Knowledge Discovery. PKDD 2002. Lecture Notes in Computer Science, vol 2431*, 15–26. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45681-3_2.

43. Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. *In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. ACM SIGMOD Record, 29(2)*, 427–438. https://doi.org/10.1145/335191.335437.

44. Breunig, M., Kriegel, H., Ng, R., & Sander, J. (2000). LOF: identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. ACM SIGMOD Record, 29(2), 93–104. https://doi.org/10.1145/342009.335388.

45. Hoffmann, H. (2007). Kernel PCA for novelty detection. *Pattern Recognition, 40(3)*, 863–874. https://doi.org/10.1016/j.patcog.2006.07.009.

46. Shyu, M-L., Chen, S-C., Sarinnapakorn, K., Chang, L. (2003). A Novel Anomaly Detection Scheme Based on Principal Component Classifier. *IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM-03)*.

47. Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., & Williamson, R. (2001). Estimating the Support of a High-Dimensional Distribution. *Neural Computation, 13(7), 1443–1471*. https://doi.org/10.1162/089976601750264965.

48. Arning, A., Agrawal, R., & Raghavan, P. (1996). A linear method for deviation detection in large databases. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)* (pp. 164–169). AAAI Press.

49. Cook, R. (1977). Detection of Influential Observation in Linear Regression. *Technometrics*, *19*(1), 15–18. https://doi.org/10.1080/00401706.1977.10489493.

50. Goodge, A., Hooi,B., See Kiong Ng, & Wee Siong Ng. (2021). LUNAR: Unifying Local Outlier Detection Methods via Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36, 6737–6745. https://doi.org/10.48550/arXiv.2112.05355.

51. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27(8), 861–874.* https://doi.org/10.1016/j.patrec.2005.10.010.

52. Hosmer, D., Lemeshow, S., & Sturdivant, R. (2013). *Applied Logistic Regression (3rd edition). Chapter 5*, 173-182. John Wiley & Sons.

53. Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2(11)*, 559–572. https://doi.org/10.1080/14786440109462720.

54. Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning*

Research, *9(86)*, 2579–2605.

55. McInnes, L., Healy, J., & Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*. https://doi.org/10.48550/arXiv.1802.03426.

References

1. Kennedy, B. (2024). *Outlier Detection in Python*. Manning.

2. Muñoz-Garcia, J., Moreno-Rebollo, J. L., Pascual-Acosta, A., & Munoz-Garcia, J. (1990). Outliers: A Formal Approach. *International Statistical Review / Revue Internationale de Statistique*, *58*(3), 215. https://doi.org/10.2307/1403805.

3. Barnett, V., & Lewis, T. (1994). *Outliers in statistical data* (3rd edition). John Wiley & Sons, Chichester.

4. Kucherenko, V., Kvach, Ya., Smentyna, N., Ulybina, V., & Andreichenko, A. (2013). *Otsinka biznesu ta nerukhomosti: navch. pos.* (2-he vyd.). Astroprynt [in Ukrainian].

5. Kucherenko, V., Zaiets, M., Zakharchenko, O., Smentyna, N., & Ulybina, V. (2013). *Otsinka ta upravlinnia nerukhomistiu: navch. pos.* TOV «Leradruk» [in Ukrainian].

6. Aggarwal, C. (2017). *Outlier Analysis. Second Edition.* Springer.

7. *Pyod 2.0.3 documentation*. (б. д.). pyod 2.0.3 documentation. https://pyod.readthedocs.io/en/latest/.

8. *Scikit-learn: machine learning in Python — scikit-learn 1.6.1 documentation*. (б. д.). scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation. https://scikit-learn.org/stable/.

9. Çılgın, C., Gökşen, Y., & Gökçen, H. (2023). The Effect of Outlier Detection Methods in Real Estate Valuation with Machine Learning. *İzmir Sosyal Bilimler Dergisi*, 5(1), 9–20. https://doi.org/10.47899/ijss.1270433.

10. Çetiner, M., Dinçsoy, Ö., & Toraman, T. (2020). Outlier Detection for Analysis of Real Estate Price. *In 28th Signal Processing and Communications Applications Conference (SIU)* (pp. 1–4). https://doi.org/10.1109/SIU49456.2020.9302110.

11. Abut, F., Arlı, H., Akay, M., & Adıgüzel, Y. (2023). A New Hybrid Approach for Real Estate Price Prediction Using Outlier Detection, Feature Selection, and Clustering Techniques. *In 8th International Conference on Computer Science and Engineering (UBMK)* (pp. 1–6). https://doi.org/10.1109/UBMK59864.2023.10286673.

12. Śpiewak, B., & Barańska, A. (2020). Chosen statistical methods for the detection of outliers in real estate market analysis. *Acta Scientiarum Polonorum Administratio Locorum*, *19*(2), 109–118. https://doi.org/10.31648/aspal.4608.

13. Morano, P., De Mare, G., & Tajani, F. (2013). LMS for Outliers Detection in the Analysis of a Real Estate Segment of Bari. *Computational Science and Its Applications (ICCSA 2013). Lecture Notes in Computer Science, 7974*. (pp. 457–472). https://doi.org/10.1007/978-3-642-39649-6_33.

14. Sisman, S., & Aydinoglu, A. (2022). Improving performance of mass real estate valuation through application of the dataset optimization and Spatially Constrained Multivariate Clustering Analysis. *Land Use Policy*, *119*, 106–167. https://doi.org/10.1016/j.landusepol.2022.106167.

15. Gnat, S. (2020). Testing the effectiveness of outlier detecting methods in property classification. *Real Estate Management and Valuation*, *28(4)*, 81–92. https://doi.org/10.1515/remav-2020-0033.

16. Domingues, R., Filippone, M., Michiardi, P., & Zouaoui, J. (2018). A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, *74*, 406–421. https://doi.org/10.1016/j.patcog.2017.09.037.

17. Steinbuss, G., & Böhm, K. (2021). Benchmarking Unsupervised Outlier Detection with Realistic Synthetic Data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *15(4)*, 1–20. https://doi.org/10.1145/3441453.

18. Han, S., Hu, X., Huang, H., Jiang, M., & Zhao, Y. (2022). ADBench: Anomaly Detection Benchmark. *In 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*. https://doi.org/10.48550/arXiv.2206.09426.

19. *GitHub - Minqi824/ADBench: Official Implement of "ADBench: Anomaly Detection Benchmark", NeurIPS 2022*. GitHub. https://github.com/Minqi824/ADBench.

20. *Software for real estate agencies "MyHome"*. Real estate agency "Leader" Ternopil. https://lider.org.ua/myhome.aspx.

21. *DIM.RIA™ - all real estate in Ukraine. Sale and rent of any real estate*. DOM.RIA.com. https://dom.ria.com/uk/.

22. *OLX.ua ads: Ukrainian ads service*. Purchase/sale of used and new goods, various services on the OLX.ua website. https://www.olx.ua/uk/.

23. *DIM.RIA - XML format*. DOM.RIA.com. https://dim.ria.com/uk/xmldoc/.

24. *Sale of apartments and rooms in Ternopil and Ternopil district*. Real estate agency "Leader" Ternopil. https://lider.org.ua/base.aspx?t=kva.

25. Bolikulov, F., Nasimov, R., Rashidov, A., Akhmedov, F., & Cho, Y.-I. (2024). Effective Methods of Categorical Data Encoding for Artificial Intelligence Algorithms. *Mathematics*, *12*(16), 2553. https://doi.org/10.3390/math12162553.

26. *RobustScaler. scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing. RobustScaler.html*.

27. *Compare the effect of different scalers on data with outliers*. scikit-learn. https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html.

28. Zhao, H., Nasrullah, Z., & Li, Z. (2019). PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research*, *20*, 1–7. https://doi.org/10.48550/arXiv.1901.01588.

29. Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C., & Chen, G. (2023). ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions. *IEEE Transactions on Knowledge and Data Engineering*, *35(12), 12181–12193.* https://doi.org/10.48550/arXiv.2201.00382.

30. Li, Z., Zhao, Y., Botta, N., Ionescu, C., & Hu, X. (2020). COPOD: Copula-Based Outlier Detection. In *2020 IEEE International Conference on Data Mining (ICDM)* (pp. 1118–1123). https://doi.org/10.48550/arXiv.2009.09463.

31. Sugiyama, M., & Borgwardt, K. (2013). Rapid Distance-Based Outlier Detection via Sampling. In *Advances in Neural Information Processing Systems (NIPS 2013)*. Curran Associates, Inc.

32. Janssens, J., Huszár, F., Postma, E., & Herik, H. (2012). Stochastic outlier selection. *Technical report TiCC TR 2012-001*. Tilburg Center for Cognition and Communication, The Netherlands.

33. Kriegel, H., Schubert, M., & Zimek, A. (2008). Angle-based outlier detection in high-dimensional data. *In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 444–452). Association for Computing Machinery. https://doi.org/10.1145/1401890.1401946.

34. Latecki, L., Lazarevic, A., Pokrajac, D. (2007). Outlier Detection with Kernel Density Functions. *In: Perner, P. (eds) Machine Learning and Data Mining in Pattern Recognition. MLDM 2007. Lecture Notes in Computer Science, 4571*, 61–75. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-73499-4_6.

35. Fang, K-T., & Ma, C-X. (2001). Wrap-Around L2-Discrepancy of Random Sampling, Latin Hypercube and Uniform Designs. *Journal of Complexity*, *17(4), 608–624.* https://doi.org/10.1006/jcom.2001.0589.

36. Yang, X., Latecki, L., Pokrajac, D. (2009). Outlier Detection with Globally Optimal Exemplar-Based GMM. *Proceedings of the SIAM International Conference on Data Mining (SDM 2009)* (pp. 145–154). https://doi.org/10.1137/1.9781611972795.13.

37. Almardeny, Y., Boujnah, N., & Cleary, F. (2022). A Novel Outlier Detection Method for Multivariate Data. *IEEE Transactions on Knowledge and Data Engineering*, *34(9), 4052–4062.* https://doi.org/10.1109/TKDE.2020.3036524.

38. Goldstein, M., & Dengel, A. (2012). Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm. *In 35th German Conference on Artificial Intelligence* (pp. 59–63).

39. Kriegel, HP., Kröger, P., Schubert, E., Zimek, A. (2009). Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data. *In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, TB. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2009. Lecture Notes in Computer Science, 5476*, 831–838. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01307-2_86.

40. He, Z., Xu, X., & Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters, 24(9), 1641–1650.* https://doi.org/10.1016/S0167-8655(03)00003-5.

41. Tang, J., Chen, Z., Fu, A.Wc., Cheung, D.W. (2002). Enhancing Effectiveness of Outlier Detections for Low Density Patterns. *In: Chen, MS., Yu, P.S., Liu, B. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2002. Lecture Notes in Computer Science, 2336*, 535–548. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-47887-6_53.

42. Angiulli, F., Pizzuti, C. (2002). Fast Outlier Detection in High Dimensional Spaces. *In: Elomaa, T., Mannila, H., Toivonen, H. (eds) Principles of Data Mining and Knowledge Discovery. PKDD 2002. Lecture Notes in Computer Science, vol 2431*, 15–26. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45681-3_2.

43. Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. *In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. ACM SIGMOD Record, 29(2)*, 427–438. https://doi.org/10.1145/335191.335437.

44. Breunig, M., Kriegel, H., Ng, R., & Sander, J. (2000). LOF: identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. ACM SIGMOD Record, 29(2), 93–104. https://doi.org/10.1145/342009.335388.

45. Hoffmann, H. (2007). Kernel PCA for novelty detection. *Pattern Recognition, 40(3),* 863–874. https://doi.org/10.1016/j.patcog.2006.07.009.

46. Shyu, M-L., Chen, S-C., Sarinnapakorn, K., Chang, L. (2003). A Novel Anomaly Detection Scheme Based on Principal Component Classifier. *IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM-03)*.

47. Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., & Williamson, R. (2001). Estimating the Support of a High-Dimensional Distribution. *Neural Computation, 13(7), 1443–1471.* https://doi.org/10.1162/089976601750264965.

48. Arning, A., Agrawal, R., & Raghavan, P. (1996). A linear method for deviation detection in large databases. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)* (pp. 164–169). AAAI Press.

49. Cook, R. (1977). Detection of Influential Observation in Linear Regression. *Technometrics*, *19*(1), 15–18. https://doi.org/10.1080/00401706.1977.10489493.

50. Goodge, A., Hooi,B., See Kiong Ng, & Wee Siong Ng. (2021). LUNAR: Unifying Local Outlier Detection Methods via Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36, 6737–6745. https://doi.org/10.48550/arXiv.2112.05355.

51. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27(8), 861–874.* https://doi.org/10.1016/j.patrec.2005.10.010.

52. Hosmer, D., Lemeshow, S., & Sturdivant, R. (2013). *Applied Logistic Regression (3rd edition). Chapter 5*, 173-182. John Wiley & Sons.

53. Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11)*, 559–572. https://doi.org/10.1080/14786440109462720.

54. Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research, 9(86)*, 2579–2605.

55. McInnes, L., Healy, J., & Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*. https://doi.org/10.48550/arXiv.1802.03426.