УДК 004.89

Выявление аномалий в масштабных данных с применением Isolation Forest и Autoencoder

Герасимов Максим — студент бакалавриата 3-го курса направления 09.03.01 «Информатика и вычислительная техника». Научные интересы: интеллектуальные информационные системы, машинное обучение. E-mail: maxger60@gmail.com

Забродин Андрей Владимирович — канд. ист. наук, доцент кафедры «Информационные и вычислительные системы». Научные интересы: информационные системы, аналитика данных, проектирование баз данных, веб-разработка, облачные технологии. E-mail: zabrodin@pgups.ru

Петербургский государственный университет путей сообщения Императора Александра I, Россия, 190031, Санкт-Петербург, Московский пр., 9

Для цитирования: Герасимов М., Забродин А. В. Выявление аномалий в масштабных данных с применением Isolation Forest и Autoencoder // Интеллектуальные технологии на транспорте. 2025. № 4 (44). С. 17–25 . DOI: 10.20295/2413-2527-2025-444-17-25

Аннотация. ***Цель:*** *сравнительный анализ двух методов обнаружения аномалий в больших массивах данных — ансамблевого алгоритма Isolation Forest и нейросетевого Autoencoder.* ***Методы:*** *проведено моделирование и экспериментальное сравнение алгоритмов на реальном датасете транзакций по кредитным картам. Использованы стандартные метрики эффективности (precision, recall, F1-score, ROC-AUC), а также матрица ошибок для анализа структуры ложных срабатываний и пропусков аномалий.* ***Результаты:*** *обе модели достигли высоких значений ROC-AUC, что подтверждает их способность надежно различать нормальные и аномальные транзакции.* ***Практическая значимость:*** *разработанные подходы применимы для автоматизированного мониторинга транзакционных потоков, предотвращения мошенничества и анализа больших данных. Наиболее эффективно комбинированное использование Isolation Forest и Autoencoder в гибридных системах, что позволяет повысить точность и снизить количество ложных тревог при обнаружении аномалий.*

Ключевые слова: *большие данные, аномалии, машинное обучение, нейронные сети, Autoencoder, Isolation Forest, транзакционные данные*

1.2.1 *— искусственный интеллект и машинное обучение (технические науки)*

Введение

В условиях роста объемов и сложности данных задача своевременного обнаружения аномалий становится особенно важной. Аномалии — это наблюдения, отклоняющиеся от нормального поведения системы, возникающие из-за сбоев, ошибок или мошеннических действий. Их игнорирование может исказить анализ и привести к неверным решениям.

Обнаружение аномалий актуально в самых разных сферах — от финансов и кибербезопасности до медицины и промышленности. Особый интерес представляют методы без учителя, не требующие разметки и способные выявлять новые типы отклонений.

Цель исследования — сравнить два современных алгоритма: Isolation Forest и Autoencoder. Рассматриваются их принципы, применение к задаче выявления мошеннических транзакций и оценка эффективности по метрикам precision, recall, F1-score и ROC-AUC, а также с использованием матрицы ошибок.

Теоретические основы

Алгоритмы машинного обучения без учителя для обнаружения аномалий

Для обнаружения аномалий при обучении без учителя применяются методы, не требующие разметки данных. Их делят на несколько групп [1]:

- кластеризационные методы предполагают, что нормальные объекты образуют плотные группы, а аномалии — разрозненные точки. Пример — k-means, где выбросы не вписываются в структуру кластеров;
- ансамблевые методы, такие как Isolation Forest, изолируют редкие объекты с помощью случайных деревьев;
- нейросетевые методы, например Autoencoder и вариационные автокодировщики, восстанавливают исходные данные и считают аномалиями объекты с высокой ошибкой реконструкции.

Выбор алгоритма зависит от структуры данных и требований: одни методы проще и понятнее, другие — гибче и лучше выявляют сложные зависимости [2].

Алгоритм Isolation Forest

Isolation Forest (iForest) — алгоритм обнаружения аномалий, предложенный Фэй Тони Лю (Fei Tony Liu) и др. в 2008 году. Его идея в том, что аномалии проще изолировать, так как они редки и отличаются от большинства данных. Алгоритм строит ансамбль случайных бинарных деревьев (Isolation Trees), где данные рекурсивно делятся по случайным признакам и порогам. Чем короче путь до изоляции объекта, тем выше вероятность, что он аномален. Средняя длина пути по всем деревьям определяет аномальный рейтинг [3].

Алгоритм реализован на Python [4–8]. Визуализация результатов (график, матрица ошибок) реализована на Matplotlib и Seaborn [9–11].

Преимущества:

- линейная вычислительная сложность $O(n)$ и низкие требования к памяти — подходит для больших данных и потоковой обработки;
- не требует предположений о распределении данных, хорошо работает с высокоразмерными и сложными выборками;
- устойчив к шуму, прост в интерпретации (через длину пути) и имеет минимум настраиваемых параметров.

Ограничения:

- возможное переобучение при малом объеме или несбалансированности данных;
- менее эффективен для глобальных аномалий и многомерных зависимостей, так как разбиения выполняются по отдельным признакам;
- не учитывает временные зависимости;
- при высоком уровне шума возможны ложные срабатывания, поэтому важна предварительная очистка данных и настройка порога аномальности.

Несмотря на ограничения, Isolation Forest остается быстрым, масштабируемым и широко применяемым методом для задач вроде выявления мошенничества и мониторинга оборудования.

Алгоритм Autoencoder

Autoencoder — нейросеть без учителя, обучающаяся восстанавливать исходные данные. Она состоит из кодировщика, сжимающего вход до компактного представления, и декодировщика, восстанавливающего исходный сигнал. Модель обучается минимизировать ошибку реконструкции — разницу между входом и выходом [12].

В задачах обнаружения аномалий Autoencoder обучается на нормальных данных. Если подать на вход нетипичный объект, ошибка реконструкции возрастает. Значения ошибки выше заданного порога считаются индикатором аномалии [13].

Как и Isolation Forest, алгоритм реализован на Python. Визуализация результатов (график, матрица ошибок) реализована на Matplotlib и Seaborn.

Преимущества:

- не требует размеченных данных;
- адаптируется к разным типам (изображения, временные ряды, табличные данные);
- выполняет нелинейное понижение размерности, выявляя скрытые зависимости, недоступные линейным методам;
- полученные коды можно использовать для визуализации или других алгоритмов.

Ограничения:

- требует большого объема данных и вычислительных ресурсов;
- при избыточной мощности может восстанавливать и аномалии («перестаравшийся декодер»);
- чувствителен к выбору порога ошибки;
- труден для интерпретации.

Несмотря на ограничения, Autoencoder — мощный инструмент для поиска сложных нелинейных аномалий в данных.

Методика исследования

Описание датасета Credit Card Fraud Detection

В исследовании используется открытый набор Credit Card Fraud Detection [14, 15]: 284 807 транзакций европейских держателей за два дня сентября 2013 года: 492 (~ 0,17 %) — мошеннические (Class = 1), остальные — нормальные (Class = 0). Каждая транзакция описана 30 признаками: 28 анонимизированных компонент PCA (V1–V28) и два «сырых» — Time (секунды от начала периода) и Amount (сумма в евро) [16]. Набор резко несбалансирован, что усложняет обучение и оценку, поэтому все процедуры и метрики учитывают редкость целевого класса [17].

Подготовка данных: очистка, масштабирование, разметка

Перед обучением проведена предобработка данных. Пропуски не выявлены, а 1081 дубликат был удален.

Для признака Amount выполнена стандартизация (вычитание среднего и деление на стандартное отклонение), чтобы устранить различие масштабов и улучшить сходимость моделей [16].

Целевой признак Class не использовался при обучении, так как методы работают без учителя, но применялся для оценки качества на тестовой выборке.

Данные были разделены на обучающую и тестовую части с сохранением доли классов: обучение проводилось без меток, а проверка — по фактическим значениям Class, что соответствует реальному сценарию применения моделей.

Метрики оценки качества моделей

Для оценки эффективности моделей использовались стандартные метрики бинарной классификации. Основой анализа служит матрица ошибок (confusion matrix), показывающая количество:

- верно обнаруженных аномалий (TP);
- ложных тревог (FP);
- корректно распознанных нормальных транзакций (TN);
- пропущенных аномалий (FN).

На основе этих показателей рассчитываются метрики качества модели [18]:

1. Precision (точность) — доля корректно выявленных мошеннических транзакций среди всех, отмеченных моделью как аномальные:

$$\text{Precision} = \frac{TP}{TP + FP}.$$

2. Recall (полнота) — доля выявленных моделью мошеннических транзакций среди всех реально имеющих место мошеннических операций:

$$\text{Recall} = \frac{TP}{TP + FN}.$$

Эта метрика отражает чувствительность алгоритма к аномалиям, показывая, какую часть от общего числа мошенничеств удалось обнаружить. Формально полнота равна доле верно найденных аномалий от общего числа аномалий. В контексте рассматриваемой задачи Recall отвечает на вопрос: насколько хорошо модель покрывает все случаи мошенничества, не пропуская их.

3. F1-мера (F1-score) — гармоническое среднее между точностью и полнотой. Эта интегральная метрика позволяет сбалансированно оценить алгоритм по совокупности точности и полноты:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

F1-score особенно важна при сильной несбалансированности классов, как в исследуемом случае: она показывает баланс между точностью и полнотой, когда высокая точность может сопровождаться пропуском аномалий [18].

4. ROC-AUC, близкое к 1, свидетельствует о высокой способности модели различать нормальные и аномальные объекты. При сильной несбалансированности также анализируется кривая Precision-Recall и AUC-PR, однако основным показателем остается ROC-AUC.

В задачах обнаружения аномалий важно учитывать стоимость ошибок: в банковской сфере пропущенное мошенничество критичнее, чем ложная тревога. Поэтому, помимо интегральных метрик, анализируется и структура ошибок по матрице ошибок.

Этапы построения моделей

Практическая часть исследования включала реализацию и сравнение двух моделей обнаружения аномалий — Isolation Forest и Autoencoder. Работа проводилась в несколько этапов:

1–2. Загрузка и разделение данных

Исходный датасет транзакций был случайным образом разделен на обучающую и тестовую выборки в пропорции 70/30 с сохранением исходной доли мошеннических операций (~ 0,17 %), чтобы обеспечить репрезентативность данных и исключить смещение результатов. Признак Time не использовался при разделении, так как транзакции упорядочены по времени [16].

На этапе предобработки были удалены дубликаты, проверено отсутствие пропусков и выполнена стандартизация признаков. Целевая метка Class не применялась при обучении, так как модели обучались без учителя, но сохранялась для последующей оценки качества по показателям TP, FP, FN и TN.

3. Обучение модели Isolation Forest

Алгоритм Isolation Forest реализован с помощью библиотеки sklearn.ensemble. Модель обучалась на данных без меток мошенничества. Основные параметры — n_estimators (число деревьев — 100–200) и max_samples (размер подвыборки, равный обучающей выборке) [19].

После обучения каждая транзакция получила оценку аномальности, основанную на средней длине пути изоляции: чем короче путь, тем выше вероятность аномалии. Порог определялся параметром contamination = 0,0017 (0,17 % данных), что соответствует доле выбросов [8].

Модель пометила около 0,17 % транзакций как аномальные. Эти результаты сравнивались с реальными метками для построения confusion matrix и расчета метрик Precision, Recall, F1 и ROC-AUC.

4. Обучение модели Autoencoder

Разработка и обучение модели проведены с использованием фреймворка глубокого обучения TensorFlow/Keras [8]. Архитектура — полносвязная (MLP): входной слой на 30 признаков → несколько скрытых слоев с убывающим числом нейронов → узкое «горлышко» ($d \approx 8$) → симметричный декодер до 30-мерного выхода. Обучение велось на стандартизированных данных; функция потерь — MSE, оптимизатор — Adam [8, 13]. Метка Class в обучении не использовалась; явные аномалии не исключались из-за их крайне малой доли (< 0,2 %). После обучения для каждого объекта вычислялась ошибка реконструкции; порог для решения «аномалия/норма» подбирался по валидации (ориентир — верхние ~ 0,17 % ошибок либо максимум F1). Превышение порога трактовалось как аномалия. На тестовой выборке по предсказаниям была построена матрица ошибок и рассчитаны Precision, Recall, F1 и ROC-AUC.

5. Сравнение и интерпретация результатов

На заключительном этапе обе модели были протестированы на отложенной выборке. Их эффективность оценивалась по метрикам Precision, Recall, F1 и ROC-AUC. Дополнительно анализировалась матрица ошибок для разграничения ложных тревог и пропущенных случаев. Также сопоставлялось пересечение выявленных транзакций, что позволило оценить комплементарность подходов и вклад каждой модели в общую детекцию.

На базе проведенного эксперимента формулируются выводы о применимости каждого метода к задаче обнаружения аномалий в транзакционных данных и рекомендации по их использованию (возможно, совместно) для повышения эффективности системы обнаружения мошенничества.

Результаты экспериментов

Результаты Isolation Forest

Алгоритм Isolation Forest был применен для выявления выбросов без использования разметки (обучение на всех данных как на нормальных). Распределение оценок аномальности (рис. 1) показывает, что основная часть наблюдений характеризуется низкими значениями, тогда как небольшой фрагмент выборки имеет заметно более высокие оценки и выделяется как потенциальные аномалии. Пороговое значение для разделения «норма/аномалия» определялось по статистике распределения (например, 99-й перцентиль) либо по результатам валидации.

Результаты показали, что Isolation Forest изолирует аномальные транзакции быстрее и эффективнее. Алгоритм выявил до 85–90 % мошеннических операций, при точности ~ 0,81–0,85, F1 ~ 0,84–0,85 и ROC-AUC = 0,95, что подтверждает высокое качество разделения нормальных и аномальных данных. Итоговое распределение ошибок классификации представлено на рис. 2, где показана матрица ошибок алгоритма Isolation Forest.

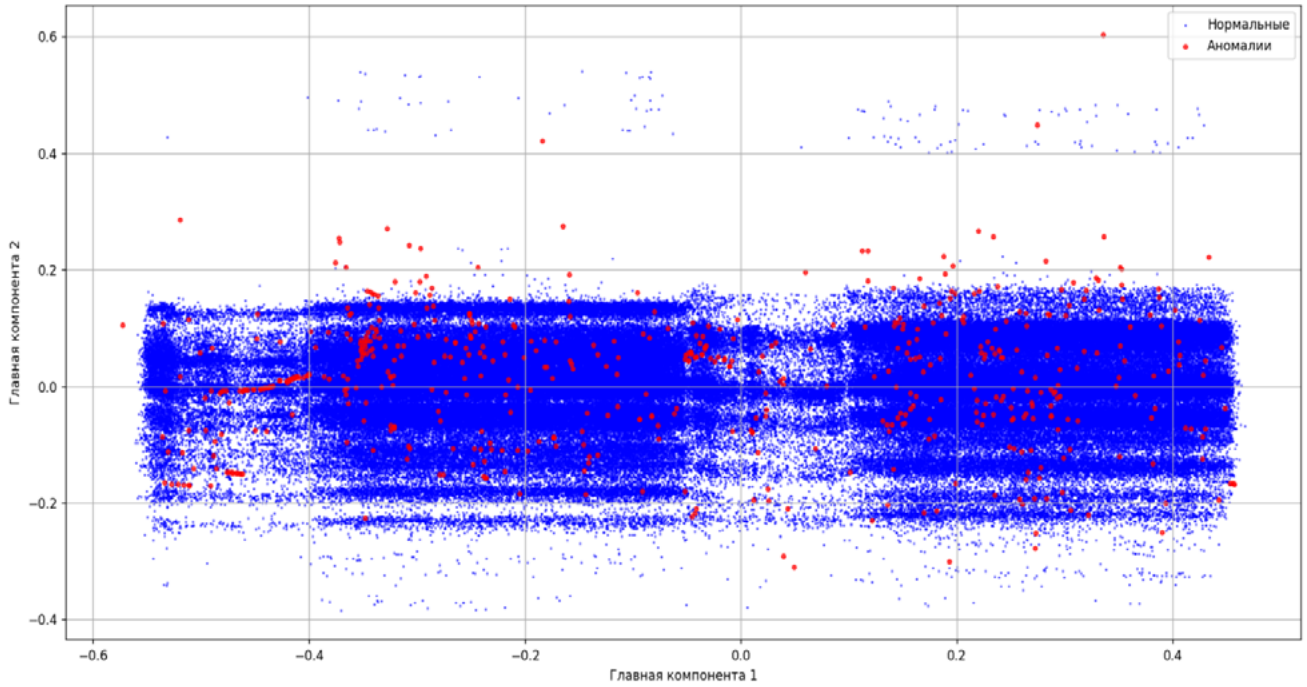

Рис. 1. Результаты алгоритма Isolation Forest (PCA-визуализация)

Рис. 2. Матрица ошибок алгоритма Isolation Forest

Результаты Autoencoder

Autoencoder обучался на данных, отражающих нормальное поведение системы, формируя компактное представление закономерностей [13]. Для обычных транзакций ошибка реконструкции была низкой, а для аномальных — заметно выше. Порог классификации определялся по статистике ошибок (например, 95-й перцентиль), что позволяло отделить редкие нетипичные наблюдения от основной массы нормальных.

Итоговые результаты и матрица ошибок алгоритма Autoencoder показаны на рис. 3 и 4.

Autoencoder выявлял аномалии по превышению порога ошибки реконструкции. Модель показала более высокую точность (~ 0,88–0,90) и меньше ложных тревог по сравнению с Isolation Forest. Полнота составила ~ 0,81–0,85, F1 — ~ 0,84–0,87, ROC-AUC — ~ 0,94–0,96, что подтверждает высокую эффективность модели.

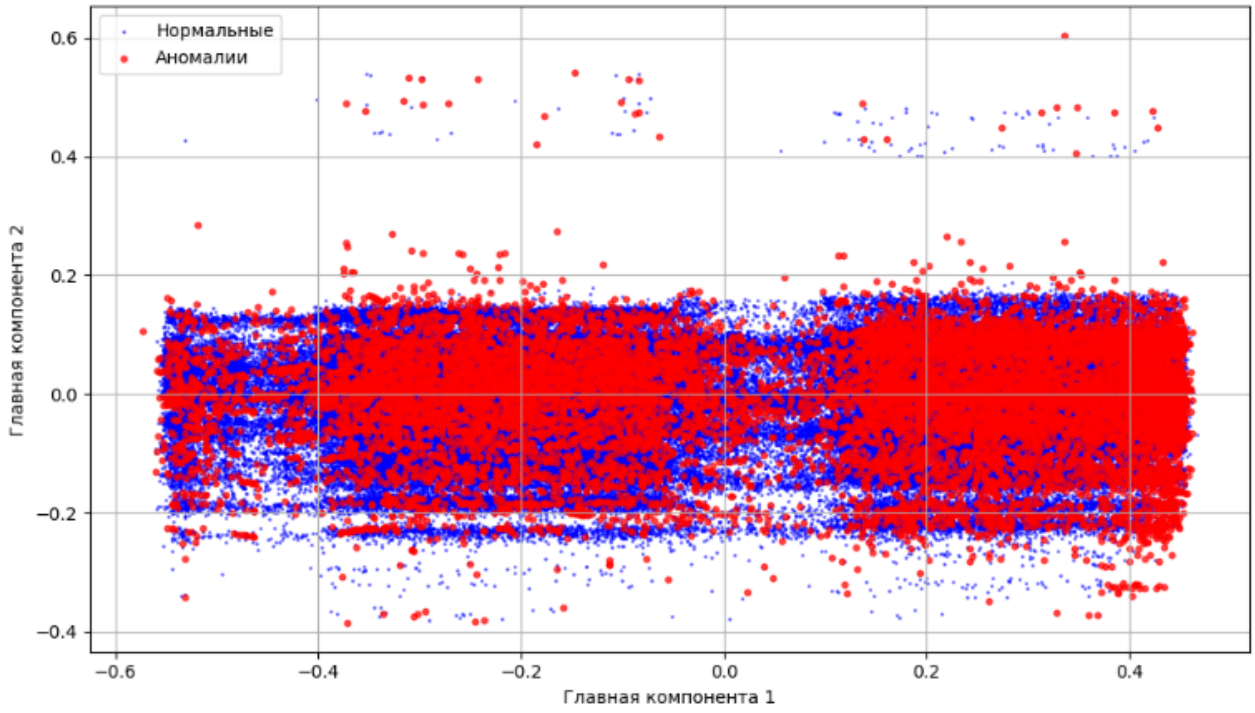

Рис. 3. Результаты алгоритма Autoencoder (PCA-визуализация)

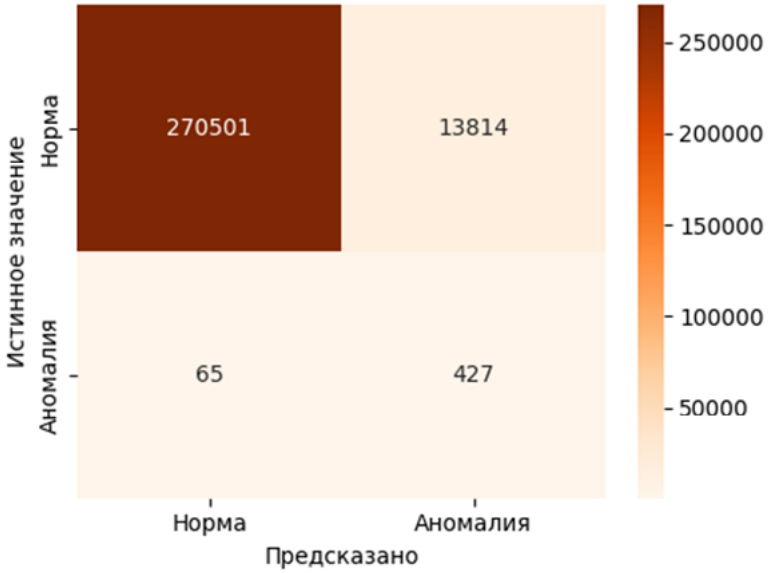

Рис. 4. Матрица ошибок алгоритма Autoencoder

Сравнительный анализ моделей

Для наглядности сравнения качества работы моделей табл. 1 суммирует полученные метрики для Isolation Forest и Autoencoder.

Таблица 1

Сравнительная таблица алгоритмов

Критерий	Isolation Forest	Autoencoder
Точность	Высокая	Очень высокая
Работа с временными рядами	Неприменим напрямую	Подходит (через окна)
Интерпретируемость	Хорошая	Ограниченная
Скорость обучения	Высокая	Средняя/низкая
Универсальность	Средняя	Высокая

Основные показатели на контрольном наборе данных можно обобщить следующим образом:

1. Точность (Precision): Autoencoder (~ 0,88) превосходит Isolation Forest (~ 0,81) по точности обнаружения аномалий, что указывает на меньшую долю ложных срабатываний.

2. Полнота (Recall): Isolation Forest (~ 0,87) превышает Autoencoder (~ 0,81) по полноте выявления, то есть охватывает большее число истинных аномалий.

3. F1-мера: Autoencoder показал слегка более высокое значение F1 (~ 0,84–0,87) по сравнению с Isolation Forest (~ 0,84–0,85), отражая лучшую гармоническую комбинацию Precision и Recall.

4. ROC-AUC: Метрика ROC-AUC оказалась высокой для обеих моделей (> 0,9), что свидетельствует о сопоставимой эффективности в ранжировании нормальных и аномальных объектов.

Стоит отметить вычислительные аспекты. Isolation Forest проще и быстрее обучается и хорошо масштабируется, что делает его удобным для анализа больших данных при ограниченных ресурсах. Autoencoder требует больше вычислительных мощностей и настройки архитектуры, но лучше выявляет сложные и тонкие аномалии в данных.

Заключение

В исследовании сравнивались два метода обнаружения аномалий в транзакциях — Isolation Forest и Autoencoder. Обе модели показали высокую эффективность, надежно различая нормальные и аномальные операции.

Isolation Forest выявляет большинство аномалий, но дает больше ложных срабатываний (высокая полнота выявления при умеренной точности). Autoencoder снижает число ложных тревог, но пропускает часть редких событий (высокая точность при сопоставимой F1-мере). Выбор метода зависит от приоритетов: при важности полноты — Isolation Forest, при акценте на точность — Autoencoder.

Оба алгоритма подходят для анализа больших данных и обнаружения нетипичных событий. Перспективным является их комбинированное использование: Isolation Forest — для первичного выявления, Autoencoder — для уточняющей фильтрации. Такой подход повышает общую эффективность систем обнаружения аномалий.

СПИСОК ИСТОЧНИКОВ

1. Обзор методов обнаружения аномалий в потоках данных / В. П. Шкодырев, К. И. Ягафаров, В. А. Баштовенко, Е. Э. Ильина // Proceedings of the Second Conference on Software Engineering and Information Management (SEIM-2017), (Saint Petersburg, Russia, 21 April 2017). CEUR Workshop Proceedings. 2017. Vol. 1864. Pp. 50–56.

2. Анализ данных и процессов: учебное пособие / А. А. Барсегян, М. С. Куприянов, И. И. Холод [и др.]. 3-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2009. 512 с.

3. Liu F. T., Ting K. M., Zhou Z.-H. Isolation Forest // Proceedings of the Eighth IEEE International Conference on Data Mining (Pisa, Italy, 15–19 December 2008). Institute of Electrical and Electronics Engineers, 2008. Pp. 413–422. DOI: 10.1109/ICDM.2008.17.

4. Scikit-learn: Machine Learning in Python. URL: http://scikit-learn.org (дата обращения: 18.10.2025).

5. Pandas: Python Data Analysis Library. URL: http://pandas.pydata.org (дата обращения: 18.10.2025).

6. NumPy v2.3 Documentation. URL: http://numpy.org/doc/2.3 (дата обращения: 18.10.2025).

7. SciPy v1.16.2 Documentation. URL: http://docs.scipy.org/doc/scipy (дата обращения: 18.10.2025).

8. PyOD V2 Documentation. URL: http://pyod.readthedocs.io (дата обращения: 18.10.2025).

9. Matplotlib: Visualization with Python. URL: http://matplotlib.org (дата обращения: 18.10.2025).

10. Seaborn: Statistical Data Visualization. URL: http://seaborn.pydata.org (дата обращения: 18.10.2025).
11. Plotly Open Source Graphing Library for Python. URL: http://plotly.com/python (дата обращения: 18.10.2025).
12. Goodfellow I., Bengio Y., Courville A. Autoencoders // Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge (MA): MIT Press, 2016. Pp. 499–523.
13. Hinton G. E., Salakhutdinov R. R. // Science. 2006. Vol. 313, Iss. 5786. Pp. 504–507. DOI: 10.1126/science.112764.
14. Credit Card Fraud Detection: Anonymized Credit Card Transactions Labeled as Fraudulent or Genuine // Kaggle. URL: http://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (дата обращения: 18.10.2025).
15. The Numenta Anomaly Benchmark // GitHub. URL: http://github.com/numenta/NAB (дата обращения: 18.10.2025).
16. Imbalanced-learn v0.14.0 Documentation. URL: http://imbalanced-learn.org (дата обращения: 18.10.2025).
17. Макшанов А. В., Журавлев А. Е., Тындыкарь Л. Н. Большие данные. Big Data: учебник для вузов. 4-е изд., стер. Санкт-Петербург: Лань, 2024. 188 с.
18. Фельдман Е. В., Ручай А. Н., Чербаджи Д. Ю. Модель выявления аномальных банковских транзакций на основе машинного обучения // Вестник УрФО. Безопасность в информационной сфере. 2021. № 1 (39). С. 27–35. DOI: 10.14529/secur210104.
19. Novelty and Outlier Detection — Scikit-learn 1.7.2 Documentation. URL: http://scikit-learn.org/stable/modules/outlier_detection.html (дата обращения: 18.10.2025).

Дата поступления: 29.10.2025
Решение о публикации: 22.11.2025

Anomaly Detection in Large-Scale Data Using Isolation Forest and Autoencoder

Maksim Gerasimov — 3rd year Bachelor`s Degree Student in 09.03.01 Informatics and Computer Technology. Research interests: intellectual information systems, machine learning. E-mail: maxger60@gmail.com

Andrey V. Zabrodin — PhD in History, Associate Professor of the “Information and Computing Systems” Department. Research interests: information systems, data analytics, database design, web development, cloud technologies. E-mail: zabrodin@pgups.ru

Emperor Alexander I St. Petersburg State Transport University, 9, Moskovsky ave., Saint Petersburg, 190031, Russia

For citation: Gerasimov M., Zabrodin A. V. Anomaly Detection in Large-Scale Data Using Isolation Forest and Autoencoder. *Intellectual Technologies on Transport*, 2025, No. 4 (44), Pp. 17–25. DOI: 10.20295/2413-2527-2025-444-17-25. (In Russian)

Abstract. ***Purpose:*** *the study aims to compare two anomaly detection methods applied to large-scale datasets, the ensemble-based Isolation Forest and the neural network-based Autoencoder.* ***Methods:*** *the investigation entailed modelling and empirical assessment of the algorithms utilizing a genuine credit card transaction dataset. Standard performance metrics such as precision, recall, F1-score, ROC-AUC were used accompanied by a confusion matrix to explore the occurrences of false positives and overlooked anomalies.* ***Results:*** *the findings revealed that both models achieved elevated ROC-AUC scores, confirming their robustness in differentiating between typical and anomalous transactions.* ***Practical significance:*** *the proposed methods can be suitable for the automated supervision of transactional flows, the prevention of fraud, and the analysis of large datasets. The integration of Isolation Forest and Autoencoder in hybrid systems has demonstrated superior effectiveness, enhancing detection accuracy while minimizing the occurrence of false positives in anomaly detection.*

Keywords: *large datasets, machine learning, anomalies, machine learning, neural network, Autoencoder, Isolation Forest, transaction data*

REFERENCES

1. Shkodyrev V. P., Yagafarov K. I., Bashtovenko V. A., Ilyina E. E. Obzor metodov obnaruzheniya anomaliy v potokakh dannykh [The Overview of Anomaly Detection Methods in Data Streams], *Proceedings of the Second Conference on Software Engineering and Information Management (SEIM-2017), Saint Petersburg, Russia, April 21, 2017. CEUR Workshop Proceedings*, 2017, Vol. 1864, Pp. 50–56. (In Russian)

2. Barsegyan A. A., Kupriyanov M. S., Kholod I. I., et al. Analiz dannykh i protsessov: uchebnoe posobie [Data and Process Analysis: A Tutorial]. Saint Petersburg, BHV-Peterburg Publishing House, 2009, 512 p. (In Russian)

3. Liu F. T., Ting K. M., Zhou Z.-H. Isolation Forest, *Proceedings of the Eighth IEEE International Conference on Data Mining, Pisa, Italy, December 15–19, 2008*. Institute of Electrical and Electronics Engineers, 2008, Pp. 413–422. DOI: 10.1109/ICDM.2008.17.

4. Scikit-learn: Machine Learning in Python. Available at: http://scikit-learn.org (accessed: October 18, 2025).

5. Pandas: Python Data Analysis Library. Available at: http://pandas.pydata.org (accessed: October 18, 2025).

6. NumPy v2.3 Documentation. Available at: http://numpy.org/doc/2.3 (accessed: October 18, 2025).

7. SciPy v1.16.2 Documentation. Available at: http://docs.scipy.org/doc/scipy (accessed: October 18, 2025).

8. PyOD V2 Documentation. Available at: http://pyod.readthedocs.io (accessed: October 18, 2025).

9. Matplotlib: Visualization with Python. Available at: http://matplotlib.org (accessed: October 18, 2025).

10. Seaborn: Statistical Data Visualization. Available at: http://seaborn.pydata.org (accessed: October 18, 2025).

11. Plotly Open Source Graphing Library for Python. Available at: http://plotly.com/python (accessed: October 18, 2025).

12. Goodfellow I., Bengio Y., Courville A. Autoencoders. In: *Goodfellow I., Bengio Y., Courville A. Deep Learning*. Cambridge (MA), MIT Press, 2016, Pp. 499–523.

13. Hinton G. E., Salakhutdinov R. R., *Science*, 2006, Vol. 313, Iss. 5786, Pp. 504–507. DOI: 10.1126/science.112764.

14. Credit Card Fraud Detection: Anonymized Credit Card Transactions Labeled as Fraudulent or Genuine, *Kaggle*. Available at: http://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (accessed: October 18, 2025).

15. The Numenta Anomaly Benchmark, *GitHub*. Available at: http://github.com/numenta/NAB (accessed: October 18, 2025).

16. Imbalanced-learn v0.14.0 Documentation. Available at: http://imbalanced-learn.org (accessed: October 18, 2025).

17. Makshanov A. V., Zhuravlev A. E., Tyndykar L. N. Bolshie dannye. Big Data: uchebnik dlya vuzov [Big Data. Big Data: a textbook for universities]. Saint Petersburg, LAN Publishing House, 2024, 188 p. (In Russian)

18. Feldman E. V., Ruchay A. N., Cherbadzhi D. Y. Model vyyavleniya anomalnykh bankovskikh tranzaktsiy na osnove mashinnogo obucheniya [Model for Detecting Abnormal Banking Transactions Based on Machine Learning], *Vestnik UrFO. Bezopasnost v informatsionnoy sfere [Journal of the Ural Federal District. Information Security]*, 2021, No. 1 (39), Pp. 27–35. DOI: 10.14529/secur210104. (In Russian)

19. Novelty and Outlier Detection — Scikit-learn 1.7.2 Documentation. Available at: http://scikit-learn.org/stable/modules/outlier_detection.html (accessed: October 18, 2025).

Received: 29.10.2025
Accepted: 22.11.2025