

**Панов Владислав Дмитриевич**

аспирант, Российский государственный социальный университет, Москва. ORCID: 0009-0005-9308-7241.

Электронный адрес: vladislavpanovvp7@mail.ru

**Vladislav D. Panov**

Postgraduate, Russian State Social University, Moscow. ORCID: 0009-0005-9308-7241.

E-mail address: vladislavpanovvp7@mail.ru

---

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ОБНАРУЖЕНИЯ АНОМАЛИЙ В МЕТРИКАХ ПРОИЗВОДИТЕЛЬНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ

---

**Аннотация.** В статье описывается разработка и апробация адаптивного программного инструментария, предназначенного для автоматизированного выявления аномальных отклонений в метриках производительности информационных систем. Основу работы составляет фреймворк CPAF (Context-aware Performance Analysis Framework), в который заложен модуль контекстно зависимого конструирования признаков, а также стратегия мультиоценочного стейкинга – объединение оценок аномальности пяти различных детекторов (Isolation Forest, LOF, ECOD, COPOD, KNN) и их подача в качестве входного вектора метаклассификатору, построенному на градиентном бустинге. Экспериментальная часть выполнена на двух открытых бенчмарках: NAB (Numenta Anomaly Benchmark – 15 одномерных временных рядов метрик AWS CloudWatch) и SKAB (Skoltech Anomaly Benchmark – 26 многомерных рядов с 8 сенсорами). Всего сопоставлено 11 методов в рамках единой методологии: темпоральное разбиение 60/40, посерийное обучение, исключение утечек данных. Показано, что supervised-подходы на многомерных данных дают наиболее высокую точность (Precision = 0,782 у Random Forest – на 4,1 % выше, чем у Isolation Forest), что существенно для задач промышленного мониторинга. Сформулированы рекомендации по выбору метода в зависимости от размерности данных и доступного объема разметки.

**Ключевые слова:** обнаружение аномалий, машинное обучение, производительность веб-приложений, фреймворк, Isolation Forest, мониторинг, временные ряды, ансамблевые методы, NAB, SKAB.

**Для цитирования:** Панов В.Д. Сравнительный анализ методов обнаружения аномалий в метриках производительности информационных систем на основе машинного обучения // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2026. № 3. С. 174–182. DOI: 10.18137/RNU.V9187.26.03.P.174

---

## COMPARATIVE ANALYSIS OF ANOMALY DETECTION METHODS IN INFORMATION SYSTEM PERFORMANCE METRICS USING MACHINE LEARNING

---

**Abstract.** This paper describes the development and empirical validation of an adaptive toolkit designed for automated anomaly detection in information system performance metrics. At its core lies the CPAF framework (Context-aware Performance Analysis Framework), which incorporates a context-dependent

---

Сравнительный анализ методов обнаружения аномалий в метриках  
производительности информационных систем на основе машинного обучения

---

feature engineering module and a Multi-Score Stacking strategy - the latter aggregates anomaly scores from five distinct detectors (Isolation Forest, LOF, ECOD, COPOD, KNN) and passes them as feature input to a gradient boosting meta-classifier. Experiments were run on two publicly available benchmarks: NAB (15 univariate AWS CloudWatch time series) and SKAB (26 multivariate series, 8 sensors each). A total of eleven methods were evaluated under a shared strict protocol. The results demonstrate that supervised approaches attain the highest Precision on multivariate data (0.782 for Random Forest), which matters for production monitoring pipelines.

**Keywords:** anomaly detection, machine learning, web application performance, framework, Isolation Forest, monitoring, time series, ensemble methods, NAB, SKAB.

**For citation:** Panov V.D. (2026) Comparative Analysis of Anomaly Detection Methods in Information System Performance Metrics Using Machine Learning. *Vestnik of Russian New University. Series: Complex Systems: Models, analysis, management*. No. 3. Pp. 174–182. DOI: 10.18137/RNU.V9I87.26.03.P.174 (In Russian).

### **Введение**

Вопрос производительности информационных систем давно перестал быть только техническим – он напрямую влияет на пользовательский опыт и, как следствие, на финансовые показатели бизнеса [1, с. 1]. Известна оценка, согласно которой задержка загрузки веб-страницы всего на одну секунду способна снизить конверсию примерно на 7 % [2, с. 1]. Если раньше это звучало как маркетинговый аргумент, то в условиях современных микросервисных архитектур, CDN-сетей и клиентского рендеринга ситуация лишь обострилась: телеметрических данных стало так много, что их ручной разбор превращается в задачу, не решаемую силами одного-двух инженеров. На практике инструменты вроде Google Lighthouse, WebPageTest, New Relic или Datadog до сих пор опираются преимущественно на статические пороги [3, с. 1]. Типичный пример: если Largest Contentful Paint превышает 4 с или Time to First Byte уходит за 1,8 с, система сигнализирует о проблеме. Но такие правила игнорируют контекст: устройство пользователя, качество канала, географию, суточную динамику нагрузки. Отсюда избыток ложных срабатываний, подрывающий доверие команды к мониторингу [4, с. 3].

Методы обнаружения аномалий на базе машинного обучения в последние годы привлекают всё большее внимание [5, с. 268; 6, с. 3]. Однако в литературе по-прежнему не хватает масштабных сравнительных исследований, проведенных на реальных данных мониторинга. Нередко авторы ограничиваются двумя-тремя алгоритмами, а эксперименты ставят на синтетике, что существенно осложняет выбор подхода на практике.

*Цель* настоящей работы – создать адаптивный инструмент (фреймворк CRAF) со стратегией мультиоценочного стекинга и провести его экспериментальную проверку на реальных данных, сопоставив с десятью альтернативными методами.

### **Обзор существующих подходов**

Обнаружение аномалий во временных рядах метрик производительности – частный случай более общей задачи поиска выбросов [7, с. 2]. Накопленный на сегодня арсенал подходов можно условно разбить на несколько групп.

**Статические пороги.** Самый прямолинейный путь, реализованный, к примеру, в Google Lighthouse, – загрузка страницы меньше 2,5 с – «хорошо», больше 4 с – «пло-

хо». Похожие схемы используются в WebPageTest и ряде АРМ-решений. Плюс очевиден – простота внедрения; минус столь же очевиден – контекст полностью игнорируется [3, с. 1].

**Статистические подходы.** Z-score, межквартильный размах, CUSUM – классический набор инструментов для выявления отклонений от «нормы». Проблема в том, что эти методы молчаливо предполагают стационарность ряда и нормальность распределения, тогда как реальные метрики производительности демонстрируют суточную, недельную и сезонную цикличность [8, с. 92].

**Isolation Forest (IF).** Неконтролируемый алгоритм, суть которого – изолировать аномальные наблюдения рекурсивным разбиением пространства случайными деревьями. Логика проста – аномалии «отрезаются» быстрее, требуя меньшего числа расщеплений. Вычислительная сложность –  $O(n \log n)$ , что делает метод применимым к высокоразмерным данным [9, с. 414].

**Local Outlier Factor (LOF).** Плотностной подход – для каждой точки сравнивается локальная плотность с плотностью её  $k$ -соседей. Особенно хорошо LOF справляется с контекстными аномалиями, когда точка выглядит необычно на фоне ближайшего окружения, но вполне нормальна в глобальном масштабе [10, с. 94].

**ECOD и COPOD.** Два метода, привлекательных в первую очередь отсутствием необходимости в тонкой настройке гиперпараметров. ECOD опирается на эмпирические кумулятивные функции распределения, COPOD – на копулы. Оба быстры, но на данных со сложными взаимозависимостями признаков могут заметно уступать конкурентам [11, с. 2].

**KNN-детекторы.** Расстояние до  $k$  ближайших соседей трактуется как мера аномальности. Подход хорошо работает при наличии выраженных кластеров, однако чувствителен к выбору  $k$  и масштабированию признаков [7, с. 8].

**Ансамблевые стратегии.** Идея комбинирования нескольких детекторов, будь то мажоритарное голосование или стекинг (anomaly scores как признаки для метаклассификатора), рассматривалась в ряде публикаций [11, с. 4; 12, с. 40], но применительно к метрикам производительности информационных систем проработана слабо.

Отдельно стоит отметить роль инженерии признаков. В работе [6, с. 403] продемонстрировано, что конструирование лаговых и скользящих статистик заметно повышает качество детекции. Семёнов В.В. [13, с. 890] на данных киберфизических систем зафиксировал рост  $F$ -меры на 4,8 % относительно «чистого» Isolation Forest за счет формирования информативного признакового пространства. Марков Г.А. [14, с. 4] сопоставил ряд unsupervised-методов на телеметрии промышленных IoT-узлов.

Анализ существующих работ выявил отсутствие систематического сравнения широкого спектра подходов (unsupervised, supervised, гибридных) на реальных данных мониторинга, которое позволило бы обоснованно очертить границы применимости каждого из них.

### **Предлагаемый метод: CPAF и его архитектура**

Фреймворк CPAF (Context-aware Performance Analysis Framework) организован в виде конвейера из четырех последовательных этапов. Первый – сбор сырых данных метрик; второй – контекстно-зависимое конструирование признаков; третий – собственно

---

Сравнительный анализ методов обнаружения аномалий в метриках  
производительности информационных систем на основе машинного обучения

---

детекция аномалий; четвертый – анализ первопричин через интерпретацию Feature Importance.

Признаковая инженерия реализована следующим образом. Из каждой метрики автоматически извлекаются лаговые значения (шаги 1 и 3), скользящие средние и стандартные отклонения по окнам 6 и 12, z-score относительно скользящего среднего, темп изменения, временные признаки (час, день недели, sin/cos-кодировки). Для многомерных данных добавляются кросс-метрические взаимодействия – отношения пар метрик. Принципиально, что все вычисления производятся строго по прошлым данным: перед применением rolling-операции выполняется shift(1), что гарантирует отсутствие look-ahead bias.

Центральная идея CPAF – агрегация anomaly scores от пяти детекторов (IF, LOF, ECOD, COPOD, KNN). Каждый из них возвращает нормализованную к  $[0, 1]$  оценку. Пять полученных скоров включаются в общий набор признаков и передаются метаклассификатору – Gradient Boosting Classifier (80 деревьев, глубина 4, learning rate 0,05). Порог бинарной классификации подбирается на обучающей части данных по максимуму F1.

Выбор такого подхода объясняется тем, что каждый из базовых детекторов обладает собственным индуктивным смещением. IF хорошо ловит глобальные выбросы, LOF – контекстные, ECOD – маргинальные отклонения на хвостах распределений, KNN – аномалии вблизи границ кластеров. Задача метаклассификатора – научиться взвешивать эти разнородные сигналы, причем с учетом контекстных признаков, не доступных базовым детекторам по отдельности.

### *Экспериментальное исследование*

Эксперименты проводились на двух открытых бенчмарках, намеренно выбранных контрастными.

NAB (Numenta Anomaly Benchmark) – коллекция из 58 размеченных временных рядов, подготовленная компанией Numenta [15, с. 39]. Для настоящего исследования были отобраны 15 рядов, непосредственно связанных с мониторингом информационных систем: CPU utilization (EC2), Network In (EC2), ELB Request Count, RDS CPU utilization из AWS CloudWatch. Все ряды одномерные, доля аномалий колеблется от 1 до 10 %.

SKAB (Skoltech Anomaly Benchmark) – 34 многомерных ряда с 8 сенсорами (акселерометры, ток, давление, температура, напряжение, расход), созданных для задач промышленного мониторинга в Сколтехе. Из них задействованы 26 рядов (подкатегории valve1, valve2, other) [16, с. 1]. Доля аномалий здесь заметно выше – от 25 до 49 %.

Подобный выбор позволяет проверить методы в двух принципиально разных режимах: скудная разметка при одномерных данных (NAB) и более обильная разметка при многомерной структуре (SKAB). Методологический протокол был единым для всех 11 методов: темпоральный split 60/40 (первые 60 % – обучение, последние 40 % – тест, без перемешивания); посерийное обучение и оценка; случайный oversampling с гауссовым шумом ( $\sigma = 0,02$ , target ratio 0,35) для supervised-методов; запрет look-ahead при конструировании признаков; макро- и микроусредненные Precision, Recall, F1 в качестве метрик.

Сопоставляемые методы распределены по четырем категориям: Rule-based (пороги  $2,5\sigma$ ); Unsupervised (IF, LOF, ECOD, COPOD, KNN); ансамбль unsupervised (голосова-

ние  $\geq 3$  из 5); Supervised (Random Forest, CPAF-GB); гибридные (CPAF Stacking, CPAF Score Fusion).

Результаты макроусреднения на 10 рядах NAB с аномалиями в тестовой выборке представлены в Таблице 1.

Таблица 1

## Результаты на NAB (одномерные метрики AWS CloudWatch)

| Метод                        | Precision | Recall | F1     |
|------------------------------|-----------|--------|--------|
| KNN                          | 0,1959    | 0,4171 | 0,2101 |
| Голосование (5)              | 0,2112    | 0,3342 | 0,2053 |
| LOF                          | 0,1731    | 0,4324 | 0,1997 |
| COPOD                        | 0,2432    | 0,2118 | 0,1913 |
| IF                           | 0,1851    | 0,3247 | 0,1867 |
| ECOD                         | 0,2280    | 0,2014 | 0,1753 |
| Стат. пороги ( $2,5\sigma$ ) | 0,2326    | 0,2028 | 0,1215 |
| CPAF ScoreFusion             | 0,0697    | 0,2217 | 0,0918 |
| CPAF Stacking                | 0,0234    | 0,0497 | 0,0318 |
| CPAF-GB                      | 0,0236    | 0,0207 | 0,0204 |
| RF                           | 0,0049    | 0,0097 | 0,0065 |

Источник: здесь и далее таблицы составлены автором по результатам исследования.

Картина на одномерных данных вполне ожидаемая. При менее чем 1 % аномалий в обучающей части unsupervised-детекторы уверенно лидируют: KNN показал  $F1 = 0,2101$ , LOF – 0,1997. Supervised-методы (RF, CPAF-GB) при столь мизерном обучающем сигнале фактически не работают – их  $F1$  не дотягивает до 0,03.

Результаты макроусреднения на 26 рядах SKAB представлены в Таблице 2.

Таблица 2

## Результаты на SKAB (многомерные данные, 8 сенсоров)

| Метод                        | Precision | Recall | F1     |
|------------------------------|-----------|--------|--------|
| IF                           | 0,7516    | 0,7485 | 0,7257 |
| KNN                          | 0,6994    | 0,7033 | 0,6865 |
| LOF                          | 0,6564    | 0,6601 | 0,6434 |
| CPAF-GB                      | 0,7136    | 0,6491 | 0,6296 |
| CPAF Stacking                | 0,7068    | 0,6641 | 0,6284 |
| Голосование (5)              | 0,7078    | 0,5831 | 0,6163 |
| RF                           | 0,7822    | 0,5591 | 0,5847 |
| CPAF ScoreFusion             | 0,6429    | 0,4702 | 0,5115 |
| COPOD                        | 0,6592    | 0,2831 | 0,3820 |
| ECOD                         | 0,6267    | 0,2414 | 0,3334 |
| Стат. пороги ( $2,5\sigma$ ) | 0,4770    | 0,0467 | 0,0643 |

На многомерных данных расстановка сил меняется. По  $F1$  лидирует IF (0,7257). Однако если посмотреть на Precision изолированно, первую позицию занимает RF (0,7822), разрыв с IF составляет 4,1 %. CPAF-GB с Precision = 0,7136 также попадает в тройку. Ми-

---

Сравнительный анализ методов обнаружения аномалий в метриках  
производительности информационных систем на основе машинного обучения

---

кроусреднённые результаты эту тенденцию подтверждают: у RF Micro Precision = 0,8054, что на 8,3 % выше, чем у IF.

Отдельный интерес представляет анализ Feature Importance в компоненте Gradient Boosting фреймворка CРАF; он позволяет понять, за счёт чего достигается результат, и различия между датасетами здесь весьма показательны.

На NAB (одномерные ряды) наибольший вклад вносят временные контекстные признаки: день недели (importance = 3,33), z-score от скользящего среднего (1,04), час суток (0,73), циклические кодировки (0,33). Anomaly scores детекторов – LOF\_score (0,15) и KNN\_score (0,09) – тоже попадают в десятку, но явно уступают контекстным переменным.

На SKAB ситуация зеркальная: доминируют скользящие статистики сенсорных метрик: Volume Flow RateRMS MA<sub>6</sub> (5,47), Thermocouple (2,52), Temperature MA<sub>12</sub> (1,89). Сконструированные признаки оказываются систематически важнее сырых значений.

### *Результаты исследования*

Обобщая полученные данные, можно выделить несколько ключевых выводов.

Во-первых, unsupervised-методы задают весьма сильный базовый уровень. И на NAB, и на SKAB тройка лидеров по F1 включает IF и KNN. Практический вывод: любая новая разработка в области детекции аномалий в метриках производительности обязана включать сравнение хотя бы с Isolation Forest, а не ограничиваться сопоставлением со статическими порогами, как это нередко встречается в литературе.

Во-вторых, размерность данных оказывается решающим фактором для supervised-подходов. На SKAB (8 сенсоров) CРАF-GB достигает F1 = 0,630, что составляет 86,8 % от лидера (IF с 0,726). На NAB (один признак) supervised-методы проигрывают на порядок. Объяснение двояко: на NAB аномалий в обучающей выборке менее 1 %, тогда как на SKAB 25–49 %; кроме того, на многомерных данных supervised-модели способны улавливать межметрические зависимости, которые unsupervised-детекторы, работающие с каждым признаком независимо, попросту «не видят».

В-третьих, для production-систем мониторинга критична именно Precision. Каждое ложное срабатывание – это алерт, отвлекающий дежурного инженера. RF на SKAB показал Precision 0,782 (Macro) и 0,805 (Micro) – лучшие значения среди всех 11 методов. В сравнении с IF это означает на 8–12 % меньше ложных тревог при сопоставимом уровне покрытия реальных аномалий.

Четвертое наблюдение: контекстная инженерия признаков влияет на результат сильнее, чем выбор алгоритма. На NAB временные признаки (день недели, час) формируют около 60 % суммарной важности top-10 признаков. На SKAB скользящие статистики (MA<sub>6</sub>, MA<sub>12</sub>) уверенно обходят сырые метрики. Это перекликается с результатами [6, с. 403; 13, с. 891] и лишний раз подтверждает: вложения в конструирование признаков окупаются лучше, чем замена алгоритма.

Наконец, пятое – о границах применимости. На одномерных данных с малой разметкой оптимален KNN или LOF. При наличии многомерности и приемлемого объёма разметки стоит отдать предпочтение IF (если приоритет F1) или RF/CРАF-GB (если приоритет Precision). Когда важна интерпретируемость, логичным выбором становится CРАF Stacking с анализом Feature Importance.

### Заключение

В работе представлен фреймворк CRAFT, объединяющий модуль контекстно-зависимой инженерии признаков и стратегию Multi-Score Stacking. Проведено экспериментальное исследование на двух бенчмарках (NAB – 15 рядов AWS CloudWatch; SKAB – 26 многомерных рядов), охватывающее 11 методов при единой строгой методологии.

Основные итоги таковы. Unsupervised-методы (IF, LOF, KNN) формируют сильный базовый уровень, с которым необходимо сравнивать любые новые решения. Supervised-подходы обеспечивают наивысшую Precision на многомерных данных (RF 0,782 – прирост 4,1 % относительно IF), что особенно важно в production-среде. Контекстная инженерия признаков и multi-detector scores обогащают пространство для классификации. Определены границы применимости каждого подхода в зависимости от размерности и объёма разметки.

В качестве направлений продолжения исследования видится валидация на данных Core Web Vitals, эксперименты с онлайн-обучением, внедрение SHAP для объяснимости на уровне отдельных инцидентов, а также использование active learning для более эффективного накопления разметки.

### Литература

1. Алексеев П.О. Оптимизация производительности веб-приложений с использованием современных фреймворков // Universum: технические науки: электронный научный журнал. 2025. № 4(133) [Электронный ресурс]. URL: <https://7universum.com/ru/tech/archive/item/19668> (дата обращения: 08.02.2025).
2. Лутаев Д.О. Исследование и анализ методов оптимизации производительности веб-приложений // Современные научные исследования и инновации. 2023. № 10 [Электронный ресурс]. URL: <https://web.snauka.ru/issues/2023/10/100843> (дата обращения: 14.11.2025).
3. Google Developers. Web Vitals [Электронный ресурс]. URL: <https://web.dev/vitals/> (дата обращения: 14.11.2025).
4. Попова И. Обнаружение аномалий в наборе данных с помощью алгоритмов машинного обучения без учителя Isolation Forest и Local Outlier Factor // StudNet. 2020. № 12. С. 1–8.
5. Кузнецов А.С., Семенов Е.Ю. Выбор метода обнаружения аномалий в образовательных данных // Программная инженерия. 2017. № 6. С. 267–274.
6. Линдигрин А.Н. Сравнительный анализ методов машинного обучения в задачах обнаружения сетевых аномалий // Известия ТулГУ. Технические науки. 2019. Вып.12 С. 400–404.
7. Chandola V, Banerjee A., Kumar V. (2009) Anomaly detection: A survey // ACM Computing Surveys. Vol. 41. No. 3. Art. 15.
8. Чернышев Ю.Ю. Применение автокодировщиков для выявления аномалий в киберфизических системах // Вестник Пермского университета. Математика. Механика. Информатика. 2022. Вып. 4(59). С. 89–94. DOI: 10.17072/1993-0550-2022-4-89-94
9. Liu F.T., Ting K.M., Zhou Z.-H. (2008) Isolation Forest // Proceedings of the 8th IEEE International Conference on Data Mining (ICDM). Pp. 413–422.
10. Breunig M.M., Kriegel H.-P., Ng R.T., Sander J. (2000) LOF: Identifying density-based local outliers // Proceedings of ACM SIGMOD. Pp. 93–104.
11. Zhao Y., Nasrullah Z., Li Z. (2019) PyOD: A Python Toolbox for Scalable Outlier Detection // Journal of Machine Learning Research. Vol. 20. Pp. 1–7.

Сравнительный анализ методов обнаружения аномалий в метриках  
производительности информационных систем на основе машинного обучения

12. Шукуров Н.А. Алгоритмы обнаружения аномалий в разнородных наборах данных // Информационные технологии и безопасность. 2024. № 1. С. 38–41.
13. Семенов В.В. Подход к формированию информативных признаков в задачах мониторинга информационной безопасности киберфизических систем // Научно-технический вестник информационных технологий, механики и оптики. 2021. Т. 21, № 6. С. 887–894. DOI: 10.17586/2226-1494-2021-21-6-887-894
14. Марков Г.А. Исследование результативности обнаружения аномалий в телеметрии узлов промышленных систем интернет вещей методами машинного обучения без учителя // Безопасность информационных технологий. 2026. № 33(2). С. 158–167. DOI: 10.26583/bit.2026.2.15
15. Lavin A., Ahmad S. (2015) Evaluating Real-Time Anomaly Detection Algorithms - The Numenta Anomaly Benchmark // Proceedings of the 14th IEEE International Conference on Machine Learning and Applications (ICMLA). Pp. 38–44.
16. Katser I., Kozitsin V. (2020) Skoltech Anomaly Benchmark (SKAB) [Электронный ресурс]. URL: <https://github.com/waico/SKAB> (дата обращения: 14.11.2025).

### References

1. Alekseev P.O. (2025) Optimization of web application performance using modern frameworks. *Universum: Technical Sciences: Electronic Scientific Journal*. No. 4(133) [Electronic resource]. Available at: <https://7universum.com/ru/tech/archive/item/19668> (accessed: 08.02.2025). (In Russian).
2. Lutaev D.O. (2023) Research and analysis of web application performance optimization methods. *Modern Scientific Research and Innovations*. No. 10 [Electronic resource]. Available at: <https://web.snauka.ru/issues/2023/10/100843> (accessed: 14.11.2025) (In Russian).
3. Google Developers. Web Vitals. Available at: <https://web.dev/vitals/> (accessed: 14.11.2025).
4. Popova I. (2020) Anomaly detection in datasets using unsupervised machine learning algorithms Isolation Forest and Local Outlier Factor. *StudNet*. No. 12. Pp. 1–8. (In Russian).
5. Kuznetsov A.S., Semenov E.Yu. (2017) Selection of anomaly detection methods in educational data. *Software Engineering*. No. 6. Pp. 267–274. (In Russian).
6. Lindigrin A.N. (2019) Comparative analysis of machine learning methods in network anomaly detection tasks. Proceedings of Tula State University. *Technical Sciences*. Is.12. Pp. 400–404. (In Russian).
7. Chandola V., Banerjee A., Kumar V. (2009) Anomaly detection: A survey. *ACM Computing Surveys*. Vol. 41. No. 3. Art. 15.
8. Chernyshev Yu.Yu. (2022) Application of autoencoders for anomaly detection in cyber-physical systems. *Bulletin of Perm University. Mathematics. Mechanics. Computer Science*. Is. 4(59). Pp. 89–94. DOI: 10.17072/1993-0550-2022-4-89-94 (In Russian).
9. Liu F.T., Ting K.M., Zhou Z.-H. (2008) Isolation Forest. Proceedings of the 8th IEEE International Conference on Data Mining (ICDM). Pp. 413–422.
10. Breunig M.M., Kriegel H.-P., Ng R.T., Sander J. (2000) LOF: Identifying density-based local outliers. *Proceedings of ACM SIGMOD*. Pp. 93–104.
11. Zhao Y., Nasrullah Z., Li Z. (2019) PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research*. Vol. 20. Pp. 1–7.
12. Shukurov N.A. (2024) Algorithms for anomaly detection in heterogeneous datasets. *Information Technologies and Security*. No. 1. Pp. 38–41. (In Russian).
13. Semenov V.V. (2021) An approach to informative feature construction in cybersecurity monitoring of cyber-physical systems. *Scientific and Technical Bulletin of Information Technologies, Mechanics and Optics*. Vol. 21. No. 6. Pp. 887–894. DOI: 10.17586/2226-1494-2021-21-6-887-894 (In Russian).

14. Markov G.A. (2026) Study of the effectiveness of anomaly detection in telemetry of industrial IoT nodes using unsupervised machine learning methods. *Information Technology Security Journal*. Vol. 33(2). Pp. 158–167. DOI: 10.26583/bit.2026.2.15 (In Russian).
15. Lavin A., Ahmad S. (2015) Evaluating real-time anomaly detection algorithms: The Numenta Anomaly Benchmark. Proceedings of the 14th IEEE International Conference on Machine Learning and Applications (ICMLA). Pp. 38–44.
16. Katser I., Kozitsin V. (2020). Skoltech Anomaly Benchmark (SKAB) [Electronic resource]. Available at: <https://github.com/waico/SKAB> (accessed: 14.11.2025).

Поступила в редакцию: 17.06.2026

Received: 17.06.2026

Поступила после рецензирования: 22.07.2026

Revised: 22.07.2026

Принята к публикации: 03.08.2026

Accepted: 03.08.2026