МРНТИ 50.41.01
УДК 004.42 **DOI 10.70239/arsu.2024.t78.n4.02**

ВЫЯВЛЕНИЕ АНОМАЛИЙ В ДАННЫХ МОНИТОРИНГА ПРОИЗВОДИТЕЛЬНОСТИ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА ISOLATION FOREST: ВОЗМОЖНОСТИ И ОГРАНИЧЕНИЯ

КЕРЕЕВ А.К.[1], МИХЕЛЬСОН О.Ю.[2]*

Кереев Адилжан Кутымович[1] - PhD, доцент кафедры Информатики и информационных технологий, Актюбинский региональный университет им. К. Жубанова, Актобе, Казахстан
E-mail: akereyev@zhubanov.edu.kz, https://orcid.org/0000-0002-8283-5807;
***Михельсон Олег Юрьевич[2]** - Старший инженер инфраструктуры, ActivSoft, Актобе, Казахстан
E-mail: miol@activsoft.kz, https://orcid.org/0009-0009-6753-3120;

Аннотация. Статья посвящена применению алгоритма Isolation Forest для выявления аномалий в данных мониторинга серверов SaaS проекта. Основная гипотеза исследования заключается в том, что алгоритм может обнаруживать ранние признаки деградации производительности и потенциальные сбои, анализируя базовые метрики, такие как загрузка процессора, использование оперативной памяти, сетевого трафика и дискового пространства. В исследовании тестировались два подхода: первый предполагал анализ каждой метрики отдельно, второй — агрегирование всех метрик в единый показатель для оценки общего состояния системы. Результаты экспериментов показали, что Isolation Forest демонстрирует высокую чувствительность к резким изменениям метрик, что приводит к значительному числу ложных срабатываний. Это особенно актуально при краткосрочных всплесках метрик, которые не обязательно свидетельствуют о реальных проблемах в системе. В статье рассматриваются ограничения этого подхода, включая необходимость точной настройки гиперпараметров, а также предложены возможные решения для улучшения точности выявления аномалий, такие как предварительная обработка данных и комбинирование с другими методами. Данное исследование подчеркивает важность применения продвинутых методов машинного обучения для мониторинга производительности серверов, особенно в условиях ограниченных метрик, характерных для систем с закрытым исходным кодом.

Ключевые слова: выявление аномалий, isolation forest, мониторинг производительности, метрики серверов, prometheus, ложные срабатывания

Введение

Эксплуатация программных систем с закрытым исходным кодом часто сталкивается с проблемами недостаточности передаваемых метрик в систему мониторинга. В отсутствие доступа к внутренним метрикам программной системы инженеры вынуждены полагаться на базовые метрики производительности серверов. Это усложняет своевременное выявление сбоев и может привести к значительным последствиям, таким как потеря данных или остановка сервисов. Разработка методов для раннего обнаружения таких проблем является актуальной задачей.

Одним из актуальных подходов для решения задач раннего обнаружения проблем производительности является выявление аномалий методами машинного обучения. Например, в работе Ronchieri et al. (2024) [1, с. 4] применены статистические подходы и алгоритмы кластеризации (DBSCAN) для выявления аномалий в инфраструктуре дата-центров. Исследование подтверждает важность предсказательной аналитики для предотвращения критических инцидентов в IT-инфраструктуре.

В другой работе, Bursic, S., Cuculo, V., D'Amelio, A. (2020) [2, с. 2], рассматриваются комбинированные подходы для выявления аномалий в лог-файлах и метриках мониторинга серверов, демонстрируя эффективность обработки временных рядов и текстовых данных для своевременного выявления проблем.

Исследование Gu et al. (2023) [3, с. 9] показало, что методы глубокого обучения успешно применяются для выявления аномалий в облачных системах, хотя эти модели требуют значительных вычислительных ресурсов и больших объемов данных для обучения.

Алгоритм Isolation Forest [4, с. 10] широко используется для обнаружения аномалий в различных областях, таких как анализ сетевых данных, финансы и производственные системы. Хотя исследований по применению этого алгоритма достаточно много, они в

основном фокусируются на других типах данных. Текущее исследование предлагает использование Isolation Forest для анализа базовых метрик производительности серверов.

Гипотеза исследования заключается в том, что использование алгоритма Isolation Forest для выявления аномалий в данных мониторинга производительности серверов позволяет заранее обнаруживать деградацию производительности и потенциальные сбои в эксплуатации систем с закрытым исходным кодом. Предполагается, что аномальные отклонения в базовых метриках, таких как использование процессора, оперативной памяти, сетевого трафика и дискового пространства, могут служить ранними индикаторами возможных проблем.

Материалы и методы

Исследование проводилось на данных мониторинга двух физических серверов производственной системы действующего SaaS проекта. Для мониторинга использовалась система Prometheus [5], собирающая данные каждую минуту. Данные запрашивались непосредственно с системы мониторинга с помощью API. Были использованы следующие метрики:

Нагрузка на процессор (CPU busy): процент использования процессора. PromQL запрос:

```
'(((count(count(node_cpu_seconds_total{instance="host"}) by (cpu))) - avg(sum by
(mode)(rate(node_cpu_seconds_total{mode="idle",instance="host"}[5m15s])))) * 100) /
count(count(node_cpu_seconds_total{instance="host"}) by (cpu))'
```

Использование оперативной памяти (RAM used): объем занятой оперативной памяти, в процентах от общего объема. PromQL запрос:

```
'100 - ((node_memory_MemAvailable_bytes{instance="host"} * 100) / node_memory_MemTotal_bytes{instance="host"})'
```

Сетевой трафик входящий (Network traffic in): объем входящего сетевого трафика, в процентах от емкости канала. PromQL запрос:

```
'rate(node_network_receive_bytes_total{instance="host"}[2m15s]) * 8 / 1024 /1024'
```

Сетевой трафик исходящий (Network traffic out): объем исходящего сетевого трафика, в процентах от емкости канала. PromQL запрос:

```
'rate(node_network_transmit_bytes_total{instance="host"}[2m15s]) * 8 / 1024 /1024'
```

Использование дискового пространства (Disk usage): процент использования дискового пространства. PromQL запрос:

```
'100 - ((node_filesystem_avail_bytes{instance="host",mountpoint="/",fstype!="rootfs"} * 100) /
node_filesystem_size_bytes{instance="host",mountpoint="/",fstype!="rootfs"})'
```

Использование файла подкачки (Swap usage): процент использования файла подкачки. PromQL запрос:

```
'((node_memory_SwapTotal_bytes{instance="host"} - node_memory_SwapFree_bytes{instance="host"}) /
(node_memory_SwapTotal_bytes{instance="host"} )) * 100'
```

В один набор помещались данные за определенные временные интервалы наблюдений. Все получаемые значения метрик округлялись до двух знаков после запятой.

В ходе исследования были опробованы два подхода:

1. Показатели без суммирования: В этом подходе на вход алгоритму передавались все показатели одновременно, без их суммирования. Предполагалось, что данный подход позволит учитывать индивидуальные характеристики каждой метрики и их взаимодействие.

2. Суммирование показателей: В этом подходе все показатели суммировались в единое среднее значение. Предполагалось, что данный подход позволит учитывать общее состояние системы на основе всех метрик.

Для проверки гипотезы использовалась реализация алгоритма Isolation Forest из библиотеки scikit-learn [6] на языке программирования python.

```
import pandas as pd
from sklearn.ensemble import IsolationForest

df = pd.DataFrame(data)
iso = IsolationForest()
iso.fit(df)

ndf = df.copy()
ndf['scores'] = iso.decision_function(df)
ndf['anomaly'] = iso.predict(df)
```

Показатель и признак аномальности сохранялись в копии набора данных для анализа. Полученные результаты сравнивались с журналом инцидентов SaaS проекта.

Образец набора данных:

№	memory	swap	cpu_b	net_in	net_out	scores	anomaly
0	34.81	0.0	7.44	16.41	2.43	0.008121	1
1	34.81	0.0	7.47	16.29	2.75	0.010787	1
2	34.81	0.0	7.75	23.59	2.98	-0.044432	-1
3	34.83	0.0	8.22	23.15	2.92	-0.008589	-1
4	34.92	0.0	8.92	19.03	2.67	0.070903	1
5	34.93	0.0	9.85	29.75	4.55	-0.173069	-1
6	34.95	0.0	10.03	19.41	2.91	0.029035	1
7	35.02	0.0	9.88	19.84	1.59	0.000030	1
8	34.96	0.0	9.88	20.75	2.80	0.056216	1
9	35.00	0.0	9.22	21.64	2.82	0.086186	1
10	34.98	0.0	8.47	16.66	1.37	0.069377	1
…							

Таблица 1. - Образец набора данных

Результаты

В результате применения алгоритма Isolation Forest на данных мониторинга производственных серверов были выявлены несколько аномалий. Использование подхода без суммирования метрик продемонстрировало высокую чувствительность алгоритма к отдельным метрикам, что привело к большому количеству ложных срабатываний. Это можно объяснить тем, что резкие, но кратковременные изменения в одной метрике, такие как скачки использования процессора или сетевого трафика, не всегда указывают на реальную деградацию производительности.

При суммировании показателей в единое среднее значение алгоритм показал меньшую чувствительность к кратковременным скачкам, однако это привело к упущению некоторых потенциально значимых аномалий, которые могли предвещать проблемы на уровне отдельных метрик.

Ограничения

Одним из главных ограничений использования Isolation Forest является его чувствительность к внезапным изменениям в данных. В условиях, когда метрики могут резко меняться, например, при кратковременных пиках нагрузки, алгоритм может интерпретировать такие скачки как аномалии, что приводит к увеличению количества ложных срабатываний. Это особенно актуально для метрик сетевого трафика и использования процессора, где даже незначительные колебания могут быть ошибочно приняты за аномалии.

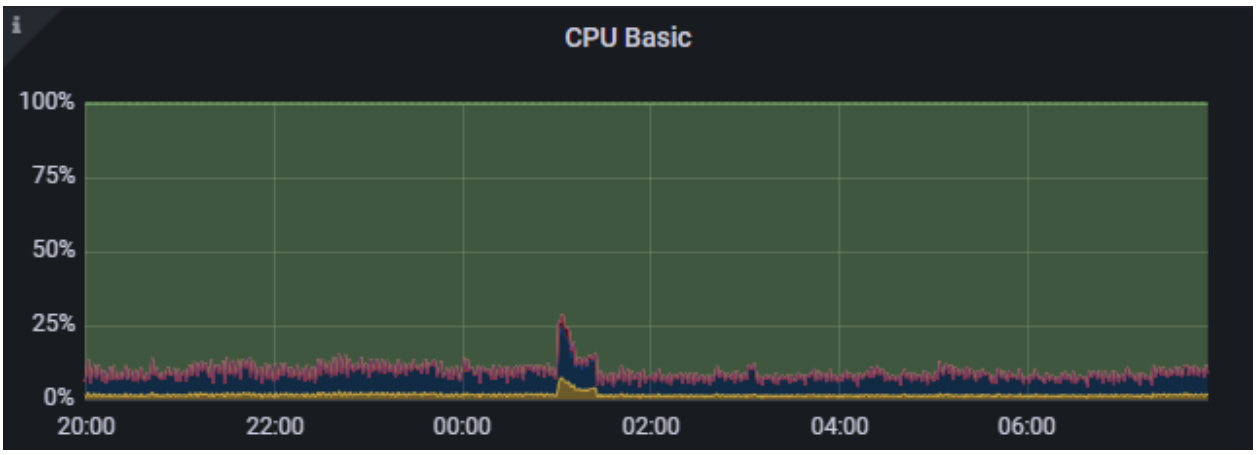

Рисунок 1. - График использования процессора

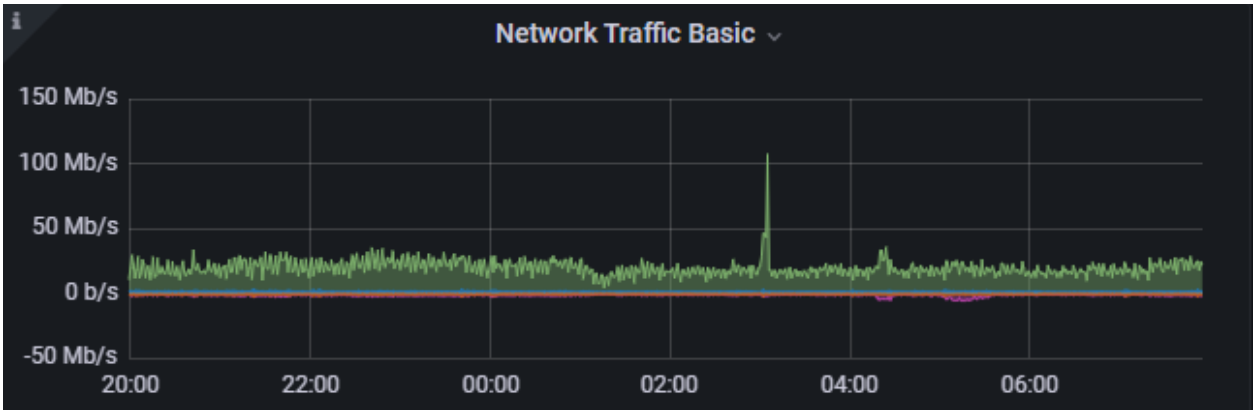

Рисунок 2. - График сетевого траффика

Еще одним ограничением является необходимость предварительной настройки гиперпараметров алгоритма, таких как количество и глубина деревьев. В данном исследовании использовались параметры по умолчанию из библиотеки scikit-learn, что могло повлиять на итоговые результаты.

Обсуждение

В ходе исследования стало очевидным, что Isolation Forest эффективен для выявления аномалий в условиях, когда наблюдаются длительные отклонения метрик от нормы. Однако кратковременные изменения и всплески в данных часто приводят к ложным срабатываниям. В случае реальных эксплуатационных систем, где колебания метрик могут быть обусловлены различными процессами, это может затруднять использование алгоритма в его стандартной конфигурации.

Кроме того, результаты показывают, что подходы с суммированием и без суммирования метрик имеют свои преимущества и недостатки. Подход без суммирования позволяет более точно выявлять аномалии на уровне отдельных метрик, но повышает количество ложных срабатываний. В то же время суммирование метрик снижает чувствительность алгоритма, но может упустить важные аномалии.

Заключение

Алгоритм Isolation Forest показал себя как потенциально полезный инструмент для выявления аномалий в данных мониторинга производительности серверов. Однако его высокая чувствительность к кратковременным изменениям может ограничивать его применение в условиях нестабильных систем. Для улучшения результатов может быть предложено комбинирование Isolation Forest с другими методами выявления аномалий, а также более тщательная настройка гиперпараметров алгоритма.

В будущем целесообразно рассмотреть использование методов предобработки данных для сглаживания кратковременных всплесков в метриках, а также исследовать возможность использования более сложных ансамблевых методов для уменьшения ложных срабатываний.

Список литературы

1. Ronchieri E. Anomaly Detection in Data Center IT & Physical Infrastructure / Elisabetta Ronchieri, Luca Giommi, Luigi Benedettto Scarponi, Luca Torzi, Alessandro Costantini, Doina Cristina Duma, Davide Salomoni // EPJ Web of Conf. 295 07004 (2024) DOI: 10.1051/epjconf/202429507004

2. Bursic S. Anomaly Detection from Log Files Using Unsupervised Deep Learning. / Bursic Sathya, Cuculo Vittorio, D'Amelio Alessandro // Formal Methods. FM 2019 International Workshops. FM 2019. Lecture Notes in Computer Science (), vol 12232. Springer, Cham. DOI: 10.1007/978-3-030-54994-7_15

3. Gu W. Performance Issue Identification in Cloud Systems with Relational-Temporal Anomaly Detection / Wenwei Gu, Jinyang Liu, Zhuangbin Chen, Jianping Zhang, Yuxin Su, Jiazhen Gu, Cong Feng, Zengyin Yang, Michael Lyu // arXiv, 2023. https://arxiv.org/abs/2307.10869

4. Liu F. "Isolation Forest" / Fei Tony Liu, Kai Ming Ting, Zhi-Hua Zhou // 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 2008, pp. 413-422, DOI: 10.1109/ICDM.2008.17.

5. Prometheus. Prometheus documentation. [Электронный ресурс]. – Режим доступа: https://prometheus.io/docs/ (дата обращения: 07.10.2024).

6. Scikit Learn. Scikit Learn user guide. [Электронный ресурс]. – Режим доступа: https://scikit-learn.org/stable/user_guide.html (дата обращения: 07.10.2024).

References

1. Ronchieri E. Anomaly Detection in Data Center IT & Physical Infrastructure / Elisabetta Ronchieri, Luca Giommi, Luigi Benedettto Scarponi, Luca Torzi, Alessandro Costantini, Doina Cristina Duma, Davide Salomoni // EPJ Web of Conf. 295 07004 (2024) DOI: 10.1051/epjconf/202429507004

2. Bursic S. Anomaly Detection from Log Files Using Unsupervised Deep Learning. / Bursic Sathya, Cuculo Vittorio, D'Amelio Alessandro // Formal Methods. FM 2019 International Workshops. FM 2019. Lecture Notes in Computer Science (), vol 12232. Springer, Cham. DOI: 10.1007/978-3-030-54994-7_15

3. Gu W. Performance Issue Identification in Cloud Systems with Relational-Temporal Anomaly Detection / Wenwei Gu, Jinyang Liu, Zhuangbin Chen, Jianping Zhang, Yuxin Su, Jiazhen Gu, Cong Feng, Zengyin Yang, Michael Lyu // arXiv, 2023. https://arxiv.org/abs/2307.10869

4. Liu F. "Isolation Forest" / Fei Tony Liu, Kai Ming Ting, Zhi-Hua Zhou // 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 2008, pp. 413-422, DOI: 10.1109/ICDM.2008.17.

5. Prometheus. Prometheus documentation. [Электронный ресурс]. – Режим доступа: https://prometheus.io/docs/ (дата обращения: 07.10.2024).

6. Scikit Learn. Scikit Learn user guide. [Электронный ресурс]. – Режим доступа: https://scikit-learn.org/stable/user_guide.html (дата обращения: 07.10.2024).

DETECTION OF ANOMALIES IN PERFORMANCE MONITORING DATA USING THE ISOLATION FOREST ALGORITHM: CAPABILITIES AND LIMITATIONS

KEREEV A.K.[1], MIKHELSON O.Yu.[2*]

Kereev Adilzhan Kutymovich[1] - PhD, Associate Professor of the Department of Computer Science and Information Technology, Aktobe Regional University named after K. Zhubanov, Aktobe, Kazakhstan
E-mail: akereyev@zhubanov.edu.kz, https://orcid.org/0000-0002-8283-5807
***Mikhelson Oleg Yuryevich[2]** - Senior Infrastructure Engineer, ActivSoft, Aktobe, Kazakhstan
E-mail: miol@activsoft.kz, https://orcid.org/0009-0009-6753-3120

Abstract. This paper explores the application of the Isolation Forest algorithm for detecting anomalies in

performance monitoring data of a SaaS project's servers. The main hypothesis suggests that the algorithm can identify early signs of performance degradation and potential failures by analyzing basic metrics such as CPU load, memory usage, network traffic, and disk space. Two approaches were tested: analyzing each metric separately and aggregating them into a single indicator to assess the overall system state. The results showed that Isolation Forest demonstrates high sensitivity to sudden changes in metrics, leading to a significant number of false positives. This issue is particularly relevant when dealing with short-term metric spikes that do not necessarily indicate real system problems. The paper discusses the limitations of this approach, including the need for fine-tuning hyperparameters, and suggests possible solutions for improving anomaly detection accuracy, such as preprocessing data and combining it with other methods. This study highlights the importance of advanced machine learning techniques in server performance monitoring, especially in conditions with limited metrics, typical of closed-source systems.

Key words: anomaly detection, Isolation Forest, performance monitoring, server metrics, Prometheus, false positives

ISOLATION FOREST АЛГОРИТМІН ҚОЛДАНА ОТЫРЫП, ӨНІМДІЛІКТІ БАҚЫЛАУ ДЕРЕКТЕРІНДЕГІ АУЫТҚУЛАРДЫ АНЫҚТАУ: МҮМКІНДІКТЕР МЕН ШЕКТЕУЛЕР

КЕРЕЕВ А.К.[1], МИХЕЛЬСОН О.Ю.[2]*

Кереев Адилжан Кутымович[1] - PhD, «Информатика және ақпараттық технологиялар» кафедрасының доценті, Қ.Жұбанова атындағы Ақтөбе өңірлік университеті, Ақтөбе қ., Қазақстан
E-mail: akereyev@zhubanov.edu.kz, https://orcid.org/0000-0002-8283-5807;
***Михельсон Олег Юрьевич[2]** - Инфрақұрылымның аға инженері, ActivSoft, Ақтөбе қ., Қазақстан
E-mail: miol@activsoft.kz, https://orcid.org/0009-0009-6753-3120;

Аңдатпа. Серверлердің өнімділігін бақылау – қазіргі заманғы SaaS жобалары үшін аса маңызды мәселе. Мұндай жүйелердің тұрақты жұмыс істеуі мен тиімділігі үшін серверлердің күйін тұрақты түрде қадағалау қажет

Бұл мақалада Isolation Forest алгоритмінің серверлердің өнімділігі мен күйін бақылауға негізделген аномалияларды анықтау мүмкіндіктері қарастырылады. Зерттеудің негізгі гипотезасы – аталмыш алгоритмнің процессор жүктемесі, жедел жадының пайдалану деңгейі, желілік трафик және дискілік кеңістік сияқты негізгі метрикаларды талдау арқылы өнімділіктің төмендеуі мен мүмкін болатын ақауларды ерте кезеңде анықтауға қабілеттілігі.

Зерттеу барысында екі түрлі тәсіл сынақтан өтті: біріншісі әрбір метриканы бөлек талдау, екіншісі – барлық метрикаларды бір көрсеткішке агрегаттау арқылы жүйенің жалпы жағдайын бағалау. Эксперименттердің нәтижелері Isolation Forest алгоритмінің метрикалардағы күрт өзгерістерге өте сезімтал екенін көрсетті, бұл өз кезегінде көптеген жалған дабылдарға әкеледі. Бұл мәселе әсіресе қысқа мерзімді метрикалық серпіндер кезінде маңызды, өйткені олар жүйеде нақты мәселелердің болуын білдірмейді. Мақалада осы тәсілдің шектеулері, соның ішінде гиперпараметрлерді дәл баптаудың қажеттілігі қарастырылады. Сонымен қатар, аномалияларды анықтау дәлдігін арттыру үшін деректерді алдын ала өңдеу және басқа әдістермен үйлестіру сияқты ықтимал шешімдер ұсынылады.

Бұл зерттеу серверлердің өнімділігін бақылау кезінде машиналық оқытудың алдыңғы қатарлы әдістерін қолданудың маңыздылығын көрсетеді, әсіресе жабық бастапқы кодты жүйелер үшін шектеулі метрикалар жағдайында маңызды.

Түйін сөздер: аномалияларды анықтау, Isolation Forest, өнімділікті бақылау, сервер метрикалары, Prometheus, жалған дабылдар