

Применение машинного обучения для прогнозирования загрузки вычислительных ресурсов сервера

Кокухина Софья Александровна

Приамурский государственный университет им.Шолом-Алейхема

Студент

Аннотация

В статье рассматривается применение методов машинного обучения для краткосрочного прогнозирования загрузки вычислительных ресурсов сервера. На основе сформированного набора данных, содержащего сведения о загрузке центрального процессора, использовании оперативной памяти, количестве активных пользователей и процессов, сетевой и дисковой активности, разработана модель прогнозирования на основе алгоритма Random Forest Regression. Проведена оценка точности прогнозирования загрузки центрального процессора и оперативной памяти с использованием метрик MAE, RMSE и коэффициента детерминации R². Полученные результаты демонстрируют возможность применения методов машинного обучения для заблаговременного обнаружения роста нагрузки и последующей автоматизации управления серверными ресурсами.

Ключевые слова: машинное обучение, Random Forest, сервер, прогнозирование нагрузки, CPU, RAM, системное администрирование, Python, мониторинг ресурсов.

Application of Machine Learning for Predicting Server Computing Resource Utilization

Kokukhina Sofya Alexandrovna

Sholom-Aleichem Priamursky State University

Student

Abstract

The article considers the application of machine learning methods for short-term prediction of server computing resource utilization. Based on a generated dataset containing information about CPU utilization, memory usage, number of active users and processes, network activity and disk input/output, a prediction model based on the Random Forest Regression algorithm was developed. The accuracy of CPU and RAM utilization prediction was evaluated using MAE, RMSE and the coefficient of determination R². The obtained results demonstrate the possibility of using machine learning methods for early detection of increasing server load and further automation of computing resource management.

Keywords: machine learning, Random Forest, server, workload prediction, CPU, RAM, system administration, Python, resource monitoring.

1 Введение

1.1 Актуальность

Современные информационные системы предъявляют повышенные требования к доступности и производительности серверной инфраструктуры. Серверы обеспечивают работу веб-приложений, баз данных, корпоративных информационных систем, файловых хранилищ и других сервисов. При этом характер нагрузки может существенно изменяться в течение суток в зависимости от количества пользователей, выполняемых процессов и объёма обрабатываемых данных.

Традиционный мониторинг вычислительных ресурсов позволяет системному администратору наблюдать текущее состояние оборудования, однако чаще всего носит реактивный характер. Администратор получает уведомление о высокой загрузке CPU или недостатке оперативной памяти уже после достижения заданного порогового значения. В результате между возникновением повышенной нагрузки и реакцией специалиста может пройти определённое время.

Одним из способов решения данной проблемы является применение методов машинного обучения. Анализируя накопленные показатели работы сервера, модель может определять закономерности изменения нагрузки и прогнозировать состояние вычислительных ресурсов на некоторый промежуток времени вперёд.

Такой подход может применяться для формирования предупредительных уведомлений, динамического масштабирования вычислительных ресурсов, управления виртуальными машинами и контейнерами, а также планирования технического обслуживания серверной инфраструктуры.

1.2 Обзор исследований

Проблема прогнозирования нагрузки вычислительных систем исследуется на протяжении длительного времени. П.А. Динда и Д.Р.О'Хэлларон исследовали возможность прогнозирования нагрузки вычислительного узла на основе исторических значений и показали, что даже сравнительно простые модели позволяют получать практически применимые результаты прогнозирования [1]. С. Ислам, Дж. Кеунг, К. Ли и А. Лю рассматривали прогнозирование потребления ресурсов в облачной инфраструктуре с использованием линейной регрессии и нейронных сетей. Авторы предложили использовать прогноз будущей нагрузки для заблаговременного выделения вычислительных ресурсов [2]. Р.Н. Калейрос, Э. Масуми, Р. Ранджан и Р. Буйя исследовали прогнозирование облачной нагрузки с применением модели ARIMA [3]. М. Борковски, С. Шульте и К. Хохрайнер предложили применение методов машинного обучения для прогнозирования будущего потребления облачных ресурсов на уровне отдельных вычислительных задач. Проведённые эксперименты показали возможность снижения ошибки оценки потребления ресурсов за счёт применения методов машинного обучения [4]. Основой применяемого в

настоящей работе алгоритма является метод Random Forest, предложенный Л. Брейманом. Метод основан на объединении большого количества деревьев решений и может применяться как для классификации, так и для решения задач регрессии [5].

1.3 Цель исследования

Целью исследования является разработка и экспериментальная оценка модели машинного обучения для краткосрочного прогнозирования загрузки центрального процессора и использования оперативной памяти сервера на основе текущих и исторических показателей его работы.

2 Материалы и методы

Для проведения эксперимента разработана программа на языке Python с использованием библиотек NumPy, Pandas, Matplotlib и scikit-learn.

В качестве алгоритма машинного обучения выбран Random Forest Regression, представляющий собой ансамбль деревьев решений. Каждое дерево формирует собственный прогноз, после чего результат определяется на основании совокупности полученных значений.

В рамках быстрого воспроизводимого эксперимента сформирован синтетический набор показателей работы сервера за 30 суток. Измерения выполняются с интервалом в пять минут. Такой подход позволяет получить достаточно большой объём экспериментальных данных без необходимости длительного фактического мониторинга сервера.

Итоговый набор после подготовки содержит 8636 записей.

Для каждой записи используются следующие характеристики состояния системы:

- загрузка центрального процессора CPU, %;
- использование оперативной памяти RAM, %;
- количество активных пользователей;
- количество запущенных процессов;
- объём сетевого трафика;
- интенсивность дисковых операций;
- температура процессора;
- час суток;
- день недели;
- значения CPU, RAM, сетевой активности и числа пользователей за предыдущие 5, 10 и 15 минут.

Целевыми параметрами являются загрузка CPU и использование RAM через следующие пять минут.

Для сохранения временной последовательности записи перед обучением модели не перемешивались. Первые 80 % набора данных использованы для обучения модели, оставшиеся 20 % — для тестирования.

Количество записей обучающей выборки составило 6908, тестовой — 1728 (Рис.1).

Первые строки исходного набора данных:

	timestamp	cpu_usage	ram_usage	active_users	process_count	\
0	2026-01-01 00:00:00	27.255641	37.457424	9	147.404539	
1	2026-01-01 00:05:00	22.540636	37.781832	7	114.754902	
2	2026-01-01 00:10:00	23.928351	41.399959	10	143.720338	
3	2026-01-01 00:15:00	21.570347	38.902975	14	140.218304	
4	2026-01-01 00:20:00	18.711354	37.785414	7	127.026338	

	network_mbps	disk_io	temperature
0	15.464225	14.686179	43.012213
1	14.968465	17.534267	41.388752
2	19.725504	10.992425	40.436384
3	9.728126	25.512434	39.862213
4	17.092693	14.427473	36.404326

Количество обучающих записей: 6908
Количество тестовых записей: 1728

Рисунок 1 – Формирование обучающей и тестовой выборок

Для прогнозирования CPU и RAM создаются две модели RandomForestRegressor. Каждая модель содержит 150 деревьев решений с максимальной глубиной 12.

Фрагмент конфигурации модели имеет следующий вид:

```
cpu_model = RandomForestRegressor(
    n_estimators=150,
    max_depth=12,
    random_state=42,
    n_jobs=-1
)
```

Качество полученных моделей оценивается с использованием трёх метрик: средней абсолютной ошибки (MAE), корня из средней квадратичной ошибки (RMSE) и коэффициента детерминации R^2

3 Результаты

3.1 Формирование исходного набора данных

На первом этапе программой сформированы временные ряды характеристик сервера (Табл.1). В генерации учитывается изменение активности в зависимости от времени суток и дня недели. В рабочие часы увеличивается количество пользователей, процессов, сетевых операций и, как следствие, вычислительная нагрузка (Рис.2).

Фрагмент набора данных имеет следующую структуру.

Таблица 1 – Параметры и назначения исходного набора

Параметр	Назначение
timestamp	Дата и время измерения
cpu_usage	Загрузка CPU, %
ram_usage	Использование RAM, %
active_users	Количество активных пользователей

process_count	Количество процессов
network_mbps	Сетевая активность, Мбит/с
disk_io	Интенсивность дисковых операций
temperature	Температура CPU

```

1 timestamp,cpu_usage,ram_usage,active_users,process_count,network_mbps,disk_io,temperature
2 2026-01-01 00:00:00,27.2556405599249,37.45742375766615,9,147.4045390942602,15.464225376449345,14.686178557271994,43.01221272674552
3 2026-01-01 00:05:00,22.540635702174914,37.78183237056095,7,114.75490241034545,14.968465226252576,17.53426728879338,41.38875184735349
4 2026-01-01 00:10:00,23.928351448753574,41.39995945977927,10,143.72033836246993,19.725504358753525,10.992425480386709,40.43638366465411
5 2026-01-01 00:15:00,21.570347158767404,38.90297497802817,14,140.2183038487244,9.728125600386122,25.512434493542045,39.86221305618935
6 2026-01-01 00:20:00,18.711354461403968,37.785414254379475,7,127.02633825081143,17.09269264435411,14.427473267315943,36.4043257340329
7 2026-01-01 00:25:00,28.747953031675937,37.24660409456156,7,100.35629702378318,12.925230627010187,6.122352495864771,41.240854315874444
8 2026-01-01 00:30:00,31.41880909804218,38.60165106114436,14,111.56503976153138,15.91790892353113,13.019207543657775,42.078313091018224
9 2026-01-01 00:35:00,32.5500292522505,42.37258788069949,11,129.35320871679224,10.502664392432688,20.283996656867274,42.719802838056
10 2026-01-01 00:40:00,21.824295281900135,37.68755894259835,6,124.96287276099935,6.327575343545236,16.572916571849575,40.94676454922173
11 2026-01-01 00:45:00,26.80034398289028,35.113946883791264,10,116.49762606714732,4.6100665071160645,14.07237392624201,43.302677389564096
12 2026-01-01 00:50:00,31.121665087781423,35.16294714713758,6,108.9356790742839,22.778139127896203,12.199670812231407,43.80844414132744
13 2026-01-01 00:55:00,31.718910848617085,37.35624076203211,6,128.7376715579136,20.889210601146754,6.941558905096115,42.61650903824119
14 2026-01-01 01:00:00,30.854689347267975,41.04662243285814,8,132.75425376548557,15.613601607807514,18.085770960941367,41.679896214043495
15 2026-01-01 01:05:00,18.22702651292892,33.34429022871929,0,108.34710727534488,0.0,4.048931652902131,38.51524539894098
16 2026-01-01 01:10:00,21.039998742908605,33.421335023251316,1,121.36242065423289,0.0,11.97896444449367,39.80878293234186
17 2026-01-01 01:15:00,24.363193869100822,37.11637087697747,5,105.90957245135824,13.564525649391744,16.408169474660554,40.19564424765848
18 2026-01-01 01:20:00,21.103741090812594,32.71187168779486,3,111.72878009495635,18.562615196015607,9.992622216878036,38.397710369787475
19 2026-01-01 01:25:00,26.586193412366697,34.74715333183282,9,147.77453126940406,8.936826083226055,17.270445464488795,42.2280509323831
20 2026-01-01 01:30:00,26.030465059168396,35.71808619014511,4,108.81986555927782,0.0,17.790558832102775,42.18094920402042
21 2026-01-01 01:35:00,15.62376740826863,31.997630610481593,2,115.51575017504399,15.381126188762169,10.271875196842952,35.50041805943289
22 2026-01-01 01:40:00,38.24923236452116,46.367919908907496,13,163.67135903161287,5.2028912993243885,15.098792734272216,44.811627541036856
23 2026-01-01 01:45:00,22.753163756217788,35.53470112549791,7,107.20789400981475,2.4317535339788297,11.932588569793934,41.13045985409699
24 2026-01-01 01:50:00,21.52365514998759,38.11917322348427,8,135.55716439970533,10.025136859913177,9.828048389311647,41.01615112226231
25 2026-01-01 01:55:00,18.316372337098414,32.018061496988885,2,133.9678353576972,0.0,6.137664829052507,38.507580535153046
26 2026-01-01 02:00:00,20.119229095932,37.47506072794992,5,116.10170488201744,16.397192502452786,12.769627055302813,39.589731656885036
27 2026-01-01 02:05:00,25.22150043682077,36.303672280401895,8,136.31556944225815,18.8552421640408,12.144200711407862,41.20632326689928
28 2026-01-01 02:10:00,17.534074226924795,32.686210841595326,3,104.21576603346914,0.0,11.1107676707405053,30.568703037248557

```

Рисунок 2 – Фрагмент сформированного набора показателей работы сервера

Следует отметить, что применяемый в эксперименте набор данных является синтетическим. Его использование позволяет обеспечить воспроизводимость исследования, однако при практическом внедрении модели необходимо выполнять дополнительное обучение и тестирование на показателях конкретного сервера.

3.2 Обучение модели прогнозирования

После формирования исходной выборки для каждого временного момента дополнительно рассчитываются значения характеристик за предыдущие 5, 10 и 15 минут.

Таким образом, модель получает информацию как о текущем состоянии вычислительной системы, так и о динамике изменения нагрузки.

На основании сформированных признаков обучены две модели:

первая модель предназначена для определения загрузки центрального процессора через пять минут;

вторая модель используется для прогнозирования использования оперативной памяти через пять минут.

После завершения обучения модели применены к тестовой части набора данных, которая ранее не использовалась в процессе обучения.

3.3 Результаты прогнозирования загрузки CPU

В результате эксперимента получены следующие показатели для прогнозирования центрального процессора:

MAE = 5,50 %

$$\text{RMSE} = 7,39 \%$$

$$R^2 = 0,929$$

Коэффициент детерминации 0,929 показывает, что модель описывает значительную часть вариации целевого показателя на сформированной тестовой выборке.

Для визуального анализа выполнено сравнение фактической и прогнозируемой загрузки CPU в течение суток (Рис.3).

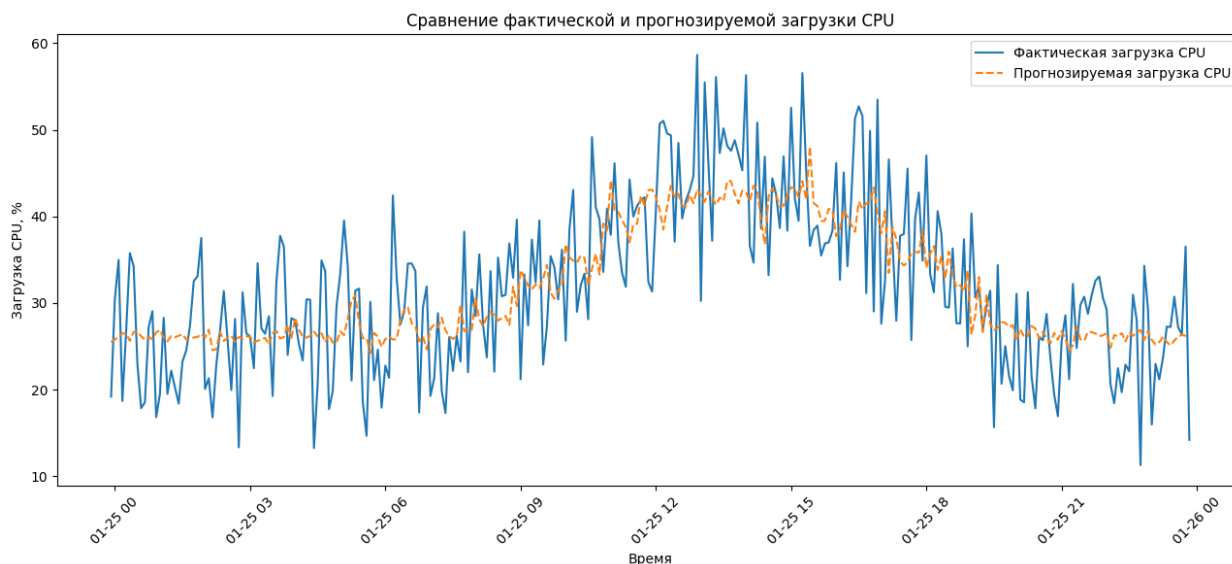


Рисунок 3 – Сравнение фактической и прогнозируемой загрузки центрального процессора

Полученный график показывает, что прогнозируемые значения повторяют основную динамику изменения нагрузки. Наибольшие расхождения наблюдаются при кратковременных резких изменениях нагрузки, которые сложнее определить на основании предыдущего состояния системы.

3.4 Результаты прогнозирования использования RAM

Для прогнозирования использования оперативной памяти получены следующие показатели:

$$\text{MAE} = 2,84 \%$$

$$\text{RMSE} = 3,83 \%$$

$$R^2 = 0,919$$

Модель продемонстрировала меньшую среднюю абсолютную ошибку при прогнозировании оперативной памяти по сравнению с загрузкой процессора. Это связано с более плавным характером изменения объема используемой оперативной памяти в сформированной модели нагрузки (Рис.4).

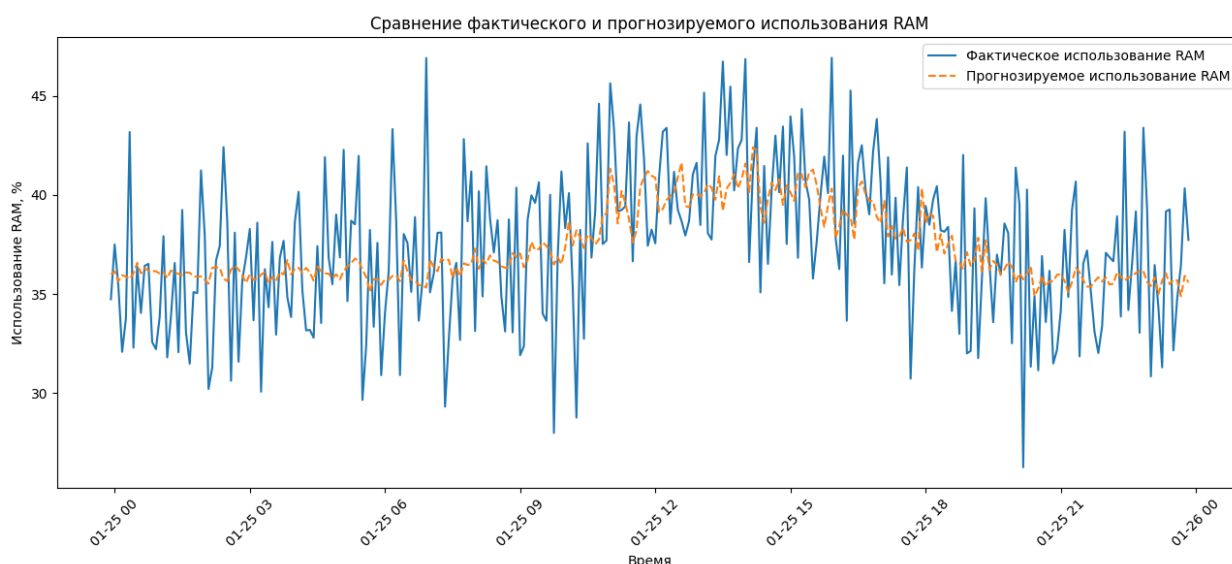


Рисунок 4 – Сравнение фактического и прогнозируемого использования оперативной памяти

Фактические и прогнозируемые значения располагаются близко друг к другу, что свидетельствует о способности выбранного алгоритма выявлять закономерности изменения используемой памяти.

3.5 Анализ значимости признаков

Одной из особенностей Random Forest является возможность оценки относительной значимости входных признаков (Табл.2).

В проведённом эксперименте наиболее значимыми признаками для прогнозирования загрузки CPU оказались текущая сетевая активность и количество активных пользователей.

Получены следующие значения значимости основных признаков.

Таблица 2 – Значимость параметров модели

Признак	Значимость
network_mbps	0,488
active_users	0,439
hour	0,015
cpu_usage_lag1	0,012
cpu_usage_lag3	0,009
cpu_usage_lag2	0,009

Высокая значимость параметров `network_mbps` и `active_users` обусловлена способом формирования экспериментального набора данных, в котором рост пользовательской и сетевой активности непосредственно связан с увеличением вычислительной нагрузки (Рис.5).

Наиболее важные признаки для прогноза CPU:

	feature	importance
4	network_mbps	0.488445
2	active_users	0.438920
0	hour	0.015417
7	cpu_usage_lag1	0.011874
9	cpu_usage_lag3	0.008953
8	cpu_usage_lag2	0.008872
6	temperature	0.003323
18	network_mbps_lag3	0.002900
5	disk_io	0.002585
1	day_of_week	0.002364

Рисунок 5 – Значимость признаков при прогнозировании загрузки центрального процессора

При использовании реальных серверных показателей распределение значимости признаков может существенно отличаться и будет зависеть от назначения конкретной системы.

3.6 Сравнительный анализ результатов

Для обобщения результатов эксперимента проведено сравнение качества прогнозирования двух основных вычислительных ресурсов (Табл.3).

Таблица 3 – Результаты прогнозирования загрузки вычислительных ресурсов

Прогнозируемый ресурс	MAE	RMSE	R ²
Центральный процессор CPU	5,50	7,39	0,929
Оперативная память RAM	2,84	3,83	0,919

Полученные значения показывают, что модель Random Forest способна с высокой степенью соответствия воспроизводить общую динамику изменения загрузки на тестовой части сформированного набора данных.

При этом прогнозирование оперативной памяти характеризуется меньшей абсолютной ошибкой. Для CPU наблюдается более высокая ошибка, что объясняется наличием кратковременных изменений вычислительной активности.

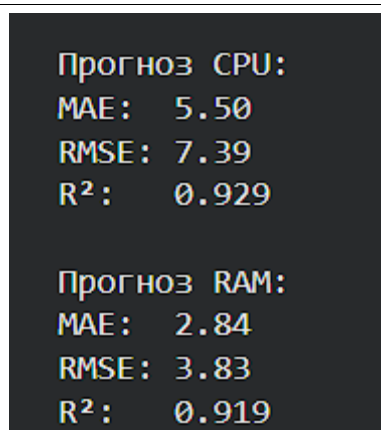


Рисунок 6 – Результаты оценки точности моделей машинного обучения

Полученные результаты демонстрируют принципиальную возможность применения методов машинного обучения для прогнозирования состояния вычислительной инфраструктуры. На основе такого прогноза в дальнейшем возможно создание системы, которая при ожидаемом превышении заданного уровня нагрузки заранее формирует уведомление администратору или выполняет автоматическое масштабирование ресурсов.

4 Выводы

В рамках исследования разработана модель машинного обучения для краткосрочного прогнозирования загрузки вычислительных ресурсов сервера. Для решения задачи применён алгоритм Random Forest Regression, использующий информацию о текущем состоянии сервера и показателях его работы за предыдущие временные интервалы.

Эксперимент показал, что для сформированного набора данных коэффициент детерминации составил 0,929 при прогнозировании CPU и 0,919 при прогнозировании RAM. Средняя абсолютная ошибка составила соответственно 5,50 и 2,84 процентного пункта.

Полученные результаты показывают возможность использования методов машинного обучения в задачах системного администрирования и мониторинга серверной инфраструктуры. В отличие от классического порогового мониторинга прогнозирование позволяет оценивать вероятное состояние системы заранее, что потенциально даёт возможность предотвращать перегрузки до их фактического возникновения.

Ограничением проведённого исследования является использование синтетического набора данных, воспроизводящего основные закономерности серверной нагрузки. В дальнейшем разработанный подход может быть дополнен сбором реальных показателей Linux-сервера с использованием систем мониторинга или библиотеки psutil. Обучение модели на реальной статистике позволит оценить эффективность алгоритма при наличии нерегулярной нагрузки, фоновых процессов и кратковременных пиков потребления ресурсов.

Библиографический список

1. Dinda P. A., O'Hallaron D. R. Host load prediction using linear models // Cluster Computing. 2000. Т. 3. № 4. С. 265280. DOI: 10.1023/A:1019048724544. URL: <https://link.springer.com/article/10.1023/A:1019048724544> (дата обращения: 20.09.2026).
2. Islam S., Keung J., Lee K., Liu A. Empirical prediction models for adaptive resource provisioning in the cloud // Future Generation Computer Systems. 2012. Т. 28. № 1. С. 155162. DOI: 10.1016/j.future.2011.05.027. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0167739X11001129> (дата обращения: 20.09.2026).
3. Calheiros R. N., Masoumi E., Ranjan R., Buyya R. Workload prediction using ARIMA model and its impact on cloud applications' QoS // IEEE Transactions on Cloud Computing. 2015. Т. 3. № 4. С. 449458. DOI: 10.1109/TCC.2014.2350475. URL: <https://researchers.westernsydney.edu.au/en/publications/workload-prediction-using-arima-model-and-its-impact-on-cloud-app/> (дата обращения: 20.09.2026).
4. Borkowski M., Schulte S., Hochreiner C. Predicting cloud resource utilization // Proceedings of the 9th IEEE/ACM International Conference on Utility and Cloud Computing. 2016. С. 3742. DOI: 10.1145/2996890.2996907. URL: <https://dl.acm.org/doi/abs/10.1145/2996890.2996907> (дата обращения: 20.09.2026).
5. Breiman L. Random forests // Machine Learning. 2001. Т. 45. С. 532. DOI: 10.1023/A:1010933404324. URL: <https://link.springer.com/article/10.1023/A:1010933404324> (дата обращения: 20.09.2026).