ПРОГНОЗИРОВАНИЕ ЗАГРУЗКИ ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА МЕТОДАМИ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ

Д. А. Михалевич
Финансовый университет при Правительстве Российской Федерации, Москва, Российская Федерация
ORCID: https://orcid.org/0009-0009-4758-4510, ✍ dmitriymikhalevich02@gmail.com

Аннотация: управление ресурсами вычислительного кластера требует прогноза будущей загрузки: недооценка спроса приводит к деградации сервисов, переоценка — к избыточному резервированию и лишним затратам. В работе на данных производственной трассы Alibaba GPU Trace 2020 решается задача краткосрочного прогнозирования почасовой загрузки процессоров, графических ускорителей и оперативной памяти кластера с горизонтом в один час и тестовым окном в 168 часов. События задач агрегированы в почасовые ряды усреднением по времени. Сопоставлены пять моделей: наивный прогноз по последнему наблюдению, модель проинтегрированного скользящего среднего, рекуррентная нейронная сеть с долгой краткосрочной памятью, байесовская динамическая линейная модель и многомерная модель векторной авторегрессии с отбором лагов глубиной до 168 часов. Критерии сравнения — точность прогноза, качество 99-процентного доверительного интервала, точность предсказания направления изменения и экономия резервируемых ресурсов. Наивысшую точность дает байесовская динамическая линейная модель: средняя абсолютная ошибка на 6–9 процентов ниже, чем у наивного прогноза; для оперативной памяти конкурентоспособна рекуррентная сеть, а векторная авторегрессия полезна для прогнозирования загрузки процессоров и ускорителей. Переход от резервирования по историческому максимуму к резервированию по верхней границе прогнозного интервала сокращает объем резерва на 22–30 процентов при покрытии 96–99 процентов.

Ключевые слова: прогнозирование временных рядов, загрузка вычислительного кластера, байесовские динамические линейные модели, векторная авторегрессия, планирование ресурсов.

Для цитирования: Михалевич Д. А. Прогнозирование загрузки вычислительного кластера методами анализа временных рядов. *Успехи кибернетики.* 2026;7(3):101–109.

Поступила в редакцию: 09.09.2026. *В окончательном варианте: 24.09.2026.*

COMPUTING CLUSTER LOAD PREDICTION FROM TIME-SERIES DATA

D. A. Mikhalevich
Financial University under the Government of the Russian Federation, Moscow, Russian Federation
ORCID: https://orcid.org/0009-0009-4758-4510, ✍ dmitriymikhalevich02@gmail.com

Abstract: we studied short-term computing cluster load prediction using the Alibaba GPU Trace 2020 production trace. We considered hourly processor, graphics accelerator, and random-access memory (RAM) loads, with a one-hour prediction horizon and a 168-hour test window. We aggregated task events into hourly time series by averaging the concurrent load over time. We compared five prediction models: a naive last-value model, an integrated moving-average model, a long short-term memory (LSTM) recurrent neural network, a Bayesian dynamic linear model, and a multivariate vector autoregression model with lag selection up to 168 hours. We evaluated prediction accuracy, 99% prediction-interval coverage, accuracy in predicting the direction of load changes, and the potential savings in reserved resources. The Bayesian dynamic linear model achieved the lowest errors, with mean absolute errors 6–9% lower than those of the naive model. The LSTM model was competitive for RAM, whereas the multivariate model performed well for processor and accelerator loads. Replacing reservation based on the historical maximum with reservation based on the upper bound of the prediction interval reduced the reserve by 22–30% while maintaining 96–99% coverage.

Keywords: time series forecasting, computing cluster load, Bayesian dynamic linear models, vector autoregression, capacity planning.

Cite this article: Mikhalevich D. A. Computing Cluster Load Prediction from Time-Series Data. *Russian Journal of Cybernetics.* 2026;7(3):101–109.

Original article submitted: 09.09.2026. *Revision submitted: 24.09.2026.*

Введение

Развитие облачных платформ и кластеров с графическими ускорителями делает управление ресурсами одной из ключевых задач эксплуатации инфраструктуры. Статическое резервирование по пиковому значению исключает нехватку ресурсов, но систематически завышает затраты: реальная загрузка подвержена выраженной суточной и недельной цикличности [1, 2]. Альтернатива — динамическое резервирование по краткосрочному прогнозу с интервальной оценкой.

Для прогнозирования нагрузки применялись классические модели временных рядов — линейные модели загрузки узла [3] и модели проинтегрированного скользящего среднего ARIMA [4], — и методы глубокого обучения, прежде всего рекуррентные сети с долгой краткосрочной памятью LSTM [5, 6]; превосходство сложных моделей при этом не универсально [7]. Зачастую модели сравниваются по точности прогноза, тогда как для планирования ресурсов важнее верхняя граница интервала: она определяет объем резерва.

Подход, принятый в работе, состоит в том, чтобы свести планирование ресурсов к задаче анализа временных рядов. Журнал событий кластера сам по себе временным рядом не является, и способ его свертки задает то, что затем прогнозируется. Здесь для каждого ресурса строится ряд средней одновременной заявленной загрузки за час, а резервирование сводится к интервальному прогнозу этого ряда на шаг вперед. Такое сведение определяет единицу планирования, горизонт и смысл верхней границы интервала как объема резерва.

Постановка задачи

Кластер обслуживает поток задач. Каждая задача при запуске заявляет требуемые объемы процессорного времени, занятость графических ускорителей и оперативной памяти. Суммарная заявленная загрузка, усредненная по часу, образует три временных ряда — по одному на ресурс.

Первый шаг решения — переход от журнала событий к временным рядам. Задачи запускаются и завершаются в произвольные моменты, поэтому загрузка задана не последовательностью наблюдений, а кусочно-постоянной функцией времени; выбор шага агрегации фиксирует объект прогноза и горизонт планирования. Час взят как естественная единица: за него планировщик успевает перераспределить резерв, и сохраняется суточная цикличность, ради которой прогноз и строится.

Требуется на каждом шаге по истории вплоть до предыдущего часа построить точечный прогноз на час вперед и доверительный интервал уровня 99%, верхняя граница которого назначается объемом резервируемого ресурса. Практическая цель — сократить резерв, не увеличивая риск нехватки: чем уже интервал, тем больше экономия, но тем чаще фактическая загрузка выходит за его границы.

Цель работы — разработать методологию прогнозирования спроса на ресурсы кластера: свести задачу планирования к анализу временных рядов, отобрать модели, сопоставить их на единой производственной трассе по общему набору критериев и выбрать наиболее подходящие для каждого из трех ресурсов.

Методология

Данные и агрегация

Использована трасса Alibaba GPU Trace 2020 — таблица *pai_task_table* кластера платформы PAI (около 1800 машин, более 6500 ускорителей): 1,24 млн записей с временем запуска, статусом и заявленными объемами спроса:

- *plan_cpu* — на процессорные мощности в процентах, одно ядро эквивалентно 100%;
- *plan_mem* — на оперативную память в гигабайтах;
- *plan_gpu* — на мощности графического ускорителя в процентах, 100% эквивалентно запросу на использование всех ядер и памяти графического ускорителя.

События приведены к почасовым рядам усреднением одновременной загрузки. Внутри часа строится равномерная сетка с шагом, равным минимальной длительности активной в этом часе задачи. В каждой точке берется суммарный заявленный объем ресурса работающих в этот

момент задач. Значение ряда за час — среднее по точкам:

$$y_t = \frac{1}{|S_t|} \sum_{s \in S_t} R(s), \tag{1}$$

где S_t — точки сетки часа t, $R(s)$ — суммарный заявленный объем ресурса момента s.

Полный ряд покрывает 07.07 — 13.09.2020; значения по краям отражают запуск и остановку сбора трассы, поэтому оставлен интервал 15.07 — 01.09.2020 — устойчивое окно в 49 суток, или 1176 часовых точек. Последние 168 точек образуют тестовую выборку, предшествующие 1008 — обучающую.

Разведочный анализ

Почасовые диаграммы размаха выявили выраженную суточную сезонность: спрос на все три ресурса нарастает к 6–8 часам, снижается и вновь подходит к пику около 18 часов; тепловые карты «час — день недели» дают более слабую недельную составляющую. По расширенному тесту Дики–Фуллера ряды нестационарны в уровнях (p-значение от 0,10 до 0,19) и стационарны в первых разностях. Автокорреляционные функции выделяют лаги 1, 2, 3, 24 и 168 часов, вторые разности обнаруживают признаки скользящего среднего второго порядка с сезонным шумом — отсюда спецификация ARIMA (0,2,2); лаг 24 в остальных моделях оказался незначимым, рабочий набор регрессоров составили лаги 1, 2 и 168.

Схема эксперимента и метрики

Прогноз строится на один час вперед по скользящему окну на протяжении 168 тестовых часов: после каждого шага фактическое значение добавляется к истории. ARIMA переобучается на каждом шаге; LSTM, BDLM и Subset VAR оцениваются однократно и далее только прогнозируют по наблюдаемым лагам. Ни одна оценка не использует сведений о моментах позже прогнозируемого.

Точность точечного прогноза измеряют MAE, RMSE и MAPE; качество интервального прогноза — доля тестовых точек, покрытых 99-процентным интервалом, и его средняя ширина, а показатель MDA (Mean Directional Accuracy) — доля шагов с верно предсказанным направлением изменения.

Прикладная метрика экономии — средний по тестовым шагам процент экономии верхней границы доверительного интервала относительно текущего исторического максимума:

$$Saving = \frac{100\%}{n} \sum_{t=1}^{n} \left(\frac{M_t - \hat{y}_t^{upper}}{M_t} \right), \tag{2}$$

где M_t — исторический максимум ряда на шаге t (по обучающей выборке и уже пройденной части теста), $\hat{y}_t^{upper}$ — верхняя граница интервала, n — число тестовых шагов. Покрытие и экономия (2) — конкурирующие требования: сужение интервала повышает экономию, но увеличивает риск нехватки ресурсов, поэтому модели сопоставляются по обоим показателям.

Моделирование

Базовая модель (T-1) — прогноз по последнему наблюдению:

$$y_t = y_{t-1} + v_t,\ v \sim N\left(v; 0, \sigma_v^2\right). \tag{3}$$

В формуле (3) v_t — ошибка прогноза с нулевым средним и дисперсией σ_v^2. Границы интервала задает стандартное отклонение σ_v, оцениваемое по накопленным одношаговым ошибкам с экспоненциальным затуханием веса (коэффициент 0,95). Так же строится интервал для LSTM.

Модель ARIMA (0,2,2) — проинтегрированное скользящее среднее второго порядка, записанное через вторую разность:

$$y_t = 2y_{t-1} - y_{t-2} + \varepsilon_t - \beta_1 \varepsilon_{t-1} - \beta_2 \varepsilon_{t-2}. \tag{4}$$

В формуле (4) ε_t — белый шум, β_1 и β_2 — коэффициенты скользящего среднего. Модель оценивается заново на каждом шаге, интервал берется из самой модели.

Сеть LSTM [5] — двухслойная, 64 скрытых состояния, прореживание 0,2, окно из 168 стандартизованных значений ряда, однократное обучение методом Adam (10^{-3}, 100 эпох).

Байесовская динамическая линейная модель BDLM [8] описывает ряд регрессией на лаги, коэффициенты которой меняются во времени:

$$y_t = x_t^{LL} + x_t^{AR1} y_{t-1} + x_t^{AR2} y_{t-2} + x_t^{AR168} y_{t-168} + v_t,\ v \ \sim \ N(v; 0, \sigma_v^2). \tag{5}$$

В формуле (5) x_t^{LL} — коэффициент при локальном уровне, остальные — при лагах $t-1$, $t-2$, $t-168$; скорость их изменения задается дисконтированием 0,99. Прогноз и границы интервала — среднее и квантили 0,5 и 99,5% по 10 000 выборкам апостериорного распределения.

Модель Subset VAR (1…168) многомерная: она описывает три ряда совместно и потому использует зависимости между ресурсами. Ряды берутся в первых разностях и стандартизуются; для каждого уравнения строится регрессия наименьших квадратов по лагам с 1-го по 168-й всех трех переменных с пошаговым исключением незначимых при уровне 0,01. Интервал строится по дисперсии оценки коэффициентов и остаточной дисперсии.

Результаты

Итоговые метрики пяти моделей приведены в таблицах 1–5, строки — три ресурсных ряда, значения в единицах трассы. Для каждого ресурса разбирается модель, выбранная по нему лучшей: оперативная память — LSTM, графические ускорители — BDLM, процессоры — Subset VAR; для каждой из них приведены прогноз на тестовой неделе и диагностика остатков.

Базовая модель

Таблица 1

Метрики тестирования базовой модели

Ряд	MAE	RMSE	MAPE, %	Покрытие, %	Ширина ДИ	MDA, %	Экономия, %
plan_cpu	58 907	74 405	6,7	98,8	385 483	—	23,3
plan_gpu	4 943	6 732	6,7	97,0	34 033	—	27,6
plan_mem	2 438	3 039	7,7	99,4	15 371	—	28,6

Наивный прогноз уже дает содержательный результат: средняя абсолютная процентная ошибка по трем ресурсам в пределах 6,7–7,7%, покрытие — 97,0–99,4%, экономия — 23,3–28,6%. Распределение остатков близко к нормальному, однако их автокорреляционные функции белому шуму не соответствуют: структура ряда исчерпана не полностью. Остальные модели сравниваются, прежде всего, с этой.

ARIMA (0,2,2)

Таблица 2

Метрики тестирования модели ARIMA (0,2,2)

Ряд	MAE	RMSE	MAPE, %	Покрытие, %	Ширина ДИ	MDA, %	Экономия, %
plan_cpu	59 895	75 176	6,8	99,4	494 736	52,1	19,4
plan_gpu	4 730	6 529	6,4	96,4	31 992	59,9	28,3
plan_mem	2 487	3 068	7,8	99,4	17 079	53,3	27,1

Коэффициенты модели значимы, но их сумма по модулю с учетом стандартных ошибок превышает единицу — процесс с точки зрения модели неустойчив, что делает ее применение рискованным. Точечный прогноз лучше базового только по *plan_gpu*; покрытие интервала выше базового для *plan_cpu*, направление изменения предсказывается верно в 52,1–59,9% случаев. Главный недостаток — самый широкий интервал: 494 736 по *plan_cpu*, на 20–36% шире, чем у остальных моделей, отчего экономия наименьшая: 19,4%.

Таблица 3

Метрики тестирования модели LSTM

Ряд	MAE	RMSE	MAPE, %	Покрытие, %	Ширина ДИ	MDA, %	Экономия, %
plan_cpu	57 195	74 598	6,4	99,4	364 535	63,5	25,3
plan_gpu	4 934	6 947	6,5	95,2	33 113	57,5	29,2
plan_mem	2 331	2 856	7,4	99,4	14 230	61,1	30,2

LSTM

Точечный прогноз лучше базового по всем трем ресурсам; интервал уже базового, покрытие не ниже базового, кроме *plan_gpu*, экономия выше базовой для всех ресурсов. Лучший результат модель показывает на оперативной памяти: самый узкий интервал во всем сравнении, 14 230, при покрытии 99,4% и наибольшей экономии 30,2%. Прогноз уверенно повторяет и суточные колебания, и снижающийся тренд конца августа (рис. 1). В остатках, однако, сохраняется значимая автокорреляция на лаге 24 (рис. 2): суточная сезонность моделью, вероятно, исчерпана не полностью, и это оставляет запас для улучшения.

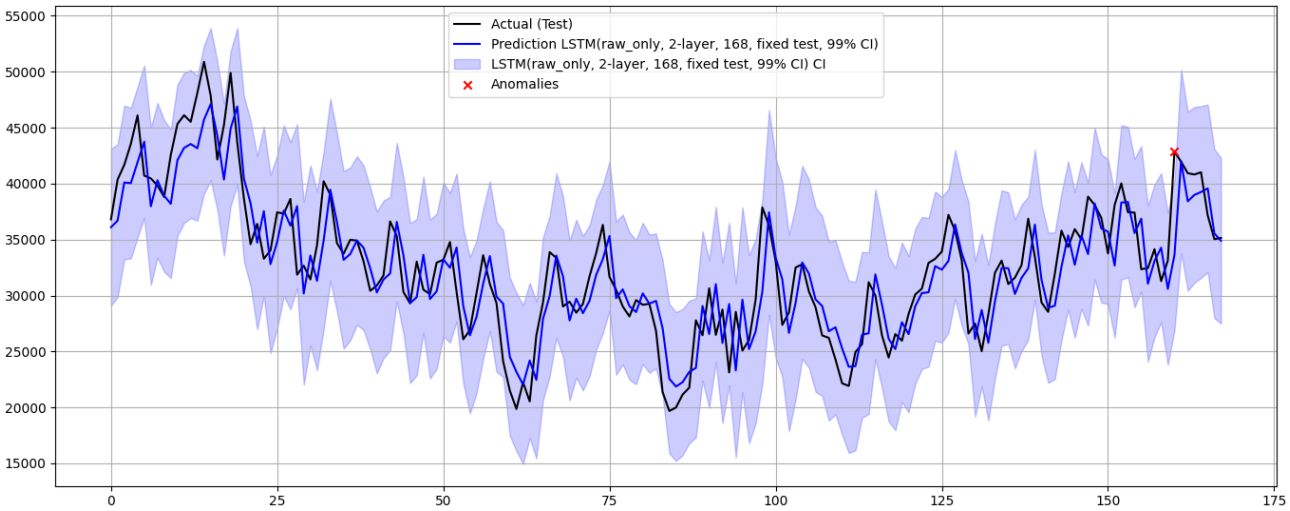


Рис. 1. *Прогноз почасовой загрузки оперативной памяти моделью LSTM на тестовой неделе: факт, точечный прогноз и 99-процентный доверительный интервал*

BDLM

Таблица 4

Метрики тестирования модели BDLM

Ряд	MAE	RMSE	MAPE, %	Покрытие, %	Ширина ДИ	MDA, %	Экономия, %
plan_cpu	55 024	69 336	6,3	99,4	413 374	64,1	22,2
plan_gpu	4 492	6 247	6,1	96,4	30 318	62,3	28,9
plan_mem	2 240	2 806	7,1	98,8	15 354	61,1	28,5

Модель точнее базовой по всем трем ресурсам: 55 024 против 58 907 по *plan_cpu* (на 6,6% ниже) и 4 492 против 4 943 по *plan_gpu* (на 9,1% ниже). Наибольшее преимущество — на графических ускорителях: там у нее лучшие в сравнении MAE, RMSE, MAPE и MDA при интервале уже базового. Прогноз воспроизводит суточные колебания и снижающийся тренд конца августа, фактические значения почти всюду остаются внутри интервала (рис. 3), а остатки по виду автокорреляционных функций близки к белому шуму (рис. 4): регулярная структура ряда исчерпана, что не удалось остальным моделям.

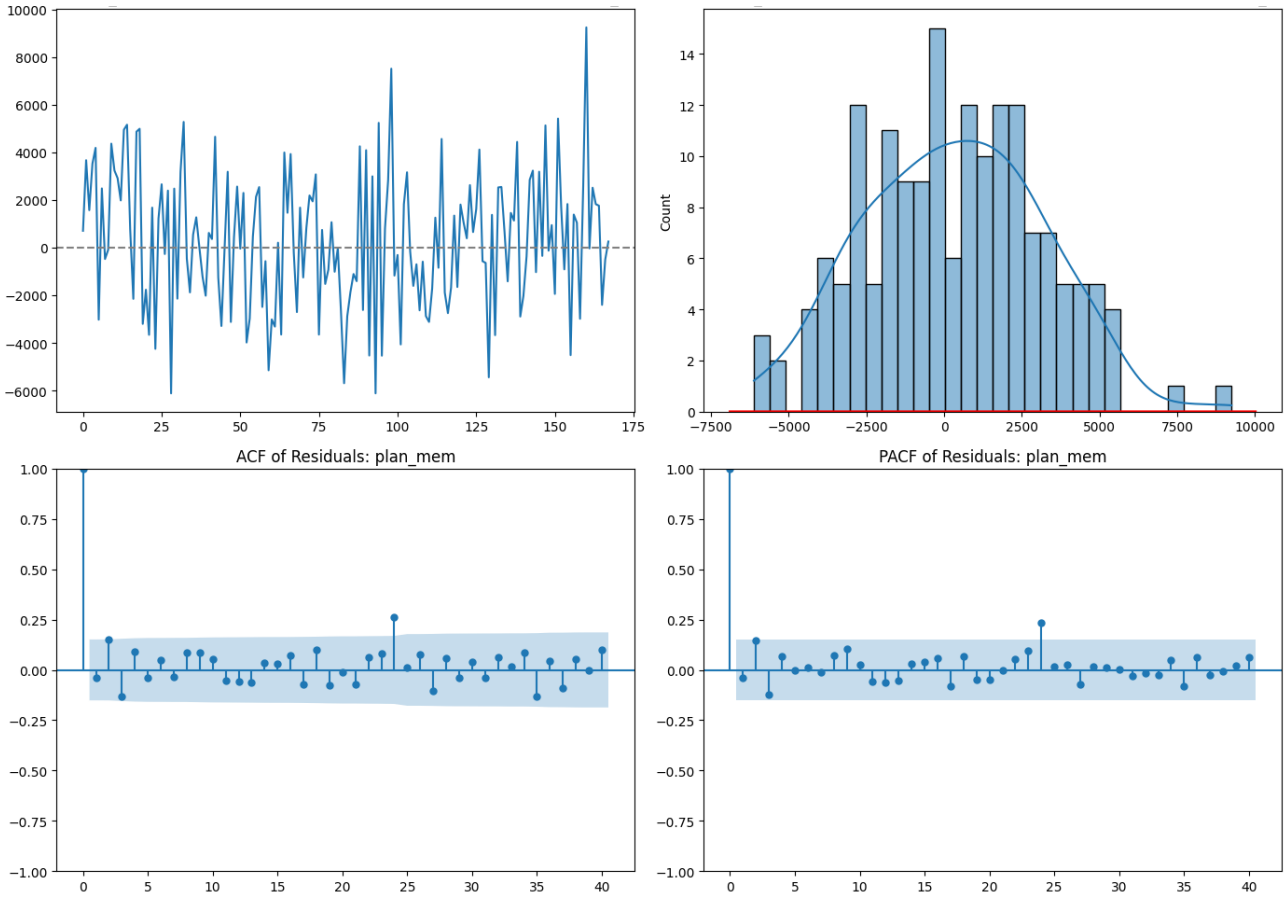


Рис. 2. *Диагностика остатков модели LSTM для прогноза почасовой загрузки оперативной памяти*

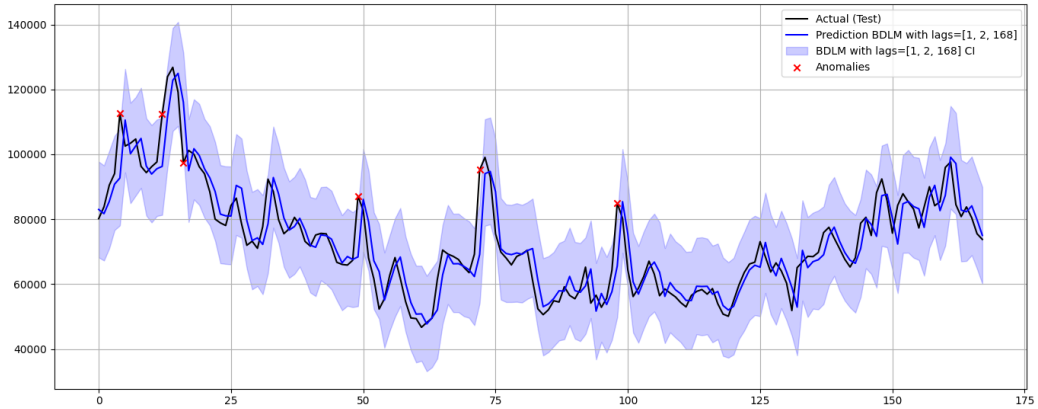


Рис. 3. *Прогноз почасовой загрузки графических ускорителей моделью BDLM на тестовой неделе*

Subset VAR

Таблица 5

Метрики тестирования модели Subset VAR (1. . . 168)

Ряд	MAE	RMSE	MAPE, %	Покрытие, %	Ширина ДИ	MDA, %	Экономия, %
plan_cpu	56 402	72 774	6,4	99,4	395 313	62,9	23,0
plan_gpu	4 885	6 704	6,6	95,8	26 590	61,1	29,6
plan_mem	2 237	2 834	7,0	97,6	14 373	65,3	29,5

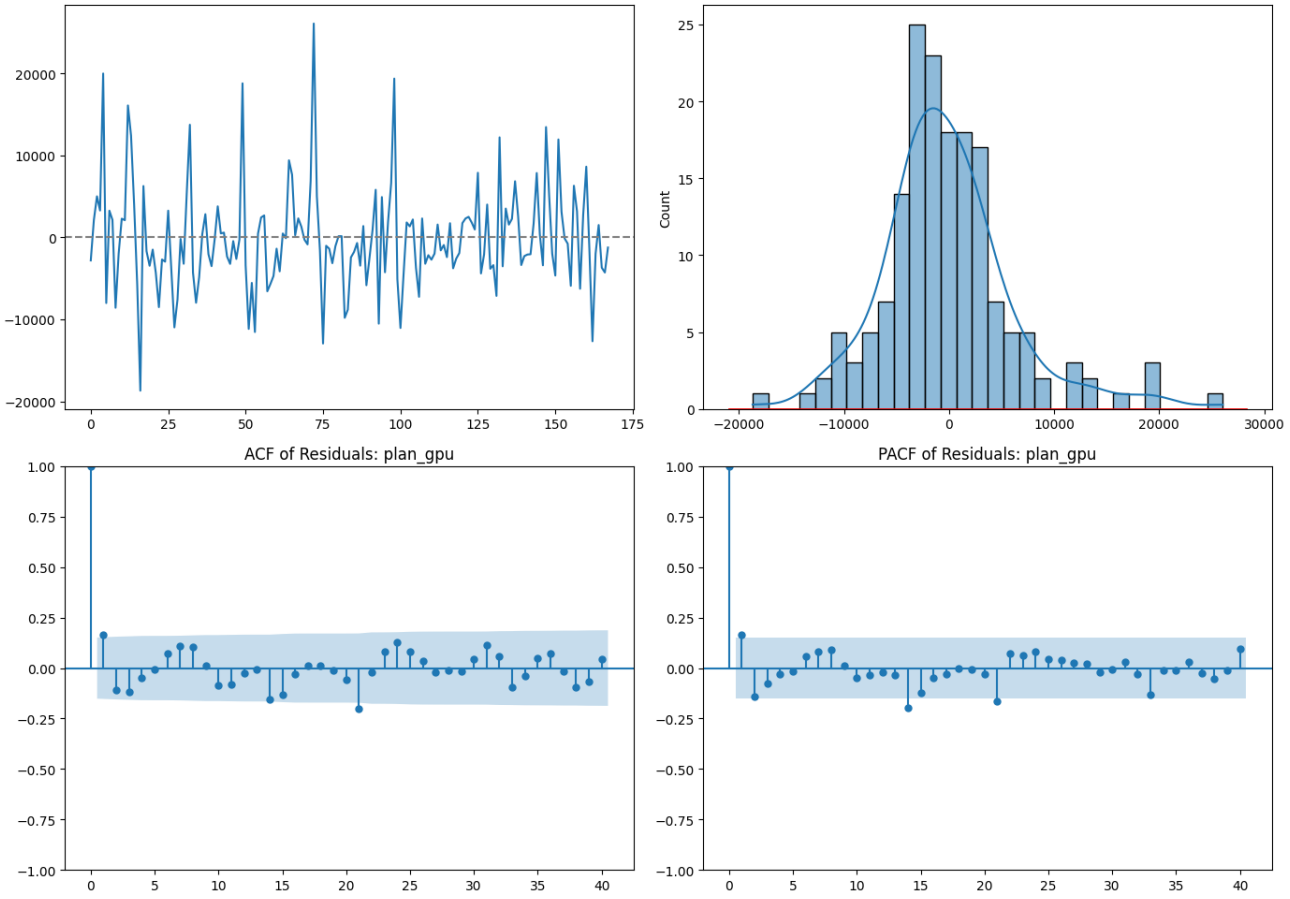


Рис. 4. *Диагностика остатков модели BDLM для прогноза почасовой загрузки графических ускорителей*

Точечный прогноз лучше базового по всем трем ресурсам; покрытие выше базового для *plan_ cpu*, интервал уже для *plan_ gpu* и *plan_ mem*, экономия выше для *plan_ gpu* и *plan_ mem*. Направление изменения предсказывается верно в 61,1–65,3% случаев, причем для *plan_ mem* это лучший результат в сравнении. Наиболее полезна модель для *plan_ cpu*: прогноз повторяет и суточные колебания, и снижающийся тренд конца августа (рис. 5), а остатки по виду автокорреляционных функций близки к белому шуму (рис. 6). По *plan_ mem* на лаге 3 значимая автокорреляция сохраняется.

Обсуждение

Наилучшие результаты по точности точечного прогноза показывает BDLM; она же лидирует по MDA для процессоров и ускорителей. LSTM дает достойный результат для *plan_ mem*, особенно по ширине и покрытию интервала, а Subset VAR — для *plan_ cpu* и *plan_ gpu*. Доверительные интервалы этих трех моделей имеют практическое преимущество: благодаря высокой точности точечного прогноза их проще откалибровать. ARIMA (0,2,2) уступает по ключевым метрикам, базовая модель служит ориентиром. Итоговая рекомендация: для процессоров и ускорителей — Subset VAR и BDLM, для памяти — LSTM.

Преимущество BDLM и Subset VAR объясняется тем, что обе явно используют выявленную лаговую структуру и описывают неопределенность внутри модели — через апостериорное распределение и через дисперсию оценки коэффициентов. Отсюда более калиброванные и при этом узкие интервалы, чем у ARIMA, где неопределенность накапливается из-за двукратного дифференцирования, и чем у моделей с интервалом, построенным отдельно по остаткам. Subset VAR дополнительно учитывает зависимости между ресурсами: некоторые заявки порождаются одними и теми же задачами и коэффициенты зависимости между ресурсами в многомерной модели доходят до 0,27.

У результатов есть ограничения. Прогнозируется заявленный, а не фактически потреб-

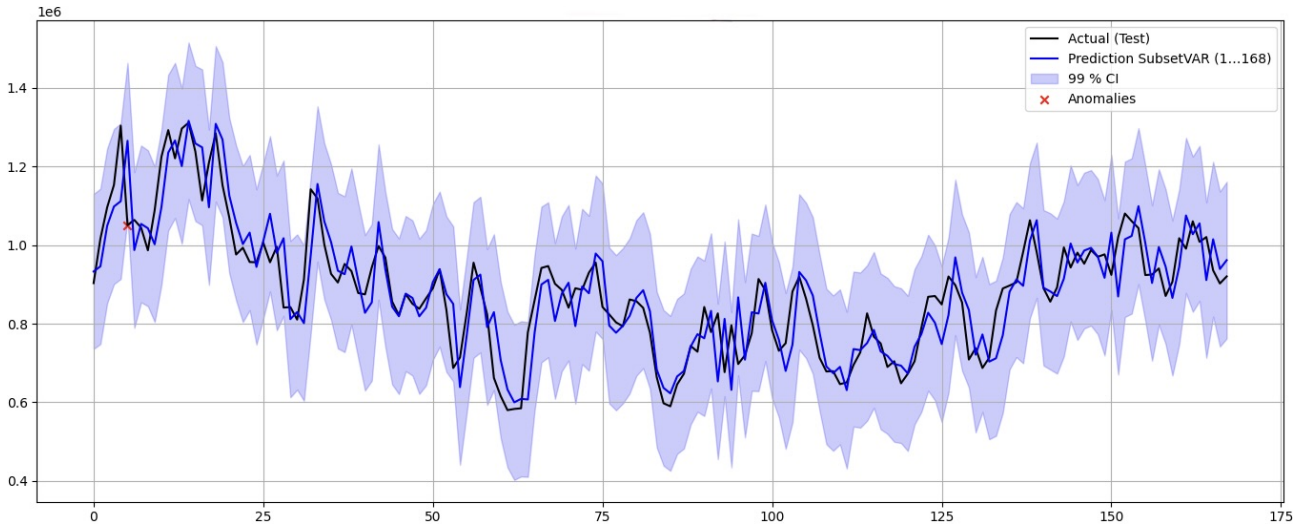


Рис. 5. *Прогноз почасовой загрузки процессоров моделью Subset VAR (1…168) на тестовой неделе*

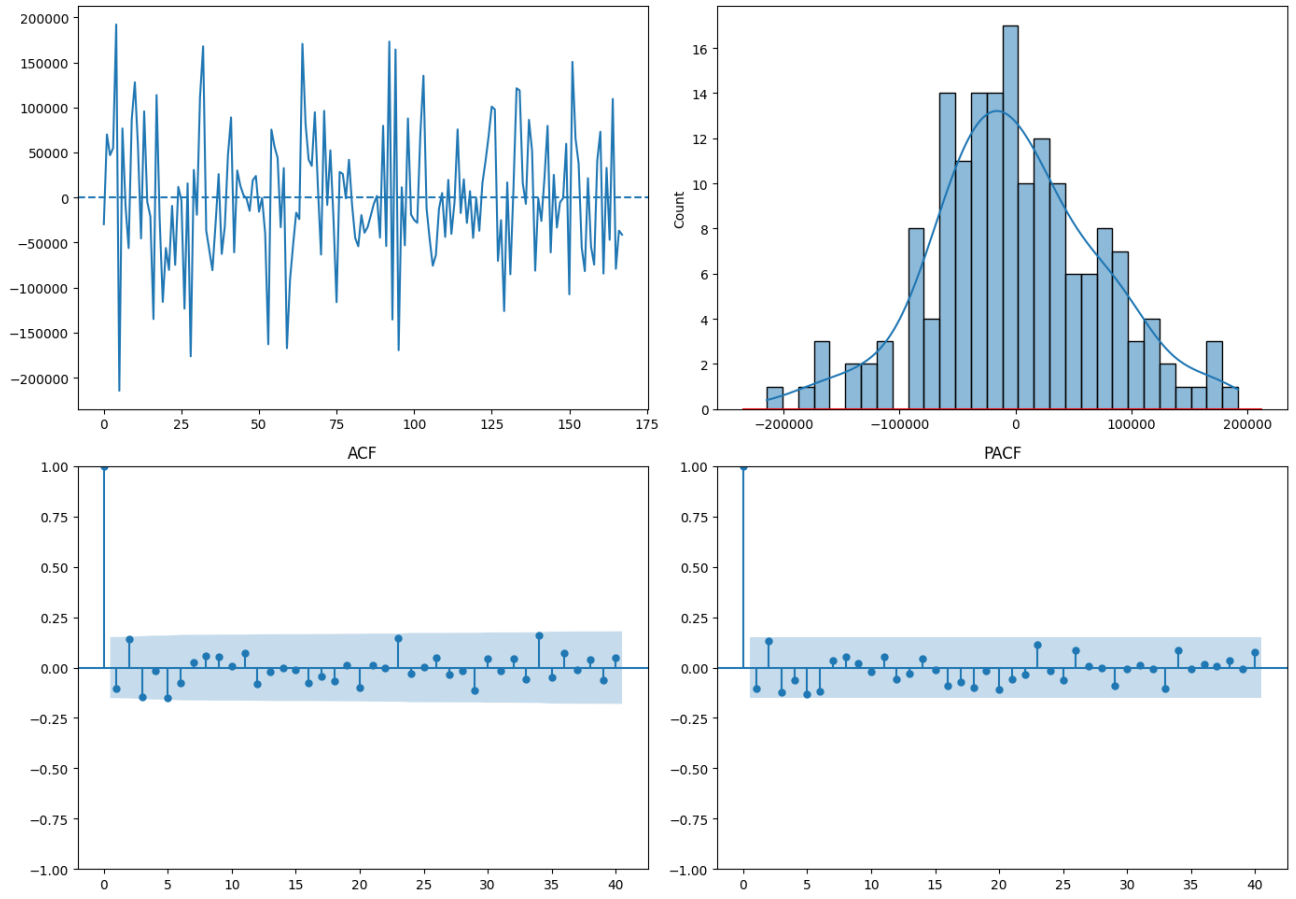


Рис. 6. *Диагностика остатков модели Subset VAR (1…168) для прогноза почасовой загрузки процессоров*

ленный объем ресурсов: для резервирования это корректно, но ряды не следует трактовать как утилизацию. Выводы получены на одной трассе за 49 суток и на горизонте в один час.

Заключение

На данных производственной трассы Alibaba GPU Trace 2020 сопоставлены пять моделей краткосрочного прогнозирования почасовой загрузки кластера. Наилучшие результаты по большинству метрик показали байесовская динамическая линейная модель, Subset VAR и LSTM: у

байесовской модели средняя абсолютная ошибка на 6–9% ниже, чем у наивного прогноза, направление изменения выбранные модели предсказывают верно в 61–63% случаев, остатки BDLM и Subset VAR близки к белому шуму; ARIMA (0,2,2) оказалась менее точной и устойчивой. Замена статического резервирования по историческому максимуму резервированием по верхней границе прогнозного интервала сокращает объем резерва на 22–30% при покрытии 96–99%. Дальнейшее развитие работы связано с переходом к различным временным масштабам, единой многомерной байесовской моделью для пошагового прогноза, прогнозированием на несколько шагов и анализом непосредственной утилизации ресурсов.

ЛИТЕРАТУРА

1. Weng Q., Xiao W., Yu Y. et al. MLaaS in the Wild: Workload Analysis and Scheduling in Large-Scale Heterogeneous GPU Clusters. *Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI'22)*; 2022. Режим доступа: https://www.usenix.org/system/files/nsdi22-paper-weng.pdf.
2. Amiri M., Mohammad-Khanli L. Survey on Prediction Models of Applications for Resources Provisioning in Cloud. *Journal of Network and Computer Applications*. 2017;82:93–113. DOI: 10.1016/j.jnca.2017.01.016.
3. Dinda P. A., O'Hallaron D. R. Host Load Prediction Using Linear Models. *Cluster Computing*. 2000;3(4):265–280. DOI: 10.1023/A:1019075903624.
4. 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;3(4):449–458. DOI: 10.1109/TCC.2014.2350475.
5. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997;9(8):1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
6. Li Y. Y., Do T. V., Nguyen H. T. A Comparison of Forecasting Models for the Resource Usage of MapReduce Applications. *Neurocomputing*. 2020;418:36–55. DOI: 10.1016/j.neucom.2020.07.059.
7. Makridakis S., Spiliotis E., Assimakopoulos V. Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward. *PLoS ONE*. 2018;13(3):e0194889. DOI: 10.1371/journal.pone.0194889.
8. Prado R., West M. *Time Series: Modeling, Computation, and Inference*. New York; 2010. 368 p.