Федеральное государственное автономное образовательное учреждение
высшего образования «Московский физико-технический институт
(национальный исследовательский университет)»

Физтех-школа аэрокосмических технологий

Кафедра Аэрофизики и летательных аппаратов

Направление подготовки: 09.03.01 Информатика и вычислительная техника (бакалавриат)

Направленность (профиль) подготовки: Компьютерное моделирование

Форма обучения: очная

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«Алгоритм предиктивного анализа отказов системы видеоаналитики в режиме времени по данным от систем мониторинга»

(бакалаврская работа)

Студент:

Боровец Николай Васильевич

_____

(подпись студента)

Научный руководитель:

Гришин Никита Александрович,

программист ПИШ РПИ

_____

(подпись научного руководителя)

Жуковский

2025

АННОТАЦИЯ

Выпускная квалификационная работа посвящена разработке метода предиктивного анализа задержек в конвейере видеоаналитики для мониторинга объектов инфраструктуры. **Цель работы** — создать алгоритм прогнозирования метрики $common\_event\_delay$ с автоматическим обнаружением аномалий для предупреждения операторов о потенциальных сбоях. В работе применяются **методы исследования**, включающие анализ временных рядов Prometheus-метрик, сравнение архитектур ML-моделей (таких как LSTM и градиентный бустинг), временную кросс-валидацию, развертывание в Docker и A/B-тестирование. В результате исследования разработан MLOps-конвейер с точностью прогнозирования, превышающей базовые методы, и временем отклика менее 1 секунды. Создана система оповещений с адаптивными порогами. Проведена валидация разработанного решения на исторических данных объемом 90643 точки, собранных за 16 дней. **Практическая значимость** работы заключается в создании готового к использованию решения для предиктивного мониторинга видеосистем с возможностью адаптации для применения в телекоммуникациях и промышленной автоматизации.

Ключевые слова: предиктивный анализ, задержки, видеоаналитика, Prometheus, временные ряды, аномалии.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Обоснование выбора темы и актуальность

Современные системы видеоаналитики играют критически важную роль в обеспечении безопасности и мониторинга объектов критической инфраструктуры, включая аэродромы, железнодорожные станции, морские порты, промышленные предприятия и нефтеперерабатывающие комплексы [1]. Эти системы обрабатывают огромные объемы видеоданных в режиме реального времени, что предъявляет высокие требования к производительности и надежности всего технологического конвейера.

С ростом масштабов развертывания и усложнением архитектуры видеоаналитических систем возрастает и сложность их мониторинга. Современные решения часто включают в себя многоуровневые конвейеры обработки, начиная от захвата видеопотоков с камер, их предварительной обработки, применения алгоритмов машинного обучения для детекции объектов и событий, передачи результатов через брокеры сообщений в бэкенд-системы и далее к конечным пользователям через веб-интерфейсы.

Повышение объемов данных и жестких требований к end-to-end-задержкам (от момента возникновения события на видео до его отображения оператору) делает необходимым переход от реактивного к предиктивному подходу в управлении производительностью. Традиционные методы мониторинга, основанные на статических пороговых значениях и алертах по факту превышения SLA, не способны предотвратить деградацию качества обслуживания до ее критических проявлений.

В данном контексте особую важность приобретает разработка интеллектуальных систем предиктивного анализа, способных на основе пото-

ковых метрик мониторинга (например, собираемых системой Prometheus [2] и визуализируемых в Grafana [3]) заблаговременно оценивать потенциальные проблемы производительности и инициировать превентивные меры по их устранению.

Цель и задачи исследования

Цель работы: разработать и внедрить комплексный метод предиктивного анализа задержек в конвейере видеоаналитики, способный прогнозировать конечную метрику $common\_event\_delay$ с заданной точностью и автоматически детектировать аномальные паттерны в работе системы для предупреждения операторов о потенциальных сбоях до их фактического проявления.

Достижение поставленной цели требует решения комплекса взаимосвязанных *задач:*

1. Проведение аналитического обзора и систематизация современной литературы по предиктивному анализу временных рядов, машинному и глубокому обучению в контексте мониторинга и диагностики производительности систем реального времени.

2. Проведение анализа структуры и взаимных корреляций временных рядов метрик, собираемых на этапах видеоконвейера, включая выявление скрытых зависимостей между компонентами системы и идентификацию наиболее информативных признаков для прогнозирования.

3. Систематический обзор и сравнительный анализ современных методов прогнозирования временных рядов и обнаружения аномалий, включая классические статистические подходы, методы машинного обучения и глубокие нейронные сети, с оценкой их применимости к специфике видеоаналитических конвейеров.

4. Обоснованный выбор оптимальной архитектуры модели (трансформер, градиентный бустинг или их гибридная комбинация) с учетом требований к точности и скорости inference, а также определение необходимого объёма обучающих данных и оптимальной периодичности переобучения модели.

5. Проектирование и реализация полноценного MLOps-конвейера, включающего автоматизированный feature-engineering, механизмы периодического дообучения модели на новых данных, высокопроизводительный inference-сервис и системы мониторинга качества оценок.

6. Всестороннее экспериментальное исследование точности и производительности разработанной модели на обширных исторических данных с использованием методов временной кросс-валидации и оценкой устойчивости к различным типам аномалий в данных.

7. Разработка и внедрение интеллектуальной системы оповещений с адаптивными порогами, а также формулирование практических рекомендаций по эксплуатации, настройке и масштабированию решения в производственной среде.

Методология и методы исследования

Для достижения поставленной цели и решения сформулированных задач применяется комплексная методология, сочетающая теоретические исследования с практическими экспериментами:

1. Организация непрерывного сбора и интеллектуальной предобработки потоковых метрик из системы мониторинга Prometheus [2], включая очистку от выбросов, нормализацию, обработку пропущенных значений и синхронизацию временных рядов различных компонентов системы.

2. Разработка специализированного модуля построения многомерных временных рядов с интеллектуальной генерацией признаков, включая временные лаги различной глубины, скользящие статистические агрегаты, спектральные характеристики и высокоразмерные эмбеддинги для захвата сложных временных зависимостей.

3. Реализация и экспериментальное сравнение различных архитектур моделей (трансформеры с механизмом внимания, ансамбли градиентного бустинга LightGBM/CatBoost, гибридные нейро-символьные подходы) с применением строгих методов перекрёстной валидации по времени для обеспечения корректной оценки обобщающей способности.

4. Контейнеризация решения с использованием технологии Docker и проведение детальных измерений latency inference в условиях, максимально приближенных к производственным, включая тестирование под нагрузкой и оценку масштабируемости.

5. Организация и проведение A/B-тестирования в реальной производственной среде с использованием методов статистической оценки значимости результатов и анализа влияния на ключевые показатели эффективности системы.

Теоретическая и практическая значимость

Теоретическая значимость работы заключается в сравнительном анализе методов прогнозирования временных рядов для систем мониторинга и определении их применимости к задачам предиктивной диагностики в условиях жестких временных ограничений.

Практическая значимость определяется разработкой готового к промышленному использованию решения для мониторинга и предупрежде-

ния отказов видеоконвейера с гарантированным соблюдением SLA по конечной метрике $common\_event\_delay$. Созданная система может быть адаптирована и масштабирована для применения в различных отраслях, где критична надежность систем обработки потоковых данных в реальном времени, включая телекоммуникации, финансовые технологии и промышленную автоматизацию.

1 Общие положения

Данная глава посвящена формальной постановке задачи предиктивного анализа задержек в конвейере видеоаналитики и представлению архитектуры исследуемой системы. В рамках главы вводятся ключевые математические обозначения, определяются целевые метрики и ограничения, формулируются требования к разрабатываемому алгоритму. Особое внимание уделяется описанию структуры видеоконвейера и точек сбора телеметрических данных, которые лягут в основу построения прогностической модели.

1.1 Архитектура системы видеоаналитики

Исследуемая система видеоаналитики представляет собой многокомпонентный конвейер, предназначенный для обработки видеопотоков в режиме реального времени с применением алгоритмов компьютерного зрения для детекции событий и объектов. Архитектура системы строится по принципу микросервисной архитектуры, что обеспечивает масштабируемость и отказоустойчивость, но одновременно усложняет задачи мониторинга и диагностики производительности.

Видеоконвейер включает следующие основные компоненты: модуль захвата видеопотока с IP-камер (получающий данные по протоколу RTSP), ML-pipeline для применения алгоритмов компьютерного зрения, брокер сообщений Apache Kafka [4] для асинхронной передачи результатов обработки, бэкенд-сервисы для бизнес-логики и сохранения данных, а также WebSocket-клиенты для доставки уведомлений конечным пользователям. Каждый компонент генерирует множество метрик производительности, которые собираются централизованной системой мониторинга Prometheus [2].

Критической характеристикой системы является end-to-end-задержка, измеряемая как время от момента возникновения события в видеопотоке до его отображения на интерфейсе оператора. Данная метрика, обозначаемая как $common\_event\_delay$, напрямую влияет на эффективность работы операторов и качество принимаемых ими решений в критических ситуациях.

1.2 Постановка задачи

Для формальной постановки задачи прогнозирования введем необходимые математические обозначения и определения. Пусть $T = \{t_1, t_2, \ldots, t_n\}$ — упорядоченное множество временных меток наблюдений, соответствующих моментам сбора метрик из системы мониторинга с фиксированным интервалом дискретизации. Обозначим через d общее число различных метрик, одновременно собираемых системой мониторинга со всех компонентов видеоконвейера.

Для каждой временной метки t_i формируется d-мерный вектор наблюдений:

$$\mathbf{x}_i = [m_i^{(1)}, m_i^{(2)}, \ldots, m_i^{(d)}] \in \mathbb{R}^d, \tag{1.1}$$

где каждая компонента $m_i^{(j)}$ представляет значение j-й метрики в момент времени t_i.

Компоненты вектора наблюдений соответствуют различным категориям метрик, характеризующих работу отдельных подсистем видеоконвейера:

- **Метрики ML-конвейера:** $vidcap\_delay$ (задержка видеозахвата), $vidcap\_fps$ (частота кадров видеозахвата), $vidcap\_fps\_avg$ (средняя частота кадров видеозахвата), характеризующие производительность модулей ком-

пьютерного зрения;

- **Метрики бэкенда:** $ml\_to\_backend\_kafka\_delay$ (задержка передачи результатов ML через Kafka), db_insert_delay (время записи в базу данных), отражающие эффективность серверной части системы;

- **Метрики WebSocket-клиента:** $common\_event\_delay$ (целевая end-to-end-задержка), $heartbeat\_*$ (метрики жизнеспособности соединений), $event\_counter$ (счетчики событий), seq_events_health (показатели корректности последовательности событий), характеризующие качество доставки результатов до конечных пользователей.

Для учета временных зависимостей в данных введем понятие скользящего окна наблюдений. Определим окно длины L и шаг сдвига s, где L представляет глубину истории, необходимую для прогнозирования, а s — частоту обновления прогнозов. Каждое k-е скользящее окно определяется как матрица:

$$X_k = [\mathbf{x}_{t_k-L+1}, \ldots, \mathbf{x}_{t_k}] \in \mathbb{R}^{L\times d}, \tag{1.2}$$

содержащая L последовательных векторов наблюдений, предшествующих моменту прогнозирования. На основе данной матрицы формируется расширенное множество признаков для обучения модели, включающее различные статистические агрегаты, временные лаги и производные характеристики, детальное описание которых приводится в главе 2.

Целевая переменная для задачи прогнозирования определяется как значение критической метрики end-to-end-задержки в будущий момент времени:

$$y_k = common\_event\_delay(t_k + \Delta), \tag{1.3}$$

где $\Delta = 900 \times 15$ с $= 13500$ с $\approx 3{,}75$ ч представляет горизонт прогнозирования, выбранный исходя из требований к заблаговременности предупреждений о потенциальных проблемах в системе. Данный горизонт соответствует прогнозу на 900 временных шагов вперед при интервале дискретизации 15 секунд.

Обучающая выборка для построения прогностической модели формируется как множество пар «окно-целевое значение»:

$$\mathcal{N} = \{(X_k, y_k)\}_{k=1}^{N}, \tag{1.4}$$

где N — общее количество доступных обучающих примеров, определяемое длиной исторических данных и параметрами скользящего окна.

В рамках данной постановки предполагается существование неизвестной целевой функции:

$$f^* : \mathbb{R}^{L \times d} \to \mathbb{R}, \tag{1.5}$$

которая отображает текущее состояние системы (представленное матрицей метрик скользящего окна) в прогнозируемое значение end-to-end-задержки.

Основная задача исследования состоит в построении алгоритма $A : \mathbb{R}^{L \times d} \to \mathbb{R}$, аппроксимирующего неизвестную функцию f^* с заданной точностью:

$$|A(X_k) - f^*(X_k)| \leq \varepsilon \quad \forall k, \tag{1.6}$$

где ε — допустимая погрешность прогнозирования, определяемая практическими требованиями к системе предупреждения.

К разрабатываемому алгоритму A предъявляется ряд требований:

1. **Точность прогнозирования:** обеспечение качества прогноза целевой

метрики $common\_event\_delay$ с MAPE $< 10\%$ и других метрик (MAE, RMSE) на валидационной выборке;

2. **Производительность:** время формирования прогноза < 5 с при развертывании в контейнеризованной среде [5] для практического применения в системе мониторинга;

3. **Интерпретируемость:** возможность анализа важности признаков и понимания логики принятия решений моделью;

4. **Практичность:** простота интеграции в существующую инфраструктуру мониторинга и возможность автоматизации процесса обновления модели.

Исходные данные для обучения и валидации алгоритма представляют собой многомерный временной ряд $X \in \mathbb{R}^{n \times d}$ с элементами типа FLOAT64, формируемый из системы мониторинга Prometheus с периодичностью сбора 15 секунд. Объем доступных исторических данных составляет приблизительно 90643 точки, накопленные за период 16 дней непрерывной работы системы.

Итоговая формализация задачи: построить алгоритм A, наилучшим образом аппроксимирующий неизвестную функцию f^* и одновременно удовлетворяющий всем указанным ограничениям по точности, производительности и адаптивности для обеспечения надежного предиктивного мониторинга систем видеоаналитики.

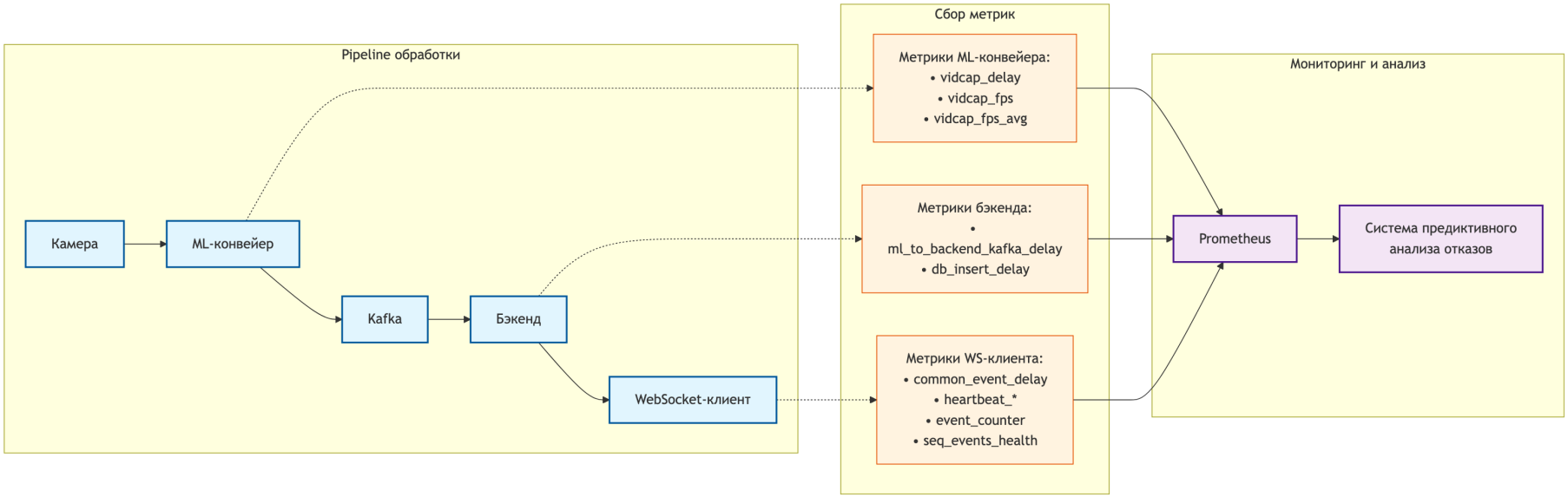

Рисунок 1.1 — Схема видеоконвейера и точки сбора метрик

2 Анализ данных и выбор методов

2.1 Анализ структуры данных видеоконвейера

Для построения эффективной модели прогнозирования необходимо провести анализ структуры и характеристик доступных данных. Исходный набор данных представляет собой многомерный временной ряд, собираемый системой мониторинга Prometheus [2] с различных компонентов видеоконвейера с периодичностью 15 секунд.

2.1.1 Описание набора метрик

Система мониторинга Prometheus [2] собирает широкий спектр метрик, характеризующих работу различных подсистем видеоконвейера, включая метрики ML-конвейера ($vidcap\_delay$, $vidcap\_fps$), бэкенда ($ml\_to\_backend\_ka$ db_insert_delay), и WebSocket-клиентов ($heartbeat\_*$, $event\_counter$, seq_events_he

Для построения модели прогнозирования отобраны следующие ключевые метрики, наиболее релевантные для задачи предсказания end-to-end-задержки:

- $common\_cad$ — целевая метрика end-to-end-задержки, усредненная за 1 час (мс);
- db_insert_cad — задержка записи в базу данных, усредненная за 1 час (мс);
- $kafka\_network\_cad$ — сетевая задержка Kafka [4], усредненная за 1 час (мс);

- $counter\_events\_total$ — общий счетчик обработанных событий в системе.

2.1.2 Временные характеристики данных

Исходный набор данных охватывает период с 25 ноября по 11 декабря 2024 года (16 дней непрерывной работы системы) и содержит 90543 временных точек. При интервале дискретизации 15 секунд это соответствует полному покрытию анализируемого периода без пропусков в данных.

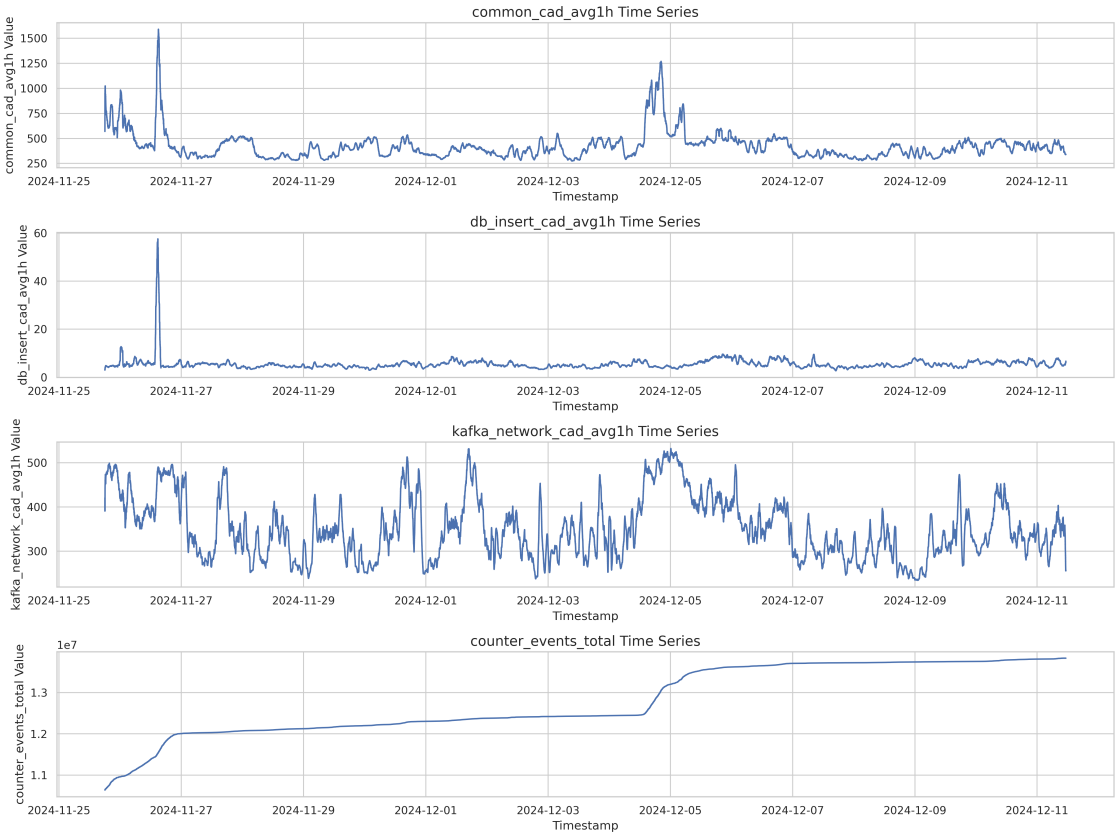

Рисунок 2.1 — Обзор временных рядов основных метрик видеоконвейера

Анализ временных характеристик показывает наличие различных паттернов в поведении метрик: циклические колебания, связанные с суточной активностью системы, периодические всплески нагрузки и редкие аномальные события, требующие особого внимания при построении модели.

2.1.3 Статистический анализ метрик

Для понимания распределения значений каждой метрики проведен описательный статистический анализ, результаты которого представлены в таблице 2.1.

Таблица 2.1 — Описательная статистика основных метрик

Метрика	**Среднее**	**Медиана**	**Мин.**	**Макс.**	**Std**
common_cad	423.72	400.63	273.66	1591.18	138.11
db_insert_cad	5.51	5.07	2.74	57.56	2.73
kafka_network_cad	351.51	339.84	234.22	532.15	68.76
counter_events_total	1.28×10^{7}	1.24×10^{7}	1.06×10^{7}	1.38×10^{7}	8.21×10^{5}

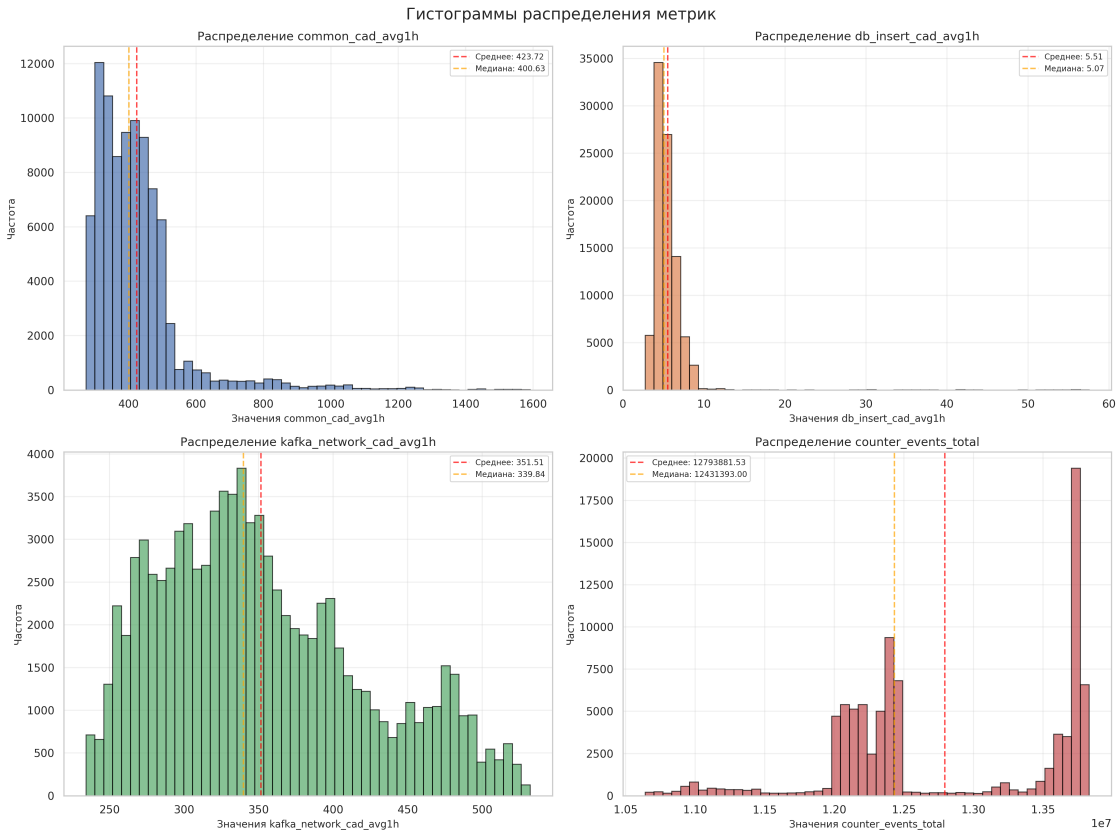

Рисунок 2.2 — Гистограммы распределения ключевых метрик системы

2.1.4 Анализ пропусков и качества данных

Качество исходных данных является критическим фактором для построения надежной прогностической модели. Анализ показывает наличие пропусков данных только в начале и конце временных рядов, что связано с особенностями синхронизации сбора различных метрик. В середине периода наблюдения пропуски отсутствуют, что свидетельствует о стабильной работе системы мониторинга.

Для обеспечения единообразия временных рядов из каждой метрики было исключено следующее количество точек:

- $common\_cad$ — 2 точки;
- db_insert_cad — 188 точек;
- $kafka\_network\_cad$ — 188 точек;
- $counter\_events\_total$ — 216 точек.

Данная стратегия обработки пропусков путем обрезания краевых значений является предпочтительной по сравнению с интерполяцией, поскольку сохраняет естественную структуру временных зависимостей в данных и исключает внесение искусственных артефактов в модель.

2.1.5 Выявление аномалий в данных

Для обнаружения аномальных значений в данных применен метод межквартильного размаха (IQR). Точки, выходящие за границы $Q_1 - 1.5 \times IQR$ и $Q_3 + 1.5 \times IQR$, рассматриваются как потенциальные выбросы.

Рисунок 2.3 — IQR-диаграммы (диаграммы размаха) и boxplots для выявления выбросов в метриках

IQR-диаграммы наглядно демонстрируют квартили и выбросы для каждой метрики, позволяя оценить степень вариабельности данных и выявить аномальные периоды работы системы.

Обнаруженные аномалии требуют детального анализа для определения их природы: являются ли они результатом реальных событий в системе

(пиковые нагрузки, сбои) или ошибками измерения. В зависимости от результатов анализа принимается решение о сохранении, корректировке или исключении аномальных точек из обучающей выборки.

2.2 Корреляционный анализ метрик

3 Глава n

3.1 Секция n

4 Глава n

4.1 Секция n

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Jain K., Adapa K.S., Grover K., Sarvadevabhatla R.K., Purini S. A Cloud-Fog Architecture for Video Analytics on Large Scale Camera Networks Using Semantic Scene Analysis // 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid). — 2023. — P. 513–523. DOI: 10.1109/CCGrid57682.2023.00054.

2) Prometheus monitoring system and time series database. — URL: https://prometheus.i (дата обращения: 05.06.2025).

3) Grafana: The open observability platform. — URL: https://grafana.com/docs/ (дата обращения: 05.06.2025).

4) Apache Kafka: A distributed streaming platform. — URL: https://kafka.apache.org/do (дата обращения: 05.06.2025).

5) Docker: Accelerated Container Application Development. — URL: https://docs.docke (дата обращения: 05.06.2025).

6) Third

Список сокращений и условных обозначений

Сокращения:

API	Application Programming Interface — программный интерфейс приложения
Docker	платформа контейнеризации приложений
FPS	Frames Per Second — кадры в секунду
Kafka	Apache Kafka — распределенный брокер сообщений
LoRA	Low-Rank Adaptation — адаптация с низкоранговой аппроксимацией
ML	Machine Learning — машинное обучение
MLOps	Machine Learning Operations — операции машинного обучения
MSE	Mean Squared Error — среднеквадратическая ошибка
Prometheus	система мониторинга и оповещений с открытым исходным кодом
SLA	Service Level Agreement — соглашение об уровне обслуживания
WS	WebSocket — протокол полнодуплексной связи

Условные обозначения:

T	множество временных меток наблюдений
d	число метрик, собираемых системой мониторинга
L	длина скользящего окна наблюдений
s	шаг сдвига скользящего окна
$\mathbf{x}_i$	d-мерный вектор наблюдений в момент времени t_i
X_k	матрица скользящего окна размерности $L \times d$
y_k	целевая переменная (значение $common\_event\_delay$)
$\mathcal{N}$	обучающая выборка
N	общее количество обучающих примеров
f^*	неизвестная целевая функция
A	разрабатываемый алгоритм прогнозирования
ε	допустимая погрешность прогнозирования
Δ	горизонт прогнозирования (15 секунд)
end-to-end	сквозной (от начала до конца процесса)
inference	процесс получения оценок от обученной модели
warm-start	инициализация обучения с предобученными параметрами