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.

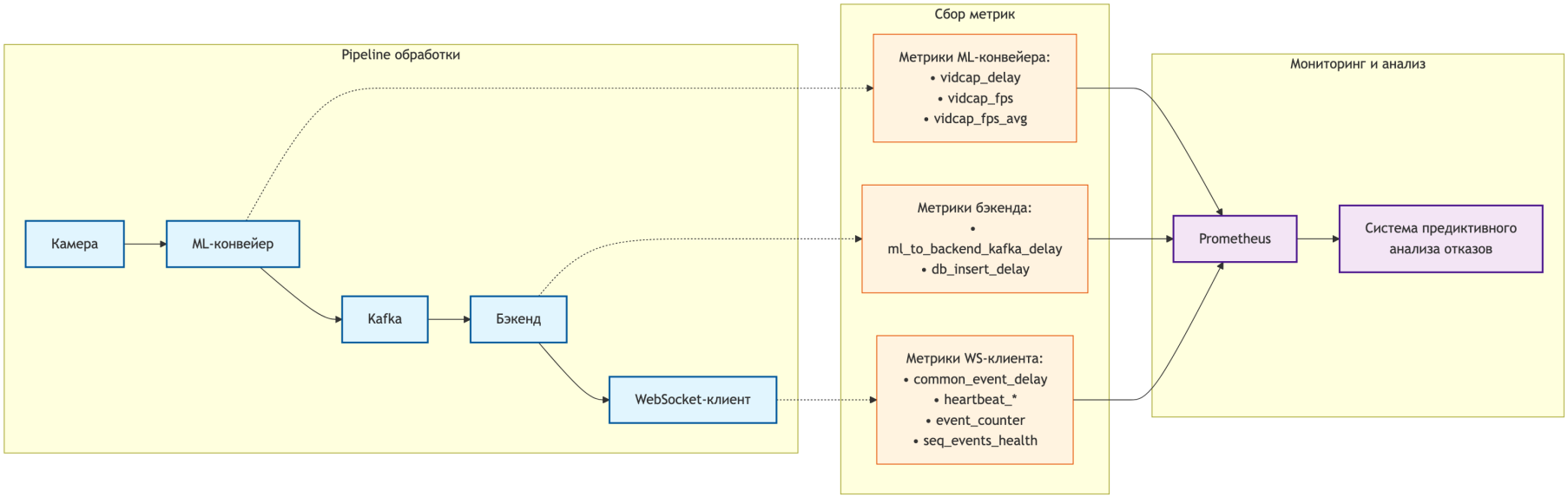

Рисунок 1.1 — Схема видеоконвейера и точки сбора метрик

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.

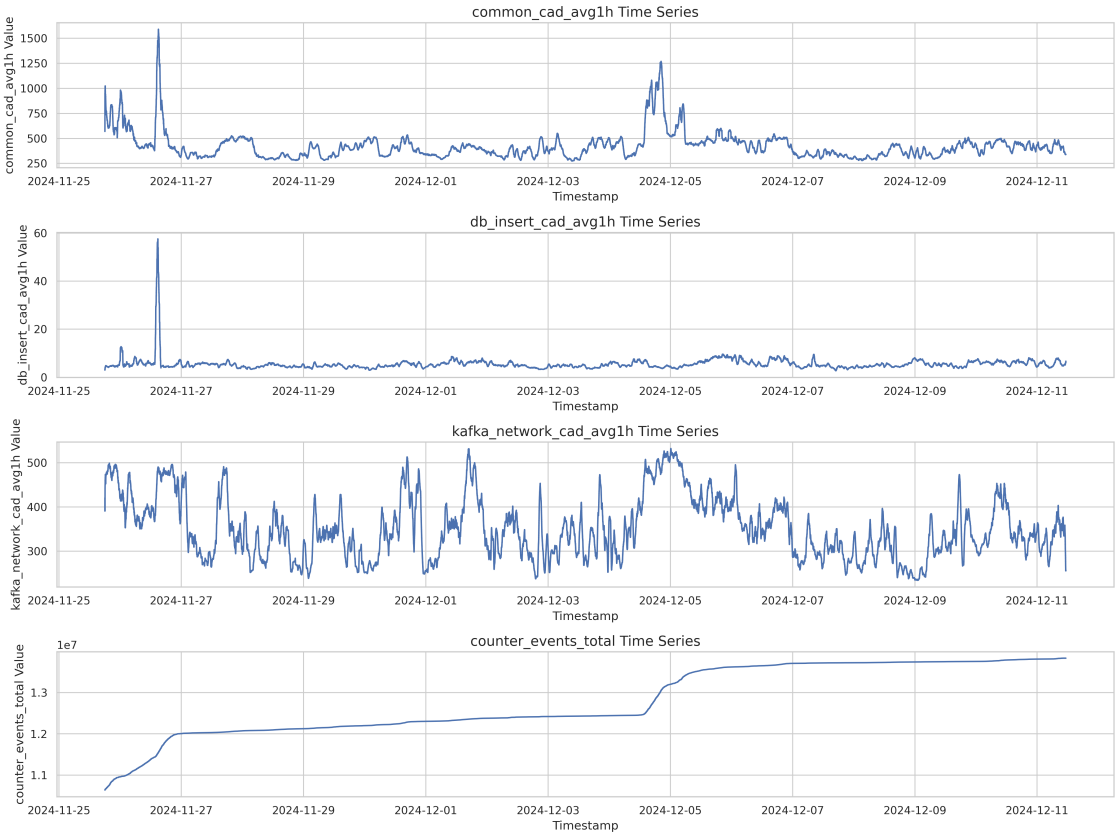

Рисунок 2.1 — Обзор временных рядов основных метрик видеоконвейера

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}

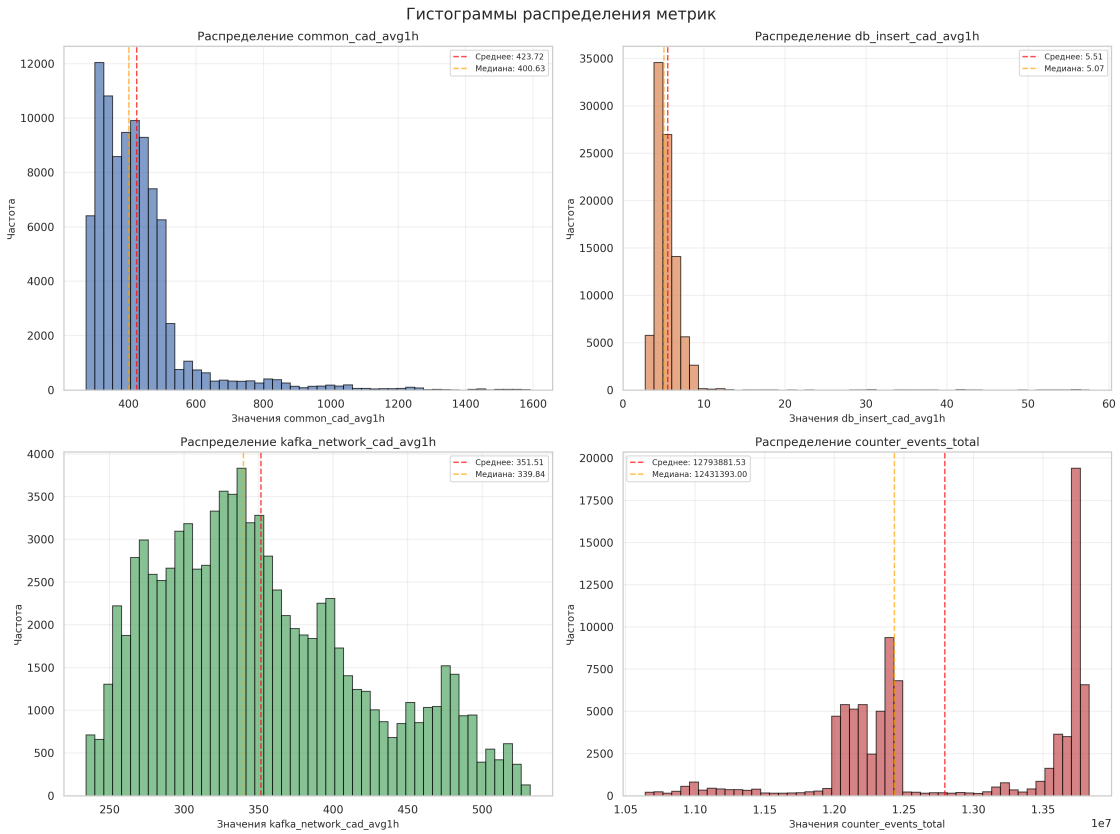

Рисунок 2.2 — Гистограммы распределения ключевых метрик системы

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

Для выявления взаимосвязей между метриками и определения наиболее информативных признаков для прогнозирования целевой переменной $common\_cad$ проведен корреляционный анализ временных рядов.

2.2.1 Матрица корреляций Пирсона

Вычисление коэффициентов корреляции Пирсона между всеми парами метрик позволяет оценить степень линейной взаимосвязи между переменными. Результаты анализа представлены в виде тепловой карты корреляций.

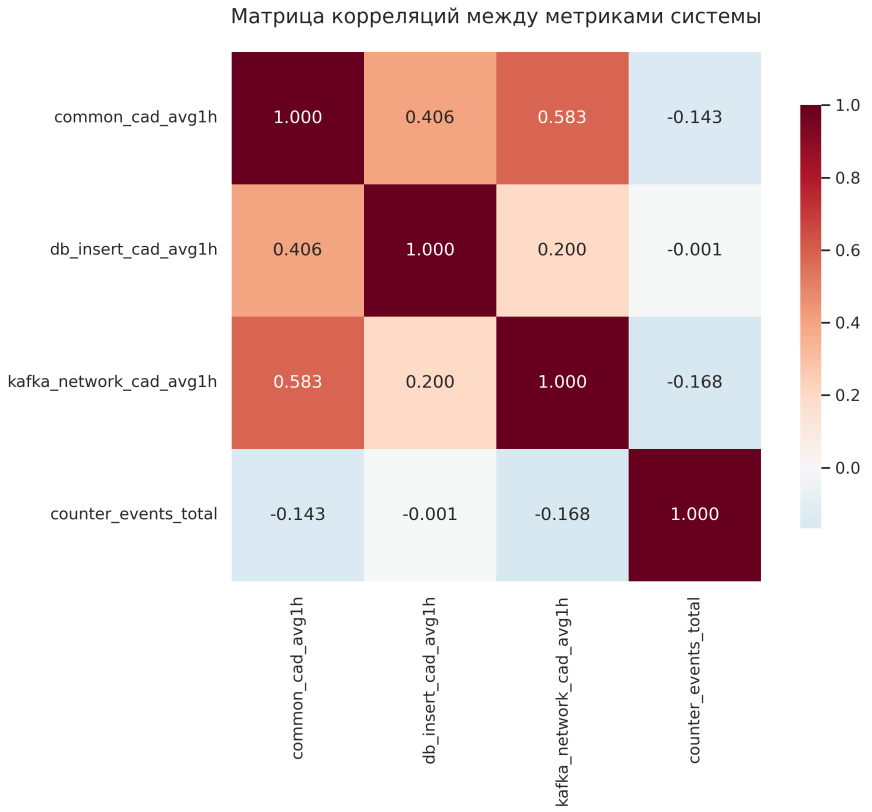

Рисунок 2.4 — Матрица корреляций между метриками системы

Анализ матрицы корреляций показывает наличие значимых взаимосвязей между отдельными метриками, что свидетельствует о взаимозависимости различных компонентов видеоконвейера. Наиболее сильные корреляции наблюдаются между метриками задержек ($common\_cad$, db_insert_cad, $kafka\_network\_cad$), что логично с точки зрения архитектуры системы.

2.2.2 Анализ связей с целевой переменной

Особое внимание уделено корреляциям с целевой метрикой $common\_cad$, поскольку они определяют потенциальную предсказательную способность признаков:

Таблица 2.2 — Корреляции метрик с целевой переменной $common\_cad\_avg1h$

Метрика	**Корреляция с** $common\_cad\_avg1h$
$kafka\_network\_cad\_avg1h$	+0.583
$db\_insert\_cad\_avg1h$	+0.406
$counter\_events\_total$	-0.143

2.2.3 Анализ временной структуры рядов

Для более глубокого понимания временных зависимостей был проведен анализ автокорреляционной функции (ACF), частной автокорреляционной функции (PACF) и сезонная декомпозиция для ключевых временных рядов.

Сезонная декомпозиция

Сезонная декомпозиция позволяет разложить временной ряд на три компоненты: тренд, сезонность и остаток (шум). Это помогает выявить долгосрочные тенденции и периодические колебания в данных. На *рисунке 2.5* представлена декомпозиция для целевой метрики $common\_cad\_avg1h$.

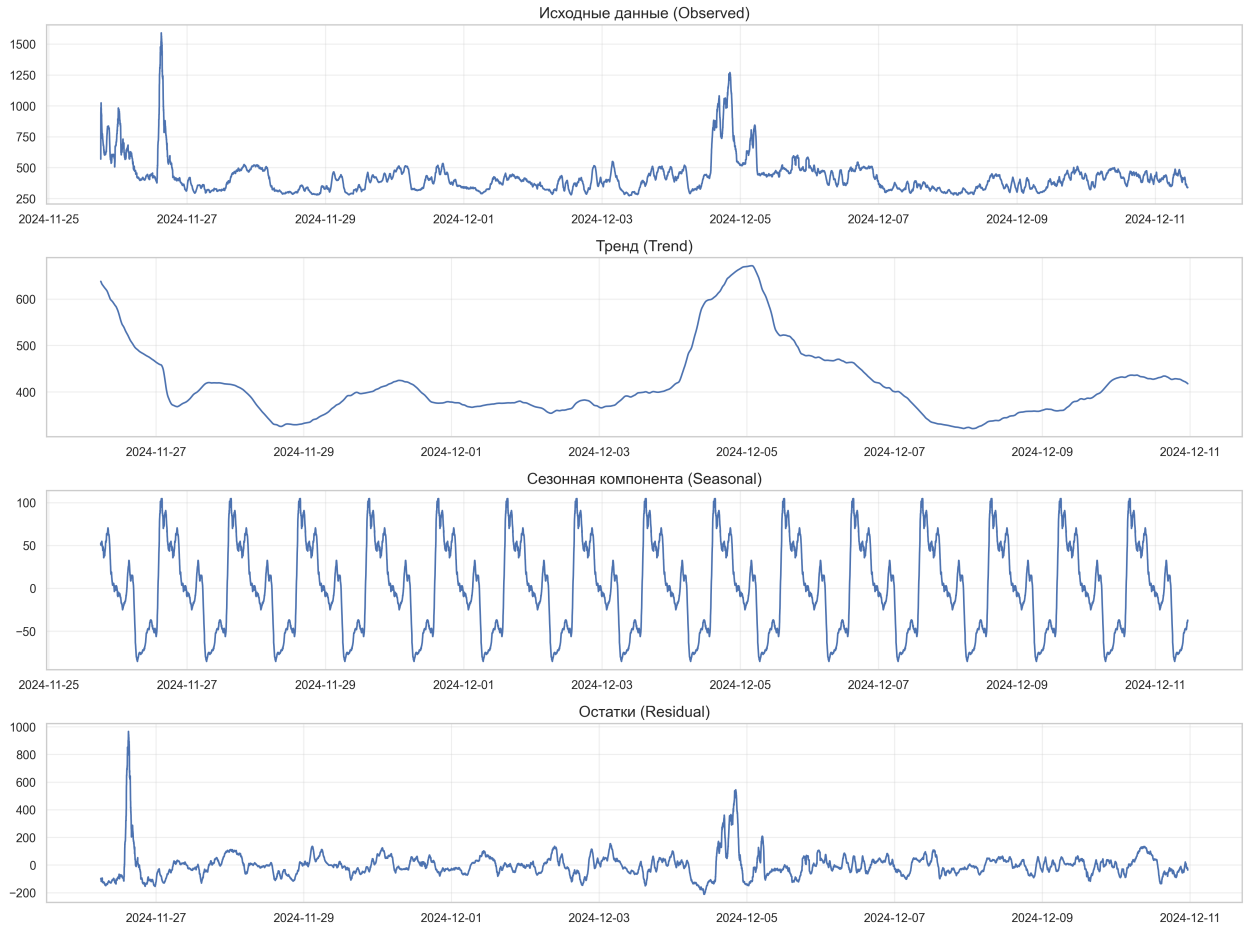

Рисунок 2.5 — Сезонная декомпозиция метрики $common\_cad\_avg1h$

Анализ показывает наличие выраженного нелинейного тренда с характерным ростом в начале декабря и последующим спадом. Наиболее важной особенностью является доминирующая суточная сезонность с четким повторяющимся паттерном, что характерно для систем с циклической нагрузкой. В остатках наблюдаются аномальные выбросы (например, 26.11 и 04.12), которые модель декомпозиции не смогла объяснить трендом и сезонностью.

Анализ автокорреляций

Функции ACF и PACF используются для определения порядка авторегрессионных (AR) и скользящих средних (MA) компонентов в моделях

временных рядов, таких как ARIMA. На *рисунке 2.6* показаны графики ACF и PACF.

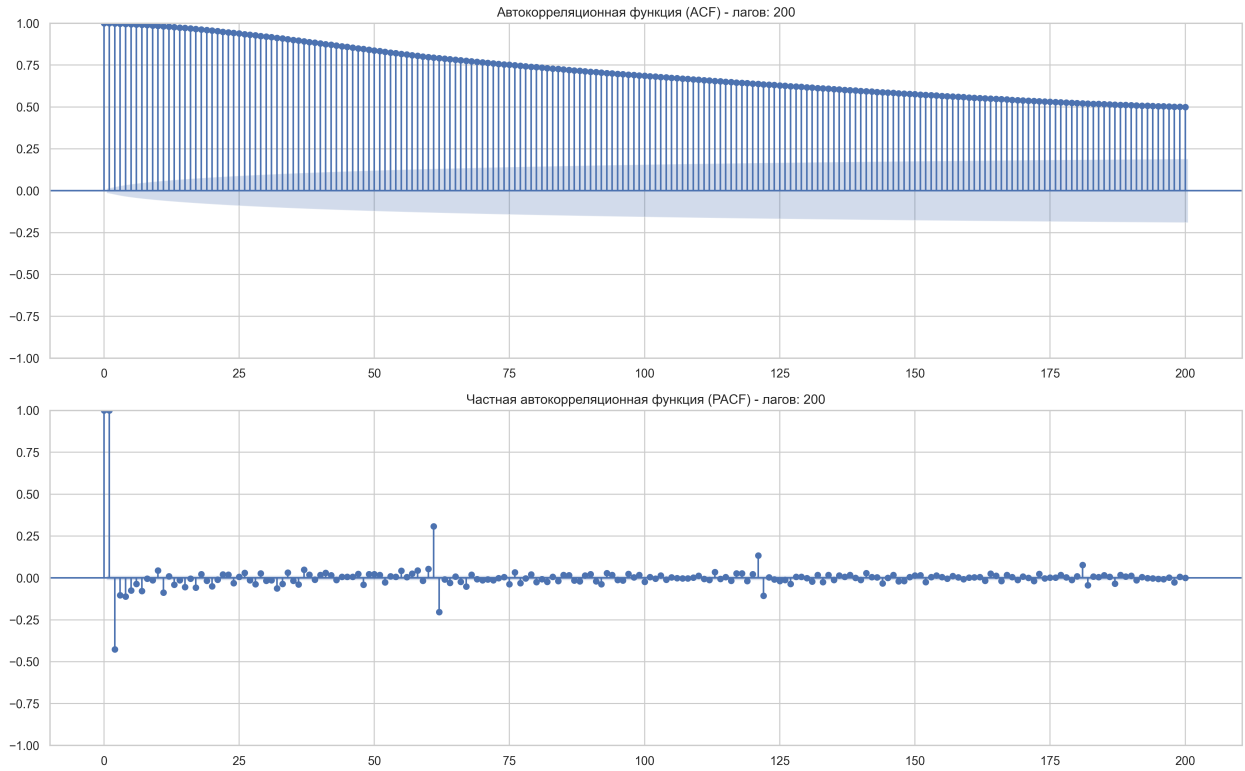

Рисунок 2.6 — Графики ACF и PACF для метрики $common\_cad\_avg1h$

Анализ автокорреляционных функций выявил ключевые характеристики временного ряда:

- **ACF медленно убывает** на протяжении всех 200 лагов, что является классическим признаком нестационарности ряда и наличия тренда. Волнообразная структура ACF подтверждает сильную сезонность;
- **PACF имеет резкий всплеск на лаге 1** с последующим обрывом, что указывает на авторегрессионный процесс первого порядка (AR(1)). Это означает сильную зависимость текущего значения от предыдущего;
- Совместный анализ ACF/PACF предполагает использование модели SARIMA

с начальными параметрами p=1, d=1, q=0 для несезонной части и дополнительными сезонными параметрами.

Однако учитывая сложность выявленных паттернов (нелинейный тренд, аномалии, сильная сезонность), для достижения высокой точности прогнозирования целесообразно рассмотреть как классические статистические методы (SARIMA), так и современные подходы машинного обучения (LSTM, Transformer), способные улавливать нелинейные зависимости.

2.2.4 Выводы по итогам анализа данных

На основе проведенного анализа данных сделаны следующие выводы:

- Корреляционный анализ подтвердил наличие статистически значимой связи между системными метриками и целевой переменной. Наиболее сильное влияние оказывает задержка в Kafka ($kafka\_network\_cad\_avg1h$) с корреляцией +0.583, что логично с точки зрения архитектуры системы;

- Сезонная декомпозиция выявила доминирующую суточную сезонность и нелинейный тренд с пиком в начале декабря. Обнаружены аномальные выбросы, требующие специальной обработки при моделировании;

- Анализ ACF/PACF показал нестационарность ряда (медленно убывающая ACF) и авторегрессионную структуру первого порядка (резкий обрыв PACF после лага 1). Это указывает на возможность применения модели SARIMA(1,1,0) с сезонными компонентами;

- Сложность выявленных паттернов (нелинейность, аномалии, сильная

сезонность) обосновывает необходимость сравнения классических статистических методов с современными подходами машинного обучения;

- Отсутствие сильной мультиколлинеарности между признаками позволяет использовать их все в модели без предварительного отсева.

Полученные результаты формируют основу для этапа feature engineering и выбора архитектуры модели прогнозирования, которые будут рассмотрены в следующей главе.

3 Разработка моделей для оценки задержек

После всестороннего анализа данных на предыдущем этапе, текущая глава посвящена практической разработке и реализации моделей для оценки задержки в видеоаналитическом конвейере. Основное внимание уделяется двум ключевым этапам: инжинирингу признаков, направленному на извлечение максимального количества полезной информации из временных рядов, и выбору, описанию и реализации моделей машинного обучения.

3.1 Инжиниринг признаков

Инжиниринг признаков — это процесс преобразования исходных временных рядов в структурированный набор данных (таблицу), пригодный для обучения моделей машинного обучения. Качество и информативность признаков напрямую влияют на точность и обобщающую способность итоговой модели [6]. На основе анализа, проведенного в Главе 2, был сформирован следующий набор признаков.

3.1.1 Календарные и временные признаки

Эти признаки позволяют модели учитывать зависимости, связанные со временем суток, днем недели и общим течением времени. Они особенно полезны для моделей на основе деревьев решений, таких как CatBoost.

- **Час дня** (`hour`) и **день недели** (`day_of_week`): категориальные признаки, позволяющие модели улавливать суточные и недельные паттерны.
- **Признак выходного дня** (`is_weekend`): бинарный флаг, принимающий значение 1, если день является субботой или воскресеньем, и 0 в про-

тивном случае.

- **Временной индекс** (`time_idx`): монотонно возрастающая переменная, представляющая собой количество времени (в часах), прошедшее с начала обучающего периода. Этот признак помогает модели аппроксимировать долгосрочный тренд в данных.

3.1.2 Циклические признаки

Календарные признаки, такие как час дня или день недели, по своей природе цикличны (после 23:00 идет 00:00). Чтобы донести эту информацию до моделей, особенно нейронных сетей, используются тригонометрические преобразования.

$$x_{sin} = \sin\left(\frac{2\pi x}{P}\right), \quad x_{cos} = \cos\left(\frac{2\pi x}{P}\right) \tag{3.1}$$

где x — исходное значение (например, час), а P — период цикла (24 для часов, 7 для дней недели). Такой подход преобразует одну переменную в две, представляя ее на единичной окружности.

3.1.3 Лаговые признаки (Lag features)

Лаговые признаки — это значения временного ряда из прошлого, используемые в качестве предикторов для будущих значений. Они являются ключевым способом информирования модели об авторегрессионной структуре данных, выявленной при анализе ACF/PACF. Признак создается путем сдвига временного ряда на k шагов назад:

$$\text{lag}_k(t) = y(t-k) \tag{3.2}$$

где $y(t-k)$ — значение целевой переменной в момент времени $t-k$.

В данном исследовании для CatBoost-модели использовались лаги: 1, 2, 4, 96, 192, 5760 шагов назад, что соответствует интервалам от 15 секунд до 24 часов. Такой выбор позволяет модели учитывать как непосредственную зависимость от предыдущих значений, так и суточную сезонность (лаг 5760 = 24 часа $\times$ 240 точек/час).

3.1.4 Признаки на основе скользящего окна (Rolling-window features)

Для захвата локальной динамики и структуры временного ряда вычисляются статистические показатели в пределах скользящего окна.

- **Скользящее среднее** (`rolling_mean`): сглаживает краткосрочные флуктуации и помогает выявить локальный тренд.
- **Скользящее стандартное отклонение** (`rolling_std`): характеризует волатильность (изменчивость) ряда в недавнем прошлом.

Размер окна w является гиперпараметром, который выбирается в зависимости от специфики модели и характера данных. В данном исследовании использовались различные наборы параметров для разных типов моделей:

- **Для LSTM-модели**: окна размером 20 и 240 точек данных (соответствующие 5 минутам и 1 часу при 15-секундном интервале);
- **Для CatBoost-модели**: более широкий набор окон — 4, 96, 192, 1920, 2880, 4320, 5760, 8640 точек данных (от 1 минуты до 36 часов), что позволяет модели улавливать как краткосрочные, так и долгосрочные паттерны.

3.2 Выбор и описание моделей

На основе выводов, сделанных в Главе 2, для решения задачи прогнозирования были выбраны модели, представляющие два разных подхода: классическую статистику и современное машинное обучение.

3.2.1 Модель SARIMA

Сезонная авторегрессионная интегрированная скользящая средняя (SARIMA) — это статистическая модель, которая является расширением модели ARIMA и предназначена для работы с временными рядами, обладающими ярко выраженной сезонностью. Выбор этой модели обоснован анализа ACF/PACF, который указал на наличие тренда, авторегрессионной зависимости и сезонных колебаний.

3.2.2 Модель CatBoost

CatBoost — это высокопроизводительная реализация градиентного бустинга над деревьями решений. Она хорошо зарекомендовала себя в работе с разнородными табличными данными, эффективно обрабатывает категориальные признаки и не требует тщательной настройки гиперпараметров.

3.2.3 Модель LSTM

Сети с долгой краткосрочной памятью (Long Short-Term Memory, LSTM) — это разновидность рекуррентных нейронных сетей (RNN), специально разработанная для улавливания долгосрочных зависимостей в последовательных данных. Архитектура LSTM позволяет эффективно бороться с проблемой исчезающих градиентов, что делает ее мощным инструментом для моделирования временных рядов.

В данной работе используется LSTM-архитектура со следующими характеристиками:

- Входной слой принимает последовательности длиной `window_size` временных шагов с количеством признаков, определяемым этапом feature engineering;
- Один LSTM-слой с 64 нейронами;
- Полносвязный скрытый слой с 8 нейронами и функцией активации ReLU;
- Выходной слой с одним нейроном и линейной функцией активации для регрессии;
- Оптимизатор Adam с learning rate 0.0001, функция потерь — Mean Squared Error.

3.3 Метрики оценки качества

Для оценки качества моделей прогнозирования используется набор метрик, позволяющих комплексно оценить точность оценки задержек. Выбор метрик обусловлен спецификой временных рядов и требованиями к практическому применению системы.

3.3.1 Средняя абсолютная процентная ошибка (MAPE)

MAPE является основной метрикой для оценки качества, поскольку обеспечивает интерпретируемость результатов в процентах:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{3.3}$$

где y_i — истинное значение, $\hat{y}_i$ — прогнозируемое значение, n — количество наблюдений. Согласно техническим требованиям, целевое значение MAPE должно быть менее 10%.

3.3.2 Среднеквадратичная ошибка (RMSE)

RMSE чувствительна к выбросам и позволяет оценить общую точность модели:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.4}$$

3.3.3 Средняя абсолютная ошибка (MAE)

MAE менее чувствительна к выбросам и показывает среднее отклонение прогнозов:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.5}$$

3.4 Методология проведения экспериментов

Корректная оценка качества моделей временных рядов требует специального подхода к разделению данных, учитывающего временную структуру и предотвращающего утечку информации из будущего в прошлое.

3.4.1 Кросс-валидация для временных рядов

Для корректной оценки качества моделей применяется специализированная кросс-валидация временных рядов (TimeSeriesSplit), которая учитывает хронологический порядок данных и предотвращает утечку ин-

формации из будущего.

Метод TimeSeriesSplit работает следующим образом:

- Данные разбиваются на k фолдов, где каждый последующий фолд включает больше исторических данных для обучения;
- Для каждого фолда тестовая выборка всегда находится хронологически после обучающей;
- Внутри каждого фолда обучающие данные дополнительно разделяются на train и validation в пропорции, определяемой параметром `test_size`.

3.4.2 Процедура валидации

Для каждого фолда кросс-валидации выполняется следующая последовательность действий:

1. **Разделение данных**: фолд разбивается на train+validation и test согласно TimeSeriesSplit;
2. **Внутреннее разделение**: train+validation дополнительно разделяется на обучающую и валидационную выборки;
3. **Масштабирование**: параметры нормализации вычисляются только на обучающей выборке и применяются ко всем частям фолда;
4. **Обучение модели**: модель обучается на train с валидацией на validation выборке;
5. **Оценка качества**: финальная оценка производится на тестовой части фолда;
6. **Сохранение результатов**: метрики каждого фолда сохраняются для последующего усреднения.

Итоговые метрики качества вычисляются как среднее арифметическое соответствующих метрик по всем фолдам, что обеспечивает более надежную и несмещенную оценку производительности модели.

3.4.3 Горизонт прогнозирования

Все модели настраиваются для прогнозирования на 900 временных шагов вперед (3.75 часа), что соответствует практическим требованиям системы мониторинга для своевременного реагирования на потенциальные проблемы в видеоконвейере.

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) Renault A., Bondu A., Lemaire V., Gay D. Automatic Feature Engineering for Time Series Classification: Evaluation and Discussion // 2023 International Joint Conference on Neural Networks (IJCNN). — 2023. — P. 1–10. DOI: 10.1109/IJCNN54540.2023.10191074.

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