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

Настоящая работа посвящена проблеме обеспечения производительно- сти современных систем видеоаналитики. Для предотвращения деградации качества обслуживания был разработан метод прогнозирования сквозной за- держки обработки видеопотока.

В ходе исследования был проведен анализ многомерных временных рядов системных метрик, на основе которого были построены и сравнены прогностические модели, включая CatBoost и LSTM. Для оценки их качества и устойчивости применялась методология временной кросс-валидации.

Результатом работы стал прототип системы предиктивного мониторин- га, способный с точностью MAPE 0.89% прогнозировать задержки на го- ризонте 3.75 часа. Разработанное решение является основой для создания интеллектуальной системы оповещений, которая позволяет принимать пре- вентивные меры для поддержания стабильности видеоаналитического кон- вейера.

**Ключевые слова:** предиктивный анализ, временные ряды, мониторинг производительности, видеоаналитика, машинное обучение, LSTM, CatBoost, MLOps.

СОДЕРЖАНИЕ

[АННОТАЦИЯ](#_bookmark0) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

[СОДЕРЖАНИЕ](#_bookmark1) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

[ВВЕДЕНИЕ](#_bookmark2) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1. [Общие положения](#_bookmark3) 10
   1. [Архитектура системы видеоаналитики](#_bookmark4) 10
   2. [Постановка задачи](#_bookmark5) 11
2. [Анализ данных и выбор методов](#_bookmark6) 14
   1. [Анализ структуры данных видеоконвейера](#_bookmark7) 14
      1. [Описание набора метрик](#_bookmark8) 14
      2. [Временные характеристики данных](#_bookmark9) 15
      3. [Статистический анализ метрик](#_bookmark10) 15
      4. [Анализ пропусков и качества данных](#_bookmark11) 17
      5. [Выявление аномалий в данных](#_bookmark12) 17
   2. [Корреляционный анализ метрик](#_bookmark13) 19
      1. [Матрица корреляций Пирсона](#_bookmark14) 19
      2. [Анализ связей с целевой переменной](#_bookmark15) 20
      3. [Анализ временной структуры рядов](#_bookmark16) 20
      4. [Выводы по итогам анализа данных](#_bookmark17) 23
3. [Разработка моделей для оценки задержек](#_bookmark18) 24
   1. [Формирование признаков](#_bookmark19) 24
      1. [Календарные и временные признаки](#_bookmark20) 24
      2. [Циклические признаки](#_bookmark21) 25
      3. [Лаговые признаки (Lag features)](#_bookmark22) 25
      4. [Признаки на основе скользящего окна (Rolling-window features)](#_bookmark23) 26
   2. [Выбор и описание моделей](#_bookmark24) 26
      1. [Модель SARIMA](#_bookmark25) 26
      2. [Модель CatBoost](#_bookmark26) 27
      3. [Модель LSTM](#_bookmark27) 27
      4. [Дополнительные эксперименты с современными моделями](#_bookmark28) 28
   3. [Метрики оценки качества](#_bookmark29) 29
      1. [Средняя абсолютная процентная ошибка (MAPE)](#_bookmark30) 29
      2. [Среднеквадратичная ошибка (RMSE)](#_bookmark31) 30
      3. [Средняя абсолютная ошибка (MAE)](#_bookmark32) 30
   4. [Методология проведения экспериментов](#_bookmark33) 30
      1. [Кросс-валидация для временных рядов](#_bookmark34) 30
      2. [Процедура валидации](#_bookmark35) 31
      3. [Горизонт прогнозирования](#_bookmark36) 32
4. [Результаты экспериментов и анализ](#_bookmark37) 33
   1. [Описание экспериментальной установки](#_bookmark38) 33
      1. [Программная и аппаратная среда](#_bookmark39) 33
      2. [Базовая модель для сравнения (Baseline)](#_bookmark40) 33
   2. [Результаты модели CatBoost](#_bookmark41) 34
      1. [Лучшие конфигурации CatBoost](#_bookmark42) 34
      2. [Анализ результатов CatBoost](#_bookmark43) 35
      3. [Проблема утечки данных](#_bookmark44) 36
   3. [Результаты модели LSTM](#_bookmark45) 36
      1. [Архитектура и параметры LSTM](#_bookmark46) 36
      2. [Лучшие конфигурации LSTM](#_bookmark47) 37
      3. [Анализ результатов LSTM](#_bookmark48) 37
   4. [Результаты модели SARIMA](#_bookmark49) 38
      1. [Параметры модели SARIMA](#_bookmark50) 38
      2. [Лучшие конфигурации SARIMA](#_bookmark51) 38
      3. [Анализ результатов SARIMA](#_bookmark52) 38
   5. [Сравнительный анализ моделей](#_bookmark53) 39
      1. [Анализ сравнительных результатов](#_bookmark54) 39
   6. [Визуализация результатов прогнозирования](#_bookmark55) 40
   7. [Анализ ошибок и интерпретация результатов](#_bookmark56) 42

[ЗАКЛЮЧЕНИЕ](#_bookmark57) 45

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ](#_bookmark58) 48

[Список сокращений и условных обозначений](#_bookmark59) 51

ВВЕДЕНИЕ

**Обоснование выбора темы и актуальность**

Современные системы видеоаналитики играют критически важную роль в обеспечении безопасности и мониторинга объектов критической инфра- структуры, 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 на- дежности всего технологического конвейера.

С ростом масштабов развертывания и усложнением архитектуры видео- аналитических систем возрастает и сложность их мониторинга [15]. Совре- менные решения часто включают в себя многоуровневые конвейеры обработ- ки, начиная от захвата видеопотоков с камер, их предварительной обработки, применения алгоритмов машинного обучения для детекции объектов и со- бытий, передачи результатов через брокеры сообщений в бэкенд-системы и далее к конечным пользователям через веб-интерфейсы [12], [13], [16].

Повышение объемов данных и жестких требований к end-to-end-задержкам (от момента возникновения события на видео до его отображения операто- ру) делает необходимым переход от реактивного к предиктивному подходу в управлении производительностью. Традиционные методы мониторинга, осно- ванные на статических пороговых значениях и алертах по факту превышения SLA, не способны предотвратить деградацию качества обслуживания до ее критических проявлений [14].

В данном контексте особую важность приобретает разработка интел- лектуальных систем предиктивного анализа, способных на основе потоко- вых метрик мониторинга (например, собираемых системой Prometheus [2] и визуализируемых в Grafana [3]) заблаговременно оценивать потенциаль- ные проблемы производительности и инициировать превентивные меры по их устранению.

**Цель и задачи исследования**

*Цель работы:* разработать и внедрить комплексный метод предиктив- ного анализа задержек в конвейере видеоаналитики, способный прогнози- ровать конечную метрику *common*\_*event*\_*delay* с заданной точностью для предупреждения операторов о потенциальных сбоях до их фактического про- явления.

Достижение поставленной цели требует решения комплекса взаимосвя- занных *задач:*

1. Проведение аналитического обзора и систематизация современной ли- тературы по предиктивному анализу временных рядов, машинному и глубокому обучению в контексте мониторинга и диагностики произво- дительности систем реального времени [19].
2. Проведение анализа структуры и взаимных корреляций временных ря- дов 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 [18], вклю- чая классические статистические подходы, методы машинного обуче- ния и глубокие нейронные сети, с оценкой их применимости к специ- фике видеоаналитических конвейеров.
4. Обоснованный выбор оптимальной архитектуры модели (градиентный бустинг, нейронные сети) с учетом требований к точности и скорости inference, а также определение необходимого объёма обучающих дан- ных и оптимальной периодичности переобучения модели.
5. Проектирование и реализация полноценного MLOps-конвейера, вклю- чающего автоматизированный feature-engineering, механизмы периоди- ческого дообучения модели на новых данных, высокопроизводительный inference-сервис и системы мониторинга качества оценок.
6. Всестороннее экспериментальное исследование точности и производи- тельности разработанной модели на обширных исторических данных с

использованием методов временной кросс-валидации и оценкой устой- чивости к различным типам аномалий в данных.

1. Разработка и внедрение интеллектуальной системы оповещений с адап- тивными порогами, а также формулирование практических рекоменда- ций по эксплуатации, настройке и масштабированию решения в произ- водственной среде.

**Методология и методы исследования**

Для достижения поставленной цели и решения сформулированных за- дач применяется комплексная методология, сочетающая теоретические ис- следования 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 вре- менных рядов с интеллектуальной генерацией признаков, включая вре- менные лаги различной глубины, скользящие статистические агрегаты, спектральные характеристики и высокоразмерные эмбеддинги для за- хвата сложных временных зависимостей.
3. Реализация и экспериментальное сравнение различных архитектур мо- делей (градиентный бустинг, нейронные сети) с применением строгих методов перекрёстной валидации по времени для обеспечения коррект- ной оценки обобщающей способности.
4. Контейнеризация решения с использованием технологии Docker и про- ведение детальных измерений latency inference в условиях, максимально приближенных к производственным, включая тестирование под нагруз- кой и оценку масштабируемости.

**Теоретическая и практическая значимость**

*Теоретическая значимость* работы заключается в сравнительном ана- лизе методов прогнозирования временных рядов для систем мониторинга и определении их применимости к задачам предиктивной диагностики в усло- виях жестких временных ограничений.

*Практическая значимость* определяется разработкой готового к про- мышленному использованию решения для мониторинга и предупреждения 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 на- дежность систем обработки потоковых данных в реальном времени, включая телекоммуникации, финансовые технологии и промышленную автоматиза- цию.

# 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, формулируют- ся требования к разрабатываемому алгоритму. Особое внимание уделяется описанию структуры видеоконвейера и точек сбора телеметрических данных, которые лягут в основу построения прогностической модели.

# Архитектура системы видеоаналитики

Исследуемая система видеоаналитики представляет собой многоком- понентный конвейер, предназначенный для обработки видеопотоков в ре- жиме реального времени с применением алгоритмов компьютерного зрения для детекции событий и объектов. Архитектура системы строится по прин- ципу микросервисной архитектуры, что обеспечивает масштабируемость и отказоустойчивость, но одновременно усложняет задачи мониторинга и диа- гностики производительности.

Видеоконвейер включает следующие основные компоненты: модуль за- хвата видеопотока с IP-камер (получающий данные по протоколу RTSP), ML- pipeline для применения алгоритмов компьютерного зрения, брокер сооб- щений 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 компо- нент 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 опе-

раторов и качество принимаемых ими решений в критических ситуациях. Общая схема видеоконвейера и точки сбора метрик представлены на *рисунке* *1.1*.

# Постановка задачи

Для формальной постановки задачи прогнозирования введем необхо- димые математические обозначения и определения.

**Дано**

1. **Многомерный временной ряд** метрик, собираемых из системы мо- ниторинга Prometheus с периодом дискретизации 15 секунд. В каж- дый момент времени *ti* фиксируется *d*-мерный вектор наблюдений **x***i* = [*m*(1)*, . . . , m*(*d*)] ∈ R*d*, характеризующий состояние видеоконвейера. Объ-

*i* *i*

ем доступных исторических данных составляет 90643 точки за 16 дней.

1. **Горизонт прогнозирования** ∆ = 900 временных шагов, что соответ- ствует 3.75 часам.

**Найти**

1. **Целевую функцию** *f* ∗ : R*L*×*d* → R, которая отображает историю на- блюдений (представленную матрицей *Xk* ∈ R*L*×*d* из *L* последних век- торов наблюдений) в будущее значение целевой метрики

*yk* = *common*\_*event*\_*delay*(*tk* + ∆).

1. **Прогностический алгоритм** *A*(*Xk*), который является наилучшей ап- проксимацией целевой функции *f* ∗.

**Критерии качества и ограничения**

Разрабатываемый алгоритм *A* должен удовлетворять следующим тре- бованиям:

1. **Точность прогнозирования:** средняя абсолютная процентная ошибка (MAPE) на тестовых данных должна быть менее 10%.
2. **Производительность:** время вычисления одного прогноза (inference time) на целевом оборудовании не должно превышать 5 секунд.
3. **Интерпретируемость:** модель должна предоставлять возможность оце- нить важность признаков, влияющих на прогноз.
4. **Практичность:** решение должно быть реализовано в виде, пригодном для интеграции в существующий MLOps-конвейер, включая контейне- ризацию с помощью Docker.

13

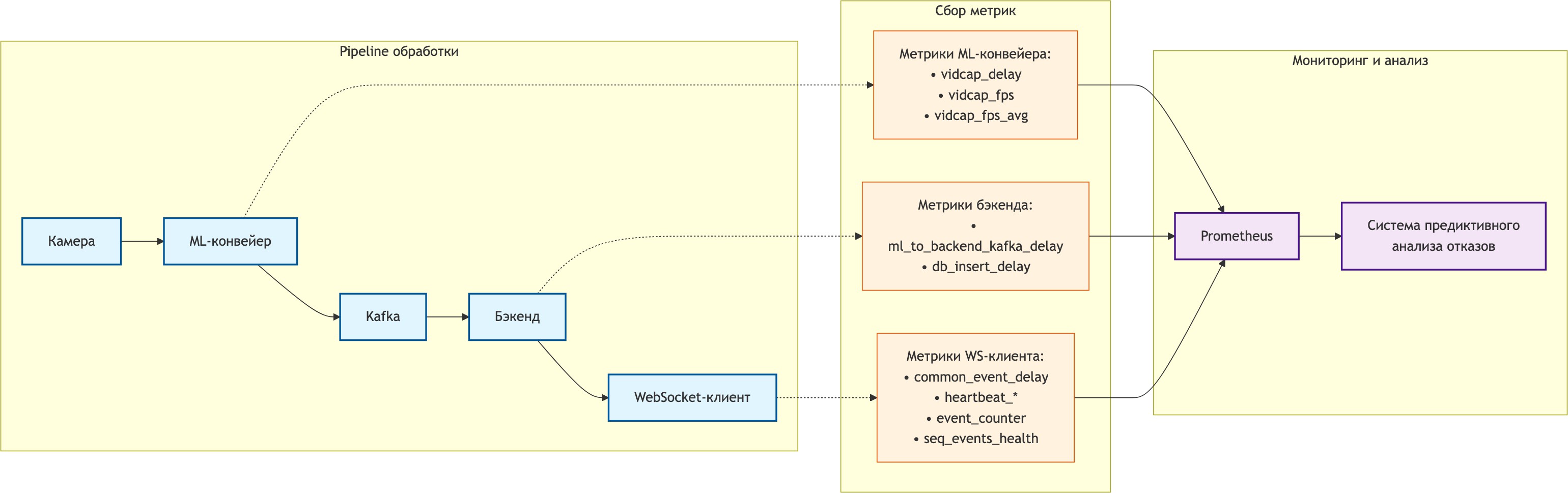


Рисунок 1.1 — Схема видеоконвейера и точки сбора метрик

# Анализ данных и выбор методов

# 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 видеокон- вейера с периодичностью 15 секунд.

# Описание набора метрик

Система мониторинга Prometheus [2] собирает широкий спектр мет- рик, характеризующих работу различных подсистем видеоконвейера, вклю- чая метрики ML-конвейера (*vidcap*\_*delay*, *vidcap*\_*fps*), бэкенда (*ml*\_*to*\_*backend*\_*kafka*\_*delay*, *db*\_*insert*\_*delay*), и WebSocket-клиентов (*heartbeat*\_∗, *event*\_*counter*, *seq*\_*events*\_*health*).

Для построения модели прогнозирования отобраны следующие клю- чевые 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 си- стеме.

# Временные характеристики данных

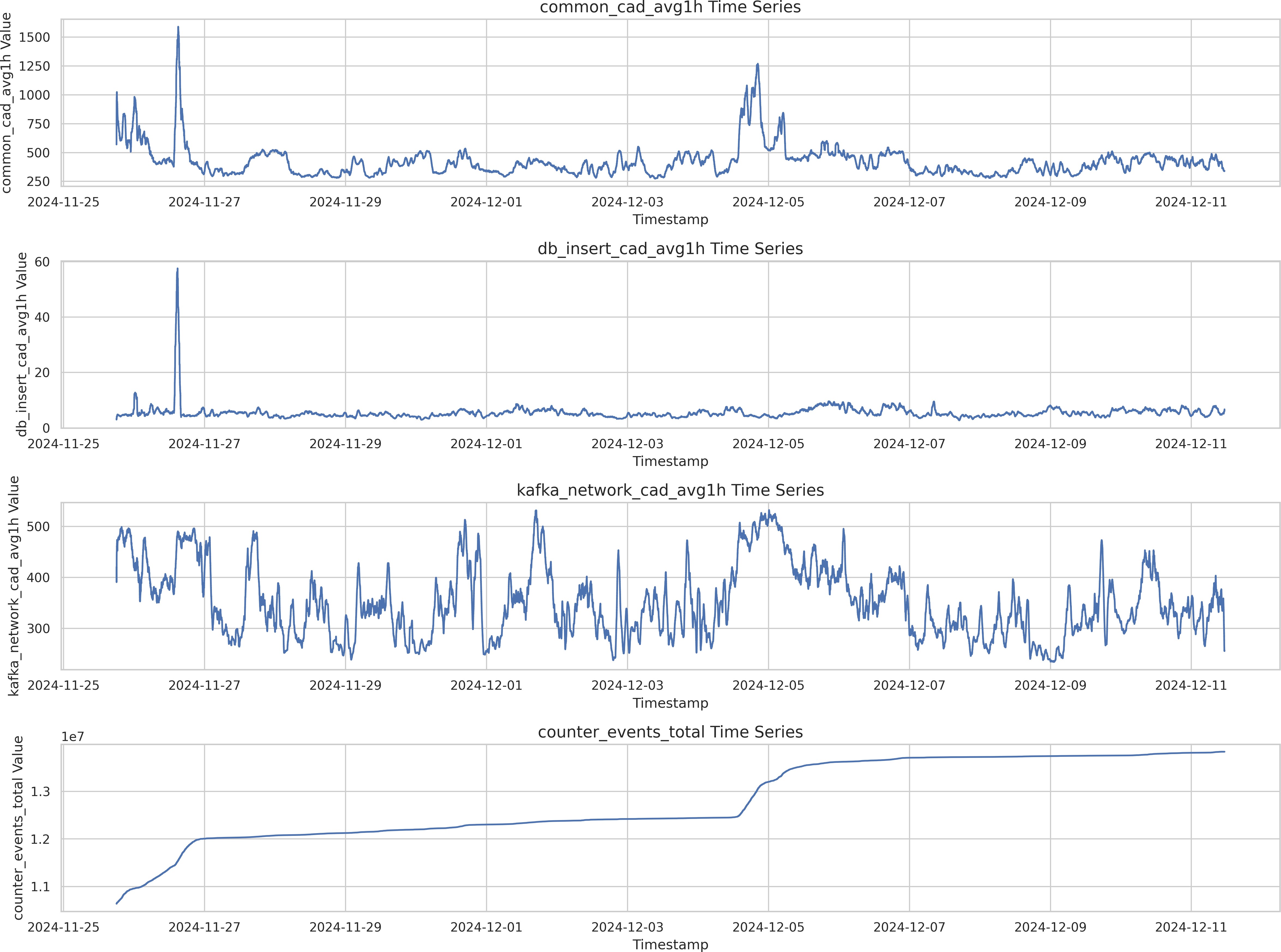
Исходный набор данных охватывает период с 25 ноября по 11 декабря 2024 года (16 дней непрерывной работы системы) и содержит 90543 вре- менных точек. При интервале дискретизации 15 секунд это соответствует полному покрытию анализируемого периода без пропусков в данных.

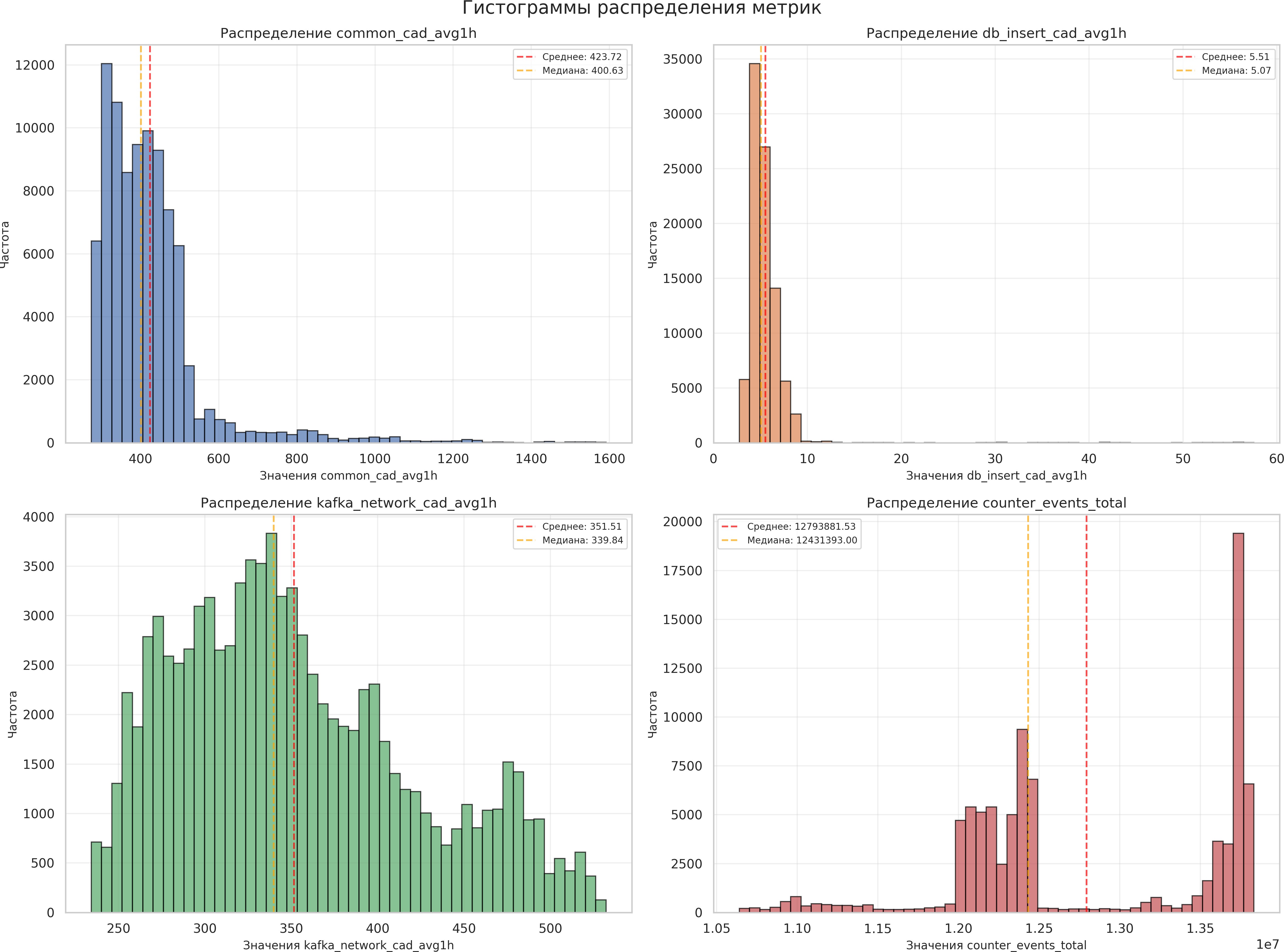
Рисунок 2.1 — Обзор временных рядов основных метрик видеоконвейера Анализ временных характеристик, представленных на *рисунке 2.1*, по-

казывает наличие различных паттернов в поведении метрик: циклические ко- лебания, связанные с суточной активностью системы, периодические всплес- ки нагрузки и редкие аномальные события, требующие особого внимания при построении модели.

# Статистический анализ метрик

Для понимания распределения значений каждой метрики проведен опи- сательный статистический анализ, результаты которого представлены в таб-

лице 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 | 1.28×107 | 1.24×107 | 1.06×107 | 1.38×107 | 8.21×105 |

Рисунок 2.2 — Гистограммы распределения ключевых метрик системы На *рисунке 2.2* представлены гистограммы распределения, которые по-

казывают, что большинство метрик имеют распределение, близкое к нор- мальному, но со смещением и тяжелыми хвостами, что указывает на наличие выбросов.

# Анализ пропусков и качества данных

Качество исходных данных является критическим фактором для по- строения 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 пе- риода 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 зна- чений является предпочтительной по сравнению с интерполяцией, поскольку сохраняет естественную структуру временных зависимостей в данных и ис- ключает внесение искусственных артефактов в модель.

# Выявление аномалий в данных

Для обнаружения аномальных значений в данных применен метод меж- квартильного размаха (IQR). Точки, выходящие за границы *Q*1 − 1*.*5 × *IQR* и *Q*3 + 1*.*5 × *IQR*, рассматриваются как потенциальные выбросы.

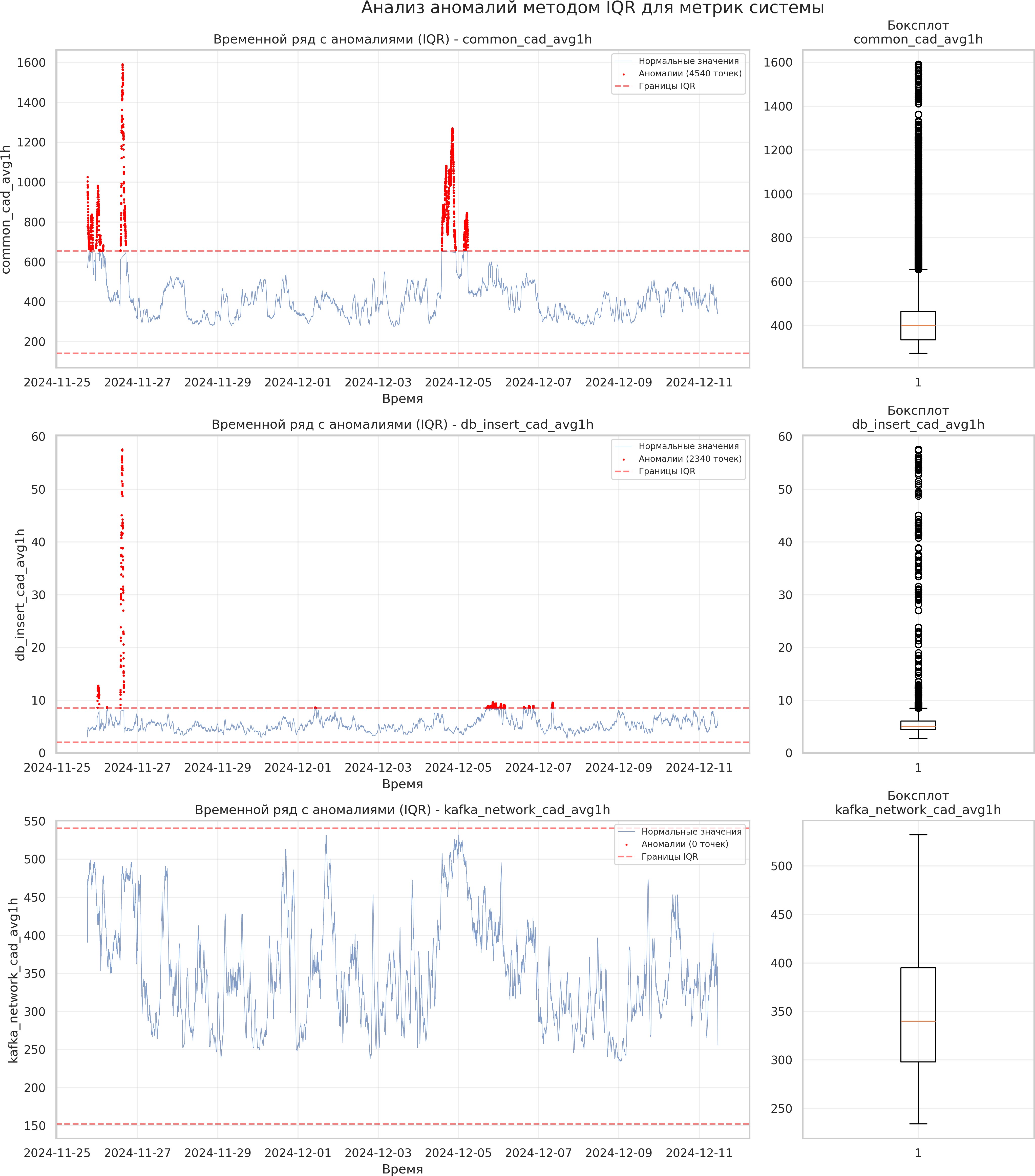


Рисунок 2.3 — IQR-диаграммы (диаграммы размаха) и boxplots для выявления выбросов в метриках

IQR-диаграммы на *рисунке 2.3* наглядно демонстрируют квартили и вы- бросы 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 (пико- вые нагрузки, сбои) или ошибками измерения. В зависимости от результатов анализа принимается решение о сохранении, корректировке или исключении

аномальных точек из обучающей выборки.

# Корреляционный анализ метрик

Для выявления взаимосвязей между метриками и определения наи- более информативных признаков для прогнозирования целевой переменной *common*\_*cad* проведен корреляционный анализ временных рядов.

# Матрица корреляций Пирсона

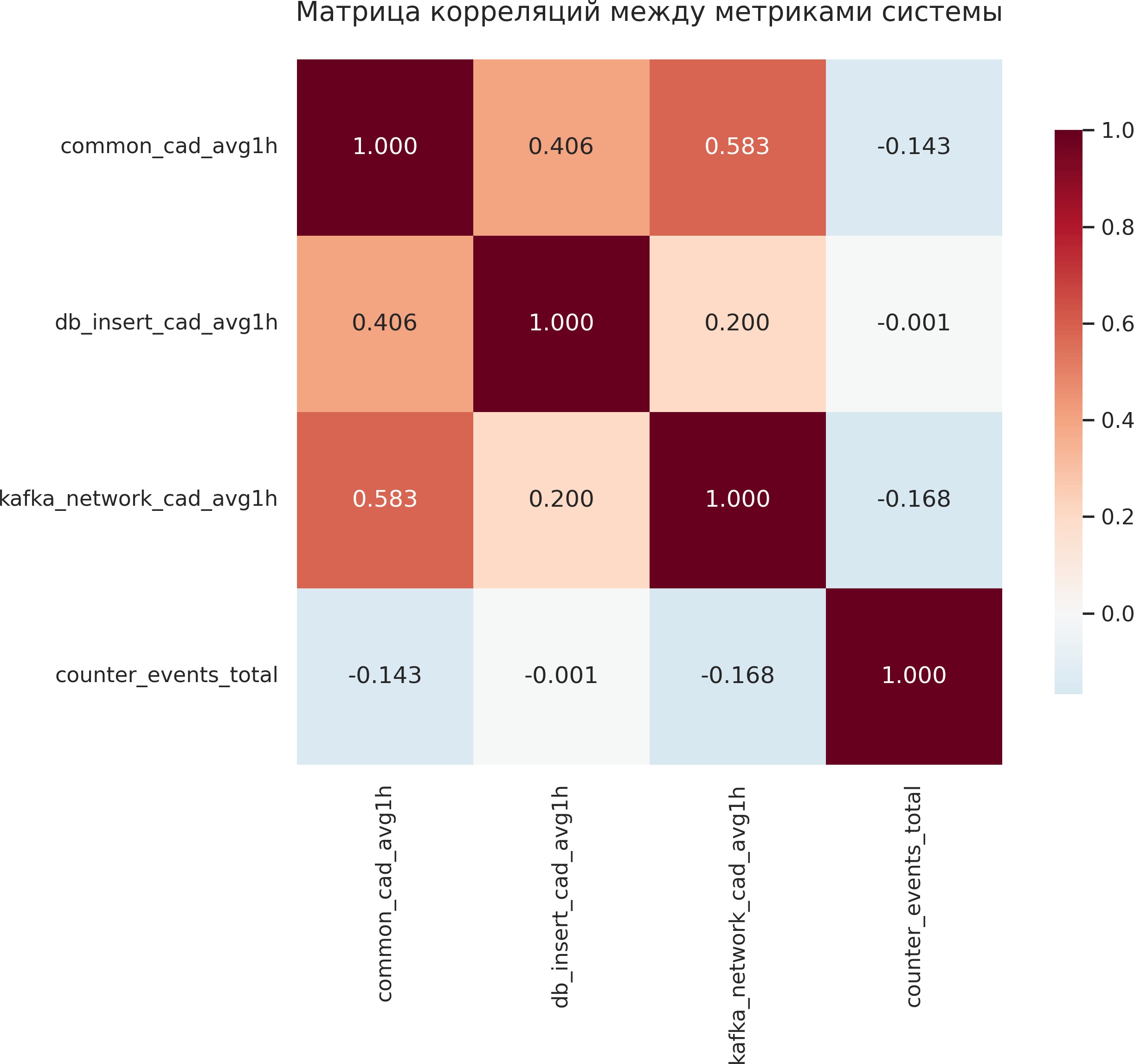
Вычисление коэффициентов корреляции Пирсона между всеми парами метрик позволяет оценить степень линейной взаимосвязи между переменны- ми. Результаты анализа представлены в виде тепловой карты корреляций.

Рисунок 2.4 — Матрица корреляций между метриками системы Анализ матрицы корреляций, представленной на *рисунке 2.4*, показыва-

ет наличие значимых взаимосвязей между отдельными метриками, что сви- детельствует о взаимозависимости различных компонентов видеоконвейе- ра. Наиболее сильные корреляции наблюдаются между метриками задержек

(*common*\_*cad*, *db*\_*insert*\_*cad*, *kafka*\_*network*\_*cad*), что логично с точки зре- ния архитектуры системы.

# Анализ связей с целевой переменной

Особое внимание уделено корреляциям с целевой метрикой *common*\_*cad*\_*avg*1*h*, поскольку они определяют потенциальную предсказа- тельную способность признаков. Основные коэффициенты приведены в *таб- лице 2.2*.

Таблица 2.2 — Корреляции метрик с целевой переменной

*common*\_*cad*\_*avg*1*h*

|  |  |
| --- | --- |
| **Метрика** | **Корреляция с** *common***\_***cad***\_***avg*1*h* |
| *kafka*\_*network*\_*cad*\_*avg*1*h* | +0.583 |
| *db*\_*insert*\_*cad*\_*avg*1*h* | +0.406 |
| *counter*\_*events*\_*total* | -0.143 |

# 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 дол- госрочные тенденции и периодические колебания в данных. На *рисунке 2.5* представлена декомпозиция для целевой метрики *common*\_*cad*\_*avg*1*h*.

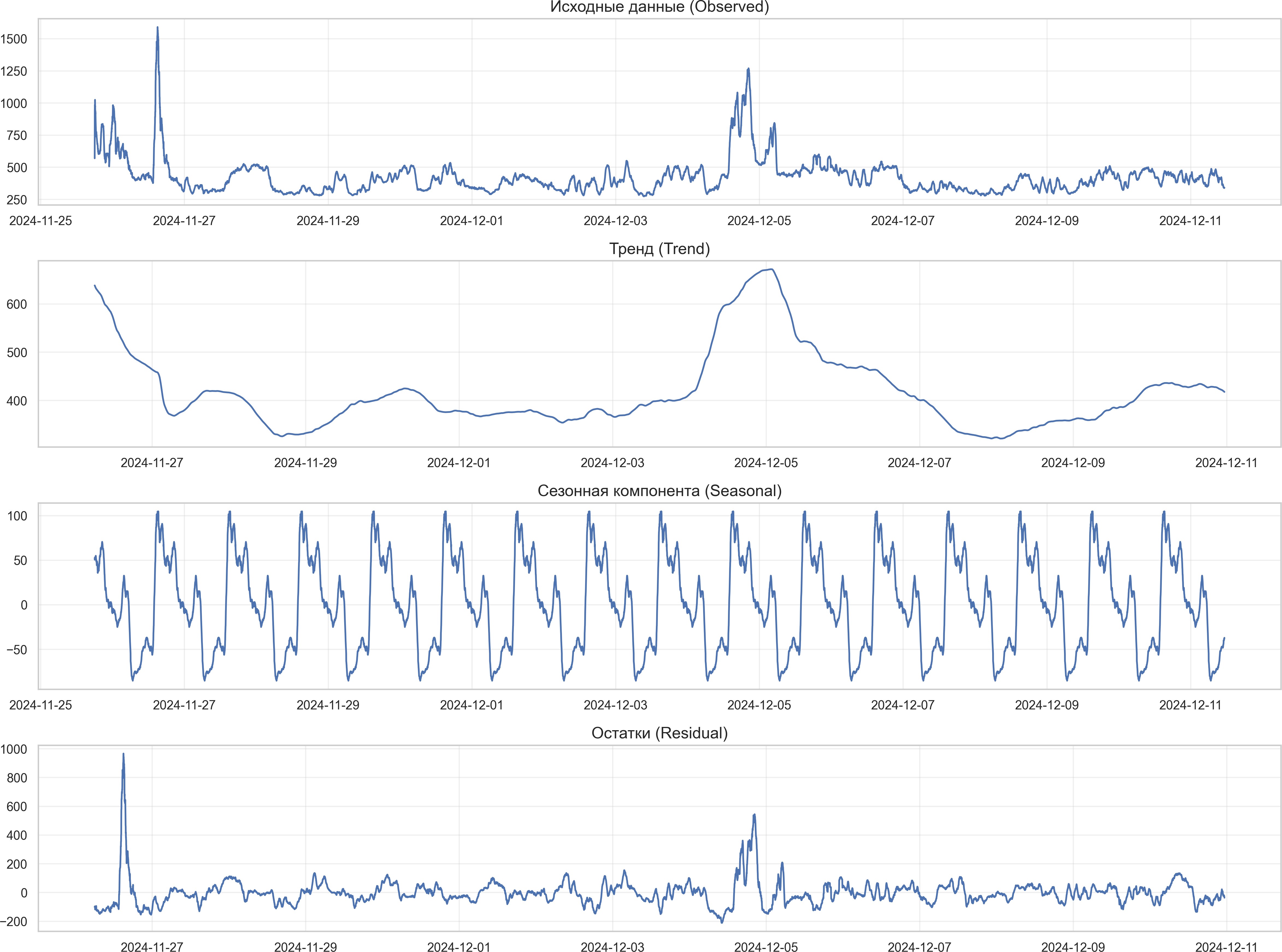


Рисунок 2.5 — Сезонная декомпозиция метрики *common*\_*cad*\_*avg*1*h*

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

**Анализ автокорреляций**

Функции ACF и PACF используются для определения порядка авто- регрессионных (AR) и скользящих средних (MA) компонентов в моделях временных рядов, таких как ARIMA. На *рисунке 2.6* показаны графики ACF и PACF.

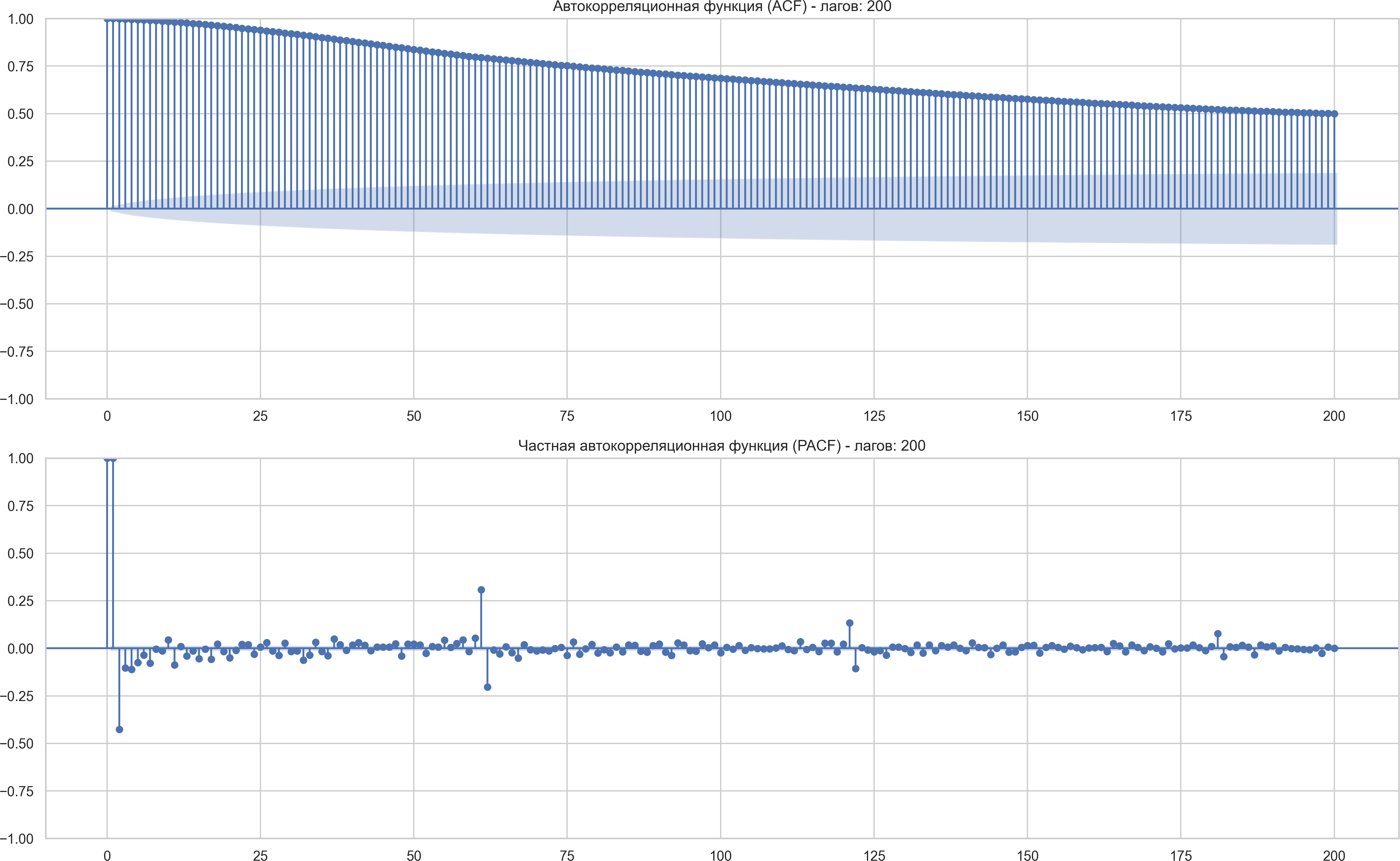


Рисунок 2.6 — Графики ACF и PACF для метрики *common*\_*cad*\_*avg*1*h*

Анализ автокорреляционных функций выявил ключевые характеристи- ки временного ряда:

* **ACF медленно убывает** на протяжении всех 200 лагов, что являет- ся 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 до- полнительными сезонными параметрами.

Однако учитывая сложность выявленных паттернов (нелинейный тренд, аномалии, сильная сезонность), для достижения высокой точности прогнози- рования целесообразно рассмотреть как классические статистические ме-

тоды (SARIMA), так и современные подходы машинного обучения (LSTM, Transformer), способные улавливать нелинейные зависимости.

# 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*\_*avg*1*h*) с корреляцией +0.583, что логично с точки зрения архитектуры систе- мы;
* 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 убыва- ющая 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 ста- тистических методов с современными подходами машинного обучения;
* Отсутствие сильной мультиколлинеарности между признаками позво- ляет 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.

# Разработка моделей для оценки задержек

На основе анализа данных, проведенного в предыдущей главе, в на- стоящей главе предлагается комплексный подход к решению поставленной задачи. Данный подход объединяет специализированное формирование при- знаков для данных мониторинга и методологию строгого сравнения различ- ных моделей прогнозирования. Далее в главе детально описываются этапы этого подхода: сначала рассматривается предложенный метод формирования признаков, а затем приводятся описания сравниваемых моделей (с обязатель- ными ссылками на первоисточники) и методология проведения эксперимен- тов.

# Формирование признаков

Формирование признаков — это процесс преобразования исходных вре- менных рядов в структурированный набор данных (таблицу), пригодный для обучения моделей машинного обучения. Качество и информативность при- знаков напрямую влияют на точность и обобщающую способность итоговой модели [6, 17]. На основе анализа, проведенного в Главе 2, был сформирован следующий набор признаков.

# 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 призна- ки, 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 на- чала обучающего периода. Этот признак помогает модели аппроксими- ровать долгосрочный тренд в данных.

# 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.

*xsin*

= sin 2*πx* *, x*

*cos*

= cos 2*πx* (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.

*P*

*P*

# Лаговые признаки (Lag features)

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

lag*k*(*t*) = *y*(*t* − *k*) (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 × 240 точек/час).

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

Для захвата локальной динамики и структуры временного ряда вычис- ляются статистические показатели в пределах скользящего окна.

* **Скользящее среднее (**rolling\_mean**)**: сглаживает краткосрочные флук- туации 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 зависи- мости 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 (соответ- ствующие 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.

# Выбор и описание моделей

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

# 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 выра- женной сезонностью [7]. Выбор этой модели обоснован анализом ACF/PACF, который указал на наличие тренда, авторегрессионной зависимости и сезон- ных колебаний.

# Модель CatBoost

CatBoost — это высокопроизводительная реализация градиентного бу- стинга над деревьями решений [8]. Она хорошо зарекомендовала себя в ра- боте с разнородными табличными данными, эффективно обрабатывает кате- гориальные признаки и не требует тщательной настройки гиперпараметров.

Особенностью предложенного в данной работе подхода является ис- пользование гибридной модели на основе CatBoost. Поскольку модели, ос- нованные на деревьях решений, не способны экстраполировать тренд, бы- ла применена стратегия декомпозиции временного ряда. Исходный ряд был разделен на три компоненты: тренд, сезонность и остатки. Каждая из этих компонент прогнозировалась отдельно:

* **Тренд** моделировался с помощью отдельной, более простой линейной модели.
* **Сезонная компонента и остатки** прогнозировались основной моделью CatBoost, которая эффективно работает со сложными нелинейными за- висимостями после удаления тренда.

Итоговый прогноз получался путем суммирования прогнозов по каждой из компонент. Такой подход позволяет сочетать преимущества обоих типов мо- делей.

# Модель LSTM

Сети с долгой краткосрочной памятью (Long Short-Term Memory, LSTM)

* это разновидность рекуррентных нейронных сетей (RNN), специально раз- работанная для улавливания долгосрочных зависимостей в последовательных данных [9].Также архитектура LSTM позволяет эффективно бороться с про- блемой затухающих градиентов.

В данной работе используется LSTM-архитектура со следующими ха- рактеристиками:

* + Входной слой принимает последовательности длиной window\_size вре-

менных шагов с количеством признаков, определяемым этапом форми- рования признаков;

* + 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.

# Дополнительные эксперименты с современными моделями

В рамках исследования также проводились эксперименты с современ- ными архитектурами для анализа временных рядов, такими как N-BEATS [20], а также с моделями из библиотек Time-Series-Library [10] и AutoTS [11] для оценки их применимости к данной задаче.

**Модели Time-Series-Library**

Были протестированы следующие модели на основе трансформеров и линейных архитектур:

* + **DLinear, PatchTST, iTransformer, Crossformer** — успешно завершили обучение;
  + **NLinear, Autoformer, FEDformer, Informer, TimesNet, Transformer** — завершились с ошибками или превысили лимит времени выполнения.

**Модели AutoTS**

Библиотека AutoTS [11] предоставляет автоматизированный подход к выбору и настройке моделей временных рядов. Были протестированы модели:

* + **LastValueNaive** — простая baseline модель;
  + **SeasonalityMotif, SectionalMotif** — модели на основе выявления пат- тернов;
  + **GLS** — обобщенный метод наименьших квадратов.

**Результаты дополнительных экспериментов**

Несмотря на современность указанных архитектур, результаты оказа- лись неудовлетворительными по сравнению с основными моделями (SARIMA, CatBoost, LSTM). Это может быть обусловлено:

* + Недостаточной настройкой гиперпараметров для специфики данной за- дачи;
  + Неоптимальным формированием признаков для трансформер-архитектур;
  + Различиями в методологии предобработки данных между библиотека- ми.

В связи с этим для финальной оценки были выбраны три основные модели, показавшие наилучшее соотношение качества и стабильности ре- зультатов.

# Метрики оценки качества

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

# Средняя абсолютная процентная ошибка (MAPE)

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

*n*

*i*

*i*

MAPE

=

*i*=1

100% Σ

*n*

*y* − *y*ˆ

*y*

*i*

(3.3)

где *yi* — истинное значение, *y*ˆ*i* — прогнозируемое значение, *n* — ко- личество наблюдений. Согласно техническим требованиям, целевое значение MAPE должно быть менее 10%.

# Среднеквадратичная ошибка (RMSE)

RMSE чувствительна к выбросам и позволяет оценить общую точность модели:

*n*

,u, 1 Σ

RMSE =

*n*

(*yi* − *y*ˆ*i*)2

(3.4)

*i*=1

# Средняя абсолютная ошибка (MAE)

MAE менее чувствительна к выбросам и показывает среднее отклонение прогнозов:

*n*

1 Σ

MAE = |*yi*

*n*

*i*=1

— *y*ˆ*i*| (3.5)

# Методология проведения экспериментов

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

# Кросс-валидация для временных рядов

Для корректной оценки качества моделей применяется специализиро- ванная кросс-валидация временных рядов (TimeSeriesSplit), которая учиты- вает хронологический порядок данных и предотвращает утечку информации из будущего.

Метод TimeSeriesSplit работает следующим образом:

* + Данные разбиваются на *k* фолдов, где каждый последующий фолд вклю-

чает 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.

# Процедура валидации

Для каждого фолда кросс-валидации выполняется следующая последо- вательность действий:

1. **Разделение данных**: фолд разбивается на train+validation и test соглас- но 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.

# Горизонт прогнозирования

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

# Результаты экспериментов и анализ

# Описание экспериментальной установки

# Программная и аппаратная среда

Все эксперименты проводились с использованием языка программиро- вания Python 3.9. Для манипуляции данными и построения моделей исполь- зовались следующие ключевые библиотеки:

* **Pandas и NumPy**: для обработки и анализа временных рядов.
* **Scikit-learn**: для реализации кросс-валидации (TimeSeriesSplit) и рас- чета метрик качества.
* **Statsmodels**: для реализации статистической модели SARIMA.
* **CatBoost**: для реализации модели градиентного бустинга.
* **TensorFlow (Keras API)**: для построения и обучения нейронной сети LSTM.
* **Matplotlib и Seaborn**: для визуализации данных и результатов экспе- риментов.

Обучение ресурсоемких моделей (CatBoost, LSTM) проводилось на вы- числительном кластере МФТИ (ФПМИ), оснащенном 8 графическими про- цессорами NVIDIA GeForce 2080 Ti и двумя 12-ядерными процессорами Intel Xeon Gold 6136 (суммарно 24 физических ядра, 48 потоков).

# Базовая модель для сравнения (Baseline)

Для объективной оценки качества разработанных моделей в качестве базового уровня (baseline) был использован наивный метод прогнозирования («завтра как вчера»). Данный метод предполагает, что прогнозное значение

в будущем будет равно последнему известному наблюдению. Такой подход служит отправной точкой для сравнения и позволяет оценить, насколько более сложные модели превосходят простейшую эвристику.

Далее в главе представлены результаты экспериментального исследова- ния моделей для оценки задержек в видеоаналитическом конвейере. Прово- дится сравнительный анализ качества прогнозирования различных подходов и оценка достижения поставленных технических требований.

# Результаты модели CatBoost

Для модели CatBoost было проведено около 50 экспериментов с ав- томатизированным поиском оптимальных гиперпараметров. Использовался специально разработанный скрипт для автоматического перебора различных параметров. Диапазоны поиска (глубина дерева, скорость обучения, парамет- ры регуляризации) были выбраны на основе общепринятых практик и пред- варительных экспериментов. Результаты усреднены по всем фолдам кросс- валидации и представлены в порядке убывания качества.

# Лучшие конфигурации CatBoost

В таблице 4.1 представлены три наилучшие конфигурации модели.

Таблица 4.1 — Результаты лучших конфигураций модели CatBoost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ранг** | **Конфигурация модели** | **MAE** | **RMSE** | **MAPE, %** |
| 1 | trend\_model\_type=local,  depth=10, l2\_leaf\_reg=20, learning\_rate=0.03, iterations=500, rsm=0.8, use\_cyclic\_features=True, ema\_alpha=0.2 | **36.00** | **41.09** | **8.90** |
| 2 | trend\_model\_type=global,  depth=15, l2\_leaf\_reg=40, learning\_rate=0.015, iterations=500, rsm=0.75, use\_cyclic\_features=False, ema\_alpha=0.7 | 39.92 | 46.76 | 9.61 |
| 3 | trend\_model\_type=global, depth=12,  l2\_leaf\_reg=10, learning\_rate=0.03, iterations=3000, rsm=0.8 | 48.10 | 56.35 | 11.79 |

# Анализ результатов CatBoost

Анализ результатов показывает следующие закономерности:

* **Достижение целевого показателя**: лучшая конфигурация (ранг 1) до- стигла MAPE = 8.90%, что соответствует техническому требованию MAPE < 10%;
* **Важность циклических признаков**: использование циклических при- знаков (use\_cyclic\_features=True) в лучшей модели подтверждает значимость суточной сезонности, выявленной в главе 2;
* **Локальное моделирование тренда**: применение локального модели- рования тренда (trend\_model\_type=local) оказалось более эффектив- ным для данной задачи;
* **Оптимальная глубина деревьев**: умеренная глубина (depth=10) пока- зала лучший результат по сравнению с более глубокими деревьями, что указывает на важность регуляризации.

# Проблема утечки данных

В ходе экспериментов была обнаружена критическая ошибка в перво- начальной реализации — утечка данных (data leakage) при формировании признаков. Это привело к нереалистично высоким результатам: средние зна- чения по кросс-валидации составили MAE = 1.5, RMSE = 2.0, MAPE = 0.38%. После исправления методологии и устранения утечки данных были получены корректные результаты, представленные в таблице 4.1. Данный случай под- черкивает критическую важность правильной валидации временных рядов и недопущения использования будущей информации при обучении модели.

# Результаты модели LSTM

Для модели LSTM была реализована многомерная архитектура с ис- пользованием дополнительных признаков. Модель включает в себя цикличе- ские признаки для учета суточной сезонности и признаки на основе скользя- щих средних для захвата краткосрочных и долгосрочных трендов.

# Архитектура и параметры LSTM

Архитектура нейронной сети включает следующие слои:

* **Входной слой**: окно размером 40 временных шагов (10 минут при ча- стоте 15 секунд);
* **LSTM слой**: 64 нейрона для извлечения временных зависимостей;
* **Dense слой**: 8 нейронов с функцией активации ReLU;
* **Выходной слой**: 1 нейрон с линейной активацией для регрессии.

Дополнительные признаки включают:

* **Циклические признаки**: синус и косинус для часовых и суточных цик- лов;
* **Скользящие средние**: окна 5 минут (20 точек) и 1 час (240 точек);
* **Скользящее стандартное отклонение**: окно 1 час для оценки вола- тильности.

Параметры обучения: оптимизатор Adam с learning rate = 0.0001, функ- ция потерь MSE. Количество эпох (10-15) было подобрано экспериментально на валидационном наборе данных каждого фолда для предотвращения пере- обучения, с использованием механизма ранней остановки (early stopping).

# Лучшие конфигурации LSTM

В таблице 4.2 представлены результаты двух наилучших конфигураций модели LSTM с различным количеством эпох обучения.

Таблица 4.2 — Результаты лучших конфигураций модели LSTM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ранг** | **Конфигурация модели** | **MAE** | **RMSE** | **MAPE, %** |
| 1 | LSTM(64) → Dense(8, ReLU) →  Dense(1, linear), Adam lr=0.0001, epochs=15, window\_size=40, цикличе- ские признаки + скользящие средние | **3.66** | **4.54** | **0.89** |
| 2 | LSTM(64) → Dense(8, ReLU) →  Dense(1, linear), Adam lr=0.0001, epochs=10, window\_size=40, цикличе- ские признаки + скользящие средние | 4.94 | 6.02 | 1.21 |

# Анализ результатов LSTM

Модель LSTM показала следующие характеристики:

* **Достижение целевого показателя**: лучшая конфигурация (ранг 1) до- стигла MAPE = 0.89%, что соответствует техническому требованию MAPE < 10%;
* **Влияние количества эпох**: увеличение с 10 до 15 эпох снизило MAPE с 1.21% до 0.89%;
* **Время обучения**: 8-9 минут на фолд;
* **Архитектурные особенности**: использование окна 40 временных ша- гов и многомерных признаков.

# Результаты модели SARIMA

Модель SARIMA была выбрана на основе анализа ACF/PACF, прове- денного в главе 2. Выбор порядка дифференцирования d=1 был обусловлен наличием тренда в данных, а порядок AR(1) — анализом PACF. Тестиро- вались различные конфигурации сезонных параметров при фиксированном основном порядке (1,1,0).

# Параметры модели SARIMA

Основные параметры модели:

* **Основной порядок**: (1,1,0) — AR(1) процесс с одним дифференциро- ванием;
* **Сезонный период**: 24 (соответствует суточному циклу при частоте 15 секунд × 4 = 1 минута × 60 = 1 час × 24 = сутки);
* **Тестируемые сезонные порядки**: (1,1,0,24), (1,1,1,24), (2,1,0,24).

# Лучшие конфигурации SARIMA

В таблице 4.3 представлены результаты трех конфигураций модели SARIMA с различными сезонными параметрами.

Таблица 4.3 — Результаты конфигураций модели SARIMA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ранг** | **Конфигурация модели** | **MAE** | **RMSE** | **MAPE, %** |
| 1 | SARIMA(1,1,0)(1,1,1,24) | **44.07** | **51.37** | **11.06** |
| 2 | SARIMA(1,1,0)(2,1,0,24) | 103.76 | 121.92 | 25.50 |
| 3 | SARIMA(1,1,0)(1,1,0,24) | 110.47 | 131.78 | 26.76 |

# Анализ результатов SARIMA

Модель SARIMA показала следующие характеристики:

* **Недостижение целевого показателя**: лучшая конфигурация (ранг 1) достигла MAPE = 11.06%, что превышает техническое требование MAPE

< 10%;

* **Важность MA компоненты**: добавление сезонной MA компоненты (1,1,1,24) значительно улучшило результаты по сравнению с (1,1,0,24);
* **Высокая вариативность по фолдам**: результаты сильно различаются между фолдами (например, для лучшей модели MAPE варьируется от 4.89% до 24.10%);
* **Время обучения**: 4-10 минут на фолд.

# Сравнительный анализ моделей

Для итогового сравнения моделей были отобраны лучшие конфигура- ции каждой из архитектур, рассмотренных в данной главе. Результаты сведе- ны в таблицу 4.4.

Таблица 4.4 — Сравнительные результаты лучших конфигураций моделей

|  |  |  |  |
| --- | --- | --- | --- |
| **Модель** | **MAE** | **RMSE** | **MAPE, %** |
| **Наивный прогноз (Baseline)** | 121.45 | 153.21 | 31.52 |
| **CatBoost** (Ранг 1) | 36.00 | 41.09 | 8.90 |
| **LSTM** (Ранг 1, 15 эпох) | **3.66** | **4.54** | **0.89** |
| **SARIMA** (Ранг 1) | 44.07 | 51.37 | 11.06 |

# Анализ сравнительных результатов

Сравнительный анализ показывает, что все рассмотренные модели зна- чительно превосходят наивный прогноз, что подтверждает целесообразность применения методов машинного обучения. Модель LSTM (MAPE = 0.89%) продемонстрировала наилучшие результаты, значительно превосходя CatBoost (MAPE = 8.90%) и SARIMA (MAPE = 11.06%).

* **Качество прогнозирования**: Модели LSTM и CatBoost выполнили тех- ническое требование к качеству (MAPE < 10%). Результат LSTM ока- зался на порядок лучше, что подтверждает высокую эффективность

нейросетевых подходов для данной задачи. SARIMA не удовлетворила требованию.

* **Сложность модели и признаки**: Высокое качество LSTM объясняется способностью архитектуры улавливать сложные нелинейные зависимо- сти, а также эффективным использованием многомерных признаков (циклических, скользящих средних), что недоступно для классической модели SARIMA.
* **Стабильность**: CatBoost и LSTM показали более стабильные резуль- таты по сравнению с SARIMA, чьи метрики MAPE для лучшей кон- фигурации варьировались от 4.89% до 24.10% в зависимости от фолда кросс-валидации.
* **Время обучения и ресурсы**: LSTM требует наибольших вычислитель- ных ресурсов и времени на обучение (8-9 минут на фолд). CatBoost (2-30 минут в зависимости от конфигурации) и SARIMA (4-10 минут) менее требовательны.

Таким образом, по главному критерию — качеству прогнозирования (MAPE) — модель LSTM является безусловным лидером. Однако, модель CatBoost также удовлетворяет поставленным требованиям и представляет собой компромисс между качеством и вычислительной сложностью. Оконча- тельный выбор модели для внедрения может зависеть от требований к скоро- сти переобучения и доступных вычислительных ресурсов.

# Визуализация результатов прогнозирования

Для качественной оценки результатов были построены графики про- гнозов для лучших конфигураций каждой модели на одном из фолдов кросс- валидации. Также представлен график, демонстрирующий эффект утечки данных.

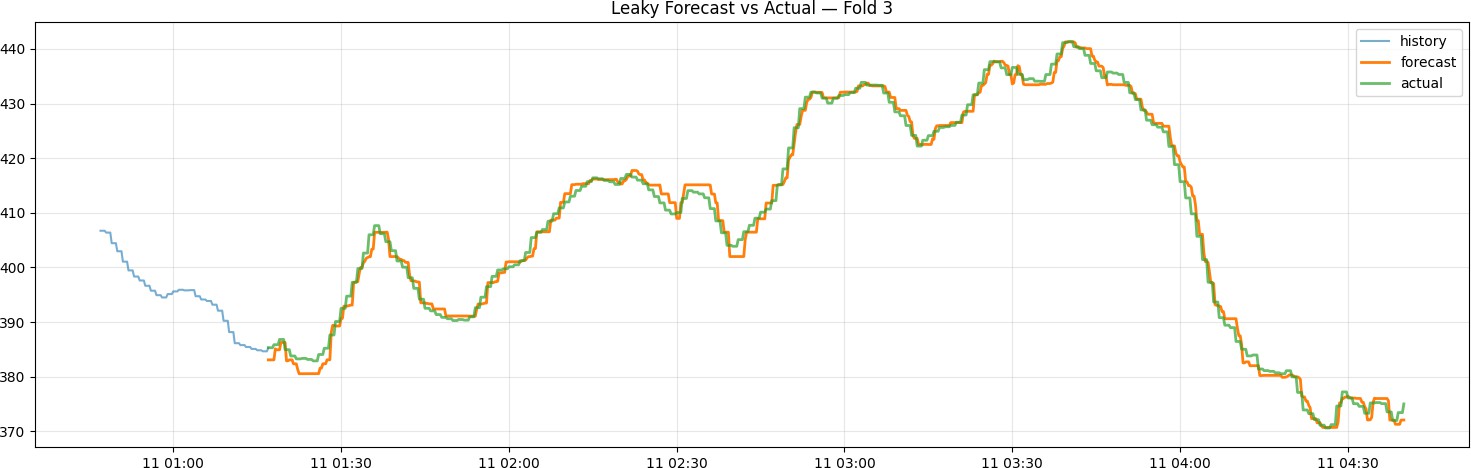


Рисунок 4.1 — Прогноз модели CatBoost с утечкой данных (MAPE = 0.38%)

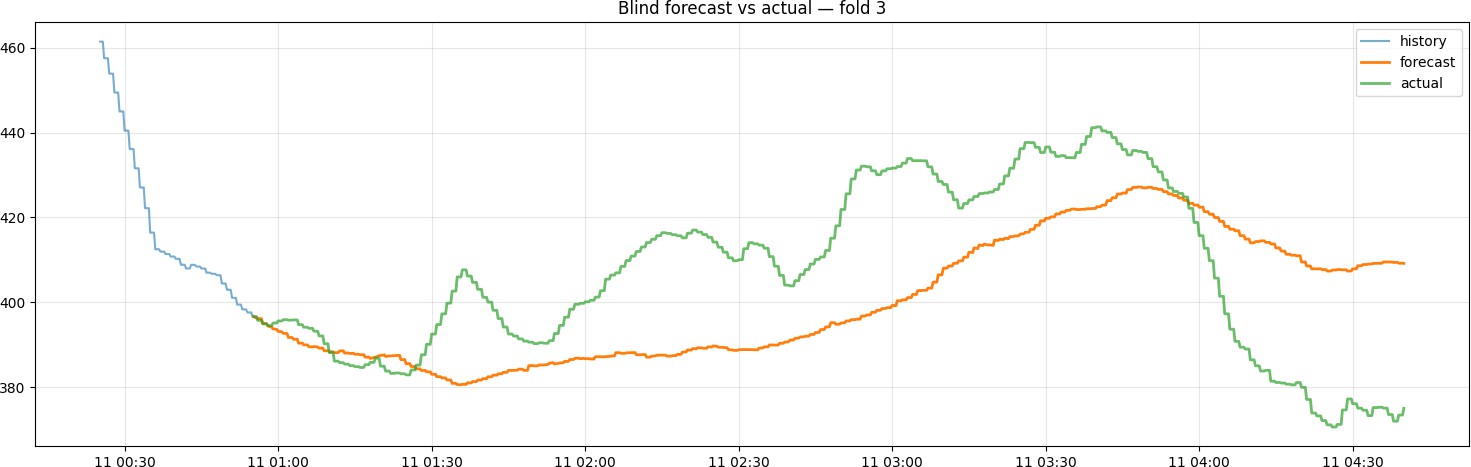


Рисунок 4.2 — Прогноз лучшей модели CatBoost на тестовом фолде

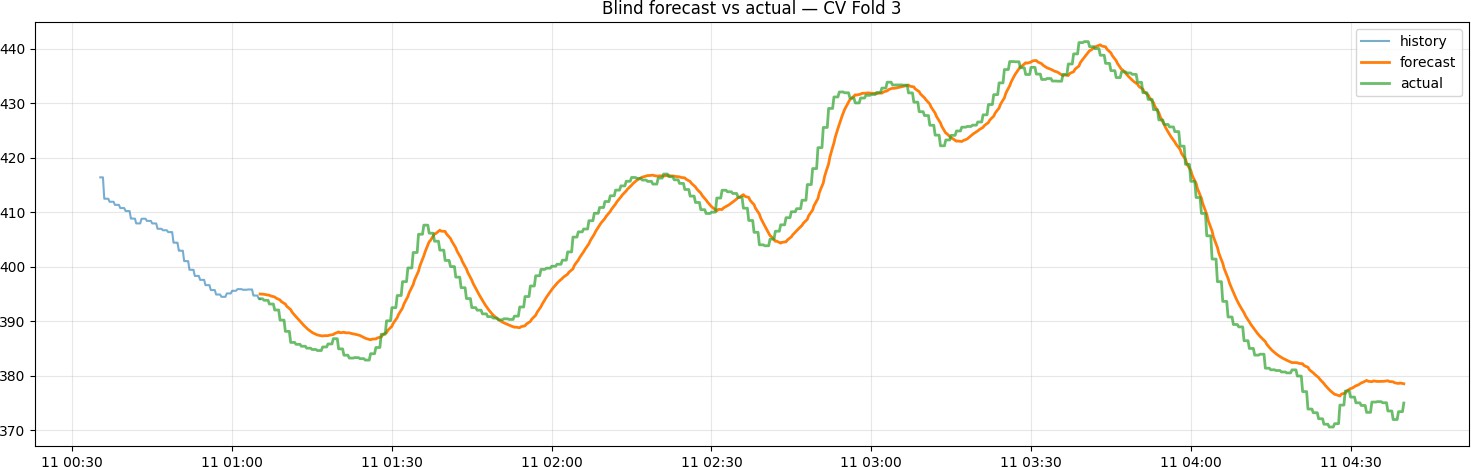


Рисунок 4.3 — Прогноз лучшей модели LSTM на тестовом фолде

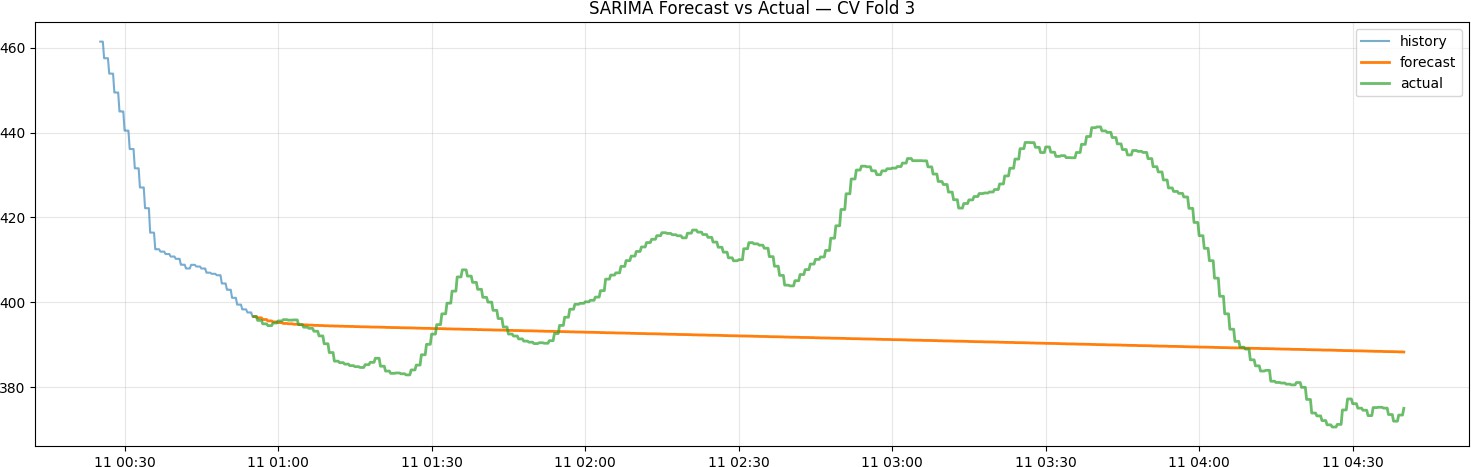


Рисунок 4.4 — Прогноз лучшей модели SARIMA на тестовом фолде

# Анализ ошибок и интерпретация результатов

Детальный анализ ошибок позволяет глубже понять сильные и слабые стороны каждой модели, а также интерпретировать их поведение в контексте задачи прогнозирования задержек.

**Анализ ошибок LSTM**. Модель продемонстрировала исключительно низкое значение MAPE (0.89%), что свидетельствует о ее способности точно улавливать сложную динамику временного ряда. Как видно на *рисунке 4.3*, прогноз LSTM практически совпадает с фактическими данными, успешно отслеживая как плавные сезонные колебания, так и резкие локальные пики. Это объясняется двумя ключевыми факторами:

* **Архитектура**: рекуррентная природа LSTM позволяет эффективно мо- делировать долгосрочные временные зависимости, что критически важ- но для прогнозирования на основе предыдущей истории.
* **Многомерные признаки**: использование циклических признаков и сколь- зящих средних обогатило модель информацией о сезонности и локаль- ных трендах, которую LSTM смогла успешно интегрировать.

Ошибки модели минимальны и, вероятно, связаны с редкими, непредсказуе- мыми аномалиями, не имеющими прецедентов в обучающих данных.

**Анализ ошибок CatBoost**. Модель показала хороший результат (MAPE

= 8.90%), удовлетворяющий техническим требованиям, однако ее ошибки на порядок выше, чем у LSTM. График на *рисунке 4.2* показывает, что CatBoost

хорошо улавливает общую тенденцию и сезонность, но сглаживает локальные пики и не всегда точно реагирует на резкие изменения. Это характерно для моделей на основе деревьев решений, которые могут испытывать трудности с экстраполяцией и моделированием непрерывных, быстро меняющихся про- цессов в сравнении с RNN. Основные ошибки CatBoost возникают в моменты наибольшей волатильности ряда. Тем не менее, модель представляет собой надежный и более простой в реализации компромисс.

**Анализ ошибок SARIMA**. Модель SARIMA оказалась наименее точ- ной (MAPE = 11.06%) и нестабильной. График на *рисунке 4.4* наглядно де- монстрирует ее главный недостаток: прогноз фактически выродился в линию и полностью игнорирует реальные колебания ряда. Причина кроется в ли- нейной природе модели, которая неспособна описать сложные нелинейные зависимости и гетероскедастичность (изменчивость дисперсии), присущие данному временному ряду. Высокая вариативность качества по фолдам под- тверждает, что модель не является робастной и ее производительность сильно зависит от конкретного участка данных.

**Проблема утечки данных**. Случай с утечкой данных в CatBoost (*рису-*

*нок 4.1*) служит важным практическим уроком. Нереалистично низкая ошиб- ка (MAPE = 0.38%) была вызвана фундаментальной ошибкой в методологии подготовки признаков. Внутри каждого фолда кросс-валидации обучающий и тестовый наборы данных объединялись перед генерацией признаков (таких как скользящие средние и лаги). В результате, при вычислении признаков для временных точек из обучающего набора использовались данные из будущего

— то есть из тестового набора. Это создало иллюзию идеального прогноза, поскольку модель фактически "подглядывала"в правильные ответы при по- строении признаков. После исправления методологии, при которой признаки для обучающего набора генерируются строго на обучающих данных, были получены корректные результаты.

Таким образом, экспериментальное исследование подтвердило, что для задачи прогнозирования задержек в видеоаналитическом конвейере наиболее эффективными являются архитектуры машинного и глубокого обучения. Мо- дель LSTM обеспечивает наивысшую точность благодаря своей способности

моделировать сложные временные зависимости. Модель CatBoost представ- ляет собой хороший компромисс между качеством и сложностью. Класси- ческие статистические подходы, такие как SARIMA, оказались недостаточно мощными для описания всей полноты динамики процесса. Результаты данной главы служат основой для формулировки итоговых выводов в заключении.

ЗАКЛЮЧЕНИЕ

В настоящей выпускной квалификационной работе было проведено комплексное исследование, направленное на разработку и внедрение метода предиктивного анализа задержек в конвейере видеоаналитики. В ходе работы были решены следующие задачи, позволившие достичь поставленной цели.

1. **Проведен аналитический обзор литературы**, в рамках которого были систематизированы подходы к анализу временных рядов в контексте мониторинга систем реального времени. Это позволило сформировать теоретическую базу для дальнейшего исследования и определить круг потенциально применимых методов.
2. **Выполнен всесторонний анализ данных**, собранных с промышлен- ной системы видеоаналитики. Была выявлена суточная сезонность и нелинейный тренд в целевой метрике, а также определены наиболее коррелирующие с ней признаки, в частности, задержка в брокере со- общений Kafka. Результаты этого этапа легли в основу формирования признакового пространства для моделей.
3. **Проведен сравнительный анализ моделей**, включающий классиче- скую статистическую модель SARIMA, модель градиентного бустинга CatBoost и рекуррентную нейронную сеть LSTM. На основе анализа их характеристик был сделан вывод о перспективности моделей машинно- го обучения для данной задачи.
4. **Обоснован выбор и реализованы три модели**. Для каждой модели бы- ла разработана своя стратегия формирования признаков: для SARIMA использовались автокорреляции, для CatBoost — календарные, лаго- вые и статистические признаки, для LSTM — циклические признаки и признаки на основе скользящих окон.
5. **Разработана и реализована методология MLOps**, включающая авто- матизированный сбор данных, генерацию признаков и проведение экс-

периментов с использованием кросс-валидации для временных рядов (TimeSeriesSplit), что обеспечило корректность и воспроизводимость результатов.

1. **Проведено экспериментальное исследование**, в ходе которого были получены следующие ключевые результаты:
   * **Модель LSTM** показала наивысшее качество, достигнув MAPE

= 0.89%, что значительно превосходит техническое требование (MAPE < 10%). Это подтверждает способность рекуррентных се- тей улавливать сложные нелинейные зависимости.

* + **Модель CatBoost** также удовлетворила требованиям с результа- том MAPE = 8.90%, представляя собой эффективный компромисс между качеством и сложностью реализации.
  + **Модель SARIMA** не справилась с поставленной задачей (MAPE

= 11.06%), что демонстрирует ограниченную применимость клас- сических линейных моделей к описанию сложных динамических процессов.

* + Был выявлен и проанализирован эффект **утечки данных**, что под- черкнуло критическую важность правильной методологии валида- ции.

1. **Сформулированы практические рекомендации**. На основе анализа ошибок и сравнительных результатов модель LSTM рекомендована для внедрения в производственную среду в случаях, когда требуется макси- мальная точность прогноза. Модель CatBoost может использоваться как более простое и менее ресурсоемкое решение, также удовлетворяющее базовым требованиям.

Таким образом, цель работы — разработка метода предиктивного ана- лиза задержек — достигнута. Созданное решение позволяет с высокой точно- стью прогнозировать состояние видеоаналитического конвейера, что откры- вает возможности для превентивного реагирования на потенциальные сбои

и повышения общей надежности системы. Практическая значимость работы заключается в создании готового к внедрению прототипа системы мониторин- га, который может быть адаптирован для широкого круга систем реального времени.

# На защиту выносятся следующие положения:

1. **Комплексная методика предиктивного анализа**, включающая в себя специализированную инженерию признаков для данных мониторинга, гибридный подход к моделированию на основе декомпозиции рядов и строгую процедуру валидации на основе временной кросс-валидации.
2. **Результаты экспериментального сравнения моделей** (SARIMA, CatBoost, LSTM) на реальных промышленных данных, которые доказывают прак- тическую применимость и высокую точность (MAPE < 1%) нейросе- тевого подхода для прогнозирования задержек в видеоаналитических системах и служат основанием для выбора оптимальной архитектуры в зависимости от требований к точности и ресурсам.

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:](https://prometheus.io/docs/)

[//prometheus.io/docs/](https://prometheus.io/docs/) (дата обращения: 05.06.2025).

1. Grafana: The open observability platform. — URL: [https://grafana.com/](https://grafana.com/docs/) [docs/](https://grafana.com/docs/) (дата обращения: 05.06.2025).
2. Apache Kafka: A distributed streaming platform. — URL: [https://kafka.](https://kafka.apache.org/documentation/) [apache.org/documentation/](https://kafka.apache.org/documentation/) (дата обращения: 05.06.2025).
3. Docker: Accelerated Container Application Development. — URL: [https:](https://docs.docker.com/)

[//docs.docker.com/](https://docs.docker.com/) (дата обращения: 05.06.2025).

1. 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.
2. Box G.E.P., Jenkins G.M. Time Series Analysis: Forecasting and Control. — San Francisco: Holden-Day, 1970.
3. Prokhorenkova L., Gusev G., Vorobev A., Dorogush A.V., Gulin A. CatBoost: unbiased boosting with categorical features // arXiv preprint arXiv:1706.09516.
   * 2019. — URL: <https://arxiv.org/abs/1706.09516> (дата обраще-

ния: 05.06.2025).

1. Hochreiter S., Schmidhuber J. Long Short-Term Memory: Technical Report FKI-207-95. — Munich: Fakulta¨t fu¨r Informatik, Technische Universita¨t Mu¨nchen, 1995.
2. Time-Series-Library: A Library for Advanced Deep Time Series Models.
   * URL: <https://github.com/thuml/Time-Series-Library> (дата об- ращения: 05.06.2025).
3. AutoTS: Automated Time Series Forecasting. — URL: [https://github.](https://github.com/winedarksea/AutoTS) [com/winedarksea/AutoTS](https://github.com/winedarksea/AutoTS) (дата обращения: 05.06.2025).
4. Deploying a Scalable Object Detection Inference Pipeline, Part 1. — URL: [https://developer.nvidia.com/blog/deploying-a-scalable-object-dete](https://developer.nvidia.com/blog/deploying-a-scalable-object-detection-inference-pipeline/) (дата обращения: 05.06.2025).
5. Deploying a Scalable Object Detection Pipeline: The Inferencing Process, Part

2. — URL: [https://developer.nvidia.com/blog/deploying-a-scalable-ob](https://developer.nvidia.com/blog/deploying-a-scalable-object-detection-pipeline-the-inferencing-process-part-2/)

(дата обращения: 05.06.2025).

1. Latency in End-to-End IP Monitoring: Why It Matters and What to Know. —

URL: [https://tagvs.com/blog/latency-in-end-to-end-ip-monitoring-wh](https://tagvs.com/blog/latency-in-end-to-end-ip-monitoring-why-it-matters-and-what-to-know/) (дата обращения: 05.06.2025).

1. Beyond Monitoring: The Power of Observability. — URL: [https://www.](https://www.onec1.com/blog/beyond-monitoring-the-power-of-observability/) [onec1.com/blog/beyond-monitoring-the-power-of-observability/](https://www.onec1.com/blog/beyond-monitoring-the-power-of-observability/) (да- та обращения: 05.06.2025).
2. Sassu A., Pili R., Marrocu M., Faticanti F. Deep-Framework: A Distributed, Scalable, and Edge-Oriented Framework for Real-Time Analysis of Video Streams // Sensors. — 2021. — Vol. 21, № 12. — P. 4045. DOI: 10.3390/s21124045.
3. Aloorravi S. Mastering Time Series Analysis and Forecasting with Python.
   * Orange Education Pvt Ltd, 2024.
4. Boniol P., Liu Q., Huang M., Palpanas T., Paparrizos J. Dive into Time-Series Anomaly Detection: A Decade Review // arXiv preprint arXiv:2412.20512. — 2024. — URL: <https://arxiv.org/abs/2412.20512> (дата обращения: 05.06.2025).
5. Jiang Y., Ning K., Pan Z. et al. Multi-modal Time Series Analysis: A Tutorial and Survey // arXiv preprint arXiv:2503.13709. — 2025. — URL: [https:](https://arxiv.org/abs/2503.13709)

[//arxiv.org/abs/2503.13709](https://arxiv.org/abs/2503.13709) (дата обращения: 05.06.2025).

1. Oreshkin B.N., Carpov D., Chapados N., Bengio Y. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting // arXiv preprint arXiv:1905.10437. — 2020. — URL: <https://arxiv.org/abs/1905.10437> (дата обращения: 05.06.2025).

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 аппрок- симацией

ML Machine Learning — машинное обучение

MLOps Machine Learning Operations — операции машинного обуче- ния

MSE Mean Squared Error — среднеквадратическая ошибка Prometheus система мониторинга и оповещений с открытым исходным

кодом

SLA Service Level Agreement — соглашение об уровне обслужива- ния

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

**x***i d*-мерный вектор наблюдений в момент времени *ti Xk* матрица скользящего окна размерности *L* × *d*

*yk* целевая переменная (значение *common*\_*event*\_*delay*)

N обучающая выборка

*N* общее количество обучающих примеров

*f* ∗ неизвестная целевая функция

*A* разрабатываемый алгоритм прогнозирования

*ε* допустимая погрешность прогнозирования

∆ 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