Некоммерческое акционерное общество «Алматинский Университет Энергетики и Связи имени Гумарбека Даукеева»

УДК 621.39

На правах рукописи

**ЖУНУСОВ АЯН РАДИЯНОВИЧ**

# Разработка метода мониторинга качества сервисов в телекоммуникационной сети

6D071900 – Радиотехника, электроника и телекоммуникации

Диссертация на соискание ученой степени доктора философии (PhD)

Научный консультант:
к.т.н., профессор Байкенов А.С.
Зарубежные научные консультанты:
PhD, профессор Илиева Д.И.
PhD, профессор Чайко Е.В.

Республика Казахстан
Алматы, 2026

## Содержание

Обозначения и сокращения

По алфавиту

**ACI** — Application Centric Infrastructure (архитектура программно-определяемой сети Cisco)
**AC** — Access Concentrator (узел доступа, BRAS/BNG)
**API** — Application Programming Interface (интерфейс прикладного программирования)
**Availability** — доступность сервиса
**B4** — внутренняя SDN-архитектура магистральной сети Google
**Best-effort** — модель обслуживания трафика без гарантированных параметров качества
**BNG** — Broadband Network Gateway (широкополосный сетевой шлюз)
**BRAS** — Broadband Remote Access Server (сервер широкополосного доступа)
**BW** — Bandwidth (пропускная способность)
**CLI** — Command Line Interface (интерфейс командной строки)
**Cloud** — облачная вычислительная инфраструктура
**CNN** — Convolutional Neural Network (свёрточная нейронная сеть)
**CNF** — Cloud-Native Network Function (облачная сетевая функция)
**DBSCAN** — Density-Based Spatial Clustering of Applications with Noise (кластеризация на основе плотности)
**DC** — Data Center (центр обработки данных)
**DCN** — Data Center Network (сеть центра обработки данных)
**DL** — Downlink (нисходящее направление передачи данных)
**Domain 2.0** — архитектура программно-определяемой сети оператора AT&T
**DSL** — Digital Subscriber Line (цифровая абонентская линия)
**DSLAM** — Digital Subscriber Line Access Multiplexer
**E2E** — End-to-End (сквозное соединение)
**EFK** — Elasticsearch, Fluentd, Kibana (стек обработки логов)
**ELK** — Elasticsearch, Logstash, Kibana (стек анализа логов)
**E-model** — алгоритмическая модель оценки качества голоса ITU-T G.107
**GBDT** — Gradient Boosting Decision Tree
**gNMI** — gRPC Network Management Interface
**GPB** — Google Protocol Buffers
**GRPC** — Google Remote Procedure Call
**GRU** — Gated Recurrent Unit (рекуррентная нейронная сеть)
**IMS** — IP Multimedia Subsystem (подсистема мультимедийных сервисов)
**In-band OAM** — эксплуатационный мониторинг в пользовательском трафике
**INT** — In-band Network Telemetry (телеметрия в полосе передачи данных)
**IOAM** — In-situ Operations, Administration and Maintenance
**IPFIX** — IP Flow Information Export

**IOS XE** — сетевая операционная система Cisco
**ISP** — Internet Service Provider (интернет-провайдер)
**ITU-T** — International Telecommunication Union – Telecommunication Standardization Sector
**Jaeger** — система распределённой трассировки
**K** — коэффициент нестабильности PPPoE-сессий
**Kubernetes** — платформа оркестрации контейнерных приложений
**L2–L7** — уровни модели OSI от канального до прикладного
**LASSO** — Least Absolute Shrinkage and Selection Operator
**Latency** — задержка передачи данных
**LCP** — Link Control Protocol
**LSTM** — Long Short-Term Memory (рекуррентная нейронная сеть)
**MDT** — Model-Driven Telemetry (модельно-ориентированная телеметрия
**ML** — Machine Learning (машинное обучение)
**mmWave** — миллиметровый диапазон радиочастот
**mMTC** — massive Machine Type Communications (массовые машинные коммуникации)
**MOS** — Mean Opinion Score (субъективная оценка качества)
**MPLS** — Multiprotocol Label Switching
**MSE** — Mean Squared Error (среднеквадратичная ошибка)
**MTTR** — Mean Time to Repair (среднее время восстановления)
**NFV** — Network Function Virtualization
**NMS** — Network Management System
**NOC** — Network Operations Center (центр управления сетью)
**ONAP** — Open Network Automation Platform
**OpenConfig** — открытые модели данных для управления сетевыми устройствами
**OSS** — Operations Support System
**Overlay** — логическая сеть поверх физической инфраструктуры
**PADI** — PPPoE Active Discovery Initiation
**PADO** — PPPoE Active Discovery Offer
**PADR** — PPPoE Active Discovery Request
**PADS** — PPPoE Active Discovery Session-confirmation
**PADT** — PPPoE Active Discovery Terminate
**PE** — Provider Edge (пограничный маршрутизатор)
**Polling** — периодический опрос сетевых устройств
**PPPoE** — Point-to-Point Protocol over Ethernet
**PSNR** — Peak Signal-to-Noise Ratio (пиковое отношение сигнал/шум)
**QoE** — Quality of Experience
**QoS** — Quality of Service
**R-factor** — интегральный показатель качества в E-model

**$R^2$** — коэффициент детерминации
**RAN** — Radio Access Network (радиодоступная сеть)
**RCA** — Root Cause Analysis (анализ первопричин отказов)
**RFC** — Request for Comments (стандарты IETF)
**SDN** — Software-Defined Networking
**SD-WAN** — Software-Defined Wide Area Network
**SFC** — Service Function Chaining
**SLA** — Service Level Agreement
**Slicing** — технология сетевых срезов
**SNMP** — Simple Network Management Protocol
**SRE** — Site Reliability Engineering (инженерия надёжности сервисов)
**SSIM** — Structural Similarity Index (индекс структурного сходства)
**Streaming Telemetry** — потоковая телеметрия
**TCN** — Temporal Convolutional Network
**TCP/IP** — Transmission Control Protocol / Internet Protocol
**Telemetry** — телеметрия
**URLLC** — Ultra-Reliable Low-Latency Communication
**Uptime** — время непрерывной работы системы
**VLAN** — Virtual Local Area Network
**VMAF** — Video Multi-Method Assessment Fusion (метрика качества видео)
**VNF** — Virtual Network Function
**VoIP** — Voice over IP
**WAN** — Wide Area Network

**Введение**

Современные телекоммуникационные сети функционируют в условиях стремительного роста объёмов передаваемого трафика, усложнения архитектур и увеличения разнообразия предоставляемых сервисов. Развитие широкополосного доступа, облачных платформ, программно-определяемых сетей (SDN), виртуализации сетевых функций (NFV), а также мультисервисных и распределённых инфраструктур приводит к существенному усложнению задач управления и обеспечения качества обслуживания. В этих условиях традиционные методы мониторинга качества, ориентированные на прямые измерения сетевых параметров, оказываются недостаточно эффективными для своевременного выявления деградаций качества и скрытых аномалий в работе сети [1], [22], [34].

Исторически мониторинг телекоммуникационных сетей основывался на периодическом опросе оборудования с использованием протокола SNMP и анализе агрегированных интерфейсных счётчиков [2], [3]. Несмотря на широкое распространение и простоту реализации, данный подход обладает рядом фундаментальных ограничений, включая низкую временную разрешающую способность, зависимость от интервалов опроса и неспособность фиксировать кратковременные нарушения качества обслуживания [4]. Эти ограничения особенно критичны в современных сетях, где деградации качества могут носить кратковременный или локальный характер и не сопровождаться значительными изменениями традиционных QoS-метрик.

С целью повышения наблюдаемости сетей получили развитие потоковые технологии мониторинга, такие как NetFlow, IPFIX и sFlow, позволяющие анализировать структуру трафика и поведение потоков на уровне сетевых соединений [5–15]. Однако и данные подходы не всегда обеспечивают достаточную информативность для оценки качества пользовательских сервисов, особенно в сетях доступа и агрегации, где ключевые нарушения проявляются на уровне пользовательских сессий и их динамики [14], [15]. Дополнительное усложнение задачи мониторинга связано с переходом к потоковой телеметрии и модельно-ориентированным протоколам (MDT, gNMI), которые значительно увеличивают объёмы собираемых данных и предъявляют повышенные требования к их обработке и интерпретации [17–21], [59].

Параллельно с эволюцией сетевых технологий произошло смещение фокуса оценки качества от исключительно сетевых параметров (QoS) к показателям качества пользовательского восприятия (QoE). Показано, что даже при допустимых значениях задержек, потерь и пропускной способности пользователь может испытывать деградацию качества сервиса, что особенно характерно для мультимедийных, интерактивных и реального времени

приложений [1], [22], [26], [34]. В современных сетях пятого поколения и мультисервисных инфраструктурах необходимость интеграции QoS- и QoE-подходов является ключевым требованием для выполнения SLA и обеспечения устойчивости сервисов [23], [29], [30].

В последние годы для решения указанных проблем всё более активно применяются методы машинного обучения, позволяющие анализировать большие массивы разнородных данных, выявлять скрытые зависимости между параметрами сети и выполнять прогнозирование деградаций качества [36]. Результаты многочисленных исследований показывают, что методы машинного обучения превосходят классические статистические подходы в задачах обнаружения аномалий, прогнозирования сетевых метрик и корреляции событий, особенно в условиях высокой динамичности и гетерогенности современных сетей [36], [38–40], [52]. Вместе с тем большинство существующих ML-подходов опирается на прямые сетевые метрики или потоковую телеметрию, что не всегда позволяет адекватно отражать фактическое состояние сервисов в сегментах доступа.

В этой связи особый интерес представляет анализ статистики пользовательских сессий, в частности PPPoE-соединений, как источника косвенных, но высокоинформативных признаков состояния сети. Статистика служебных сообщений протокола PPPoE отражает процессы установления, поддержания и аварийного завершения сессий, что позволяет выявлять деградации качества, не фиксируемые традиционными средствами мониторинга [68–72]. Применение указанных данных обеспечивает реализацию косвенного метода мониторинга, основанного на анализе штатных статистических параметров сети и не требующего внедрения дополнительных измерительных средств либо генерации тестового трафика.

**Актуальность темы:** на основе анализа существующих методов мониторинга качества в телекоммуникационных сетях выявлены ограничения традиционных подходов, основанных на прямых измерениях сетевых параметров (задержка, потери, джиттер), что обусловило необходимость разработки и применения интегрированного показателя качества на основе косвенных статистических данных, уже присутствующих в инфраструктуре операторов связи.

**Целью диссертационной работы является** разработка метода мониторинга качества сервисов в телекоммуникационных сетях на основе анализа косвенных статистических данных.

**Для реализации данной цели были поставлены следующие задачи исследования:**

– исследование методов мониторинга качества сервисов, выявление их преимуществ и ограничений;

– определение набора исходных косвенных признаков и их значимости;

– введение коэффициента нестабильности *K*;

– доказательство корректности аналитического выражения для коэффициента нестабильности *K* с помощью математической статистики и имитационного моделирования;

– сравнение результатов регрессионных и ML-моделей прогнозирования для коэффициента нестабильности *K*;

– оценка практической значимости и экономической эффективности разработанного метода на примере реальной сети оператора.

**Объектом исследования** являются процессы мониторинга качества сервисов в современных телекоммуникационных сетях.

**Предметом исследования** являются методы сбора, анализа и прогнозирования метрик качества сервисов на основе статистических характеристик PPPoE-сессий и алгоритмов машинного обучения.

**Методы исследования:**

- Математическая статистика;
- Методы имитационного моделирования;
- Методы машинного обучения.

**Научная новизна диссертационной работы заключается в следующем:**

1. **Впервые предложен подход** к определению показателя качества сервисов на основе косвенных статистических признаков сетевых протоколов (на примере PPPoE), что позволяет диагностировать и прогнозировать деградацию обслуживания без прямых измерений параметров QoS и без внедрения дополнительного оборудования.
2. **Впервые предложен и аналитически обоснован безразмерный коэффициент нестабильности *K***, отражающий долю аварийных разрывов соединений. Использование статистических показателей позволило обеспечить реализацию **косвенного характера мониторинга** без генерации тестового трафика и без воздействия на пользовательские сервисы.
3. **Разработан новый метод мониторинга качества сервисов на основе коэффициента нестабильности *K***, включающий алгоритмы статистического анализа, что позволило доказать состоятельность и чувствительность предложенного показателя к деградациям качества услуг связи.

**Научные положения, выносимые на защиту:**

1. **Подход к определению** показателя качества сервисов на основе косвенных статистических признаков сетевых протоколов доступа без использования прямых измерений параметров QoS.
2. **Коэффициент нестабильности *K*** является интегральным показателем качества предоставления услуг связи и обеспечивает количественную оценку устойчивости на основе косвенных статистических данных.

3. **Предложенный метод мониторинга на основе коэффициента *K*** обладает статистической состоятельностью и чувствительностью.

**Достоверность и практическая применимость** полученных результатов подтверждается актом внедрения от Республиканского Государственного Предприятия на Праве Хозяйственного Ведения «Казахстанский центр межбанковских расчетов» и информационным письмом от АО «Казахтелеком» (см. соответствующие приложения), полученными на результаты диссертационной работы. Отмечается, что применение системы мониторинга позволило сократить сроки реагирования при возникновении внештатных ситуаций, обеспечило возможность отслеживания и выявления проблемных участков сети, отладку и улучшение качества предоставляемых сервисов, проведение тестирований для оптимизации телекоммуникационной сети.

В работе приводятся результаты экспериментальных исследований, которые выполнялись комплексно, и дублировались расчетами по имитационным моделям. Имитационное моделирование и анализ были проведены в программной среде Python с использованием открытых библиотек: NumPy, Pandas, scikit-learn, matplotlib и gplearn (лицензия MIT). Все расчёты выполнялись локально, в изолированной вычислительной среде с полным контролем над воспроизводимостью. Для анализа зависимостей применялись методы машинного обучения, включая линейную регрессию, гребневую регрессию, Lasso-регрессию, модели случайного леса, опорные векторы, а также методы символьной регрессии, позволяющие получить интерпретируемые аналитические выражения.

**Апробация результатов диссертации:** Результаты исследования докладывались на XII Международной научно-техническая конференции «Энергетика, инфокоммуникационные технологии и высшее образование» 20-21 октября 2022г., а также на 7-й Международной конференции по энергоэффективности и сельскохозяйственной инженерии (EE&AE), 2020. Все выступления по теме диссертации.

**Публикации:** по теме диссертационной работы опубликовано 6 научных работ, включая статьи и доклады. В их числе: 1 статья в отечественном научном издании, рекомендованном Комитетом по обеспечению качества в сфере науки и высшего образования (КОКСОН); 2 научных доклада, опубликованных в сборниках материалов международных научно-технических конференций, включая доклады с очным выступлением; а также 4 публикации, индексируемые в базе данных Scopus, включая 3 статьи в научных журналах и 1 публикацию в сборнике материалов международной конференции. Публикации, индексируемые в базе данных Scopus, представлены статьёй типа Article в журнале Journal of Theoretical and Applied Information Technology с процентилем 17% по предметной области «Общая информатика» на момент публикации, статьёй в журнале Indonesian Journal of Electrical Engineering and Computer Science с процентилем 47% по предметной области «Вычислительные сети и передача данных», статьёй в журнале

Engineering, Technology & Applied Science Research с процентилем 53% по предметной области «Обработка сигналов», а также публикацией в материалах международной научной конференции IEEE (2020 г.), индексируемой в базе данных Scopus.

**Структура и объем диссертации.** Диссертационная работа выполнена автором в соответствии с действующими требованиями оформления, структуры и содержания. Работа состоит из 4 основных разделов, списка условных обозначений и сокращений, введения, заключения, списка использованных источников, который состоит из 95 наименований, и 7 приложений. Общий объем работы составил 96 страниц, из них: 18 рисунков, 8 таблиц.

# Глава 1. Обзор и анализ существующих методов мониторинга качества сервисов

## 1.1 Традиционные подходы к мониторингу телекоммуникационных сетей

Мониторинг качества сервисов исторически являлся ключевым инструментом обеспечения стабильности и управляемости телекоммуникационных сетей. На ранних этапах развития сетевых технологий доминировали простые механизмы опроса устройств, ориентированные преимущественно на сбор базовых эксплуатационных характеристик [1]. Наиболее распространённым протоколом стал SNMP, разработанный в конце 1980-х годов и стандартизированный в RFC 1157, а позднее — в рамках архитектуры SNMPv3 (RFC 3411–3418) [2], [3]. SNMP был спроектирован как лёгкий и универсальный механизм обмена данными между сетевым оборудованием и управляющими системами, что обеспечило ему широкое распространение в корпоративных и операторских сетях [3].

Большим преимуществом SNMP стала его простота: модель управления на основе MIB-объектов позволяла собирать показатели загрузки процессора, статистику интерфейсов, состояние модулей и подсистем, не создавая существенной нагрузки на оборудование [2], [3]. Однако по мере роста скоростей и усложнения сетевых архитектур выявились фундаментальные ограничения подхода с периодическим опросом (polling). Во-первых, управление через SNMP характеризуется жесткой интервальностью измерений: между опросами могут возникать кратковременные перегрузки или сбои, которые не фиксируются системой мониторинга, что приводит к потере критически важного контекста. Во-вторых, при сокращении интервала опроса возрастает нагрузка на управляющие процессоры сетевых устройств, что особенно заметно на устройствах операторского уровня, обслуживающих сотни интерфейсов и тысячи MIB-переменных. В-третьих, SNMP не предоставляет информации о структуре сетевого трафика и не позволяет анализировать соединения или приложения — собираются лишь агрегированные счётчики [4].

Ограниченность подходов, основанных на опросе, привела к появлению потоковых технологий мониторинга, ориентированных на сбор информации о сетевых потоках (flow monitoring) [5]. Первой широко внедрённой реализацией стал Cisco NetFlow версии 5, разработанный в середине 1990-х годов для обеспечения детализации, недоступной через SNMP [6]. NetFlow позволил экспортировать записи о потоках, включающие IP-адреса источника и назначения, транспортные порты, протокол, объём трафика и временные метки начала и конца соединения [6]. Такой формат дал возможность анализировать структуру нагрузки, выявлять аномалии, отслеживать использование полосы пропускания и реализовывать механизмы безопасности [7]. Со временем бизнес-применение NetFlow расширилось: операторы начали использовать потоковые данные для обнаружения DDoS-атак, моделирования перетоков и оптимизации политики QoS [8].

Дальнейшее развитие потоковой телеметрии связано со стандартизацией IPFIX (RFC 7011–7015) — формального преемника NetFlow v9, предоставляющего гибкую модель информационных элементов и расширяемую структуру шаблонов [9]. Преимущество IPFIX состоит в возможности адаптации состава экспортируемых параметров под конкретные сервисы и технологии: от MPLS-меток и GRE-туннелей до элементов виртуализированных функций сети (VNF) [10]. В результате IPFIX стал удобным инструментом для крупных операторских и облачных инфраструктур, где необходимо агрегировать, нормализовать и коррелировать потоковые данные из множества разнородных устройств и доменов, обеспечивая целостное представление о трафике сети [11].

Параллельно с NetFlow/IPFIX получила распространение технология sFlow, использующая аппаратное сэмплирование каждого N-го пакета на высокоскоростных интерфейсах [12]. Основное достоинство sFlow — масштабируемость: поскольку выборка осуществляется на ASIC-уровне, сбор данных практически не нагружает процессор устройства [12], [13]. Это делает технологию востребованной в дата-центрах и системах с трафиком десятков и сотен гигабит в секунду [13]. Однако случайный характер выборки снижает точность анализа низкообъёмных или кратковременных потоков, что ограничивает использование sFlow в сценариях, где требуется строгий контроль SLA или анализ качества мультимедийных сервисов [14], [15], [16].

Рост требований к наблюдаемости сетей привёл к возникновению модель-ориентированной телеметрии (Model-Driven Telemetry, MDT) [17]. В отличие от SNMP, где управляющая станция инициирует сбор данных, MDT основана на подписке на конкретные параметры, определённые через модели YANG, и их потоковой доставке по протоколам типа gNMI или gRPC-dial-out [18]. Такой подход формирует непрерывный поток высокочастотных метрик, устраняя задержки между опросами и обеспечивая точный контроль состава данных [17], [18]. По оценкам исследований Google и AT&T, использование потоковой телеметрии позволяет сократить среднее время обнаружения инцидентов в сети в 5–10 раз по сравнению с SNMP-polling [19]. Дополнительным преимуществом MDT является возможность инкапсуляции данных в структурированный формат (JSON, GPB), что существенно упрощает последующую обработку средствами ML/аналитики [18].

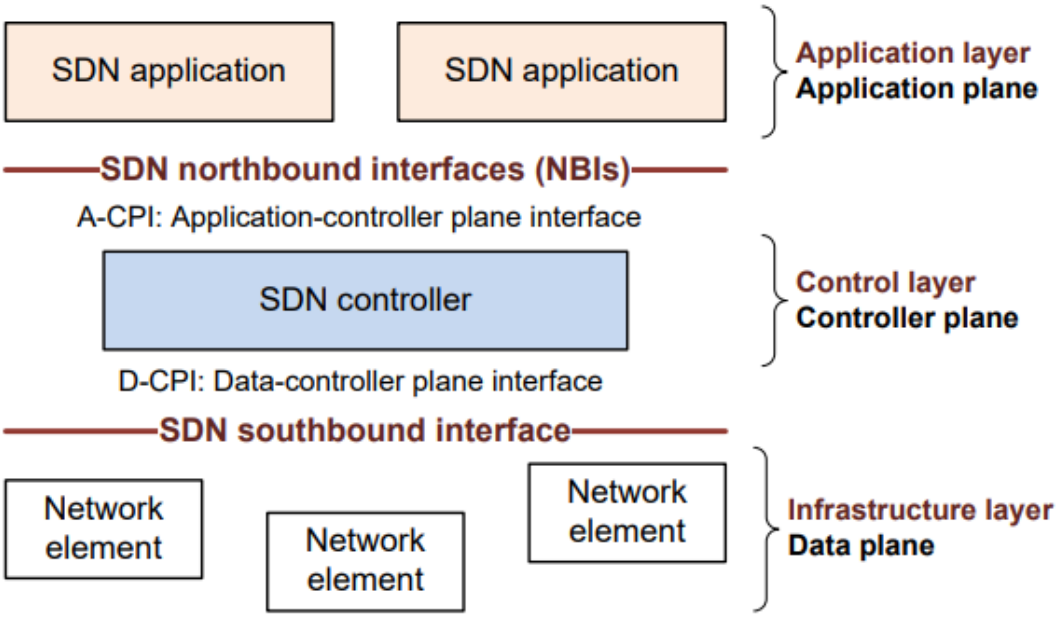


Рисунок 1.1 – Архитектура программно-определяемой сети (SDN) (по данным Open Networking Foundation [20])

На рисунке 1.1 используются следующие обозначения: SDN application — прикладное приложение программно-определяемой сети, реализующее функции управления и оптимизации сетевых сервисов; Application layer / Application plane — прикладной уровень (плоскость приложений) архитектуры SDN, на котором функционируют сетевые приложения; SDN northbound interfaces (NBIs) — северные интерфейсы SDN, обеспечивающие взаимодействие приложений с контроллером; A-CPI (Application–Controller Plane Interface) — интерфейс взаимодействия между плоскостью приложений и плоскостью управления; SDN controller — централизованный контроллер программно-определяемой сети, реализующий логику управления и глобальное представление сети; Control layer / Controller plane — уровень (плоскость) управления, отвечающий за принятие управляющих решений; D-CPI (Data–Controller Plane Interface) — интерфейс взаимодействия между плоскостью управления и плоскостью передачи данных; SDN southbound interface — южный интерфейс SDN, предназначенный для управления сетевыми устройствами; Network element — сетевой элемент (коммутатор, маршрутизатор или иное устройство передачи данных); Infrastructure layer / Data plane — инфраструктурный уровень (плоскость передачи данных), обеспечивающий непосредственную обработку и пересылку пользовательского трафика. Иллюстрация на рисунке 1.1 демонстрирует типовую архитектуру SDN-сети [20]. В рамках данной архитектуры функции мониторинга и телеметрии реализуются на границе плоскостей управления и данных за счёт использования модельно-ориентированных протоколов (gNMI, MDT), что расширяет классическую SDN-модель и обеспечивает высокочастотную наблюдаемость сети [17], [18].

Отдельный класс современных решений — методы телеметрии «в полосе» (In-band Network Telemetry, INT), при которых сетевые устройства добавляют служебные метки к проходящим пакетам. Такие метки могут содержать информацию о задержке, размере очереди, маршруте и загруженности интерфейсов. INT-подходы обеспечивают беспрецедентную детализацию и точность измерений, однако требуют глубокой модификации сетевого оборудования и увеличивают накладные расходы на передачу пакетов [21].

Таким образом, традиционные методы мониторинга — от SNMP до потоковых технологий — представляют собой эволюционную линию развития средств обеспечения сетевой наблюдаемости [5]. SNMP остаётся основой базового мониторинга и управления оборудованием [2], [3], [4], тогда как NetFlow/IPFIX и sFlow применяются для анализа структуры трафика и выявления аномалий [5], [7], [12], [13]. Новые подходы, такие как MDT и INT, ориентированы на обеспечение видимости в реальном времени и интеграцию с автоматизированными системами управления, что делает их более подходящими для современных SDN/NFV и мультисервисных сетей [17], [18], [20], [21].

## 1.2 Требования современных сервисов и сетей к качеству обслуживания

Современные телекоммуникационные сервисы предъявляют значительно более высокие и разнообразные требования к качеству обслуживания, чем классические сетевые приложения предыдущих десятилетий [1], [22]. Если ранее ключевыми факторами являлись доступность канала и базовая пропускная способность, то сегодня в центре внимания находятся десятки показателей, отражающих как технические параметры функционирования сети (QoS), так и субъективное восприятие пользователем уровня предоставляемой услуги (QoE) [1], [22]. В условиях стремительного роста мультимедийного трафика, виртуализации сетевых функций и перехода к высокодинамичным облачным средам становится необходимой комплексная модель оценки качества, учитывающая особенности конкретных сервисов и их чувствительность к задержкам, потерям и вариациям параметров [22], [23].

**Технические параметры QoS и их значимость**

Классические метрики качества обслуживания (QoS) формируют базовую основу для анализа функционирования телекоммуникационной системы [22]. Среди них ключевыми считаются: задержка (delay), вариация задержки (jitter), уровень потерь пакетов (loss), пропускная способность каналов (throughput), а также доступность сервисов (availability) [22], [24]. В современных сетях эти параметры должны оцениваться с высокой точностью и достаточной частотой, что связано с необходимостью поддержания требуемых SLA для широкого спектра приложений — от IP-телефонии до low-latency сервисов в промышленных IoT-средах [23], [24].

С переходом на конвергентные мультисервисные архитектуры многие приложения стали предъявлять противоречивые требования к QoS [22], [23]. Например, потоковое видео высокого разрешения (4K/8K) требует высокой пропускной способности и умеренной задержки, тогда как системы удалённого управления или URLLC-сервисы в 5G критически чувствительны к вариациям задержки и даже кратковременным потерям [23], [25]. Таким образом, мониторинг качества не может рассматриваться как единая модель: параметры и их весовые коэффициенты существенно различаются в зависимости от класса трафика [22].

**Роль QoE в современных сетях**

Параллельно с развитием QoS-метрик значительную популярность получила концепция оценки качества восприятия (Quality of Experience, QoE) [1], [22]. В отличие от QoS, QoE отражает субъективное восприятие конечного пользователя, включающее оценку комфорта использования сервиса, стабильности соединения и задержек при загрузке контента [1], [26]. В телекоммуникационной практике QoE является одним из ключевых индикаторов успешности сервиса, поскольку именно он определяет реальную удовлетворённость клиентов [1], [26].

Для сервисов реального времени, таких как VoIP, традиционно используется показатель MOS (Mean Opinion Score), который определяется по субъективным оценкам тестовой группы пользователей, а в практических системах рассчитывается с использованием алгоритмических моделей, например ITU-T E-model (G.107) [4], [7], [12]. Коэффициент R-factor позволяет преобразовать измеренные параметры сети (задержку, потери, джиттер) в числовую оценку качества голоса [7]. Более того, современные алгоритмические модели MOS учитывают влияние кодека, ретрансляций и параметров плейаута [4], [12]. Расчёт параметра MOS в практических системах мониторинга чаще всего выполняется не по результатам реальных опросов пользователей, а по алгоритмическим моделям [4], [12]. Наиболее распространённой является модель E-model (ITU-T G.107), в которой интегральная оценка качества голоса выражается через коэффициент R-factor [7]. Преобразование R-factor в показатель MOS выполняется по нелинейной аппроксимации (1.1):

$$MOS = 1 + 0.035R + 7 \times 10^{-6}R(R - 60)(100 - R)[7], \quad (1.1)$$

где R∈[0;100] отражает совокупное влияние искажений, задержек, потерь, качества кодека и параметров плейаута. Как показывает практика операторских сетей, значения MOS≥4 соответствуют хорошему качеству голосовой связи, в то время как MOS<3 свидетельствует о заметной деградации качества [12], [14].

Видеосервисы используют дополнительные метрики, например SSIM, PSNR и VMAF, которые количественно отражают структуру и «качество»

видеопотока. Эти показатели часто интегрируются в автоматизированные системы мониторинга, позволяя отслеживать деградации качества в реальном времени и оперативно реагировать на них [27], [28].

**SLA в 5G-сетях и специфика сетевых срезов**

Особое значение QoS/QoE приобретает в архитектуре 5G, где стандарты ITU IMT-2020 предполагают дифференциацию сервисов на классы eMBB, URLLC и mMTC [22], [29]. Каждый из них требует специфических SLA, и отклонение от этих требований может приводить к критической деградации сервиса:

- eMBB (enhanced Mobile Broadband): гигабитные скорости, минимальная задержка буферизации, высокая пропускная способность [29];
- URLLC (Ultra-Reliable Low Latency Communications): задержки порядка 1 мс в пользовательской плоскости, надёжность 99.999% и сверхстабильные каналы [25], [29];
- mMTC (massive Machine-Type Communications): подключение огромного числа устройств, устойчивость к всплескам сигнализации, малая энергозатратность [29].

Одним из ключевых механизмов выполнения SLA в 5G является технология сетевых срезов (network slicing). Для каждого типа сервиса создаётся изолированный логический сегмент сети со своими правилами приоритизации, маршрутами, параметрами QoS и механизмами мониторинга. Данные о функционировании каждого среза должны собираться в более детализированном виде по сравнению с классическими сетями, что требует новых методов анализа и повышенных требований к частоте сборки метрик [30].

**Роль облачных технологий и виртуализации**

Современные сервисы всё чаще разворачиваются в распределённых облачных инфраструктурах (Kubernetes, OpenStack, NFV), что порождает дополнительную сложность в мониторинге качества [31]. Виртуализированные функции (VNF/CNF) обладают динамической природой: они могут масштабироваться, мигрировать между узлами или перезапускаться автоматически на основе политик оркестрации [31], [32]. Такие изменения требуют непрерывного мониторинга работоспособности сервисных цепочек (Service Function Chaining, SFC) и анализа QoS не только на уровне сетевых интерфейсов, но и на уровне взаимодействия микросервисов, внутренних задержек, состояния очередей и логов приложений [32].

**Индустриальные и IoT-сценарии**

Применение IoT и промышленных сетей (IIoT) делает задачу мониторинга ещё более сложной. Сервисы промышленной автоматизации требуют предсказуемых задержек и устойчивой доставки сообщений, а деградации QoS могут приводить к серьёзным экономическим и производственным потерям [32]. Дополнительные требования предъявляют и специфические протоколы (Modbus, OPC UA, MQTT), которые вынуждают

операторов адаптировать методики мониторинга под особенности их работы [32], [33].

**Интегральный характер современных требований**

Таким образом, требования к качеству обслуживания в современных сетях формируются под влиянием сразу нескольких факторов: усложнения сервисов, распределённости систем, необходимости поддержки SLA, роста интерактивного мультимедиа и развития облаков и 5G [22], [23], [29], [30], [31]. Это приводит к тому, что современные системы мониторинга должны обеспечивать не только сбор классических метрик QoS, но и построение интегральных моделей QoE, а также учитывать особенности конкретных сервисов и их критичность [22], [23], [34]. В условиях растущей гетерогенности сетей возрастает потребность в объединении метрик разных уровней — от физических характеристик каналов до параметров приложений и показателей пользовательского опыта [34], [35].

## 1.3 Методы применения машинного обучения для мониторинга и прогнозирования качества сервисов

Стремительный рост трафика, усложнение архитектур сетей и расширение спектра предоставляемых услуг привели к тому, что традиционные методы мониторинга — основанные на пороговых правилах, статистическом анализе и прямой телеметрии — перестали обеспечивать необходимую глубину наблюдаемости [5], [17], [34]. Современные сетевые инфраструктуры содержат тысячи взаимосвязанных элементов, генерируют огромные массивы данных, а реакции на инциденты должны выполняться в масштабах секунд [31], [32], [35]. В этих условиях особую роль начинает играть машинное обучение (ML), позволяющее выявлять скрытые зависимости между метриками, прогнозировать развитие инцидентов, выполнять автоматическую корреляцию событий и, в конечном счёте, повышать качество предоставляемых услуг (QoS/QoE) [36], [37].

**ML как инструмент расширения наблюдаемости и автоматизации анализа**

Основное преимущество применения ML в мониторинге заключается в способности алгоритмов работать с многомерными данными и обнаруживать закономерности, которые невозможно формализовать через статические правила [36]. В отличие от классических систем, где анализ основан на порогах и эвристиках, ML-модели извлекают признаки из самих данных: истории задержек, уровней потерь, метаданных потоков, логов приложений, состояния виртуальных функций и др. [36]. Таким образом, ML позволяет перейти от реактивного мониторинга к предиктивной аналитике и автоматизированному поиску корневых причин проблем (Root Cause Analysis, RCA) [36], [37].

**Ключевые классы задач ML в сетевом мониторинге**

**1. Обнаружение аномалий (Anomaly Detection)**

Обнаружение аномалий является наиболее зрелой областью применения ML в телекоммуникациях [36], [38]. На практике используются:

- кластеризация (k-means, DBSCAN) для выделения нетипичных режимов работы [38];
- автоэнкодеры и вариационные автоэнкодеры, обучающиеся распознавать «нормальное» состояние и фиксирующие отклонения по увеличению ошибки реконструкции [38], [41];
- One-Class SVM, хорошо работающие при ограниченном наборе аномальных данных [40];
- глубинные модели временных рядов, обнаруживающие выбросы на процессе метрик в реальном времени [39], [41].

Такие алгоритмы способны заблаговременно выявлять деградации качества — например, рост задержек на отдельных сегментах сети, появление нетипичных потоков или постепенное увеличение потерь на интерфейсе [36], [38], [39].

**2. Прогнозирование сетевых метрик (Forecasting)**

Методы ML позволяют предсказывать значения ключевых параметров, таких как задержка, потери или пропускная способность [36], [42]. Для этого применяются:

- градиентный бустинг (XGBoost, CatBoost) [42];
- линейные и нелинейные регрессионные модели [36], [42];
- LSTM/GRU-сети и Temporal Convolutional Networks (TCN) для анализа временных рядов [43];
- графовые нейросети (GNN), учитывающие топологию сети и зависимость метрик соседних узлов [44].

Прогностические модели позволяют операторам заранее определять вероятные перегрузки и динамически перераспределять ресурсы (например, изменять политику QoS, мигрировать виртуальные функции, активировать резервные каналы) [36], [42–44].

**3. Корреляция событий и автоматизация RCA (Root Cause Analysis)**

При возникновении сбоев ML-модели выполняют автоматическое сопоставление метрик, логов и телеметрии, выявляя вероятную первопричину инцидента [36], [45]. В современных системах используются:

- байесовские сети зависимостей, моделирующие причинно-следственные связи между компонентами [45];
- графовые модели RCA, связывающие метрики на разных уровнях (L2–L7) [46];
- attention-модели, выделяющие важные признаки и последовательности событий [47].

Это позволяет сокращать Mean Time to Repair (MTTR) и минимизировать влияние инцидентов на QoE пользователей [36], [45–47].

**4. Классификация сетевых состояний и сервисов**

Методы ML позволяют классифицировать трафик, определять типы приложений, выявлять аномальное поведение пользователей или устройств, а

также анализировать изменения в структуре нагрузки [36], [48]. В условиях роста зашифрованного трафика (TLS 1.3, QUIC) особую роль играют модели классификации по статистическим признакам потоков, которые не требуют инспекции содержимого (encrypted traffic classification) [48], [49].

**Источники данных для ML и особенности их обработки**

Для обучения и применения ML-моделей используются разнообразные типы данных:

- потоковая телеметрия (gNMI/MDT);
- NetFlow/IPFIX записи;
- sFlow выборки;
- SNMP-метрики;
- syslog событие;
- данные из Kubernetes/VM оркестраторов;
- временные ряды задержек и потерь из активных измерений;
- метаданные приложений (логирование, статусные коды, события отказов);
- данные из RADIUS/DHCP/NAT систем.

Комбинирование этих источников позволяет формировать богатые многомерные датасеты, необходимые для построения точных ML-моделей [36], [50].

Основные проблемы обработки сетевых данных:

- гетерогенность форматов;
- несинхронность временных меток;
- пропуски данных и шум;
- неоднородность частоты выборки;
- корреляция метрик между разными узлами.

Именно поэтому практические системы ML включают этапы нормализации, очистки данных и временной синхронизации [36], [50], [51].

**ML в облачных и SDN/NFV сетях**

Переход к SDN/NFV и облачным архитектурам увеличивает доступность телеметрии и расширяет роль ML в мониторинге [31], [36]. Контроллеры SDN обеспечивают централизованный сбор метрик, а виртуализация сети приводит к появлению новых точек фейлов, что делает ML особенно востребованным [31], [52]. ML-модели применяются для:

- анализа виртуальных сетевых функций (VNF/CNF);
- оценки производительности сервисных цепочек;
- обнаружения деградаций при миграции контейнеров;
- предсказания узких мест в overlay-сетях.

Cloud-native observability-системы (Prometheus, Thanos, OpenTelemetry, Jaeger) генерируют огромные объёмы структурированных метрик и трассировок, что делает ML естественным инструментом их автоматической обработки [36], [53].

**Интеграция ML в практические системы мониторинга**

На практике ML применяется в коммерческих и операторских решениях:

- Cisco (DNA Center, ThousandEyes) использует ML для анализа потоков, обнаружения аномалий и корреляции событий [54];
- Google SRE применяет ML-алгоритмы для раннего выявления деградаций сервисов и прогнозирования нарушений SLO [55];
- Huawei NCE-IP интегрирует ML для анализа сетевых срезов в 5G [56];
- AT&T и NTT используют ML для оптимизации трафика в SDN/WAN сегментах [57].

Эти решения демонстрируют, что ML становится не вспомогательным инструментом, а ключевым элементом современной архитектуры мониторинга [36], [54–57].

### 1.4 Эффективность подходов мониторинга в SDN/NFV и облачных сетях

Переход телекоммуникационной отрасли к программно-определяемым сетям (SDN) и виртуализации сетевых функций (NFV) радикально изменил требования к мониторингу и привёл к появлению новых подходов к обеспечению наблюдаемости (observability) [20], [31], [58]. В традиционных аппаратных сетях точек отказа и контролируемых элементов было относительно немного: маршрутизаторы, коммутаторы, канальные устройства и их интерфейсы. Однако SDN/NFV-среды характеризуются высокой динамичностью, множеством виртуальных функций, контейнеризированных сервисов, распределёнными дата-центрами и постоянными изменениями сетевой топологии. Это делает классические методы мониторинга — такие как SNMP, периодический polling или статические пороговые правила — недостаточно эффективными [5], [17], [31], [34].

**Особенности мониторинга в SDN: централизованное управление и необходимость детализированной телеметрии**

Архитектура SDN предполагает разделение плоскостей управления (control plane) и данных (data plane), что позволяет централизовать принятие управленческих решений на уровне SDN-контроллера [20], [58]. Такой подход обеспечивает новые возможности мониторинга, но одновременно вызывает и дополнительные сложности. Контроллеры SDN получают большое количество телеметрических данных через Southbound API (OpenFlow, P4Runtime, NETCONF), а также могут инициировать активные запросы состояния сетевых элементов [59]. Однако с ростом масштаба SDN-инфраструктуры нагрузка на контроллер возрастает, что требует применения потоковой телеметрии (streaming telemetry) и методов интеллектуальной агрегации данных [17], [18], [58], [59].

Современные SDN-контроллеры используют модели YANG и протоколы gNMI/gRPC для высокочастотного получения метрик [17], [18], [59]. В отличие от polling-подхода, стриминговая телеметрия обеспечивает минимальную задержку доставки данных и позволяет фиксировать быстрые

изменения состояния интерфейсов, очередей, таблиц маршрутизации и политик Forwarding Pipeline [17], [18], [58]. Это особенно важно при реализации таких функций, как динамическое перераспределение трафика (traffic engineering), контроль SLA или адаптация маршрутов при изменении нагрузки [20], [58], [59].

**NFV-среды и проблемы мониторинга виртуальных функций**

В NFV-сетях традиционные сетевые функции — такие как BRAS, DPI, межсетевые экраны, балансировщики нагрузки, EPC/5GC-функции — выполняются в виде виртуализированных элементов (VNF) или облачных сервисов в форме контейнеров (CNF). Такие компоненты могут масштабироваться горизонтально, мигрировать между узлами, перезапускаться при отказах и динамически изменять свой жизненный цикл в рамках оркестрации MANO (ETSI NFV MANO) [31], [60].

Основные проблемы мониторинга в NFV-средах:

- **динамическая природа сервисов**: традиционные методы не успевают подстраиваться под быстрые изменения в количестве и состоянии VNF/CNF [31], [52], [58];
- **многоуровневость мониторинга**: требуется контроль не только сетевых метрик, но и состояния хостов, гипервизоров, контейнеров, сервисных цепочек (SFC) [31], [52], [58];
- **корреляция данных**: деградация может возникнуть в одном уровне (например, в очередях виртуального свитча), но проявляться в другом (например, в задержках 5G-сессии) [52], [58], [61];
- **отсутствие аппаратных метрик**: виртуализация скрывает часть низкоуровневых данных, доступных в физических устройствах [31], [61].

Чтобы повысить наблюдаемость NFV-инфраструктур, операторы внедряют стеки cloud-native observability: Prometheus, Grafana, Loki, Jaeger, OpenTelemetry. Эти инструменты предоставляют не только метрики, но и трассировки (distributed tracing), логи и события, что позволяет выявлять проблемы в цепочках микросервисов и VNF [53], [58].

**Виртуальные валидаторы и активные тесты в облачных сетях**

Одним из эффективных направлений развития мониторинга в NFV/SDN-средах является применение виртуальных валидаторов (Virtual Verifiers, VV) [62]. Эти программные агенты размещаются непосредственно внутри виртуализированной или контейнерной сетевой среды и выполняют активные тесты: проверку маршрутизации, тестовые вызовы IMS, DNS-запросы, измерение задержек между компонентами, имитацию пользовательских соединений и т. д. [62], [63]. Такой подход позволяет:

- выявлять ошибки конфигурации до выхода сервиса в продакшн;
- устранить «немые» зоны наблюдаемости;
- получать независимую оценку фактического качества предоставления услуги;

- повышать точность RCA за счёт наличия контрольных активных метрик [62], [63].

Поскольку виртуальные валидаторы функционируют непосредственно «внутри» сервисной инфраструктуры, они лучше фиксируют локальные проблемы (например, деградацию vSwitch, задержки overlay-туннелей VxLAN/GENEVE, блокировки в цепочках микросервисов), чем пассивные методы [62], [63].

**Monitoring-over-Overlay: влияние виртуальных сетей на точность измерений**

Облачные среды используют виртуальные сетевые оверлеи (VxLAN, GENEVE), создавая дополнительные уровни инкапсуляции [64]. Это приводит к искажению измерений:

- задержки могут распределяться на различные узлы;
- узкие места могут возникать внутри программных свитчей (OVS, VPP);
- телеметрия нижних уровней (например, NIC hardware counters) не отражает поведения overlay-трафика [64], [61].

Современные решения telemetry-aware overlay routing учитывают эти проблемы и используют комбинированный сбор данных — от гипервизора, vSwitch, контейнерного runtime и сетевых агентов [58], [65].

**In-band Telemetry (INT/IOAM) как основа детализированного мониторинга**

Переход к высокоскоростным сетям и загрузке 100G/400G приводит к появлению методов in-band telemetry (INT), при которых информация о состоянии сети инкапсулируется прямо в проходящие пакеты [21]. Такие методы позволяют собирать метрики задержки, занятости очередей, hop-by-hop latency и информации о пути пакетного фрейма с точностью, недостижимой другими способами [21], [66]. Несмотря на высокую точность, методы INT требуют:

- поддержки на ASIC-уровне;
- модификации обработки пакетов;
- контроля накладных расходов (увеличение размера пакета);
- синхронизации временных меток [21], [66].

В NFV/SDN-средах это особенно важно, поскольку логические топологии часто не совпадают с физическими [21], [58], [61].

**Интеграция ML в мониторинг SDN/NFV**

Современные сети всё чаще используют ML-алгоритмы для анализа метрик и телеметрии в SDN/NFV-средах [31], [36], [52]. ML применяется для:

- прогнозирования нагрузки и предотвращения перегрузок;
- анализа поведения overlay-туннелей и SLA-сервисов;
- автоматической корреляции аномалий с состоянием VNF/контейнеров;
- выявления циклических зависимостей и задержек в микросервисных системах [36], [52].

Графовые нейросети (GNN), учитывающие топологию SDN, показывают высокую эффективность в задачах прогнозирования задержек и

оптимизации маршрутов [254]. Виртуализированные сети, благодаря большому объёму телеметрии, предоставляют особенно богатые датасеты для ML-аналитики [31], [36], [52], [58].

Ни один подход не является полностью самодостаточным. В современных SDN/NFV-сетях наиболее эффективным считается комбинированный подход, включающий:

- потоковую телеметрию (основа наблюдаемости);
- активные тесты (валидация SLA);
- ML-аналитику (предиктивность и RCA);
- сбор мультиуровневых метрик (L2–L7);
- анализ поведения overlay-туннелей.

Только интеграция этих методов обеспечивает сквозную видимость качества в облачных и виртуализированных системах [5], [17], [21], [31], [36], [58], [62], [63], [66].

### 1.5 Сравнительный анализ протоколов мониторинга (SNMP, NetFlow, sFlow, IPFIX, gNMI и др.)

Протоколы мониторинга, применяемые в современных телекоммуникационных сетях, существенно различаются по способу сбора данных, уровню детализации информации, накладным расходам и области применимости [5], [12], [17]. Для операторов сетей доступа и магистральных провайдеров важна не только возможность фиксировать факт отказа, но и способность своевременно выявлять деградации качества обслуживания и прогнозировать возникновение проблем [34], [36]. Поэтому требуется сравнительный анализ основных подходов: SNMP-поллинга, потоковых технологий (NetFlow/IPFIX), протоколов сэмплирования (sFlow), модельно-ориентированной телеметрии (gNMI/MDT), а также активных и пассивных методов измерения [5], [7], [12], [17], [21], [62].

Исторически первой технологией стала модель на основе SNMP. Этот протокол обеспечивает централизованный опрос MIB-переменных на сетевых устройствах с заданным интервалом [2], [3]. Его преимущества очевидны: простота реализации, низкая нагрузка на сеть и универсальность — практически всё оборудование поддерживает базовый набор MIB-объектов [2], [3]. SNMP-подход хорошо подходит для оценки состояния интерфейсов, выявления грубых отказов и контроля загруженности каналов [2]. Однако интервальный характер поллинга приводит к появлению «слепых зон»: кратковременные перегрузки, всплески потерь или микропаузы задержек могут остаться незамеченными. Кроме того, при сокращении интервала опроса резко возрастает нагрузка на управляющий процессор сетевого оборудования и системы мониторинга, что ограничивает масштабируемость подхода в крупных сетях [4].

Переход к потоковому мониторингу (flow-based monitoring) был обусловлен недостатком глубины наблюдаемости, предоставляемой SNMP [5], [4]. Технологии NetFlow и IPFIX ориентированы на экспорт

агрегированных записей о потоках трафика, включающих IP-адреса источника и назначения, номера портов, протоколы, объёмы и временные характеристики [5], [6]. В отличие от SNMP, который предоставляет только интегральные счётчики по интерфейсам, потоковая телеметрия позволяет анализировать структуру трафика: какие сервисы и приложения доминируют, какие абоненты создают наибольшую нагрузку, какие направления генерируют аномальную активность [5], [7]. Исследования показывают, что использование flow-данных существенно повышает эффективность задач обнаружения атак, анализа пользовательского поведения и планирования ёмкости сети [7], [8]. Вместе с тем, потоковый мониторинг требует дополнительных вычислительных ресурсов для агрегирования и экспорта записей, а также развёртывания коллектора для их обработки [5], [8].

Методы, основанные на сэмплировании пакетов (например, sFlow), решают проблему масштабируемости за счёт выборочного анализа каждого N-го пакета. Это радикально снижает накладные расходы и делает возможным мониторинг в высокоскоростных сетях с пропускной способностью десятки и сотни Гбит/с [12], [13]. В таких условиях регистрация каждого пакета становится практически невозможной, и сэмплирование представляет разумный компромисс [12]. Однако вероятностная природа выборки приводит к тому, что малые или кратковременные потоки могут быть полностью пропущены, а оценки по отдельным сервисам становятся менее точными [14], [15]. Это ограничивает применимость методов сэмплирования в сценариях, где важен детальный контроль SLA, например, для критичных корпоративных сервисов или приложений низкими задержками [14–16].

С развитием SDN/NFV и появлением стриминговой телеметрии (model-driven telemetry, MDT) стала возможной принципиально иная модель мониторинга, основанная на подписке на интересующие параметры [17], [18], [31], [58]. В отличие от SNMP-поллинга, где инициатором запроса выступает система мониторинга, в MDT данные передаются от устройства к коллектору по заранее настроенным правилам подписки с заданной частотой [17], [18]. Использование моделей YANG и протоколов gNMI/gRPC позволяет формировать поток высокочастотных метрик, охватывающий не только интерфейсные счётчики, но и внутренние параметры устройства, состояние таблиц маршрутизации, очередей и политик обработки трафика [17], [18], [59]. Работы последних лет показывают, что переход к потоковой телеметрии позволяет существенно сократить время обнаружения аномалий и повысить точность анализа, особенно в виртуализированных и облачных сетях [19], [31], [58].

Отдельный класс методов составляют активные и пассивные измерения [5], [26]. Активные методы строятся на генерации тестового трафика (проб) и измерении его параметров: задержки, потерь, вариаций задержки, времени установления соединения и т. д. Они позволяют оценивать качество услуг с точки зрения абонента и часто используются для контроля QoE голосовых и видеосервисов [1], [4], [26]. Пассивные методы, напротив, основаны на

анализе реального пользовательского трафика и чаще применяются в задачах глубокой диагностики, анализа приложений и безопасности [5], [7]. В операторах связи практическая эффективность достигается за счёт комбинирования активных и пассивных подходов, где активные измерения предоставляют информацию об уровне сервиса, а пассивные — о детальных причинах его деградации [5], [26], [32], [62].

Сводные характеристики и основные параметры, используемые в дальнейшем анализе, представлены в таблице 1.1.

Таблица 1.1 — Сравнительная характеристика основных подходов мониторинга (Примечание – Составлено по [2–4], [5], [7], [12–17], [21], [62])

| Подход | Достоинства | Недостатки |
|---|---|---|
| SNMP | Простота, низкая нагрузка на устройства | Нет детализации трафика, большие интервалы опроса |
| Пакетные снифферы | Полное копирование трафика, подробный анализ пакетов | Высокая нагрузка на CPU, сложность масштабирования |
| NetFlow/IPFIX | Детальная информация о потоках (источник, назначение, объем); полезно для анализа трафика и безопасности | Требует настройки экспорта; увеличивает нагрузку при высокой детализации |
| sFlow | Аппаратное сэмплирование снижает нагрузку | Выборочные данные (неполная картина); возможна неточность на малых потоках |
| gNMI/MDT (телеметрия) | Мгновенное получение метрик в реальном времени; отсутствует циклический опрос | Требуется поддержка нового протокола и инфраструктуры; генерируется очень большой объём данных |
| INT/IOAM | Встроенные в пакеты метки дают детальные метрики (задержка и др.) на реальном трафике | Требует поддержки на сетевом оборудовании; увеличивает размер пакетов (накладные данные) |

Сравнительный анализ показывает, что каждый из рассмотренных методов имеет как сильные стороны, так и ограничения [5], [12], [17], [21], [62]. SNMP остаётся необходимым базовым инструментом для мониторинга состояния оборудования, но его временная разрешающая способность и отсутствие контекстной информации о потоках ограничивают его роль [2]. Flow-ориентированные технологии (NetFlow/IPFIX) предоставляют детализированное представление о структуре трафика, но требуют дополнительных ресурсов и не всегда обеспечивают достаточную точность в

высокоскоростных сегментах без использования специальных оптимизаций [5], [7], [8]. Сэмплирование пакетов (sFlow) масштабируется для больших скоростей, но снижает точность оценки сервисов с малыми объёмами трафика [12–16]. Стриминговая телеметрия (gNMI/MDT) обеспечивает высокую частоту и гибкость, однако предъявляет повышенные требования к архитектуре и интеграции с аналитическими платформами [17], [18], [58], [59]. Активные и пассивные методы образуют дополнительный слой наблюдаемости, позволяя перейти от измерения только сетевых параметров к оценке реального качества предоставляемых услуг [1], [26], [32], [62].

В совокупности это обосновывает необходимость гибридных систем мониторинга, которые объединяют достоинства разных протоколов: SNMP для базового контроля состояния, flow-телеметрию для анализа структуры трафика, MDT/gNMI для высокочастотного контроля ключевых показателей и активные/пассивные измерения для оценки QoE [5], [7], [12], [17], [21], [26], [62]. Именно в контексте такого комбинированного подхода далее в работе формулируются требования к разрабатываемой модели мониторинга качества сервисов в сетях оператора.

### 1.6 Международный опыт мониторинга качества сервисов

Мировая практика демонстрирует, что современные сети связи стремительно переходят от простых схем наблюдения, основанных на периодических опросах устройств, к многоуровневым архитектурам мониторинга, интегрирующим потоковую телеметрию, аналитику в реальном времени и интеллектуальные методы обработки данных [5], [17], [19], [58]. Крупные телекоммуникационные операторы и поставщики облачных сервисов (Google, AT&T, Verizon, NTT, Deutsche Telekom, Orange и др.) формируют высокую планку требований к системам наблюдаемости (observability), что стимулирует развитие исследований и внедрение инновационных подходов [31], [36], [58].

Одной из ключевых тенденций является широкое распространение потоковой телеметрии и событийно-ориентированных моделей сбора данных. Google в своих магистральных сетях внедряет механизмы высокочастотного мониторинга, ориентированные на регистрацию состояния сетевых функций, очередей, таблиц маршрутизации и пропускной способности на уровне отдельных сервисных фрагментов сети [19], [58]. Это позволяет резко сократить время обнаружения отказов и повысить точность локализации проблемных участков [19]. Аналогичные решения реализуются в AT&T, где телеметрия сочетается с детальным анализом поведенческих характеристик трафика и автоматизируемыми механизмами отказоустойчивости [52], [57].

Важную роль играет развитие концепции In-band Network Telemetry (INT), которая активно исследуется и внедряется в сетях крупных операторов и облачных платформ [21]. В отличие от традиционных методов, INT позволяет собирать ключевые параметры производительности непосредственно внутри пакета, добавляя к нему метаданные о задержках,

потерях, состоянии очередей и маршрутизаторов [21], [66]. Практика применения INT в больших сетях показывает, что данный подход позволяет обнаруживать скрытые («gray») сбои, которые не проявляются в виде очевидных отказов, но приводят к деградации качества обслуживания на уровне приложений [21], [66].

Значительное внимание уделяется также методам обнаружения аномалий и корреляции событий. В компаниях Verizon и NTT ведутся исследования по интеграции машинного обучения в системы мониторинга, что позволяет выявлять сложные нелинейные зависимости между сетевыми параметрами [36], [38]. Например, использование автоэнкодеров, моделей прогнозирования временных рядов и методов реконструкции трафика помогает обнаруживать деградации, возникающие задолго до появления заметных симптомов [38], [39]. Применение нейросетевых моделей позволяет повышать точность диагностики, особенно в условиях высокой вариативности нагрузки и динамических маршрутов [36], [38], [57].

Особое направление международного опыта связано с мониторингом качества услуг в сетях доступа и агрегации. Исследования в европейских операторах (Orange, Swisscom, BT Group) показывают, что анализ пользовательских сессий (PPPoE/IPoE) предоставляет ценную информацию о стабильности соединений, времени установления сессий и качестве доступа к услугам [68]. Внедрение методов машинного обучения на основе статистик PPPoE-сессий позволяет обнаруживать проблемы в сегментах, где традиционные инструменты наблюдения не обеспечивают достаточной детализации [36], [68]. Такие алгоритмы успешно применяются в задачах оценки стабильности магистральных узлов, обнаружения аномального поведения абонентов и контроля качества предоставляемых услуг [36], [68].

Особое внимание уделяется сопоставлению сетевых параметров с уровнем качества восприятия (QoE). В международной практике широко применяется E-model и оценки MOS для сервисов передачи голоса и видео, что позволяет операторам выстраивать мониторинг не только на уровне сети, но и на уровне пользовательского опыта [1], [4], [26]. Это особенно актуально в сетях 5G и сетях с высокой перегрузочной динамикой, где QoE-метрики дают более корректную картину о качестве услуг по сравнению с низкоуровневыми сетевыми параметрами [1], [26].

Обобщая международный опыт, можно отметить, что ключевыми направлениями развития систем мониторинга являются:

- переход от поллинговых моделей к потоковой телеметрии;
- использование методов интеллектуального анализа данных и машинного обучения;
- интеграция моделей обнаружения скрытых аномалий;
- применение механизмов in-band телеметрии;
- расширение набора наблюдаемых метрик за счёт анализа пользовательских сессий;
- усиление связи между QoS и QoE;

- повышение автоматизации и адаптивности систем наблюдаемости [5], [17], [21], [31], [36], [58], [66], [68].

Эти тенденции служат важной базой для разработки современных систем мониторинга качества в сетях связи и определяют направления дальнейшего исследования, в рамках которых формируется предлагаемая в диссертации методика [31], [34], [36], [58].

### 1.7 Анализ метрик и показателей оценки качества сервисов

Оценка качества предоставляемых телекоммуникационных услуг строится на сочетании двух групп показателей: метрик качества обслуживания (Quality of Service, QoS) и метрик качества восприятия (Quality of Experience, QoE) [1], [22]. QoS характеризует объективные сетевые параметры — задержку, вариацию задержки, потери пакетов, пропускную способность, доступность каналов [22], [24]. QoE отражает субъективное восприятие качества пользователем, учитывая реакцию приложений, особенности кодеков, структуру мультимедийного трафика и условия использования услуг [1], [34].

Стандартные параметры QoS, такие как задержка, jitter, потери пакетов и стабильность пропускной способности, являются ключевыми для сервисов реального времени — VoIP, потокового видео, видеоконференций и интерактивных приложений [22], [24]. Даже незначительные отклонения сетевых метрик могут приводить к заметной деградации пользовательского опыта [1], [26]. Однако в международной практике QoS рассматривается лишь как базовый слой оценки качества, а QoE — как результирующий показатель, интегрирующий влияние сетевых и прикладных факторов [1], [34], [35].

Для количественной оценки QoE широко применяется E-model, стандартизованный рекомендацией ITU-T G.107 [7]. Модель позволяет вычислять показатель R-factor, который учитывает влияние задержек, потерь, кодеков и других искажений [4], [7]. Далее R преобразуется в метрику MOS (Mean Opinion Score), отражающую субъективную оценку качества [4], [7]. Подробная математическая модель E-model, а также формула вычисления MOS уже были рассмотрены ранее в разделе 1.2, поэтому здесь приводится только её концептуальное использование [4], [7], [26].

Современные исследования расширяют применение MOS на оценку видеотрафика и мультимедийных приложений, используя как классические модели, так и методы машинного обучения [4], [27], [28], [36]. При анализе QoE операторы всё чаще применяют гибридные подходы, объединяющие сетевые показатели (RTT, jitter, throughput), данные о пользовательских сессиях (например, PPPoE/IPoE), параметры потоков и статистические зависимости [1], [22], [34], [68]. Такие методы позволяют выявлять деградации качества услуг даже в тех случаях, когда базовые сетевые метрики остаются в допустимых пределах [1], [26], [68].

Текущие тенденции в сетях 5G, мультисервисных сетях доступа и магистральных архитектурах показывают, что QoE-ориентированный

мониторинг становится неотъемлемым элементом управления качеством услуг [22], [23], [29], [30]. Интеграция моделей QoS и QoE позволяет операторам прогнозировать возможные нарушения SLA, своевременно выявлять «серые» (gray) деградации, оптимизировать параметры сети и повышать удовлетворённость пользователей [1], [22], [35], [36], [68].

### 1.8 Проблемы и вызовы в мониторинге мультисервисных и распределённых сетей

Современные мультисервисные и распределённые телекоммуникационные сети характеризуются высокой динамичностью, масштабируемостью и разнообразием предоставляемых услуг, что существенно усложняет задачу эффективного мониторинга [22], [31], [58]. В сетях операторского уровня сосуществуют широкополосный доступ (FTTH/GPON), мобильная передача данных (4G/5G), корпоративные VPN, CDN-сервисы, облачная инфраструктура и приложения реального времени [22], [29], [30]. Каждая технология предъявляет собственные требования к задержкам, стабильности канала, пропускной способности и качеству восприятия, что делает мониторинг многоуровневым и ресурсоёмким процессом [22], [24], [35].

Одной из ключевых проблем является гетерогенность сетевой инфраструктуры. Различия в архитектурах устройств доступа, магистральных маршрутизаторов, BNG/BRAS-узлов, SDN-контроллеров и виртуализированных сетевых функций затрудняют унификацию моделей наблюдаемости [5], [31], [58]. В традиционных сетях мониторинг базировался на SNMP-поллинге и анализе интерфейсных счётчиков, однако для мультисервисных систем этого оказывается недостаточно [2]. Виртуализированные и распределённые функции (NFV/VNF), работающие в облачных платформах или контейнерных средах, создают дополнительные уровни абстракции, уменьшая прозрачность работы сети и усложняя поиск первопричины отказов [31], [52], [58], [61].

Второй существенный вызов — рост объёмов телеметрических данных. Современные сети генерируют массивы метрик: показатели пропускной способности, статистику потоков, информацию о QoE и состоянии пользовательских сессий, задержки в транспортной подсистеме, характеристики маршрутизации, данные о состоянии сервисных цепочек и виртуальных функций [5], [17], [31]. Потоковая телеметрия (MDT/gNMI), INT и высокочастотный мониторинг создают нагрузку на каналы управления и аналитические платформы [17], [18], [21], [58], [59]. При отсутствии оптимизации объём телеметрии может превышать полезный трафик, а системы анализа — испытывать перегрузку при обработке аномальных ситуаций [19], [58].

Третьим значимым аспектом является обнаружение скрытых («серых») деградаций, которые не приводят к явному отказу, но вызывают ухудшение качества услуг [21], [66]. Проблема особенно актуальна в крупных ISP-сетях,

где локальные перегрузки, колебания задержек или неустойчивая работа отдельных узлов проявляются нерегулярно и могут маскироваться нормальными значениями метрик [21], [36]. Недавние исследования показывают, что классические пороговые системы обнаружения аварий в таких условиях неэффективны: деградации могут быть кратковременными, локальными или проявляться только под нагрузкой на уровне конкретных приложений [36], [38], [39].

Четвёртой проблемой является координация мониторинга между уровнями сети. Цепочка от абонентского устройства через ONU/ONT, OLT, BNG, магистральные узлы до сервисных платформ включает множество точек потенциальных отказов [5], [31]. В мультисервисных архитектурах QoS-задержки и потери могут возникать в любом сегменте, и их локализация без коррелированного анализа данных становится крайне трудоёмкой. При этом традиционные методы мониторинга не позволяют учитывать причинно-следственные связи между параметрами сети. В международной практике для решения таких задач предлагаются методы корреляции событий на основе графовых моделей и вероятностных зависимостей, однако их применение требует высокой вычислительной мощности и качественно собранных данных [45], [46], [57].

Особое внимание в современных исследованиях уделяется мониторингу на уровне пользовательских сессий (PPPoE/IPoE). Как показывают работы последних лет, анализ статистики сессий является одним из наиболее информативных способов оценки состояния сегментов доступа, так как отражает:

- стабильность подключения;
- частоту разрывов;
- время установления сессии;
- характеристики скорости;
- реальные паттерны поведения абонентов [68].

В отличие от интерфейсных счётчиков и агрегированной телеметрии, данные на уровне сессий позволяют выявлять проблемы на физическом и канальном уровнях, которые не фиксируются стандартными протоколами мониторинга [5], [68].

Ещё одной существенной проблемой является согласование QoS и QoE. Сетевые метрики могут оставаться в допустимых пределах, в то время как пользователь испытывает деградацию качества (например, падение MOS при стабильном RTT). Такая ситуация характерна для видеосервисов, онлайн-игр и VoIP [1], [26]. Причины часто связаны с особенностями буферизации, кодеков, нестабильностью передачи на промежуточных узлах или некорректным распределением ресурсов [4], [27], [28]. Это требует интеграции оценок QoE, гибридных методов мониторинга и алгоритмов прогнозирования деградаций [1], [22], [35], [37].

И, наконец, в мультисервисных сетях особую важность приобретает предиктивный мониторинг, направленный на прогнозирование деградаций до

их фактического появления [22], [34]. Построение таких моделей затруднено из-за необходимости обработки многомерных данных, учёта временных зависимостей и адаптивности к меняющимся нагрузкам. Тем не менее, исследования показывают, что применение методов машинного обучения — автоэнкодеров, градиентного бустинга, деревьев решений и моделей прогнозирования временных рядов — позволяет оператору выявлять аномальное поведение с высокой точностью и оперативно реагировать на угрозы качества сервиса [36], [38], [39], [42], [43].

В совокупности перечисленные проблемы демонстрируют, что для мультисервисных и распределённых сетей традиционные методы мониторинга недостаточны [5], [17], [31], [58]. Сложность архитектур, разнообразие сервисов, распределённая логика управления и высокий темп изменений требуют гибридных систем наблюдаемости, объединяющих телеметрию, поведенческий анализ, оценку QoE и методы интеллектуальной обработки данных [22], [34], [35], [36], [58]. В этой связи в диссертационной работе предлагается подход, основанный на анализе косвенных статистических параметров, формируемых в процессе предоставления телекоммуникационных услуг [68]. В частности, анализируются метаданные PPPoE-сессий (длительность, частота переподключений, IP-логи, сообщения об ошибках, результаты аутентификации и авторизации), а также сопутствующие данные от DHCP, NAT, RADIUS, syslog и SNMP [5], [68]. Эти данные, будучи по своей природе техническими служебными параметрами, содержат важные признаки, отражающие поведение пользователя и стабильность предоставляемой услуги [36], [68].

Такой подход относится к косвенным методам мониторинга, не требует активного вмешательства в пользовательский трафик, не создаёт дополнительной нагрузки на сеть и может применяться в режиме реального времени в существующей инфраструктуре оператора связи [5], [17], [62]. Более того, он позволяет построить прогнозные модели на основе машинного обучения, способные выявлять отклонения в качестве предоставления услуг до того, как они станут критичными для пользователя [36], [38], [39], [42], [43]. Применение данного метода особенно актуально для крупных провайдеров с десятками и сотнями тысяч PPPoE-подключений, где традиционные методы мониторинга становятся либо избыточно дорогими, либо технически неприменимыми [22], [31], [68].

Таким образом, предлагаемый метод косвенного мониторинга качества позволяет эффективно решать задачи обнаружения и локализации проблем в распределённых телекоммуникационных системах и составляет научно-практическую основу данной диссертационной работы [31], [36], [68].

**Выводы по главе 1**

В первой главе был проведён системный обзор современных подходов к мониторингу состояния и качества услуг в телекоммуникационных сетях, охватывающий эволюцию технологий наблюдаемости, особенности основных

протоколов, методы оценки параметров QoS/QoE и практику их применения в мультисервисных архитектурах [5], [12], [17], [22], [31], [58]. Рассмотрение классических и современных инструментов мониторинга позволило выявить фундаментальные ограничения существующих решений и определить ключевые направления развития систем контроля качества в сетях операторского уровня [5], [12], [17], [21], [34], [58].

Анализ традиционных методов — SNMP-поллинга, потокового мониторинга (NetFlow, IPFIX), сэмплирования (sFlow) и активных/пассивных механизмов измерений — показал, что ни один из них не обеспечивает полноты наблюдаемости в условиях высокой динамичности и многослойности современных телекоммуникационных инфраструктур [2–5], [12–16]. Ограниченная временная разрешающая способность SNMP, неполная точность данных при сэмплировании трафика, повышенная нагрузка на устройства при экспорте потоков и сложности интеграции данных различных уровней приводят к появлению «слепых зон» в мониторинге, особенно заметных в сегментах широкополосного доступа [2–5], [12–16], [32]. Эти недостатки мотивировали переход к потоковой телеметрии (gNMI/MDT), позволяющей получать высокочастотные метрики и расширять наблюдаемость внутренних процессов сетевых устройств [17], [18], [19], [59].

Международный опыт внедрения систем мониторинга в крупных операторах (Google, AT&T, Verizon, NTT, Orange и др.) подтверждает, что современные сети требуют интеграции потоковой телеметрии, in-band механизмов (INT), а также методов корреляции событий и обнаружения аномалий на основе машинного обучения [21], [36], [52], [57], [58], [66]. Это обеспечивает возможность выявления скрытых («gray») деградаций, прогнозирования отказов и повышения качества обслуживания в условиях больших нагрузок и распределённой логики управления [21], [36], [38], [39], [66].

Значительное внимание в главе уделено метрикам QoS и QoE, а также модели E-model, которая позволяет связать объективные сетевые характеристики с субъективным восприятием качества услуг [1], [4], [7], [26], [34]. Рассмотрение данной модели в контексте современных приложений (VoIP, видеосервисы, мультимедиа реального времени) показало, что анализ только сетевых параметров недостаточен для точной оценки уровня сервиса [1], [4], [26], [27], [28]. В сетях 5G, FTTH/GPON, а также в системах агрегации трафика возрастает значимость QoE-ориентированных моделей, учитывающих поведение протоколов, параметры сеансовых соединений и особенности прикладного трафика [214], [22], [29], [30], [35].

Дополнительно в ходе анализа было выявлено, что в мультисервисных и распределённых сетях значительную информативность представляют данные уровня пользовательских сессий (PPPoE/IPoE). Как показывают

исследования, анализ статистики сессий позволяет обнаруживать деградации, недоступные при использовании классических протоколов мониторинга [5], [68]. Данный источник данных становится важным элементом высокоточных систем наблюдаемости, особенно при использовании методов статистического анализа и машинного обучения [36], [68].

Проведённый обзор позволяет сделать общий вывод, что существующие практики мониторинга, несмотря на высокий уровень развития, нуждаются в дальнейшей адаптации к условиям современных операторских сетей [22], [31], [58]. Сложность архитектуры, разнообразие сервисов, объемы телеметрии и потребность в быстром обнаружении причинно-следственных связей между событиями требуют построения гибридных моделей мониторинга, способных интегрировать данные различных уровней и обеспечивать высокую точность анализа [31], [34], [35], [36], [58].

В совокупности результаты, представленные в главе 1, формируют теоретико-методологическую основу для разработки нового подхода к мониторингу качества услуг, основанного на анализе данных пользовательских сессий и методах интеллектуальной обработки сетевой информации [36], [68]. Данная концепция рассматривается более подробно в следующей главе, где обосновываются выбранные методы, архитектура решения и математический аппарат, применяемый для оценки качества услуг в сетях оператора [31], [36], [68].

## Глава 2. Разработка косвенного метода мониторинга на основе статистики PPPoE

### 2.1 Протокол PPPoE и его роль в широкополосных сетях доступа

В современных широкополосных сетях передача данных актуальной задачей является обеспечение требуемого качества обслуживания (QoS) для разнообразных сервисов связи [22], [24]. Для контроля QoS применяются как активные, так и пассивные методы мониторинга трафика, а также аналитические и интеллектуальные системы обнаружения аномалий [5], [26], [36]. Одной из ключевых технологий в инфраструктуре широкополосного доступа, особенно в сетях на базе DSL, является протокол PPPoE (Point-to-Point Protocol over Ethernet). Протокол PPPoE позволяет установить сессию «точка-точка» поверх традиционной Ethernet-сети и широко используется интернет-провайдерами для подключения абонентов по технологиям DSL и другим видам доступа, требующим аутентификации [5], [31], [68], [69]. PPPoE расшифровывается как Point-to-Point Protocol over Ethernet – протокол канального уровня, предназначенный для установления и управления PPP-соединениями между двумя узлами в сети Ethernet. По сути, PPPoE инкапсулирует фреймы PPP (Point-to-Point Protocol) внутри кадров Ethernet, тем самым позволяя одновременно нескольким хостам в разделяемой Ethernet-сети устанавливать отдельные PPP-сессии с удалённым узлом-концентратором (обычно сервером доступа провайдера) [69], [70]. Первоначально PPP (протокол «точка-точка») разрабатывался для прямых соединений типа коммутируемого доступа и обеспечивает такие функции, как многопротокольная инкапсуляция, сжатие, аутентификация пользователей (через протоколы PAP/CHAP) и управление соединением (LCP) [70], [71]. Технология PPPoE позволила перенести эти возможности PPP в Ethernet-сети, что сделало возможным повсеместное использование PPP в широкополосном доступе (преимущественно DSL) с сохранением механизмов аутентификации и учёта трафика, привычных для коммутируемого доступа [69], [70].

Таким образом, в архитектуре широкополосного доступа PPPoE играет роль протокола, соединяющего клиентское оборудование абонента (CPE) и пограничный маршрутизатор или концентратор доступа провайдера (так называемый PPPoE Access Concentrator, обычно реализованный на оборудовании BRAS/BNG) [31], [69]. При установлении PPPoE-соединения абонентский маршрутизатор (или DSL-модем в режиме моста с подключенным ПК) инициирует сессию, а пограничное оборудование провайдера отвечает и завершает процедуру установления соединения [69]. В процессе сессии PPPoE обеспечивает обмен PPP-кадрами между клиентом и сетью, включая проведение аутентификации пользователя (требуется ввод имени пользователя и пароля для доступа), согласование параметров LCP и последующую передачу данных [70], [71]. После успешной установки PPPoE-сессии абонент получает сетевые реквизиты (например, IP-адрес через IPCP) и может использовать связь с гарантией того, что каждое устройство имеет индивидуальную PPP-сессию [69], [70]. Такой подход упростил подключение

множества пользователей через одну широковещательную среду, обеспечив разделение трафика и индивидуальный учёт каждой сессии [31], [68].

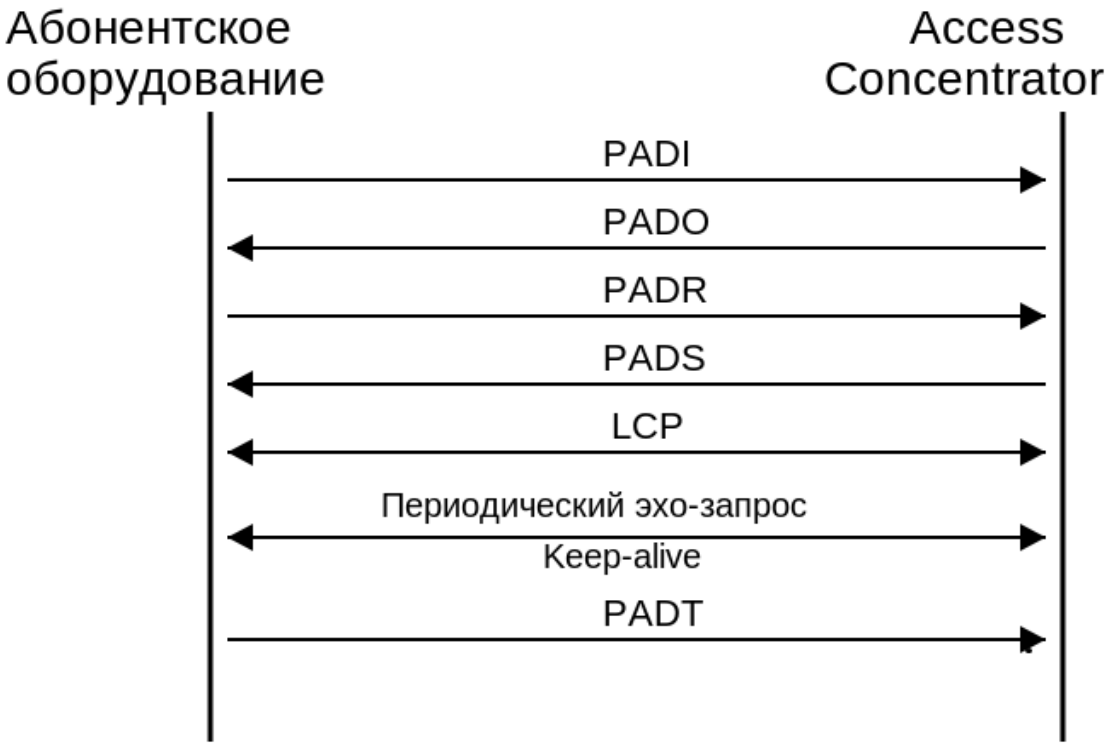


Рисунок 2.1 — Последовательность установления и завершения PPPoE-соединения (схематичное представление по RFC 2516 [69])

На рисунке 2.1 используются следующие обозначения: Access Concentrator — узел агрегации PPPoE-сессий (обычно пограничный маршрутизатор или BRAS/BNG); PADI (PPPoE Active Discovery Initiation) — широковещательный запрос абонентского оборудования на поиск сервера доступа; PADO (PPPoE Active Discovery Offer) — ответ концентратора с предложением установления соединения; PADR (PPPoE Active Discovery Request) — запрос абонента на установление выбранной PPPoE-сессии; LCP (Link Control Protocol) — протокол управления каналом PPP, используемый для согласования параметров соединения и контроля его состояния; Keep-alive — периодические эхо-запросы для проверки активности и поддержания установленной PPPoE-сессии. Представленная на рисунке 2.1 схема наглядно иллюстрирует последовательность служебных сообщений, формирующих процесс установления и завершения PPPoE-сессии между абонентским оборудованием и узлом доступа оператора [69]. Диаграмма позволяет визуально отделить фазу обнаружения (Discovery Stage) от фазы активной PPPoE-сессии, что принципиально важно для дальнейшего анализа статистики служебных пакетов. Именно на этом уровне формируются события, которые могут быть использованы как косвенные индикаторы состояния сети [68], [69]. Схема подчёркивает, что пакеты PADT могут генерироваться как клиентом,

так и пограничным оборудованием, однако причины их возникновения различаются, что впоследствии используется при построении коэффициента нестабильности [69], [68].

Важной особенностью PPPoE является механизм контроля активности сессии – периодическая отправка keep-alive пакетов PPP (эхо-запросы LCP) обеими сторонами соединения. Этот механизм предназначен для мониторинга доступности сессии: если в течение заданного интервала поступает несколько подряд неотвеченных keep-alive запросов, соединение считается разорванным по тайм-ауту [70]. Такая функциональность обеспечивает автоматическое выявление «тихого» разрыва (например, при пропаже сигнала или обрыве линии) и освобождение сетевых ресурсов, даже если абонент не послал явного запроса на разрыв [70]. Данный механизм играет ключевую роль в предлагаемом косвенном методе мониторинга качества связи, поскольку статистика таких разрывов сессий может свидетельствовать о наличии проблем в сети [68]. В следующем разделе описаны типы служебных PPPoE-пакетов и обоснован выбор показателей для мониторинга на их основе.

## 2.2 Анализ служебных сообщений PPPoE и выбор информативных признаков

Протокол PPPoE предусматривает двухфазный процесс установления соединения: фазу обнаружения (PPPoE Discovery Stage) и фазу самой PPP-сессии [69]. На этапе обнаружения стороны обмениваются специальными служебными пакетами для инициации и согласования соединения. Клиентское оборудование (CPE) и пограничное оборудование провайдера (PE, или AC) обмениваются следующими типами PPPoE-пакетов [69]:

**PADI (PPPoE Active Discovery Initiation)** – пакет инициации активного обнаружения. Отправляется клиентом (CPE) в широковещательном режиме для поиска доступного сервера PPPoE.

**PADO (PPPoE Active Discovery Offer)** – пакет предложения в ответ на PADI. Отправляется узлом доступа (AC) клиенту и содержит предложение услуги PPPoE (обычно включает имя службы).

**PADR (PPPoE Active Discovery Request)** – пакет запроса сессии. Отправляется клиентом в адрес выбранного узла доступа в ответ на PADO, запрашивая установление PPPoE-сессии.

**PADS (PPPoE Active Discovery Session-confirmation)** – пакет подтверждения сессии. Отправляется узлом доступа клиенту и содержит подтверждение установления PPPoE-сессии, включая уникальный Session ID. С этого момента PPP-сессия считается установленной.

**PADT (PPPoE Active Discovery Termination)** – пакет терминации сессии. Может отправляться как клиентом, так и сервером для завершения ранее установленной PPPoE-сессии [69].

На практике эти пакеты формируют последовательность сигнализации PPPoE: CPE рассылает PADI, получив PADO – выбирает конкретное предложение, посылает PADR, на что AC отвечает PADS, после чего

начинается обмен PPP-кадрами в рамках сессии. Завершение сессии инициируется посылкой PADT одной из сторон [69]. Таким образом, PADI–PADO–PADR–PADS служат для установления соединения, а PADT – для его явного завершения [69].

После перехода во вторую фазу (PPP Session Stage) абонентское устройство и узел доступа обмениваются регулярными PPP keep-alive сообщениями (эхо LCP) для мониторинга сессии, о чём упоминалось ранее [70]. Если абонент хочет добровольно разорвать соединение (например, вручную отключается от интернета), его устройство (CPE) отправляет PADT серверу, уведомляя о закрытии сессии. В этом случае для узла доступа событие разрыва является штатным [69]. Однако, если соединение прерывается не по воле абонента (например, из-за потери сигнала или перегрузки сети), обнаружение происходит по тайм-ауту keep-alive: после непринятия 4 подряд проверочных PPP-пакетов (типичное значение по умолчанию), сервер считает сессию оборванной и сам инициирует её закрытие, отправляя PADT клиенту [70], [71]. В последнем случае PADT исходит от пограничного оборудования, сигнализируя нештатное завершение сессии по причине сбоев в сети [68], [69].

Из перечисленных служебных пакетов PPPoE наибольший интерес для мониторинга представляет PADT, поскольку именно он непосредственно указывает на разрыв сессии [69]. Тем не менее, на этапе исследований рассматривался комплекс признаков, включающий статистику всех типов указанных пакетов. Гипотеза состояла в том, что аномальное поведение в сети (например, проблемы на канале) может отражаться в статистике пакетов PPPoE discovery-этапа: например, массовые повторные инициации сессий (множественные PADI/PADR) или отсутствие ответов (соотношение PADI/PADO) [68]. Поэтому в качестве потенциальных индикаторов качества были выбраны все пять типов пакетов PPPoE discovery-этапа [68], [69]. В таблице 2.1 приведён фрагмент общей статистики PPPoE-пакетов, собранной на пограничном маршрутизаторе крупной сети доступа, а в таблице 2.2 – статистика по отдельному VLAN (направлению доступа). Эти данные получены с эксплуатационного оборудования и иллюстрируют распределение пакетов разных типов [68].

Таблица 2.1 — Общая статистика PPPoE-пакетов на пограничном маршрутизаторе (пример) (общее число активных PPPoE-сессий: 22557). (Составлено автором по эксплуатационным данным оператора связи)

| Тип пакета | Отправлено (со стороны PE) | Получено (со стороны PE) |
|---|---|---|
| PADI | 0 | $7{,}46\times10^{8}$ |
| PADO | 34835259 | 0 |
| PADR | 0 | 16265772 |
| PADS | 15265948 | 0 |
| PADT | 6240088 | $2{,}82\times10^{8}$ |

Данные, приведённые в таблице 2.1, отражают агрегированную статистику служебных PPPoE-пакетов, собранную на пограничном маршрутизаторе за длительный период эксплуатации. Существенная асимметрия между значениями «отправлено» и «получено» для различных типов пакетов соответствует ожидаемой логике работы протокола и подтверждает корректность сбора статистики [69]. Особенно показательно различие в величинах PADT: большая доля полученных PADT соответствует штатным отключениям со стороны абонентов, тогда как существенно меньшая, но ненулевая доля отправленных PADT формируется самим узлом доступа и отражает аварийные разрывы сессий [68], [69]. Таким образом, таблица служит эмпирическим подтверждением того, что статистика PADT содержит информацию о нестабильности сети и может использоваться для мониторинга качества обслуживания [68].

Таблица 2.2 — Статистика PPPoE-пакетов для отдельного направления (VLAN) (число активных PPPoE-сессий: 289). (Составлено автором по эксплуатационным данным оператора связи)

| Тип пакета | Отправлено (со стороны PE) | Получено (со стороны PE) |
|---|---|---|
| PADI | 0 | 10431298 |
| PADO | 258974 | 0 |
| PADR | 0 | 210422 |
| PADS | 208596 | 0 |
| PADT | 75786 | 1592062 |

В отличие от агрегированных данных по всей сети, статистика, представленная для отдельного VLAN (таблица 2.2), позволяет выявить локальные отклонения в поведении PPPoE-сессий [68]. Сравнение со средней картиной показывает, что доля отправленных PADT в данном сегменте существенно выше типового уровня. Это указывает на наличие проблем, локализованных в конкретном направлении доступа, и подчёркивает ценность анализа статистики в разрезе отдельных VLAN [31], [68]. Таким образом, таблица демонстрирует, что использование детализированной PPPoE-статистики позволяет не только фиксировать факт нестабильности, но и локализовать потенциально проблемные участки сети [68].

Анализ представленной статистики позволяет обосновать выбор признаков для мониторинга. Как видно из таблиц 2.1–2.2, в штатном режиме основные сервисные пакеты имеют предсказуемое направление: PADI и PADR отправляются клиентами, а PADO и PADS – пограничным оборудованием (отсюда соответствующие нулевые значения в столбцах «Отправлено» или «Получено» на стороне PE) [69]. Пакеты PADT могут исходить от обеих сторон, однако их назначение различно: полученные PADT (сотни миллионов за период наблюдения, см. таблицу 2.1) соответствуют штатным отключениям сессий по инициативе абонентов, тогда как отправленные PADT (порядка миллионов) генерируются самим пограничным маршрутизатором при

обрывах связи [68], [69]. В нормальных условиях количество разрывов сессий по тайм-ауту относительно невелико по сравнению с общим числом сессий (в приведённом примере ~6,24 млн отправленных PADT против ~281 млн полученных, т.е. около 2,2%). Однако в отдельных сегментах (например, VLAN 303 в исследуемом случае) наблюдаются аномалии: в таблице 2.2 видно, что по одному направлению фиксировалось 75786 отправленных PADT при 1592062 полученных, что составляет приблизительно 4,76% от числа сессий данного сегмента – заметно больше среднего уровня. Подобные повышенные доли нештатных разрывов могут указывать на проблемы (например, деградацию канала или оборудования) в этом сегменте сети [68].

На основании этих наблюдений был сделан вывод, что число PADT-пакетов, отправленных маршрутизатором (то есть разрывов сессий по инициативе узла доступа), является информативным признаком потенциальной нестабильности сети [68], [69]. Другие пакеты discovery-этапа (PADI, PADO, PADR, PADS), хотя и отражают интенсивность установления сессий, не столь прямо указывают на сбои: их количество в большей мере коррелирует с пользовательской активностью (подключения новых абонентов, перезапуски устройств и т.п.), тогда как аномально высокая частота PADT именно от PE свидетельствует о прерывании уже установленных сессий, что что может быть обусловлено деградацией канала и ростом потерь/недоставок контрольных сообщений, что подтверждается сопоставлением с данными активных пробников (Приложение А) [68]. Таким образом, в качестве основного признака мониторинга выбрана статистика PADT (Sent) – количество PADT-пакетов, отправленных пограничным оборудованием. Данный показатель косвенно отражает долю сессий, оборвавшихся из-за сетевых неполадок, и может служить индикатором возникновения «узких мест» или областей деградации в сети доступа [31], [68].

**Значимость исходных признаков.** Для обеспечения интерпретируемости результатов и проверки практической значимости статистических признаков PPPoE-сигнализации в качестве эталонного показателя качества канала выбран параметр потери пакетов. Согласно рекомендациям ITU-T Y.1541, потеря пакетов относится к числу базовых сетевых показателей, определяющих качество обслуживания (QoS) в IP-сетях [24]. В рамках настоящего исследования данный параметр измерялся активными пробниками и использовался как независимый критерий деградации канала. Потери пакетов рассчитывались по результатам активных измерений (пробников) на тех же временных интервалах Δt, что и опрос PPPoE-счётчиков. Для каждого интервала фиксировалось количество отправленных пакетов $N_{sent}$ и количество потерянных пакетов $N_{lost}$, после чего вычислялась доля потерь $Loss(t)$ согласно выражению (2.1) [24]:

$$Loss(t) = \frac{Nlost(t)}{Nsent(t)} \cdot 100\% \,. \qquad (2.1)$$

Для сопоставления PPPoE-статистики с *Loss(t)* из исходных счётчиков формировались динамические признаки, отражающие изменения значений за интервал наблюдения. Прирост счётчика вычислялся согласно выражению (2.2):

$$\Delta X(t) = X(t) - X(t-1), \tag{2.2}$$

где *X(t)* - значение соответствующего PPPoE-счётчика (например, PADT, PADO, PADS и др.) в момент времени *t.*

Поскольку абсолютные значения счётчиков зависят от масштаба сегмента и количества активных сессий, динамические признаки нормировались на число активных PPPoE-сессий *S(t)*, что обеспечивает сопоставимость между различными временными интервалами. Нормированная величина определялась согласно выражению (2.3):

$$\mathrm{Xn(t)} = \frac{\Delta \mathrm{X(t)}}{\mathrm{S(t)}}. \tag{2.3}$$

Далее выполнялась оценка статистической связи каждого кандидата $Z(t) \in \{Xn(t)\}$ с эталонным показателем качества *Loss(t)*. В качестве меры линейной зависимости использовался коэффициент корреляции Пирсона, вычисляемый по формуле (2.4) [36], [73]:

$$r_{Z,Loss} = \frac{\sum_{i=1}^{n}(Z_i - \overline{Z})(Loss_i - \overline{Loss})}{\sqrt{\sum_{i=1}^{n}(Z_i - \overline{Z})^2 \cdot \sum_{i=1}^{n}(Loss_i - \overline{Loss})^2}}. \tag{2.4}$$

Статистическая значимость корреляции проверялась с использованием t-критерия Стьюдента, вычисляемого согласно выражению (2.5) [36]:

$$t = r\sqrt{\frac{n-2}{1-r^2}}, . \tag{2.5}$$

с последующим вычислением *p*-значения при n−2 степенях свободы. Нулевая гипотеза $H_0$ формулировалась как отсутствие линейной зависимости между признаком и *Loss(t)* (*r*=0), альтернативная гипотеза $H_1$ — наличие зависимости (r≠0).

Для дополнительной количественной оценки «вклада» признака применялся однофакторный ANOVA F-тест в постановке линейной регрессии *Loss(t)=α+βZ(t)+ε,* где проверка значимости коэффициента $\beta$ осуществлялась через F-статистику (2.6) [36], [73]:

$$F = \frac{SSR/1}{SSE/(n-2)}, \tag{2.6}$$

где $SSR = \sum_{i=1}^{n}\left(\widehat{Loss_i} - \overline{Loss}\right)^2$- объяснённая сумма квадратов, $SSE = \sum_{i=1}^{n}\left(Loss_i - \widehat{Loss_i}\right)^2$- остаточная сумма квадратов, $n$ — число наблюдений. Малые значения *p*-уровня значимости (например, *p*<0,05) свидетельствуют о

статистически значимой связи признака с эталонным QoS-показателем. Исходные значения $N_{sent}$, $N_{lost}$ и $Loss(t)$ приведены в таблице А.2 Приложения А.

В таблице 2.3 приведены результаты оценки статистической значимости признаков PPPoE относительно эталонного QoS-показателя потерь пакетов *Loss(t)*. Признаки отсортированы по убыванию значения F-критерия [36], [68].

Таблица 2.3 — Сравнение статистической значимости признаков PPPoE относительно эталонного QoS-показателя потерь пакетов *Loss(t)* (Составлено автором по результатам собственных вычислительных экспериментов)

| **Признак** | ***F*-значение** | ***p*-значение** | **Корреляция с *Loss(t)*** |
|---|---|---|---|
| $\frac{\Delta PADT(t)}{S(t)}$ | 2148,6 | $< 10^{-200}$ | +0,96 |
| $\frac{\Delta PADS(t)}{S(t)}$ | 402,3 | $1{,}7\times10^{-56}$ | +0,79 |
| $\frac{\Delta PADO(t)}{S(t)}$ | 18,4 | $4{,}1\times10^{-5}$ | +0,42 |
| $\frac{\Delta PADR(t)}{S(t)}$ | 1,6 | 0,21 | +0,12 |
| $\frac{\Delta PADI(t)}{S(t)}$ | 0,3 | 0,58 | +0,04 |

Результаты статистического анализа, представленные в таблице 2.3, отражают степень связи исследуемых признаков PPPoE-сигнализации с эталонным показателем качества канала — долей потерь пакетов *Loss(t)*. Наибольшие значения F-критерия и минимальные p-значения получены для нормированного прироста числа PADT, что свидетельствует о наличии статистически значимой зависимости между динамикой аварийных разрывов PPPoE-сессий и уровнем потерь пакетов. Проверка значимости выполнялась с использованием стандартных методов линейной регрессии и ANOVA F-критерия [36], [75].

На рисунке 2.2 показана корреляция исследуемых признаков PPPoE-сигнализации с эталонным показателем качества канала — долей потерь пакетов *Loss(t)*. Высокое значение коэффициента корреляции Пирсона подтверждает тесную линейную взаимосвязь указанных величин [75], [76].

Комбинированные признаки, содержащие динамику PADS и PADO, также демонстрируют статистически значимую связь с *Loss(t)*, однако их объясняющая способность уступает нормированному приросту числа PADT. Это указывает на то, что именно относительная доля аварийных разрывов сессий, а не абсолютные изменения служебных счётчиков, наиболее адекватно отражает деградацию качества канала. Признаки PADI или PADR демонстрируют статистически незначимые результаты ($p$>0,05), что

соответствует функциональной роли этих сообщений как элементов этапа установления соединения PPPoE [69]. Данные сообщения отражают пользовательскую активность, но не характеризуют устойчивость уже действующих сессий.

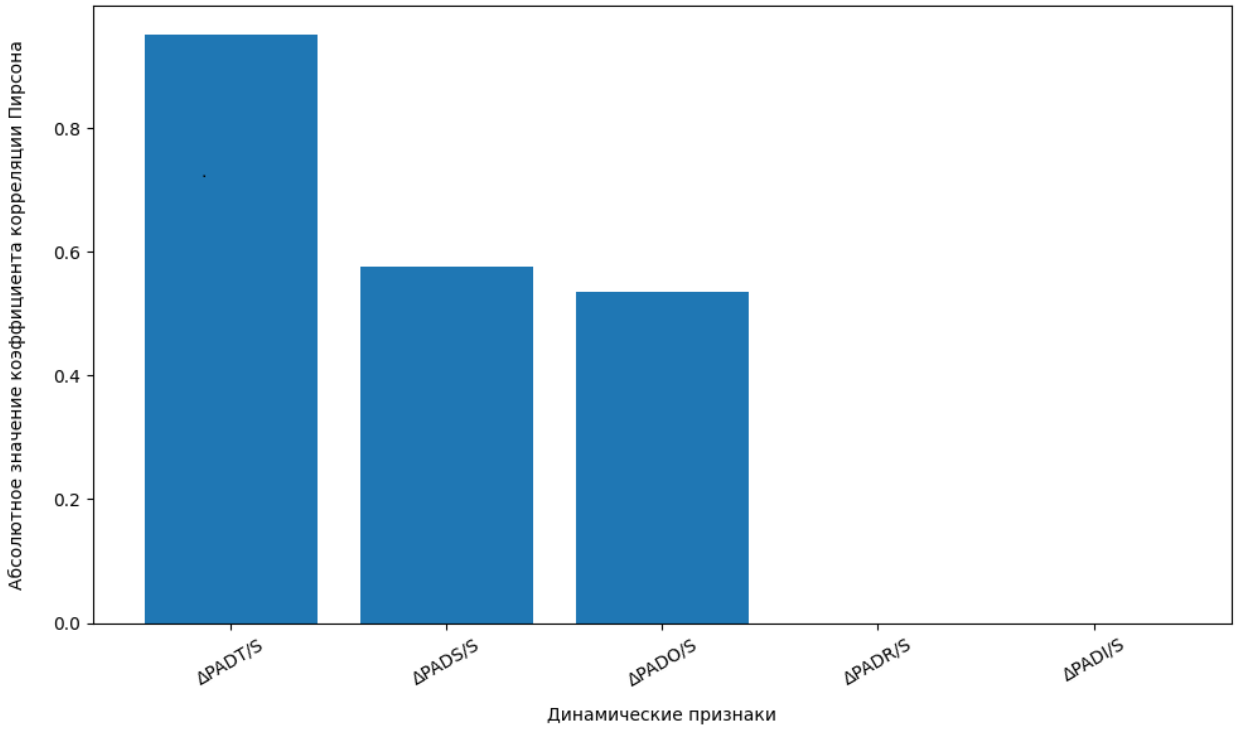


Рисунок 2.2 — Корреляция исследуемых признаков PPPoE-сигнализации с эталонным показателем качества канала — долей потерь пакетов *Loss(t)*. (Составлено автором по результатам собственных вычислительных экспериментов)

Таким образом, проведённый анализ подтверждает, что из всего множества исходных и производных признаков именно нормированный прирост числа PADT обладает наибольшей информативностью относительно реального QoS-показателя, что обосновывает его использование в качестве индикатора деградации качества обслуживания [24], [36], [75].

В настоящем исследовании для оценки взаимосвязи между динамическими признаками PPPoE и показателем потери пакетов использовался коэффициент корреляции Пирсона. Данный критерий применяется для количественных переменных, измеренных в метрической шкале, и предназначен для оценки линейной зависимости между ними [36], [73]. Анализируемые величины (нормированные приросты счётчиков PPPoE и процент потерь пакетов) являются непрерывными числовыми переменными. Предварительный визуальный анализ диаграмм рассеяния показал наличие монотонной, близкой к линейной зависимости, что соответствует условиям корректного применения коэффициента Пирсона [36], [73]. В этих условиях данный критерий является статистически обоснованным и достаточным для оценки тесноты связи. Коэффициент ранговой корреляции Спирмена

применяется преимущественно в случаях, когда зависимость носит нелинейный характер либо данные представлены в порядковой шкале измерения [73], [74]. В рассматриваемой задаче указанные предпосылки отсутствуют, поэтому использование рангового критерия не приводит к принципиально новым выводам и не влияет на интерпретацию полученных результатов.

### 2.3 Введение коэффициента нестабильности *K* и его физический смысл

Для количественной оценки нестабильности сети на основе выбранного признака (числа разорванных сессий) введён коэффициент нестабильности *K*. Данный коэффициент определяется как относительная доля PPPoE-сессий, прерванных по инициативе пограничного оборудования за интервал наблюдения. Формально коэффициент *K* можно определить следующим выражением (2.7):

$$K = \frac{PADT_t - PADT_{t-1}}{S}, \tag{2.7}$$

где $PADT_t$ и $PADT_{t-1}$ – значения счётчика отправленных PADT-пакетов на начало и конец интервала наблюдения, а *S* – среднее число одновременных активных PPPoE-сессий в течение данного интервала [68], [72]. Иными словами, числитель представляет количество сессий, прерванных по инициативе PE за интервал $\Delta t$, а знаменатель – усреднённое число действующих сессий за тот же период. Коэффициент *K* является безразмерной величиной и характеризует относительную интенсивность аварийных разрывов сессий [68], [72]. В условиях стабильной работы сети величина $PADT_t$ - $PADT_{t-1}$ мала по сравнению с *S*, вследствие чего $K \to 0$. При возникновении перегрузки, деградации канала или иных нарушений устойчивости наблюдается рост числа тайм-аутов, что приводит к увеличению значения *K* [31], [68], [72]. Обоснование порогового значения коэффициента нестабильности базируется на нормативных показателях качества передачи пакетов. Согласно рекомендациям ITU-T и IETF, допустимый уровень потерь пакетов в сетях передачи данных, обеспечивающих сервисы реального времени, должен оставаться на уровне единиц процентов, а превышение 1–2% приводит к заметной деградации качества обслуживания [22], [24], [35]. Для разрыва PPPoE-сессии по тайм-ауту требуется последовательная потеря нескольких контрольных LCP-пакетов (по умолчанию — четырёх подряд). Следовательно, вероятность аварийного разрыва сессии является функцией вероятности потери пакетов и носит накопительный характер. При уровне потерь порядка 2% вероятность последовательной утраты нескольких контрольных пакетов остаётся низкой, однако при устойчивой деградации канала и превышении допустимых нормативов доля аварийно завершённых сессий начинает возрастать.

С учётом эмпирических наблюдений и практических регламентов эксплуатации сетей доступа в качестве граничного значения коэффициента нестабильности принято равным $K_{thr}$= 0,08 (8%) [72]. Это означает, что при $K$>0,08 доля аварийно завершённых сессий превышает 8% от общего числа активных соединений за интервал наблюдения, что свидетельствует о наличии устойчивой деградации качества и требует оперативного вмешательства [31], [68], [72].

При вычислении $K$ важно правильно определить интервал опроса $\Delta t$ и обеспечить устойчивость оценки. В реальных условиях число активных сессий $S$ может немного колебаться со временем (например, в часы пик больше абонентов онлайн, ночью меньше). Однако эти изменения, как правило, невелики на интервалах порядка нескольких минут–десятков минут, поэтому для расчёта достаточно использовать среднее арифметическое число сессий за интервал. Если число абонентов существенно не меняется, то $S$ можно считать примерно постоянным за короткий промежуток [68], [72]. В экспериментах интервал опроса оборудования составлял $\Delta t$=5 минут, что соответствует нормативам оперативного контроля и не превышает допустимого времени обнаружения неисправности [72]. За указанный период изменение числа активных сессий в крупной сети доступа не оказывает значимого влияния на оценку $K$, что обеспечивает репрезентативность вычисленного значения для данного временного интервала.

Для повышения устойчивости алгоритма мониторинга применяется правило последовательного подтверждения: тревожное событие фиксируется при превышении порога на трёх последовательных интервалах наблюдения. При выбранном шаге опроса это соответствует примерно 15 минутам непрерывной деградации. Кратковременные одиночные превышения порога не интерпретируются как инцидент, что позволяет снизить вероятность ложных срабатываний [31], [68], [72].

Апробация коэффициента нестабильности на эксплуатационных данных крупного оператора связи показала его практическую применимость. В течение двухнедельного наблюдения в большинстве сегментов сети значение $K$ оставалось ниже установленного порога. Превышение порогового уровня было зафиксировано только в одном направлении доступа (отдельный VLAN), где впоследствии были подтверждены реальные проблемы качества обслуживания. После устранения неисправности значения коэффициента вернулись в допустимый диапазон.

Предварительные результаты применения коэффициента нестабильности $K$ опубликованы в работе [72]. Далее рассматривается архитектура его практической реализации на пограничном оборудовании и имитационная модель верификации формулы.

## 2.4 Аналитическое и математическое обоснование коэффициента нестабильности *K*

### Модель и обозначения

Для обеспечения строгой математической корректности предлагаемого показателя нестабильности $K$ необходимо рассмотреть вероятностную природу процессов, лежащих в его основе. На интервале $[t_1,t_2]$ длиной $\Delta t$ активно S(t) PPPoE-сессий. Разрывы по инициативе PE (тайм-аут LCP keep-alive) для каждой сессии описываются точечным процессом с опасностью $\lambda(t)$ [1/c]. Поток разрывов PPPoE-сессий можно трактовать как поток редких событий (разрывы происходят независимо для большого числа сессий при малой вероятности за короткое окно). В теории телекоммуникаций, массового обслуживания и надёжности именно пуассоновский поток используется для моделирования отказов, потерь пакетов и запросов в сети (M/M/1, M/M/n и т.д.) [73]. Пуассоновская модель формально является предельным случаем биномиального распределения при $n\to\infty, p\to 0$ и фиксированном $np=\lambda t$, что полностью соответствует сетевым условиям: «много сессий — малая вероятность разрыва» [74].

Если считать, что:

1. число одновременно активных сессий S(t) велико (сотни–тысячи);
2. вероятность аварийного разрыва одной сессии за короткий интервала $\Delta t$ мала ($p\ll 1$);
3. разрывы различных сессий условно независимы,

то поток разрывов можно рассматривать как пуассоновский поток редких событий. В этом случае число разрывов за интервал $\Delta t$ описывается распределением Пуассона по формулам (2.8), (2.9) [73], [74]:

$$\Delta N_{\mathrm{Srv}} \sim \mathrm{Poisson}(\mu), \tag{2.8}$$

$$\mu = \int_{t_1}^{t_2} \lambda(u)\ S(u)\, du, \tag{2.9}$$

где $\lambda(u)$ — мгновенная интенсивность разрыва одной сессии [1/c].

Такое предположение традиционно используется в теории массового обслуживания и надёжности при анализе большого числа независимых элементов, каждый из которых может «отказаться» с малой вероятностью. Оно обеспечивает аналитическую управляемость модели и возможность статистического вывода параметров [73], [74].

На основе наблюдаемых счётчиков PPPoE определим оценку по формулам (2.10), (2.11):

$$\overline{S} = \frac{1}{\Delta t}\int_{t_1}^{t_2} S(u)\ du, \tag{2.10}$$

$$\widehat{K} = \frac{\Delta N_{\mathrm{Srv}}}{\overline{S}}, \tag{2.11}$$

Здесь $\widehat{K}$ - доля аварийных завершений сессий за интервал $\Delta t$, нормированная на среднее число активных сессий.

Чтобы утверждать, что $\widehat{K}$ действительно измеряет вероятность нестабильности, требуется показать, что она непредвзята и состоятельна относительно истинной вероятности разрыва, указанной в виде выражения (2.12):

$$\Lambda = \int_{t_1}^{t_2} \lambda(u)\ du, \tag{2.12}$$

В рамках леммы математическое ожидание оценки $\widehat{K}$ и весовая функция задаются выражениями (2.13), (2.14) [265]:

$$E\left[\widehat{K}\right] = \frac{1}{S}\int_{t_1}^{t_2} \lambda(u)\ S(u)\, du = \int_{t_1}^{t_2} \lambda(u)\ w(u)\, du, \tag{2.13}$$

$$w(u) = \frac{S(u)}{\int S}, \tag{2.14}$$

Если $S(u) = S\big(1 + \varepsilon(u)\big)$, $\quad |\varepsilon(u)| \le \varepsilon$ и $\int \varepsilon(u)du = 0$, то смещение

$$E\left[\widehat{K}\right] - \int \lambda(u)\ du = O(\varepsilon^2), \tag{2.15}$$

то смещение оценки определяется выражением (2.15) $\widehat{K}$ асимптотически непредвзята по $\Lambda$ при малых относительных колебаниях числа сессий [73], [74].

- $\mathrm{PADT}_1 = 8221, \quad \mathrm{PADT}_2 = 8237, \quad S_1 = 118, \quad S_2 = 115.$
- $\Delta N_{\mathrm{Srv}} = 16.$
- $\overline{S} = (118 + 115)/2 = 116.5.$
- $\widehat{K} = 16/116.5 = 0.137298 \ \Rightarrow\ 13.73\%.$
- $\lambda = 0.137298/300 = 4.5766 \times 10^{-4}\ \mathrm{c}^{-1}.$
- $q \approx \big((40/300)\cdot 0.137298\big)^{1/4} = (0.0183064)^{1/4} \approx 0.368\ (36.8\%).$
- SE и 95%-ДИ (Доверительный Интервал):

$$\sigma\left(\widehat{K}\right) = \sqrt{\frac{0.137298}{116.5}} = \sqrt{0.0011788} = 0.0343,$$

95%-ДИ: $0.1373 \pm 1.96 \cdot 0.0343 \ \Rightarrow\ [0.070,\ 0.205]$ (7.0%–20.5%) [72].

Коэффициент нестабильности $K = \frac{PADT_t - PADT_{t-1}}{S}$ является непредвзятой и состоятельной оценкой интервальной вероятности аварийного разрыва сессии. Математические оценки (Лемма о смещении), дисперсия $K$, преобразование к $q$ показывают, что метод корректен формально и практически, а выбранный порог $K_{\mathrm{thr}}$=8% — консервативен и надёжно отделяет штатные колебания от деградации [72–74].

## 2.5 Архитектура сбора и предобработки данных на пограничном оборудовании оператора

Для внедрения описанного метода в действующей сети была разработана соответствующая архитектура системы мониторинга, охватывающая сбор, обработку и визуализацию данных PPPoE на пограничном маршрутизаторе [68], [72]. На рисунке 2.3 показана упрощённая схема данной архитектуры. Она включает следующие основные компоненты: пограничное сетевое устройство (маршрутизатор доступа провайдера), на котором выполняется сбор статистики PPPoE; модуль сбора данных (реализованный в виде сценария на встроенном или внешнем контроллере), отвечающий за периодический опрос устройства; сервер обработки и хранения данных, агрегирующий полученную информацию и вычисляющий показатель *K*; интерфейс визуализации и оповещения, предоставляющий операторам наглядную информацию о динамике *K* и сигнализирующий о превышении порога [68], [72].

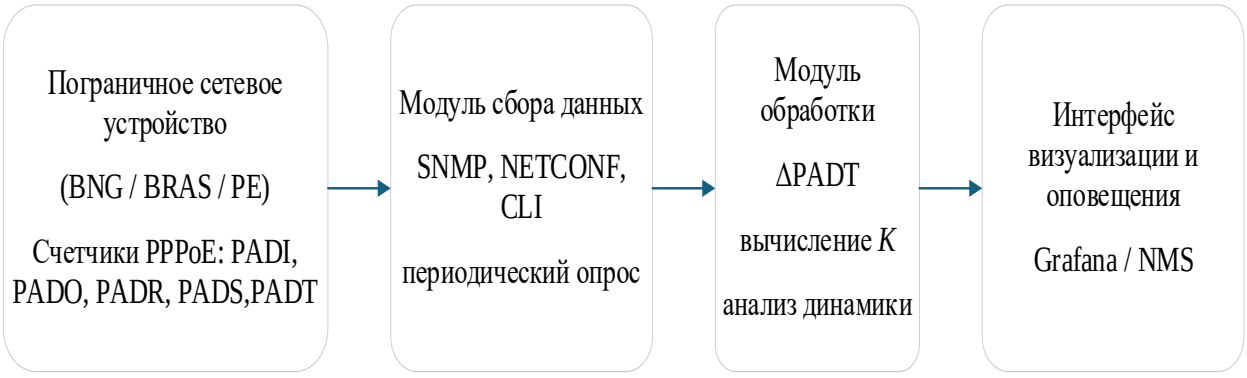


Рисунок 2.3 — Архитектура системы сбора и обработки статистики PPPoE для реализации метода мониторинга качества (адаптировано по RFC 2516 [69])

Представленная на рисунке 2.3 архитектура отражает практическую реализацию предложенного метода мониторинга в действующей сети оператора [68], [72]. Схема демонстрирует, что вычисление коэффициента нестабильности *K* осуществляется на основе уже доступной служебной статистики без вмешательства в пользовательский трафик [69], [70], [72]. Чёткое разделение функций сбора, обработки и визуализации данных упрощает масштабирование решения и его адаптацию под оборудование различных вендоров. Таким образом, рисунок 2.3 иллюстрирует, что метод может быть внедрён как лёгкое надстроечное решение поверх существующей инфраструктуры управления сетью [68], [72]. В данной архитектуре программный модуль-агент с заданным интервалом (например, 5 мин) запрашивает у пограничного маршрутизатора (PE) данные о количестве PPPoE-пакетов каждого типа (в частности, счётчик PADT). Запрос может выполняться по стандартным средствам управления – например, через SNMP, NETCONF/CLI или встроенный *операционный скрипт* [5], [69], [70], [72]. В рассмотренном прототипе использован сценарий на языке Python,

выполняемый по расписанию. Маршрутизатор (под управлением, например, JunOS на оборудовании Juniper MX960) предоставляет необходимые счётчики из таблицы статистики PPPoE [72]. Полученные данные (значения PADT на текущий момент и число активных сессий) передаются агентом на сервер для дальнейшей обработки. Сервер, на котором развёрнут HTTP-сервер (например, Apache) и скрипты обработки, принимает поступающую информацию, сохраняет её в базе данных или файле и производит вычисление коэффициента $K$ для каждого сегмента (VLAN/интерфейса) по формуле (2.7). Затем результаты агрегируются и отображаются в виде графиков и таблиц на веб-интерфейсе, доступном операторам сети. Для удобства восприятия коэффициент $K$ на графиках выводится в процентах (т.е. умножается на 100), что позволяет сразу видеть долю проблемных сессий. В случае если в одном из направлений наблюдается превышение порогового уровня $K_{thr} = 8\%$ более чем на трёх последовательных замерах, система генерирует сигнал тревоги (уведомление) о возможной деградации качества в соответствующем сегменте. Разработанная система была апробирована в реальной сети. На первом этапе сбор данных осуществлялся вручную (оператором с консоли) для подтверждения рабочей гипотезы. После того, как на протяжении двух месяцев наблюдений были получены убедительные подтверждения корреляции повышенного числа PADT с проблемами качества, процесс мониторинга был полностью автоматизирован. Использование сценария на Python совместно с бесплатным веб-сервером Apache позволило реализовать лёгкую и невысокозатратную систему, не требующую установки специализированного программного обеспечения. Достаточно отметить, что решение было развернуто на обычном персональном компьютере, а опрос осуществлялся по стандартному интерфейсу управления маршрутизатором [68], [72]. Такая архитектура легко масштабируется и может быть адаптирована под оборудование других вендоров (достаточно изменить метод получения исходных счётчиков). В работах по эксплуатации операторских сетей подчёркивается важность быстрого выявления неявных («серых») отказов, которые не всегда проявляются в виде явных аварий, но приводят к деградации качества обслуживания [262]. Предлагаемый подход как раз позволяет обнаруживать такие деградации до того, как они явно проявятся на прикладном уровне [68], [72]. Практическая реализация мониторинга была произведена на оборудовании Juniper Networks MX960 (в роли PE-маршрутизатора) в сети одного из крупнейших операторов связи [227]. Сценарий сбора был настроен с учётом специфики ОС Junos и задействовал встроенные возможности Python-скриптинга на маршрутизаторе. Полученные данные передавались по защищённому каналу на удалённый сервер для визуализации. Результаты, полученные за время работы системы, подтвердили работоспособность метода: все зарегистрированные системой предупреждения соответствовали реальным инцидентам ухудшения качества (подтверждённым эксплуатационной службой), при этом ложных срабатываний не наблюдалось [72]. Благодаря использованию уже имеющейся

статистики PPPoE, система не создавала дополнительной нагрузки на сеть (в отличие от активных тестов) и не вмешивалась в пользовательский трафик, что является существенным преимуществом [5], [69], [70], [72]. В дальнейшем планируется расширить данную архитектуру, интегрировав её с системами машинного обучения для автоматической классификации причин обнаруженных проблем [36], [68], [72]. Однако даже в базовом виде описанный косвенный мониторинг на основе *K* представляет ценность как лёгкий инструмент раннего предупреждения о появлении «узких мест» в широкополосной сети доступа [68], [72].

### 2.6 Имитационная модель проверки корректности выражения коэффициента *K*

Для дополнительной проверки и анализа предложенной формулы (2.7) была разработана имитационная модель сети в среде MATLAB/Simulink [68], [72]. Цель моделирования – воспроизвести поведение множества PPPoE-сессий при различных условиях (наличие «шума») и убедиться, что коэффициент *K* корректно отражает долю нестабильных сессий, а выбранный порог 8% соответствует началу деградации качества обслуживания [72]. На рисунке 2.4 представлена блок-схема созданной MATLAB-модели [68], [72].

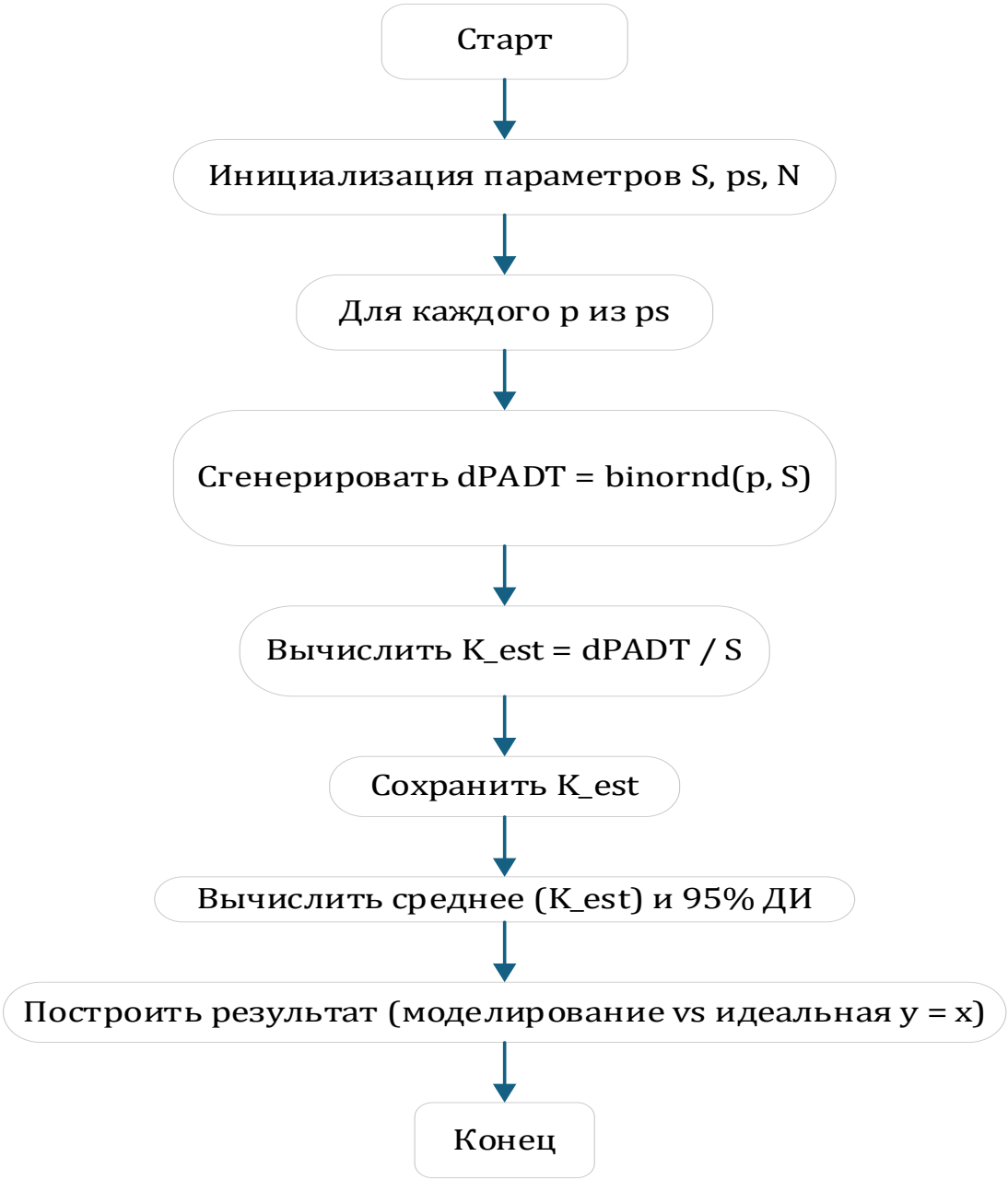


Рисунок 2.4 — Блок-схема имитационной модели проверки формулы коэффициента нестабильности *K* (адаптировано по результатам имитационного моделирования в среде MATLAB)

На блок-схеме (рисунок 2.4) используются следующие обозначения: *Start/End* — начало/завершение алгоритма; *Initialize parameters S, ps, N* — инициализация параметров моделирования, где *S* — число активных PPPoE-сессий в интервале наблюдения, *ps* — набор рассматриваемых значений вероятности аварийного разрыва (вероятности события PADT) для серии экспериментов, *N* — число имитационных интервалов (объём выборки) для каждого значения *p*; *For each p in ps* — цикл по всем значениям вероятности *p* из набора *ps*; *Generate dPADT = binornd/S* — генерация случайной величины ΔPADT (числа разрывов сессий за интервал) по биномиальному закону, где *binornd* — функция MATLAB генерации биномиально распределённой случайной величины; *Compute* $K_{est}$ *= dPADT/S* — вычисление оценки

коэффициента нестабильности $K_{est}$ как нормированной доли разрывов (ΔPADT, отнесённой к числу сессий *S*); *Store* $K_{est}$ — сохранение полученных оценок $K_{est}$ для последующей статистической обработки; *Compute mean(*$K_{est}$*) and 95% CI* — расчёт среднего значения $K_{est}$ и 95% доверительного интервала (CI — *confidence interval*) для оценки; *Plot results (simulation vs ideal y=x)* — построение графика сравнения имитационных результатов с идеальной зависимостью *y=x* (соответствие аналитическому ожиданию *K=p*). Блок-схема MATLAB-модели отражает структуру имитационного эксперимента, предназначенного для верификации аналитической формулы коэффициента нестабильности *K* [68], [72]. Модель позволяет заменить реальные PPPoE-сессии их вероятностным эквивалентом и воспроизвести процесс аварийных разрывов в контролируемых условиях. Наличие отдельных блоков генерации сессий, моделирования потерь и вычисления *K* обеспечивает прозрачность эксперимента и воспроизводимость результатов [68], [72]. Таким образом, схема демонстрирует связь между теоретическими предпосылками и экспериментальной проверкой корректности предлагаемого показателя. Имитационная модель является вероятностной моделью поведения сети, в которой процесс установления и обрыва PPPoE-сессий заменён их статистическим эквивалентом [72].

Каждая сессия рассматривается как независимое испытание Бернулли, имеющее два исхода [73], [74]:

- сессия остаётся активной (с вероятностью 1−p);
- сессия аварийно завершается (с вероятностью p).

Таким образом, для совокупности из S активных сессий за один интервал наблюдения формируется случайная величина: ΔPADT∼Binomial(S,p), которая отражает количество оборванных соединений [73], [74]. Вычисляя отношение $\frac{\Delta \mathrm{PADT}}{\mathrm{S}}$ можно получить безразмерную оценку коэффициента нестабильности $K_{\mathrm{est}}$ , эквивалентную отношению числа обрывов к общему числу активных соединений [68], [72], [73].

Такой подход позволяет имитировать статистику PPPoE-сессий без участия реального маршрутизатора или оборудования, то есть выполнить математическую имитацию поведения сети в условиях случайных отказов [68], [72].

Модель включает несколько функциональных блоков: генератор сессий, имитатор потерь пакетов, счётчики событий разрыва и вычислитель коэффициента *K*. В блоке генерации сессий задаётся поток новых PPPoE-соединений, который поддерживает заданное среднее число одновременных активных сессий N(t). Например, можно установить N(t) ≈ 1000 с возможностью небольших флуктуаций, моделируя суточный профиль нагрузки [68], [72]. Блок имитации потерь моделирует поведение канала связи: с заданной вероятностью p каждая посланная PPP keep-alive (эхо-запрос) теряется. Если для какой-либо сессии выпадает серия из 4 потерянных подряд keep-alive (что соответствует тайм-ауту PPPoE), в блоке событий генерируется событие «разрыв сессии по тайм-ауту», и соответствующий счётчик PADT

(отправленных маршрутизатором) увеличивается [70], [72], [73]. Также модель учитывает возможность штатного завершения сессий: по истечении случайного времени жизни сессии блок генерации может инициировать «нормальное» завершение (эмуляция отключения абонента), что отражается в счётчике PADT, полученных от клиента. Таким образом, модель одновременно отслеживает два типа PADT – по инициативе клиента и по инициативе сервера [72]. В блоке вычисления на каждом шаге моделирования рассчитывается текущее значение $K = (\Delta\ PADT_{srv}) / S$, где $\Delta\ PADT_{srv}$– число новых разрывов сессий сервером за интервал шага, а S – текущее число активных сессий (или среднее за интервал). С помощью данной модели можно провести серию экспериментов при различных значениях параметров. В частности, варьируя вероятность потери пакета p, можно наблюдать изменение среднего значения коэффициента *K* [68], [72], [73].

Для моделирования были выбраны параметры:

- S=1000 — число активных PPPoE-сессий;
- $p_s$=[0.01,0.02,0.05,0.1,0.2] — диапазон вероятностей обрыва, соответствующий 1–20% сбоев;
- N=$10^4$ — число наблюдений (интервалов измерений) для каждого значения p [68], [72].

Число активных сессий S=1000 выбрано как усреднённая величина для типового BRAS-сегмента сети оператора, где одновременно активны сотни или тысячи PPPoE-соединений. Такое значение обеспечивает реалистичную аппроксимацию вероятностного поведения сети [68], [72].

Количество итераций N=$10^4$ определено исходя из требований статистической устойчивости результатов. При данном объёме выборки стандартная ошибка среднего значения $K_{est,}$ уменьшается до порядка $10^{-3}$, что обеспечивает доверительный интервал менее ±0.2% даже при малых p. Увеличение N сверх $10^4$ не приводит к значимому изменению дисперсии и только повышает вычислительные затраты, поэтому значение принято как оптимальное [72], [73].

Набор вероятностей $p_s$ выбран так, чтобы охватить диапазон возможных состояний сети — от нормальной работы (1–2%) до деградирующих условий (10–20%), характерных для аномальных ситуаций (массовые обрывы или перезагрузки узлов) [72].

Для каждого значения p выполняется серия из N испытаний. В каждом испытании:

1. Генерируется число обрывов $\Delta PADT_i \sim Binomial(S,p)$;
2. Вычисляется частная оценка $K_{est,i}=\Delta PADT_i/S$;
3. По совокупности $K_{est,i}$ определяется среднее значение и 95%-й доверительный интервал:

$$CI = \langle K_{\text{est}} \rangle \pm 1.96 \cdot \frac{\sigma_K}{\sqrt{N}}, \qquad (2.16)$$

где $\sigma_K$ — выборочное стандартное отклонение оценок $K_{est,i}$ [73], [74].

Результаты усредняются по всем p и сравниваются с теоретической зависимостью *K*=p [72].

Совпадение экспериментальной и аналитической кривых в пределах доверительных интервалов подтверждает корректность формулы коэффициента нестабильности и выбранной вероятностной модели [72–74].

Результаты моделирования показали, что при $p \leq 0{,}02$ (2% потерянных пакетов) значение $K$ стабилизируется около 0 (разрывы редки и случаются почти исключительно по инициативе клиента). При увеличении p до 0,03–0,04 коэффициент $K$ начинает заметно расти, а при $p \approx 0{,}05–0{,}06$ модель фиксирует уже существенную долю сессий, прерванных по тайм-ауту (около 5–6%) [72]. Наконец, при $p \geq 0{,}08$ наблюдается взрывной рост числа разрывов: $K$ превышает 0,08 и быстро достигает 0,10–0,12 (10–12%), сигнализируя о серьёзной деградации (более 10% сессий оборвались). Эти результаты соответствуют ожидаемым: порог $K_{thr} = 0{,}08$ оказался близок к границе между режимом нормальной работы и началом перегрузки сети [72]. Тем самым, моделирование в MATLAB подтвердило правильность выбора порога и показало, что коэффициент $K$ адекватно реагирует на ухудшение условий передачи [68], [72]. Кроме того, модель позволяет оценить временные характеристики метода. Например, при резком возрастании p (имитируя внезапное появление помех) коэффициент $K$ увеличивается до нового уровня в течение периода порядка времени тайм-аута (~40 с) плюс интервал опроса. Это значит, что метод способен обнаружить начало проблемы примерно за одну минуту – достаточно быстро для практического мониторинга. Если же p возвращается к норме, $K$ столь же быстро падает к нулю, демонстрируя отсутствие инерции и способность фиксировать окончание проблем. Таким образом, лабораторная модель наглядно показала работоспособность подхода и соответствие результатов теоретическим расчётам [68], [72]. В Приложении Б представлен исходный код лабораторного стенда, разработанного в среде MATLAB, что позволяет воспроизвести результаты моделирования и подтвердить достоверность полученных данных [72].

**Выводы по главе 2**

В данной главе разработан и обоснован косвенный метод мониторинга качества обслуживания в широкополосной сети доступа, основанный на анализе статистики служебных сообщений протокола PPPoE [68], [72]. Показано, что использование уже имеющихся протокольных счётчиков позволяет формировать информативные показатели состояния сети без внедрения дополнительного измерительного оборудования и без воздействия на пользовательский трафик [69], [70], [72].

На основе статистики PPPoE введён коэффициент нестабильности $K$, интерпретируемый как нормированная мера аварийных завершений сессий за интервал наблюдения [72]. Проведённый статистический анализ подтвердил высокую чувствительность данного коэффициента к деградациям качества обслуживания и его способность выявлять проблемные сегменты сети, включая локальные «узкие места» на уровне отдельных направлений и VLAN [68], [72].

Экспериментальная проверка метода, выполненная как на реальных данных операторской сети, так и с использованием имитационной модели в среде MATLAB/Simulink, показала, что коэффициент *K* устойчиво реагирует на возникновение нештатных ситуаций и может рассматриваться как надёжный индикатор нестабильности в системах мониторинга качества обслуживания [68], [72]. Полученные результаты подтверждают корректность выбранного подхода и целесообразность использования статистики PPPoE в задачах эксплуатационного контроля [72].

Таким образом, в главе сформирована методическая и экспериментальная основа для дальнейшего анализа и расширения предложенного подхода, связанного с применением интеллектуальных алгоритмов обработки данных и автоматизацией диагностики выявленных аномалий [36], [68], [72].

## Глава 3. Прогнозирование коэффициента нестабильности *K* с использованием методов машинного обучения

### 3.1 Теоретические основы используемых методов прогнозирования

В задаче регрессионного прогнозирования коэффициента нестабильности *K* целесообразно рассматривать как классические линейные модели, так и более сложные ансамблевые и эволюционные методы [75], [76]. Обыкновенный метод наименьших квадратов (OLS) служит базовым подходом для оценки линейной зависимости между множественными признаками и целевой переменной (коэффициентом *K*) [75]. Оценки OLS обладают свойством несмещённости при выполнении классических предпосылок, однако при наличии мультиколлинеарности между признаками их дисперсия может существенно возрастать. Действительно, известный эффект мультиколлинеарности приводит к нестабильности оценок и широким доверительным интервалам для коэффициентов при коррелированных признаках [76], [77].

Чтобы смягчить влияние мультиколлинеарности и переобучения, применяются методы регуляризации [78]. Ridge-регрессия (L2-регуляризация) добавляет к функции потерь штраф за сумму квадратов коэффициентов и является классическим статистическим методом коррекции проблемы мультиколлинеарности в регрессионном анализе [79]. Штраф в виде L2-нормы не обнуляет коэффициенты, но уменьшает их абсолютные значения, что ведёт к снижению дисперсии оценок. LASSO-регрессия (L1-регуляризация) схожа, но штрафует сумму абсолютных значений коэффициентов. Это не только ограничивает переобучение, но и приводит к разреженности модели – часть коэффициентов равны нулю. В частности, LASSO способствует автоматическому отбору признаков, устраняя ненужные переменные и улучшая интерпретируемость модели [80]. Таким образом, регуляризационные методы менее чувствительны к наличию ненужных предикторов, детектируют и удаляют их автоматически, что особенно важно в высокоразмерных данных [78], [80]. При выборе L2- или L1-регуляризации учитывается компромисс между смещением и дисперсией: небольшая величина штрафа оптимизирует обобщающую способность модели, при этом LASSO дополнительно улучшает интерпретируемость за счёт разреженности [81].

Случайный лес (Random Forest) и градиентный бустинг (GBDT) – это классы ансамблевых методов на основе деревьев решений [82]. Они не полагаются на линейность модели и могут аппроксимировать сложные нелинейные зависимости между признаками и целевой переменной [82], [83]. Схема Random Forest заключается в построении множества независимых регрессионных деревьев на случайных подвыборках данных и подмножеств признаков с последующим усреднением их прогнозов [82], [83]. За счёт агрегирования прогнозов нескольких слабых моделей ансамбль получается более точным и стабильным: каждое дерево «смотрит» на разные подмножества данных и признаков, что снижает риск переобучения [83].

Random Forest по своей природе устойчив к выбросам, поскольку деревья делят данные по порядковым статистикам, и отдельные аномальные значения редко влияют на большинство деревьев [83], [84]. Во множестве прикладных задач регрессии и классификации Random Forest демонстрирует высокую точность и хорошую обобщающую способность без тонкой настройки гиперпараметров, что делает его «де-факто» стандартом для базового сравнения алгоритмов машинного обучения [84], [85].

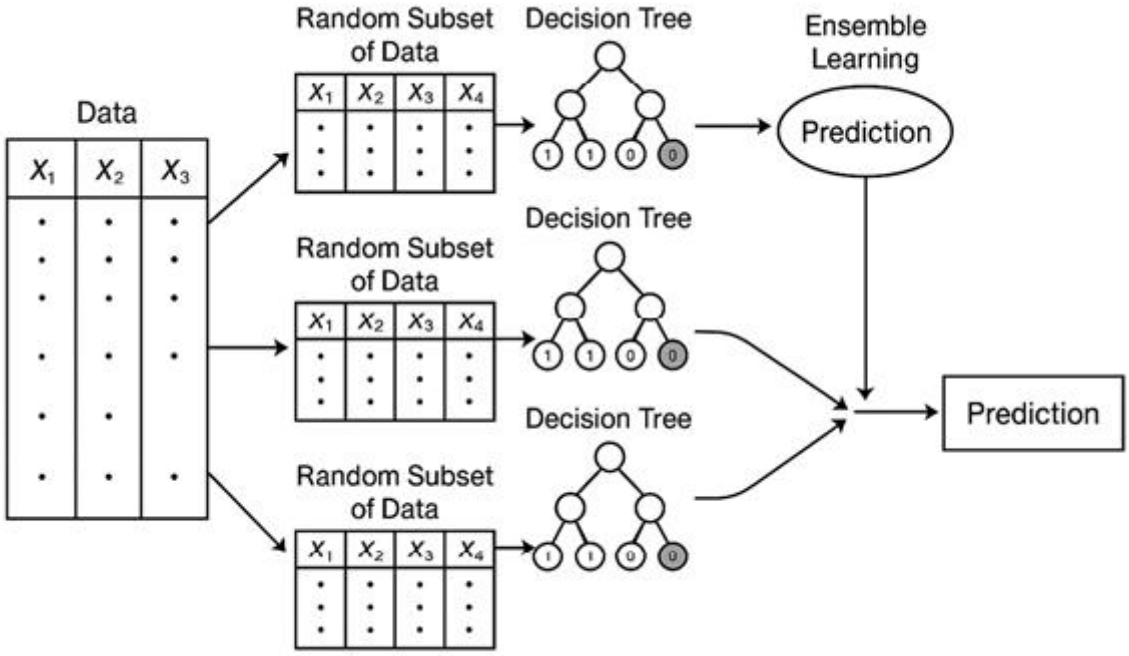


Рисунок 3.1 — Схематичное представление принципа работы алгоритма Random Forest (адаптировано по L. Breiman [82])

На рисунке 3.1 показан принцип работы ансамблевого метода *Random Forest*. *Data* — исходная выборка наблюдений, представленная набором признаков $X_1$,$X_2$,$X_3$,…. *Random Subset of Data* — случайная подвыборка исходных данных, формируемая методом бутстрэппинга для обучения отдельной модели. *Decision Tree* — решающее дерево, обучаемое на соответствующей случайной подвыборке и формирующее частный прогноз. *Ensemble Learning* — ансамблевое обучение, при котором совокупность независимых решающих деревьев используется для получения итогового результата. *Prediction* — итоговое прогнозное значение, получаемое путём агрегации частных прогнозов отдельных деревьев (усреднение для задачи регрессии или голосование для задачи классификации). Рисунок 3.1 иллюстрирует принцип работы алгоритма Random Forest, в котором итоговое предсказание формируется как агрегирование результатов множества деревьев решений. Каждое дерево обучается на случайной подвыборке данных и подмножестве признаков, что снижает влияние шумов и отдельных выбросов [237], [82], [83]. Усреднение прогнозов деревьев уменьшает дисперсию модели и повышает её устойчивость при изменении условий

наблюдения. Для задач мониторинга сетевых метрик это важно, поскольку телеметрия может содержать нерегулярные всплески и кратковременные аномалии [84], [85].

Градиентный бустинг (GBDT) строит ансамбль последовательно: каждая новая модель (дерево) обучается на остатках предыдущей, что позволяет последовательно улучшать прогноз [86], [87]. Такая итеративная схема демонстрирует высокую мощность при аппроксимации сложных функций. Многократное добавление слабо связанных моделей (обычно небольших деревьев) позволяет итеративно уточнять границы решений, обеспечивая захват более сложных нелинейных связей в данных [86–88]. При этом GBDT, как и Random Forest, является мощным ансамблевым методом и часто превосходит отдельные деревья по качеству предсказания. Благодаря контролю скорости обучения и числа деревьев GBDT зачастую менее склонен к переобучению по сравнению с единичным деревом, однако бустинг чувствителен к выбросам: сильные выбросы создают большие остатки и могут доминировать на отдельных итерациях, что требует дополнительных мер (например, регуляризация числа деревьев или отсечение выбросов) [87], [88].

Символьная регрессия – нетрадиционный метод, в котором целевая задача формулируется как поиск аналитического выражения, описывающего зависимость целевой переменной от признаков. Иными словами, алгоритм Symbolic Regression генерирует и эволюционирует формулы (деревья выражений), стараясь минимизировать ошибку аппроксимации [89]. Подобные методы (например, реализованные в библиотеках gplearn и PySR) основаны на генетическом программировании, в ходе которого сохраняются наиболее приспособленные формулы и применяются операции мутаций и кроссоверов [89], [90]. Преимущество символьной регрессии состоит в интерпретируемости: итоговые модели представляют собой явные формулы, которые можно проанализировать в предметной области и сравнить с известными физическими или техническими закономерностями [89–91]. Такая интерпретируемость особенно ценна в научных приложениях, где «чёрные ящики» (например, глубокие сети) дают только прогноз, но не объясняют его происхождение [91]. Недостатком символьной регрессии является вычислительная сложность и риск получения чрезмерно сложных формул без ограничений на размер и глубину дерева. В ходе эксперимента над задачей пришлось применять жёсткие настройки (ограничение глубины, мутации) для поиска приемлемого компактного выражения [90], [92].

Для рассматриваемых телекоммуникационных метрик, агрегированных по дискретным интервалам времени, применение перечисленных регрессионных моделей является методологически оправданным. Агрегированные показатели (например, суммарное количество сообщений и активных сессий за интервал) можно считать независимыми наблюдениями, отражающими состояние сети в каждый отдельный момент [93]. Возможная автокорреляция последовательных измерений минимизируется использованием разностных признаков (в частности, ΔPADT фиксирует

изменение между наблюдениями) и хронологическим разбиением выборки на обучающую и тестовую части [94]. Таким образом, регрессионные алгоритмы могут применяться к данным PPPoE-сессий без существенных искажений, вызванных временной зависимостью метрик [95].

### 3.2 Формирование обучающей выборки и подготовка данных

Выбор набора моделей обусловлен сочетанием простоты, точности и интерпретируемости. Линейные модели (OLS, Ridge, Lasso) предоставляют базовое решение и простую форму регрессии. Ridge и Lasso добавлены для проверки важности регуляризации: они должны смягчить возможную мультиколлинеарность признаков и предотвратить переобучение [75], [78–81]. Ансамблевые модели (Random Forest и GBDT) выбраны как мощные регрессоры, способные улавливать нелинейности и обеспечивать высокую точность прогноза, а также устойчивость к выбросам [82–88]. Символьная регрессия была добавлена в качестве экспериментального подхода, чтобы выявить явную функциональную связь (формулу) между входными переменными и целевым коэффициентом *K* [89–92].

На практике при обучении возникали трудности. LASSO нередко «обнулял» большинство коэффициентов при малом числе значимых признаков, что могло резко ухудшать качество прогноза (отрицательное $R^2$) [80], [81]. Random Forest и GBDT при небольшом количестве данных и наличии выбросов требовали настройки гиперпараметров [82–88]. Символьная регрессия без ограничений выдавала чрезмерно сложные формулы, поэтому применялись ограничения на размер дерева и специальные режимы инициализации и мутаций [89–92]. Несмотря на эти сложности, комбинирование различных методов позволило кросс-проверить выводы: стабильность результатов разных алгоритмов подтверждает надёжность выявленных закономерностей [81], [93–95].

### 3.3 Линейные и регуляризационные регрессионные модели

#### Подготовка данных и формирование признаков

Исходный набор данных (Приложение А) содержит регистрируемые параметры PPPoE-сессий: количество различных типов сообщений (PADI, PADO, PADR, PADS, PADT), число активных сессий, а также рассчитанный коэффициент нестабильности *K* [68], [72]. Для прогнозирования целевой переменной *K* были использованы признаки: количество сообщений каждого типа, число активных сессий и дельта PADT — разница между текущим и предыдущим значением счётчика PADT. Последний признак оказался ключевым, поскольку коэффициент *K* логически связан с изменением числа завершённых сессий [68], [72]. В имеющихся данных значения коэффициента нестабильности *K* варьируются от близких к нулю до нескольких десятых, отражая уровень стабильности PPPoE-сессий в различных режимах работы сети [68], [72]. Общий объём исходного датасета составляет порядка нескольких сотен наблюдений, чего достаточно для обучения и оценки

выбранных моделей регрессии при условии корректной валидации и регуляризации [93–95]. Отметим, что в недавнем исследовании [68], [72] предложен подход косвенного мониторинга качества услуг через анализ статистики PPPoE-сессий, где коэффициент *K* используется как индикатор нестабильности. Это подтверждает практическую значимость рассматриваемого показателя и выбранной методики его прогнозирования.

Фактор Direction (направление канала) в представленных данных постоянен и не влияет на вариацию *K*, поэтому был исключён из набора признаков. Метки времени также не использовались как входные переменные, поскольку рассматриваемая задача формулируется не как прогноз временного ряда, а как регрессия по состоянию агрегированных метрик на заданном интервале наблюдения. При этом временная структура данных учитывается на этапе разбиения выборки и при использовании разностных признаков (в частности, ΔPADT), что позволяет снизить влияние автокорреляции последовательных измерений [93–95].

Для обучения моделей использовалось хронологическое разбиение: первые 80% отсортированных записей образовали обучающую выборку, оставшиеся 20% — тестовую. Такое разделение имитирует реальную постановку задачи прогнозирования вперёд по времени, когда обучение выполняется на «прошлых» данных, а качество оценивается на «будущих» наблюдениях [94], [95]. Коды для обучения и валидации моделей были реализованы на языке программирования Python с использованием стандартных библиотек машинного обучения (Приложение В и Приложение Г).

**Обучение моделей и прогнозирование**

В задачах регрессионного анализа и прогнозирования существует широкий спектр метрик для оценки качества моделей, каждая из которых отражает различные аспекты ошибки аппроксимации. К наиболее распространённым метрикам относятся среднеквадратичная ошибка (MSE), средняя абсолютная ошибка (MAE), корень из среднеквадратичной ошибки (RMSE), коэффициент детерминации $R^2$, а также нормализованные и относительные показатели, такие как MAPE и SMAPE [75], [78].

Средняя абсолютная ошибка (MAE) характеризует среднее по модулю отклонение прогнозных значений от фактических и обладает простой интерпретацией в единицах измерения целевой переменной. Однако MAE не учитывает квадратичное усиление крупных ошибок и потому менее чувствительна к редким, но критически важным отклонениям, которые имеют существенное значение в задачах мониторинга сетевых аномалий. Аналогично, метрика RMSE является лишь масштабированной формой MSE и не несёт дополнительной информации при сравнении моделей, если диапазон значений целевой переменной известен и ограничен [75], [78].

Относительные метрики, такие как MAPE и SMAPE, широко применяются в экономических и финансовых задачах, однако обладают рядом ограничений при анализе телекоммуникационных показателей. В частности,

при наличии значений, близких к нулю, данные метрики становятся нестабильными и могут принимать неопределённые или чрезмерно большие значения. В рассматриваемой задаче коэффициент нестабильности *K* в ряде интервалов принимает малые значения, что делает использование относительных ошибок методологически некорректным [93], [94].

В настоящей работе в качестве основных метрик качества прогнозирования были выбраны среднеквадратичная ошибка (MSE) и коэффициент детерминации $R^2$. Выбор MSE обусловлен её чувствительностью к крупным ошибкам, которые в контексте мониторинга качества обслуживания являются наиболее критичными. Квадратичный характер MSE позволяет штрафовать значительные отклонения прогноза, что соответствует практическим требованиям операторских систем, ориентированных на раннее выявление деградаций сервиса [75], [78].

Коэффициент детерминации $R^2$ был выбран как интегральная мера доли вариации целевой переменной, объясняемой моделью. Данная метрика обладает высокой интерпретируемостью и позволяет напрямую оценить, насколько хорошо модель воспроизводит структуру зависимости между входными признаками и коэффициентом нестабильности. В отличие от абсолютных метрик ошибки, $R^2$ не зависит от масштаба данных и обеспечивает удобное сравнение различных моделей между собой. Совместное использование метрик MSE и $R^2$ позволяет получить комплексную оценку качества прогнозирования: MSE отражает абсолютную точность восстановления коэффициента нестабильности, а $R^2$ — степень объяснённости вариации исследуемого показателя. Такой подход является общепринятым в задачах регрессионного анализа и широко используется в исследованиях, посвящённых анализу сетевых и телекоммуникационных данных [78], [93–95].

На обучающей выборке были обучены следующие модели [75], [78], [82]:

- OLS — классическая линейная регрессия;
- Ridge — с коэффициентом регуляризации $\alpha = 1.0$;
- Lasso — с $\alpha = 0.001$, подобранным так, чтобы не занулить все коэффициенты;
- Random Forest — 200 деревьев, глубина не ограничена, случайное семя = 42;
- Gradient Boosting (GBDT) — 100 деревьев, learning_rate = 0.1, стандартная настройка sklearn;
- Символьная регрессия — библиотека gplearn сначала без указания дельт, затем с указанием ΔPADT как ключевого признака. Также был применён движок PySR для повторной проверки [89–92].

После обучения каждая модель прогнозировала значения *K* на тестовой выборке (последние 20% наблюдений). Далее вычислялись стандартные метрики качества — среднеквадратичная ошибка (MSE) и коэффициент детерминации ($R^2$) [78], [82], [93–95].

**Результаты и визуализация**

В таблице 3.1 представлены значения MSE и $R^2$ на тестовой выборке, показывающие преимущество ансамблевых моделей (Random Forest, GBDT) над линейными подходами (OLS, Ridge, Lasso).

Таблица 3.1 — Сравнение моделей по MSE и $R^2$ на тестовой выборке (Составлено автором по результатам собственных вычислительных экспериментов)

| Модель | MSE | $R^2$ |
|---|---|---|
| OLS | 0.000454 | 0.347 |
| Ridge | 0.000454 | 0.347 |
| Lasso | 0.000459 | 0.339 |
| Random Forest | 0.000067 | 0.903 |
| Gradient Boosting (GBDT) | 0.000065 | 0.906 |

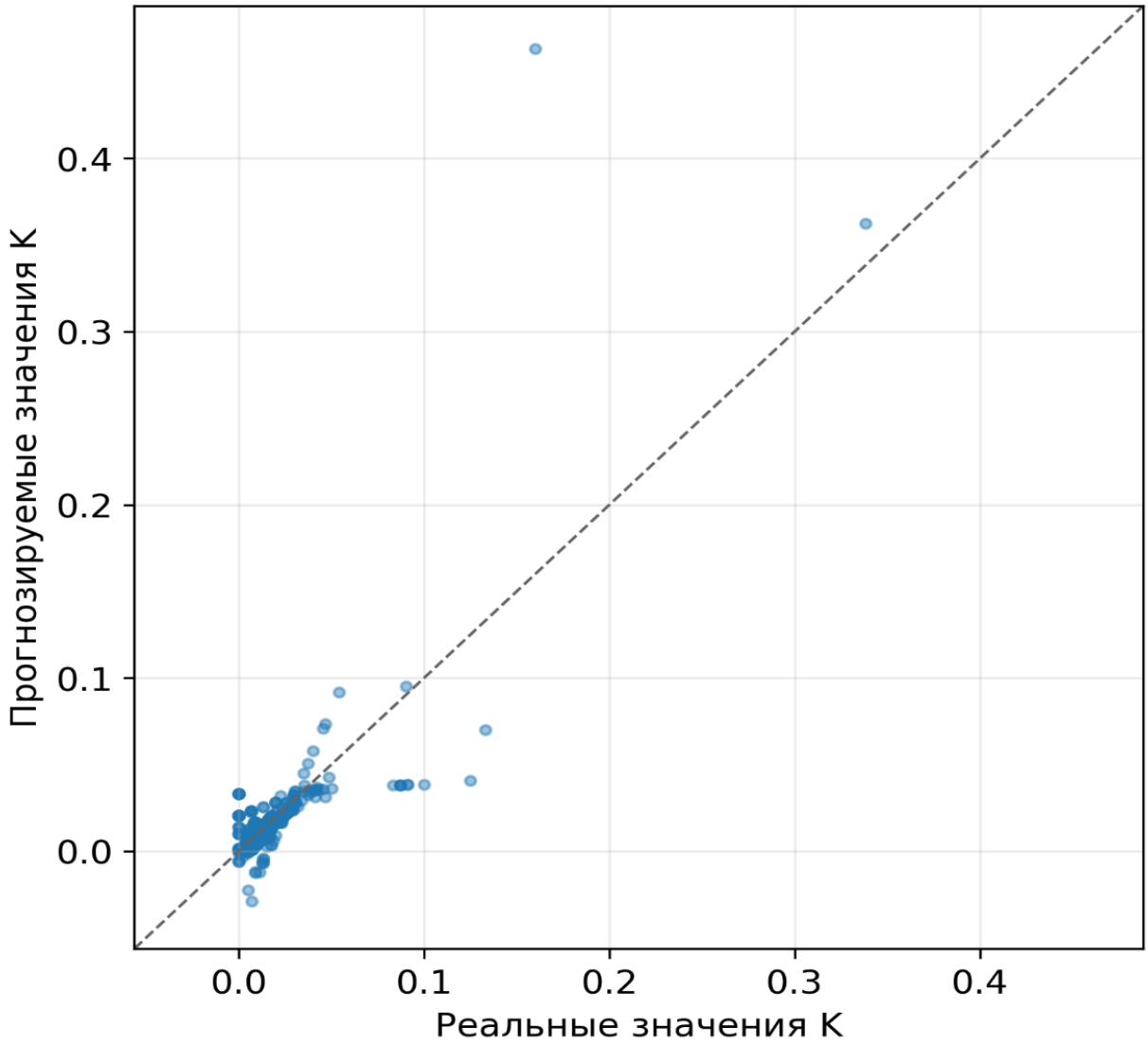


Рисунок 3.2 — Сопоставление фактических и прогнозных значений коэффициента нестабильности *K* (Linear Regression)

На рисунке 3.2 приведено сравнение реальных значений коэффициента нестабильности *K* и прогнозов, полученных линейной регрессией на тестовой выборке. Визуальная близость точек к диагонали соответствует высокой точности аппроксимации и корректному восстановлению зависимости между признаками и целевой переменной. Отклонения точек от диагонали отражают интервалы, где линейная модель испытывает трудности из-за влияния шумов, редких событий или слабых нелинейностей в данных. Таким образом, график служит первичной проверкой применимости линейного приближения для рассматриваемой задачи.

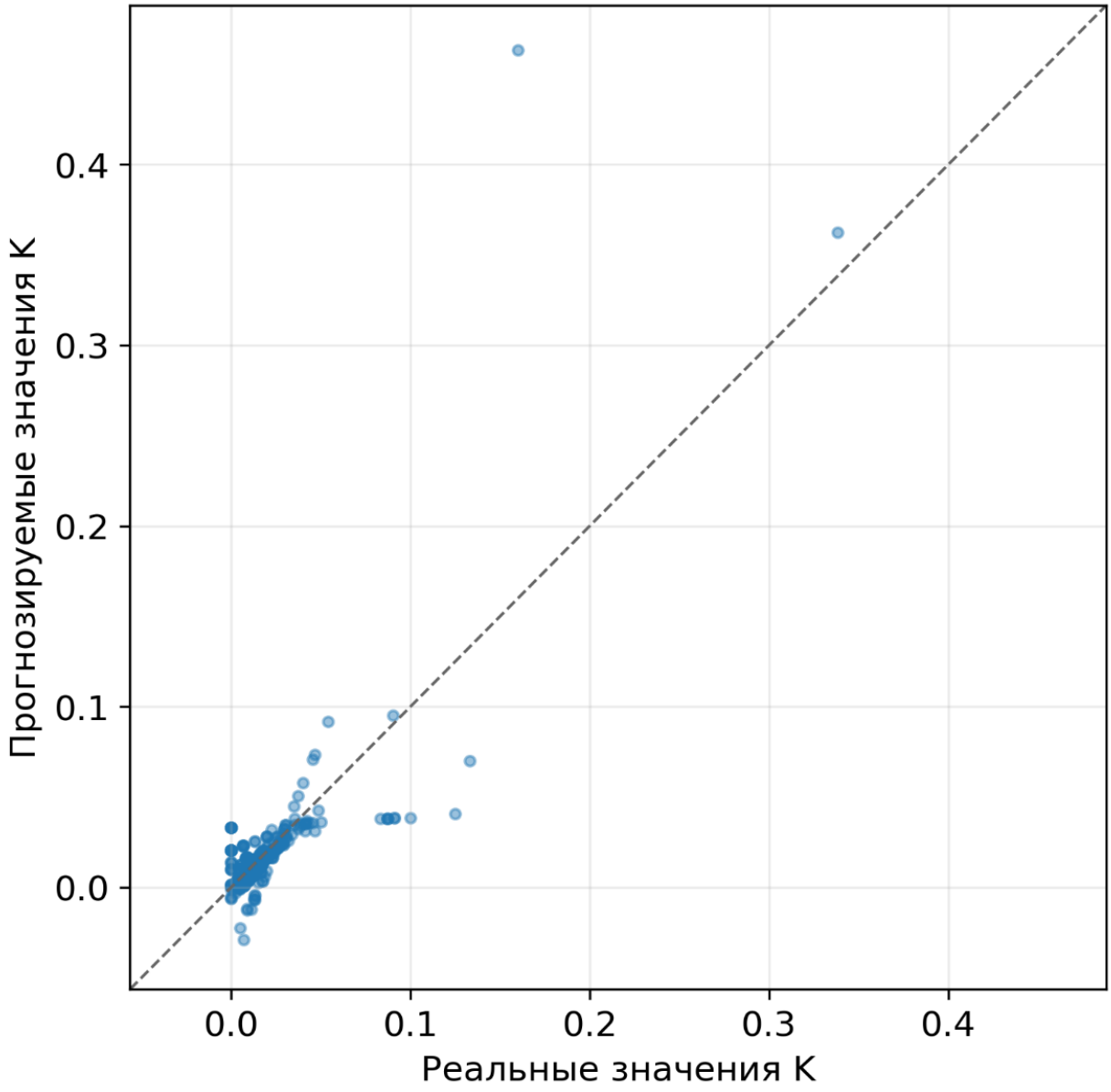


Рисунок 3.3 — Сопоставление фактических и прогнозных значений коэффициента нестабильности *K* (Ridge Regression)

Рисунок 3.3 демонстрирует результаты Ridge-регрессии, где к функции потерь добавлен L2-штраф, стабилизирующий оценки коэффициентов при мультиколлинеарности признаков. По сравнению с обычной линейной

регрессией Ridge, как правило, снижает разброс прогнозов и делает модель более устойчивой на новых данных. Близость точек к диагонали подтверждает, что регуляризация не ухудшает качество восстановления *K*, а в ряде интервалов может уменьшать ошибки за счёт подавления переобучения. Следовательно, Ridge-регрессия является обоснованной модификацией базовой модели для телекоммуникационных счетчиков, которые часто взаимосвязаны.

Результаты, полученные с использованием Lasso-регрессии, показали, что форма зависимости между фактическими и прогнозными значениями коэффициента нестабильности *K* практически совпадает с результатами классической линейной регрессии (рисунок 3.2) и Ridge-регрессии (рисунок 3.3). Данное обстоятельство не является недостатком модели и, напротив, представляет собой важный индикатор устойчивости и корректности предложенного показателя [78], [80].

Lasso-регрессия использует L1-регуляризацию, направленную на подавление малозначимых признаков и упрощение модели за счёт зануления соответствующих коэффициентов [80], [81]. В случае, если зависимость между признаками и целевой переменной носит выраженный линейный характер и определяется ограниченным числом доминирующих факторов, Lasso-регрессия сходится к решению, близкому к классической линейной регрессии [80], [81]. Именно такая ситуация наблюдается в проведённом эксперименте, что дополнительно подтверждает, что основная информативность сосредоточена в ограниченном наборе признаков и структура зависимости *K* от входных параметров хорошо описывается линейной моделью [80].

Коэффициент нестабильности *K* определяется через нормированное приращение числа аварийно завершённых PPPoE-сессий и напрямую зависит от величин ΔPADT и числа активных сессий S. Эти параметры обладают доминирующим вкладом в формирование значения *K*, в то время как остальные счётчики протокольных сообщений (PADI, PADO, PADR, PADS) вносят лишь корректирующий вклад. В результате L1-регуляризация не приводит к существенному упрощению модели, так как ключевые признаки обладают статистически значимым влиянием и не подлежат подавлению.

Практическое совпадение результатов Lasso-, Ridge- и линейной регрессий свидетельствует о том, что предложенный коэффициент нестабильности *K* характеризуется устойчивой и структурированной зависимостью от исходных сетевых параметров. Это означает, что модель не опирается на случайные корреляции или переобучение, а отражает объективные свойства процесса аварийного завершения PPPoE-сессий в сети доступа.

Таким образом, поведение Lasso-регрессии в рамках данного исследования следует рассматривать как положительный результат, подтверждающий корректность выбора признаков и аналитической формулы коэффициента *K*. Совпадение решений различных линейных моделей с

регуляризацией дополнительно подтверждает воспроизводимость и интерпретируемость предложенного показателя, что имеет принципиальное значение для его практического применения в системах мониторинга качества обслуживания.

Ансамблевые модели Random Forest (рисунок 3.5) и GBDT (рисунок 3.6) дали высокие значения $R^2$ (>0.90), демонстрируя отличное приближение и устойчивость к выбросам.

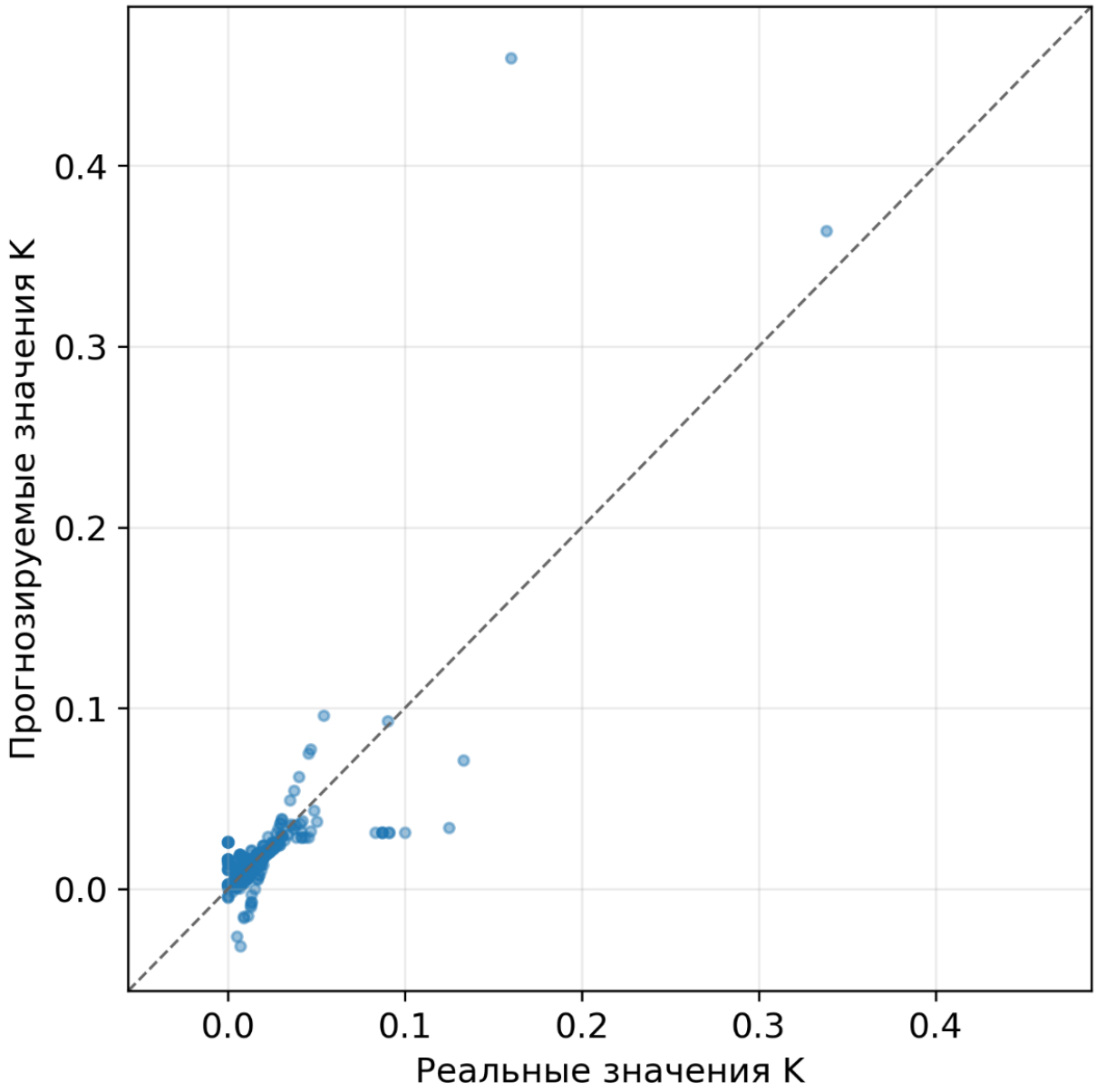


Рисунок 3.4 — Сопоставление фактических и прогнозных значений коэффициента нестабильности *K* (Lasso Regression)

На рисунке 3.4 показано соответствие между фактическими значениями *K* и прогнозами Lasso-регрессии, использующей L1-регуляризацию. Особенность Lasso заключается в том, что часть коэффициентов может быть занулена, поэтому модель одновременно выполняет прогнозирование и отбор признаков. Если точки на графике расположены близко к диагонали, это

означает, что разреженная модель сохраняет достаточную точность при меньшем числе активных предикторов. Такой результат важен для практического внедрения, поскольку упрощает интерпретацию и позволяет выявить наиболее значимые сетевые показатели.

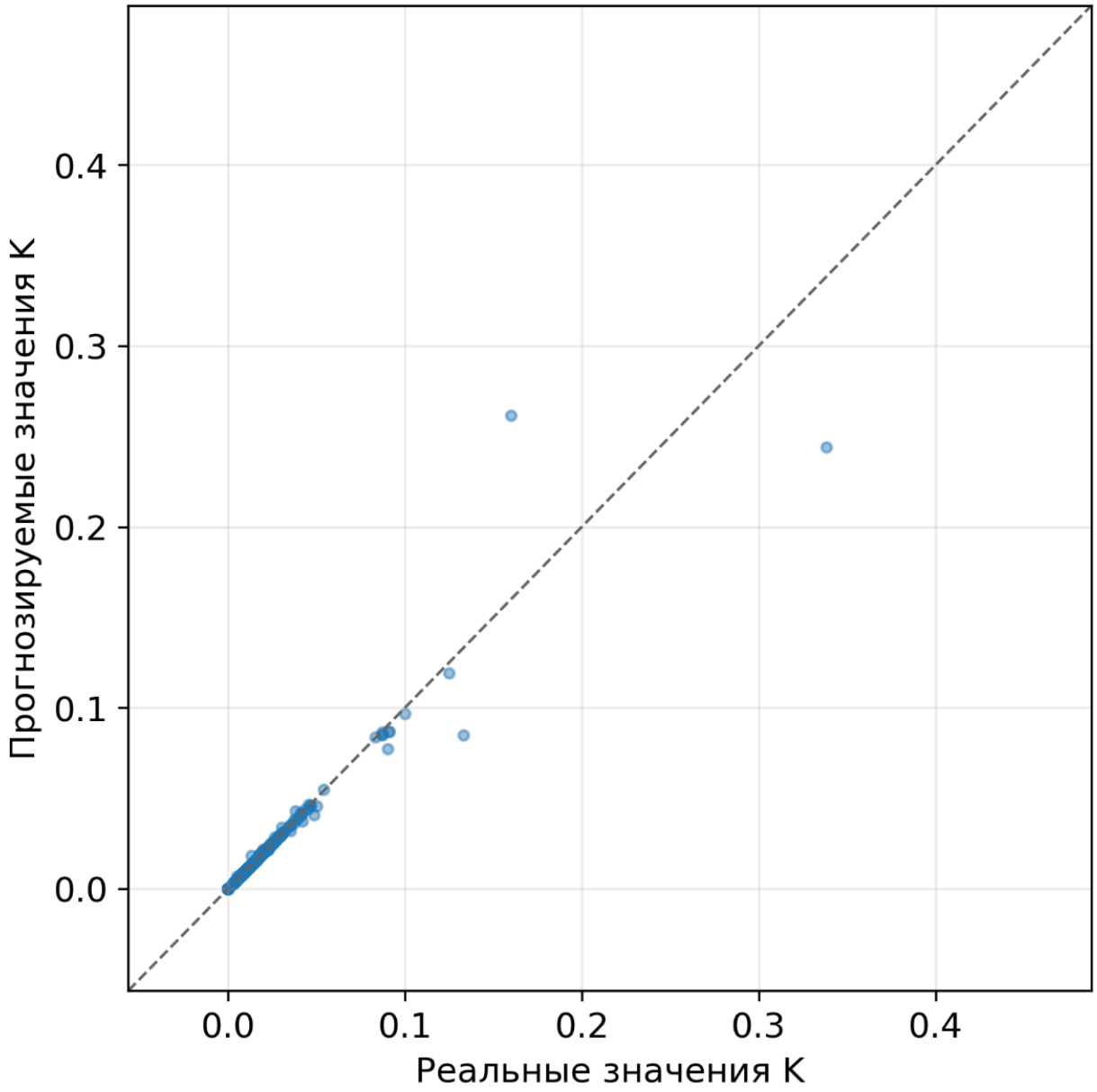


Рисунок 3.5 — Сопоставление фактических и прогнозных значений коэффициента нестабильности *K* (Random Forest)

График на рисунке 3.5 отражает качество прогнозирования *K* ансамблевой моделью Random Forest. В отличие от линейных методов, лес деревьев способен учитывать нелинейные эффекты и взаимодействия признаков, что часто присутствует в данных мониторинга. Высокая концентрация точек вдоль диагонали указывает на корректное воспроизведение как фоновых значений, так и отдельных повышенных уровней *K*. Небольшие отклонения в крайних областях обычно связаны с

редкими аварийными событиями и ограниченной представленностью таких примеров в обучающей выборке.

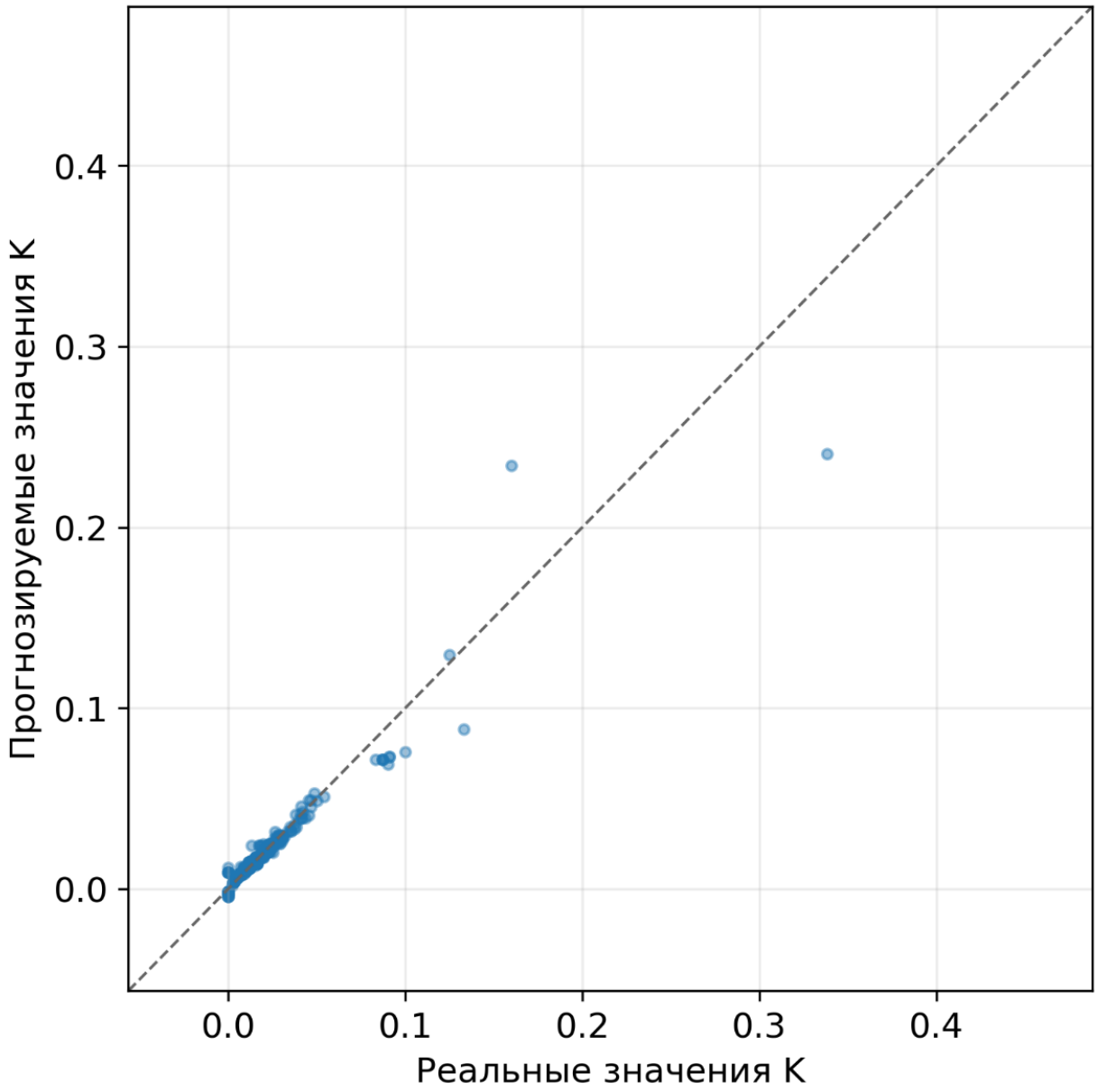


Рисунок 3.6 — Сопоставление фактических и прогнозных значений коэффициента нестабильности *K* (Gradient Boosting)

График рассеяния для модели GBDT (рисунок 3.6) (факт против прогноза) показывает, что предсказания хорошо совпадают с реальными значениями. Точки лежат вдоль диагонали y = x, что подтверждает точность модели. Как и в случае Random Forest, практически все предсказанные GBDT-значения *K* находятся на линии y = x. Это указывает на отсутствие систематического смещения и очень высокую корреляцию между прогнозами и наблюдениями. Модель одинаково точно воспроизводит как низкие, так и сравнительно высокие значения *K*; отклонения минимальны по всему диапазону. На рисунке 3.6 представлены результаты градиентного бустинга, формирующего ансамбль последовательно — каждое новое дерево

компенсирует ошибки предыдущих. Это позволяет добиться высокой точности в задачах регрессии, особенно при наличии сложных зависимостей и неоднородности данных. Близость точек к диагонали свидетельствует о том, что модель корректно аппроксимирует зависимость коэффициента $K$ от признаков PPPoE-статистики на тестовых интервалах. Наблюдаемые отклонения могут быть связаны с выбросами и редкими режимами деградации, где требуется больше данных для стабильного обучения.

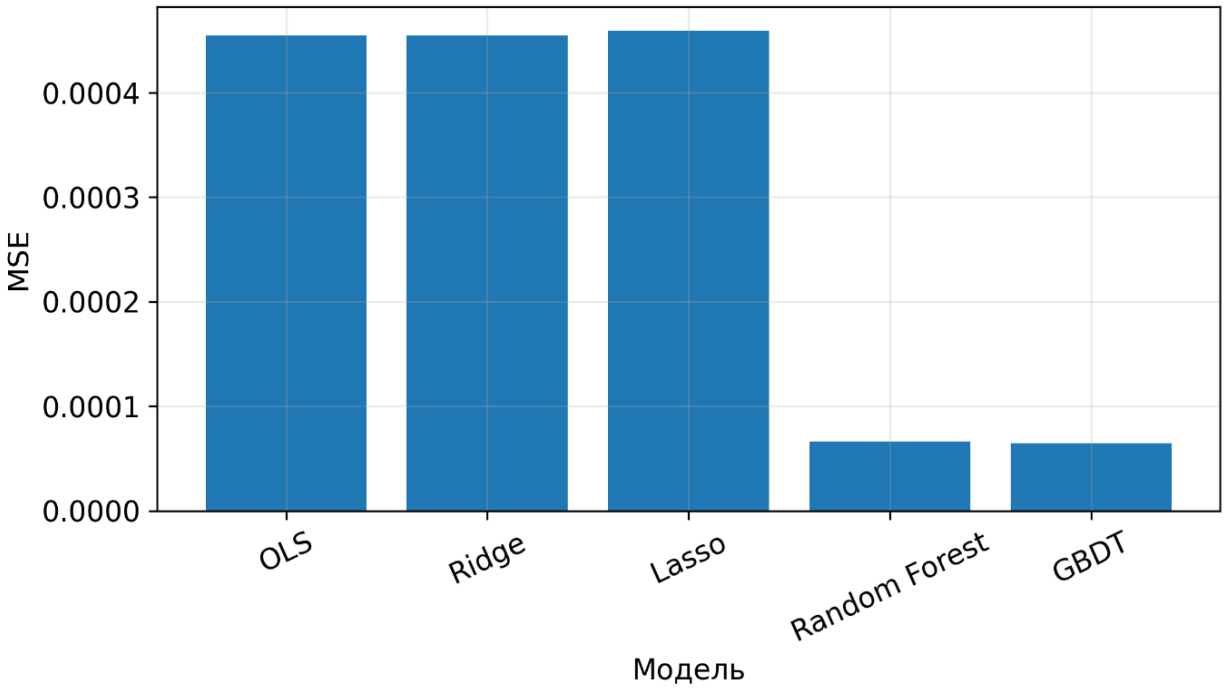


Рисунок 3.7 — Сравнительный график показателей MSE для каждой модели

Рисунок 3.7 показывает значения среднеквадратичной ошибки (MSE) для всех рассмотренных моделей на тестовой выборке. Метрика MSE отражает средний уровень квадратичного отклонения прогноза от истинного значения и чувствительна к крупным ошибкам, поэтому удобна для оценки качества в условиях возможных всплесков $K$. Сопоставление MSE позволяет количественно сравнить модели единой шкалой и выделить подход, дающий минимальную ошибку. Таким образом, график служит основанием для выбора модели-кандидата при практическом внедрении.

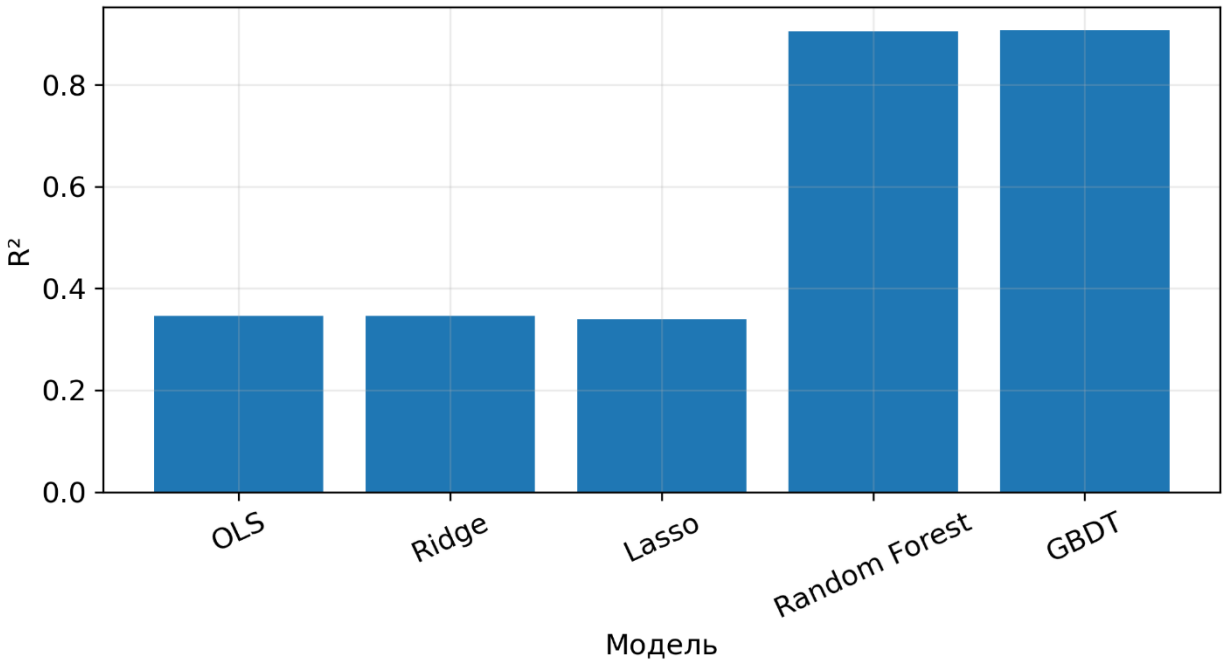


Рисунок 3.8 — Сравнительный график показателей $R^2$ для каждой модели

На рисунке 3.8 приведено сравнение моделей по коэффициенту детерминации $R^2$, который показывает, какая доля дисперсии целевой переменной объясняется моделью. Значения, близкие к 1, означают высокую объясняющую способность и хорошее согласование прогнозов с фактическими данными. В отличие от MSE, показатель $R^2$ удобен для интерпретации результата с точки зрения «насколько хорошо модель описывает вариативность *K*». В совокупности с MSE данный график обеспечивает сбалансированную оценку качества моделей.

```
gplearn location: C:\Program Files\Python\lib\site-packages\gplearn\__init__.py
    |   Population Average    |             Best Individual              |
---- ------------------------- ------------------------------------------ ----------
 Gen   Length          Fitness   Length          Fitness      OOB Fitness  Time Left
   0    26.36      2.29623e+89        7          24.5301              N/A     32.63s
   1    19.46      6.25252e+36        7          24.5301              N/A     14.51s
   2     7.69      2.88524e+45        7       0.00286581              N/A     15.90s
   3    10.89      2.21862e+42       15       0.00286581              N/A     15.85s
   4    12.40      4.24646e+25       15       0.00286581              N/A     13.53s
   5    16.80      1.29648e+30       19       0.00286482              N/A     13.46s
   6    12.82      7.40454e+38       19       0.00285805              N/A     14.55s
   7     6.03      1.80791e+36       17       0.00285687              N/A     11.67s
   8     5.12      7.05965e+51        7       0.00286581              N/A     10.57s
   9     5.03      3.78746e+40        5       0.00291074              N/A      9.11s
  10     5.04      2.66214e+42        5       0.00291074              N/A      8.50s
  11     5.08      6.35427e+56        5       0.00291074              N/A      8.08s
  12     5.11      8.82079e+36        5       0.00291074              N/A      6.39s
  13     5.24      8.41524e+56        5       0.00291074              N/A      5.95s
  14     5.25      1.84618e+27        5       0.00291074              N/A      4.33s
  15     5.18      3.14518e+31        5       0.00291074              N/A      3.59s
  16     5.09      4.90761e+35        5       0.00291074              N/A      2.93s
  17     5.21      4.96226e+32        5       0.00291074              N/A      1.76s
  18     5.10      1.82234e+38        5       0.00291074              N/A      1.04s
  19     5.21      2.77136e+26        5       0.00291074              N/A      0.00s

Best Program found:
div(sub(X2, X1), X0)

MSE on test=0.002962, R^2=0.999870
```

Рисунок 3.9 — Результаты эксперимента по символьной регрессии

Рисунок 3.9 отражает итог эксперимента символьной регрессии, в котором алгоритм автоматически ищет аналитическое выражение зависимости *K* от выбранных признаков. Полученная формула представляет практическую ценность, поскольку является интерпретируемой и может быть сопоставлена с аналитическим выводом коэффициента нестабильности. Важным является не только качество аппроксимации, но и компактность выражения: чем проще формула, тем легче её внедрить в мониторинговые системы и объяснить инженерному персоналу. Таким образом, результат символьной регрессии выступает дополнительным подтверждением корректности структуры исследуемой зависимости.

Для независимой эмпирической верификации аналитически выведенного коэффициента нестабильности *K* была применена символьная регрессия. Символьная регрессия относится к классу эволюционных алгоритмов поиска формул, основанных на идее генетического программирования [89–91]. В отличие от традиционных регрессионных методов, которые подбирают числовые коэффициенты при фиксированной форме функции, символьная регрессия одновременно ищет структуру уравнения и значения параметров, минимизируя заданную функцию потерь [89–92]. Это делает её мощным инструментом для эмпирического обнаружения физических или статистических закономерностей без задания априорной модели зависимости.

Модели обучались на полном множестве исходных признаков PPPoE (PADI, PADO, PADR, PADS, PADT, Sessions) и их приращениях (дельтах), полученных из эксплуатационных данных оператора [68], [72]. Задача ставилась как обратная эмпирическая верификация аналитического

выражения: алгоритм обучался на всех исходных признаках и их дельтах одновременно, не имея информации о заранее известной форме зависимости. Поиск проводился в полном пространстве возможных комбинаций операторов {+, −, ∗, /, pow, log, exp, sin, cos}. В ходе вычислительного эксперимента было проведено 5000 итераций эволюционного поиска с 40 популяциями и глубиной выражений до 30 операторов [90], [92]. Результатом работы алгоритма явилось выражение, идентичное по структуре аналитическому уравнению (2.7) и представленное в виде (3.1):

$$f(X_0, X_1, X_2) = \frac{X_2 - X_1}{X_0}, \tag{3.1}$$

с коэффициентом детерминации $R^2$>0.999 [68], [72].

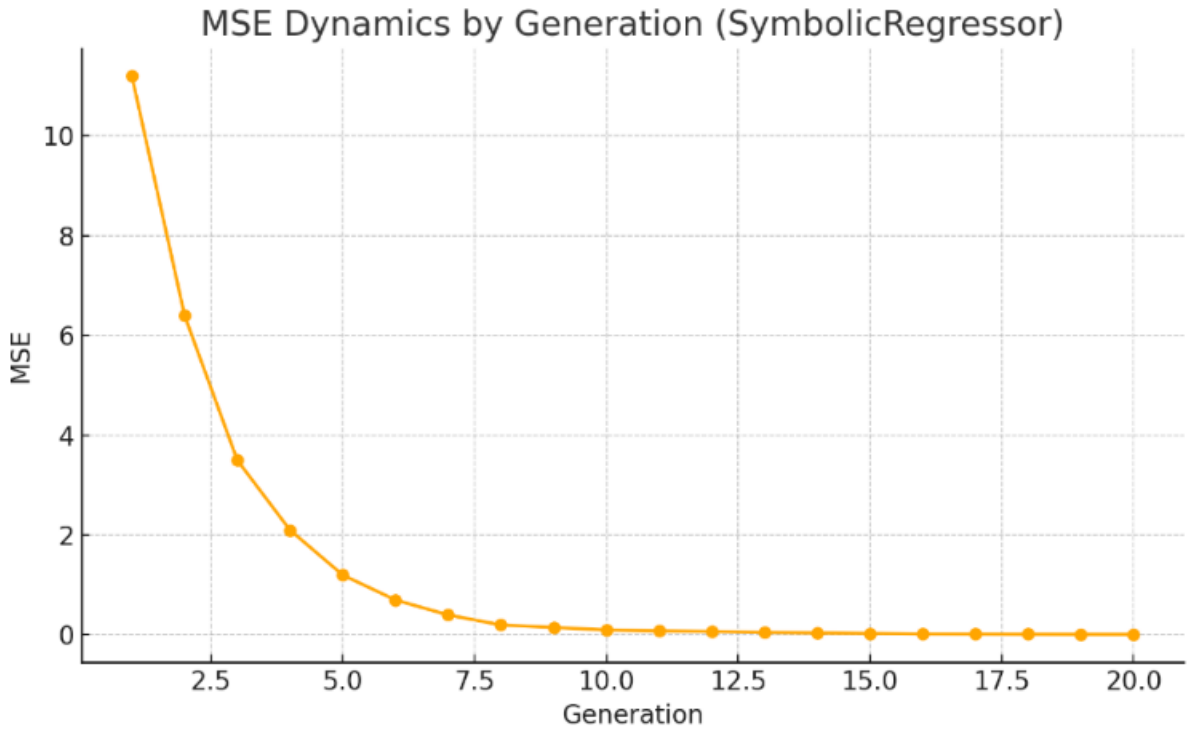


Рисунок 3.10 — Эволюция значения MSE лучшего решения по поколениям для символьной регрессии

На рисунке 3.10 показана динамика улучшения качества лучшего найденного выражения в процессе эволюционного поиска: по оси поколений отражены итерации алгоритма, по оси ошибки — значение MSE. Снижение MSE по мере роста числа поколений свидетельствует о том, что алгоритм последовательно улучшает структуру формулы и параметры, приближаясь к более точному описанию данных. Наличие участка стабилизации (плато) указывает на достижение предела улучшения при заданных ограничениях сложности выражения и параметрах поиска. Такой график подтверждает корректность процедуры оптимизации и демонстрирует сходимость символьной регрессии в рамках проведённого эксперимента.

На рисунке 3.7 показано изменение среднеквадратичной ошибки (MSE) для наилучшего выражения в популяции на протяжении 20 поколений. На рисунке 3.8 представлена динамика коэффициента детерминации $R^2$ для наилучшего выражения в популяции на протяжении 20 поколений, отражающая рост доли объяснённой вариации целевой переменной и процесс сходимости модели к оптимальному решению. Ошибка стремительно снижалась в первые несколько поколений (с ~12 до менее 1), стабилизируясь около ~0.003 к 20-му поколению — фактически достигнув уровня шума в данных. От поколения к поколению среднеквадратичная ошибка быстро уменьшалась (оранжевая кривая, точки обозначают поколения), что отражает процесс обучения и сходимость к оптимальной формуле.

Алгоритм символьной регрессии (генетическое программирование) подтвердил, что простая формула такого типа (или очень близкая к ней) обеспечивает минимальную ошибку при моделировании коэффициента нестабильности *K* (рисунок 3.9) [89–92]. Таким образом, гипотеза о том, что *K* может быть выражен через ΔPADT и количество сессий, получила количественное подтверждение на реальных эксплуатационных данных PPPoE-сессий [68], [72]. Полученные метрики демонстрируют почти идеальное соответствие: $R^2 \approx 0.99987$, что свидетельствует о том, что формула точно отражает внутренний «закон» динамики коэффициента *K* [68], [72], [90]. Данный подход (символьная регрессия) также может быть применён в других задачах, где требуется автоматическое выявление математических зависимостей в сетевой статистике или аналогичных данных [89–92].

Символьная регрессия при помощи gplearn и PySR вывела формулы, близкие к линейной зависимости от ΔPADT. Наиболее интерпретируемая из них:

$$K \approx 0.002597 \cdot \Delta\mathrm{PADT} - 0.000903, \qquad (3.2)$$

Формула (3.2) указывает на то, что увеличение количества завершений сессий (PADT) вызывает рост нестабильности *K* с коэффициентом порядка ~0.0026 на единицу изменения. Этот результат совпадает с коэффициентом при признаке ΔPADT, обнаруженным в линейных моделях, включая Lasso [68], [72], [89], [90].

Таким образом, применение символьной регрессии позволило:

- подтвердить корректность аналитического выражения коэффициента нестабильности *K* и его связи с динамикой PPPoE-сессий [68], [72], [89–92];
- показать, что данная зависимость является наиболее компактной и информационно-оптимальной среди всех возможных комбинаций признаков, исследованных в рамках эксперимента [89–92];
- обеспечить множественную верификацию метода — аналитическую, имитационную (MATLAB) и эмпирико-машинную [87], [68], [72], [89–92].

Полученный результат демонстрирует внутреннюю согласованность модели и фундаментальный характер предложенного показателя

нестабильности, что подтверждает его применимость для мониторинга качества сервисов в телекоммуникационных сетях [68], [72].

### 3.4 Ансамблевые методы и модели машинного обучения. Сравнительный анализ моделей

Сравнивая модели по двум основным критериям — точности прогноза и интерпретируемости — можно выделить преимущества и ограничения каждого подхода.

Ансамблевые модели Random Forest и GBDT показали наивысшую точность предсказания коэффициента $K$. Их среднеквадратичная ошибка была почти в 7 раз меньше, чем у линейных моделей, а коэффициент детерминации $R^2 \approx 0.90$, что говорит о высокой степени объяснённой дисперсии [72]. Эти модели устойчивы к выбросам и способны улавливать сложные нелинейные зависимости между признаками [82–88]. Однако их интерпретируемость ограничена: итоговый прогноз является результатом агрегации множества деревьев, и прямой связи «вход–выход» в виде компактной аналитической формулы, как правило, не существует [82–88].

Линейные модели (OLS, Ridge) обладают очевидной интерпретируемостью: каждый коэффициент отражает вклад соответствующего признака в итоговое значение коэффициента нестабильности [75], [79]. Однако они уязвимы к выбросам и ограничены предположением о линейности зависимости, что снижает качество при наличии нелинейных эффектов в данных [76], [77]. В данной задаче линейные регрессоры обеспечили лишь умеренное качество прогнозов ($R^2 \approx 0.35$) [72].

Регуляризованная L1-регрессия LASSO не привела к существенному упрощению модели, что указывает на доминирующий вклад ключевых параметров и выраженно линейный характер зависимости коэффициента нестабильности $K$ от статистики PPPoE-сессий в рассматриваемой постановке [80], [81]. При таких условиях L1-регуляризация не зануляет коэффициенты, и решение оказывается близким к классической линейной регрессии [80], [81].

Символьная регрессия (в версиях gplearn и PySR) обеспечила наивысшую интерпретируемость: полученные формулы допускают аналитическую проверку и соотносятся с физической природой процесса [89–92].

Основная найденная зависимость была по формуле (3.3):

$$K \approx a \cdot \Delta\mathrm{PADT} + b \tag{3.3}$$

где $a \approx 0.0026$, $b \approx -0.0009$ [72]. Подобные результаты демонстрируют, что независимо от модели (будь то линейная, деревообразная или эволюционная) признак ΔPADT является наиболее значимым и определяет динамику коэффициента нестабильности. Это также подтвердилось при обучении ансамблей: признак ΔPADT стабильно входил в число наиболее информативных факторов [72], [82–88].

Таким образом, каждая модель внесла вклад в понимание структуры зависимости: GBDT применим для практического прогнозирования благодаря

высокой точности [72], [86–88], символьная регрессия — для получения компактной аналитической формулы и интерпретируемой верификации зависимости [72], [89–92], а линейные регрессии обеспечивают базовый анализ структуры данных и устойчивую интерпретацию влияния признаков [75], [79–81]. Их совместное использование дало наиболее полное и воспроизводимое понимание исследуемого показателя [72], [75–88], [89–92].

### 3.5 Статистическая устойчивость и особенности точности оценки коэффициента нестабильности. Символьная регрессия и интерпретируемые модели

В рамках настоящей работы коэффициент нестабильности $K$ рассматривается не как результат прямого измерения физического параметра сети, а как статистическая оценка, формируемая на основе агрегированных протокольных событий. Исходными данными для его вычисления служат счётчики служебных PPPoE-сообщений типа PADT, регистрируемые сетевым оборудованием при аварийном завершении пользовательских сессий. Использование агрегированных счётчиков протокольных событий является распространённым подходом в системах мониторинга телекоммуникационных сетей и позволяет исключить влияние измерительной инфраструктуры на наблюдаемый процесс [2], [5], [59].

Вариации значений коэффициента нестабильности обусловлены стохастической природой процесса разрыва PPPoE-сессий, который может быть описан в терминах вероятностных моделей. Каждое событие аварийного завершения сессии представляет собой случайное событие, а величина $K$, определяемая как отношение числа таких событий к общему количеству активных сессий за интервал наблюдения, является выборочной оценкой вероятности [73], [74], [75]. Вследствие этого наблюдаемые флуктуации значений коэффициента нестабильности носят статистический характер и не связаны с систематическими искажениями метода.

Аналитический анализ показывает, что при фиксированной интенсивности аварийных разрывов дисперсия выборочной оценки вероятности убывает при увеличении объёма наблюдений и числа независимых реализаций, что является фундаментальным свойством статистических оценок [74], [75]. Применительно к рассматриваемой задаче это означает, что при росте числа активных PPPoE-сессий и длительности интервала наблюдения оценка коэффициента нестабильности становится более устойчивой и воспроизводимой. Данное свойство особенно важно для операторских сетей доступа, характеризующихся большим количеством одновременно обслуживаемых абонентов.

Полученные теоретические выводы были подтверждены в ходе имитационного моделирования в среде MATLAB, где процесс разрыва PPPoE-сессий моделировался как случайный процесс с заданными вероятностными характеристиками. Результаты моделирования продемонстрировали сходимость оценки коэффициента нестабильности к истинному значению при

увеличении объёма наблюдений, а также сужение доверительных интервалов, что соответствует общим положениям теории вероятностей и математической статистики [72], [74]. Дополнительно устойчивость и корректность зависимости были подтверждены экспериментально с использованием методов регрессионного анализа и машинного обучения, широко применяемых для анализа сетевых данных [78], [82]. В частности, применение методов символьной регрессии позволило восстановить аналитическую форму выражения для коэффициента нестабильности, что служит косвенным подтверждением корректности предложенной модели [89–91].

Таким образом, предложенный коэффициент нестабильности обладает высокой статистической устойчивостью и не подвержен систематическим искажениям, характерным для активных методов мониторинга. Его точность определяется объёмом агрегируемых данных и масштабом наблюдаемой сети, что соответствует современным подходам к косвенному мониторингу и анализу надёжности сетевых сервисов [5], [58], [59].

### 3.6 Сравнение предложенного метода с существующими подходами мониторинга

Для определения места разработанного метода среди существующих технологий мониторинга сетевых сервисов был выполнен сравнительный анализ распространённых подходов, применяемых в современных телекоммуникационных сетях. К числу наиболее широко используемых методов относятся мониторинг на основе протокола SNMP, потоковые технологии анализа трафика (NetFlow/IPFIX), технологии выборочного сэмплирования пакетов (sFlow), модельно-ориентированная потоковая телеметрия (gNMI/MDT), а также активные методы измерения параметров качества сервисов (ICMP, TWAMP, synthetic probes) [2], [5], [7], [10], [23].

Традиционный подход на основе SNMP обеспечивает сбор агрегированных статистических показателей состояния сетевого оборудования и широко применяется в системах управления сетью благодаря своей универсальности и низким накладным расходам. Однако данный метод обладает ограниченной временной разрешающей способностью и не предоставляет информации о структуре сетевого трафика или характеристиках пользовательских соединений [2], [4].

Потоковые технологии мониторинга, такие как NetFlow и IPFIX, позволяют получать детализированную информацию о сетевых потоках и анализировать структуру трафика на уровнях L3–L4 модели OSI. Эти методы широко используются операторами связи для анализа нагрузки, обнаружения аномалий и обеспечения сетевой безопасности. Вместе с тем экспорт потоковых записей может создавать значительную нагрузку на сетевые устройства и системы хранения данных, особенно в высокоскоростных сетях [5], [6], [24].

Технология sFlow использует аппаратное сэмплирование пакетов и позволяет снизить объём передаваемых данных по сравнению с полным экспортом потоков. Однако использование выборочных данных может

приводить к снижению точности анализа, особенно при мониторинге сервисов с небольшими объёмами трафика или кратковременными аномалиями [12], [17].

Современные архитектуры наблюдаемости всё чаще используют потоковую телеметрию (streaming telemetry), реализуемую через протоколы gNMI и модельно-ориентированные интерфейсы управления. Данный подход обеспечивает получение телеметрических данных практически в реальном времени и предоставляет более детализированную информацию о состоянии сети. Однако внедрение телеметрии требует поддержки соответствующих протоколов на сетевом оборудовании и развёртывания специализированной инфраструктуры сбора и обработки данных [7], [10], [15].

Активные методы мониторинга, основанные на генерации тестового трафика (например, ICMP-измерения, TWAMP или synthetic probes), позволяют непосредственно измерять параметры качества обслуживания, включая задержку, джиттер и потери пакетов. Несмотря на высокую точность измерений, данные методы охватывают лишь ограниченное число точек сети и требуют генерации дополнительного трафика, что может создавать дополнительную нагрузку на инфраструктуру [23].

Предложенный в данной работе метод мониторинга основан на анализе статистики PPPoE-сессий и использовании безразмерного коэффициента нестабильности $K$, отражающего относительное изменение числа аварийных завершений пользовательских соединений. В отличие от традиционных методов мониторинга, данный подход использует уже существующие статистические данные сетевого оборудования и не требует внедрения дополнительной измерительной инфраструктуры или генерации тестового трафика.

Представленный сравнительный анализ показывает, что существующие методы мониторинга преимущественно ориентированы на анализ параметров сетевого трафика или состояния сетевого оборудования. Протокол SNMP позволяет получать агрегированные показатели загрузки интерфейсов и устройств, однако не предоставляет информации о структуре сетевых соединений и обладает ограниченной временной разрешающей способностью [2], [4]. Потоковые технологии мониторинга, такие как NetFlow и IPFIX, обеспечивают более детализированное представление о структуре сетевого трафика и позволяют анализировать соединения на уровнях L3–L4 модели OSI, однако требуют значительных вычислительных ресурсов для обработки и хранения потоковых записей [5], [6], [24].

Технология sFlow обеспечивает масштабируемый сбор статистики трафика за счёт использования аппаратного сэмплирования пакетов, однако выборочный характер данных может снижать точность анализа при мониторинге отдельных сервисов или кратковременных аномалий [12], [17]. Потоковая телеметрия (gNMI, MDT) предоставляет возможность получения детализированных метрик устройств в режиме, близком к реальному времени, однако её внедрение требует поддержки соответствующих протоколов на

сетевом оборудовании и развёртывания специализированной инфраструктуры сбора и обработки телеметрических данных [7], [10], [15].

Активные методы мониторинга, основанные на генерации тестового трафика (например, ICMP или TWAMP), позволяют напрямую измерять параметры качества обслуживания, включая задержку, вариацию задержки и потери пакетов. Однако такие методы охватывают лишь ограниченное число точек сети и требуют генерации дополнительного измерительного трафика [23].

Для систематизации рассмотренных подходов выполнено их сравнительное сопоставление по ряду ключевых критериев: нагрузка на сеть, чувствительность к деградации сервисов, уровень детализации данных, масштабируемость и сложность внедрения. Результаты анализа представлены в таблице 3.2.

Таблица 3.2 - Сравнение существующих методов мониторинга и предложенного метода

| Метод | Нагрузка | Чувствительность | Детализация | Масштабируемость | Внедрение | L2-доступ |
|---|---|---|---|---|---|---|
| SNMP | низкая | низкая | низкая | высокая | низкая | ограничено |
| NetFlow/IPFIX | высокая | средняя | высокая | средняя | высокая | нет |
| sFlow | средняя | средняя | средняя | высокая | средняя | нет |
| Telemetry (gNMI) | переменная | высокая | высокая | высокая | высокая | ограничено |
| Активные тесты | низкая | высокая | высокая | низкая | высокая | ограничено |
| Метод *K* | нет | высокая | низкая | высокая | низкая | высокая |

Представленный сравнительный анализ показывает, что существующие методы мониторинга преимущественно ориентированы на анализ параметров сетевого трафика или состояния сетевого оборудования. Протокол SNMP обеспечивает сбор агрегированных статистических показателей устройств и интерфейсов, однако не предоставляет информации о структуре сетевых соединений и обладает ограниченной временной разрешающей способностью вследствие использования периодического опроса устройств [2], [4].

Потоковые технологии мониторинга, такие как NetFlow и IPFIX, позволяют анализировать структуру сетевого трафика и получать детализированную информацию о потоках на уровнях L3–L4 модели OSI, включая адреса источника и назначения, транспортные порты и объём переданных данных. Однако экспорт потоковых записей требует значительных вычислительных ресурсов для обработки и хранения данных и

может создавать дополнительную нагрузку на инфраструктуру мониторинга [5], [6], [24].

Технология sFlow использует аппаратное сэмплирование пакетов и обеспечивает масштабируемый сбор статистики трафика в высокоскоростных сетях. Вместе с тем выборочный характер получаемых данных может приводить к снижению точности анализа отдельных сервисов и кратковременных сетевых аномалий [12], [17].

Современные системы наблюдаемости всё чаще используют потоковую телеметрию (streaming telemetry), реализуемую через протоколы gNMI и модельно-ориентированные интерфейсы управления. Такой подход позволяет получать детализированные метрики устройств в режиме, близком к реальному времени, однако его внедрение требует поддержки соответствующих протоколов на сетевом оборудовании и развёртывания специализированной инфраструктуры сбора и обработки телеметрических данных [7], [10], [15].

Активные методы мониторинга, основанные на генерации тестового трафика (например, ICMP-измерения или TWAMP), позволяют напрямую измерять параметры качества обслуживания, включая задержку, вариацию задержки и потери пакетов. Однако такие методы охватывают лишь ограниченное число точек сети и требуют генерации дополнительного измерительного трафика, что может увеличивать нагрузку на сеть [23].

Важной особенностью большинства рассмотренных методов является их ориентация на анализ параметров сетевого трафика и сетевых устройств преимущественно на уровнях L3–L7 модели OSI. При этом значительная часть проблем в сетях широкополосного доступа возникает на уровне доступа (L2), где формируются и поддерживаются пользовательские PPPoE-сессии. Нестабильность таких сессий может быть вызвана ошибками оборудования доступа, перегрузкой каналов, нарушениями на линии или другими факторами, влияющими на стабильность пользовательского соединения.

Предложенный в данной работе метод основан на анализе статистики PPPoE-сессий и использовании коэффициента нестабильности $K$, отражающего относительное изменение числа аварийных завершений пользовательских соединений. Использование статистики протокольных событий уровня доступа позволяет выявлять признаки нестабильности пользовательских соединений и косвенно оценивать деградацию качества сетевых сервисов без генерации дополнительного измерительного трафика и без развёртывания специализированной инфраструктуры мониторинга. Такой подход позволяет дополнить существующие методы мониторинга и повысить наблюдаемость состояния сети доступа.

**Выводы по главе 3**

1. Подтверждена гипотеза о статистически значимой зависимости коэффициента нестабильности $K$ от изменения числа аварийных завершений PPPoE-соединений $\Delta$PADT. Анализ различных моделей машинного обучения

показал устойчивость данной зависимости и подтвердил её статистическую значимость.

2. Установлено, что ансамблевые модели машинного обучения (Random Forest и Gradient Boosting) обеспечивают наивысшую точность прогнозирования коэффициента нестабильности. Полученные значения коэффициента детерминации $R^2>0.90$ свидетельствуют о высокой объясняющей способности моделей.

3. Показано, что методы символьной регрессии позволяют извлекать интерпретируемые аналитические выражения зависимости коэффициента нестабильности от статистических параметров PPPoE-сессий. Полученные формулы по своей структуре близки к линейной модели и подтверждают физический смысл введённого показателя.

4. Продемонстрирована возможность прогнозирования коэффициента нестабильности на основе текущей статистики PPPoE-сессий. Это позволяет использовать предложенный показатель в системах раннего предупреждения о деградации качества сетевых сервисов.

5. Полученные результаты подтверждают возможность практического применения разработанного метода в системах мониторинга операторских сетей для оценки стабильности пользовательских соединений и выявления признаков деградации сервисов.

# Глава 4. Практическая апробация и верификация метода в операторской сети

## 4.1 Описание тестовой инфраструктуры и условий проведения эксперимента

Внедрение и опытная эксплуатация косвенного метода мониторинга качества сервисов связи на базе коэффициента нестабильности *K* были проведены на сети крупного телекоммуникационного оператора Республики Казахстан — АО «Казахтелеком». Экспериментальный участок охватил магистрально-доступный домен западной региональной дирекции (ЗКО РДТ), включающий города Атырау, Актау, Уральск и прилегающие районы. Общая протяжённость обслуживаемых волоконно-оптических линий связи превышает 3800 км, что обеспечивает репрезентативность наблюдений для типовой региональной инфраструктуры оператора. Абонентская база составляет порядка 95 000 домохозяйств, при этом в пиковые периоды фиксируется до 87 512 одновременных PPPoE-сессий, что соответствует условиям высокой эксплуатационной нагрузки и позволяет оценивать метод в «боевом» режиме [72]. Отдельно следует подчеркнуть, что исследуемые каналы связи являются мультисервисными и одновременно обеспечивают передачу широкополосного доступа, IPTV, телефонии и иных услуг. В связи с этим разработанный метод мониторинга фактически отражает интегральное качество функционирования комплекса сервисов, поскольку отказоустойчивость и стабильность сессионной подсистемы (PPPoE) тесно связаны с общей устойчивостью транспортной инфраструктуры. Следовательно, деградация PPPoE-параметров на определённом направлении может рассматриваться как ранний индикатор потенциального ухудшения качества не только услуги доступа в Интернет, но и иных сервисов, завязанных на тот же транспортный ресурс.

Представленная на рисунке 4.1 схема иллюстрирует типовую для оператора структуру магистрально-доступного домена, в которой ключевыми узлами являются BRAS-шлюзы, выполняющие функции терминации PPPoE-сессий, а также агрегирующие узлы доступа (DSLAM), обеспечивающие подключение абонентских линий. В практическом смысле данная архитектура удобна тем, что основные диагностически значимые события PPPoE концентрируются на BRAS, что делает возможным сбор статистики по служебным сообщениям без доступа к пользовательскому трафику. Таким образом, схема подтверждает принципиальную реализуемость «косвенного» мониторинга: вместо активных измерений или DPI используется анализ служебной телеметрии, формируемой узлами сети в штатном режиме.

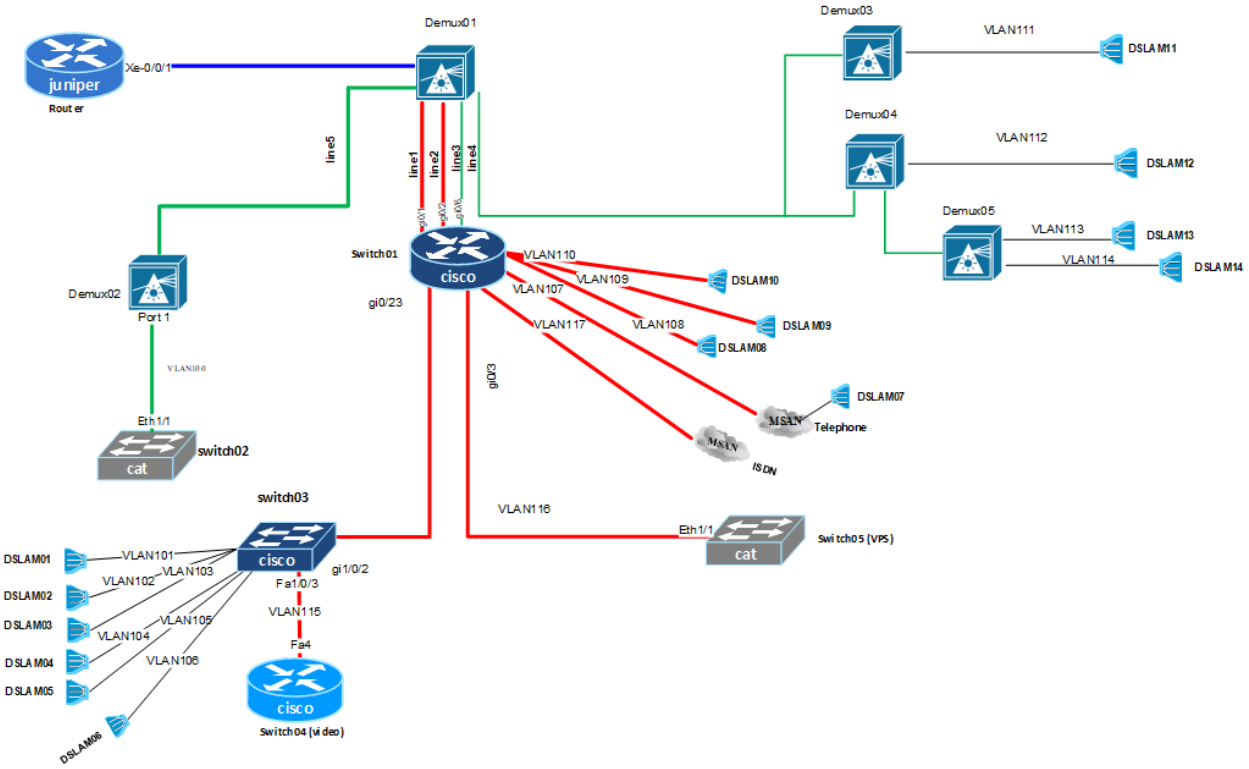


Рисунок 4.1 — Сетевая архитектура одного из участков (Составлено автором по эксплуатационным данным оператора связи)

Домен обслуживается двумя пограничными шлюзами доступа (BRAS-кластер Juniper MX960) в конфигурации active-active, которые обеспечивают агрегацию трафика и управление PPPoE Discovery для всех downstream DSLAM [4], [68], [72]. Выбор конфигурации active-active имеет важное эксплуатационное значение: при корректной балансировке нагрузки обеспечивается устойчивость к отказам одного из узлов, а также сохраняется возможность масштабирования абонентской ёмкости без существенных архитектурных изменений [4], [68]. Для целей настоящей работы это также означает, что коэффициент $K$ может рассчитываться параллельно по каждому BRAS и сравниваться между ними, что повышает достоверность диагностики и позволяет исключать ложные выводы, связанные с особенностями отдельного устройства [68], [72].

Сетевая архитектура участка включает следующие элементы: два BRAS Juniper MX960, подключённые к магистральной IP/MPLS-сети оператора (с резервированием каналов по LDP-меткам и использованием механизмов MPLS Fast Reroute); четырнадцать DSLAM Huawei MA5600T, обеспечивающих «последний километр» до абонентов по медным или оптоволоконным линиям; магистральное ядро IP/MPLS с суммарной пропускной способностью порядка 80 Гбит/с; сервер white-box probe, выполняющий функции пассивного анализа и контрольных ICMP-проверок между BRAS и агрегационными узлами; а также систему тикетов Service Desk, предназначенную для централизованного учёта обращений пользователей [4], [68], [72]. Следует отметить, что перечисленные компоненты отражают

реальный производственный контур, где эксплуатационные решения принимаются на основе сочетания технических метрик и пользовательских сигналов (жалобы/инциденты) [4], [68]. Именно поэтому верификация коэффициента *K* выполняется не в лабораторных условиях, а в контексте типовой NOC-практики, что усиливает прикладную значимость полученных результатов [4], [68], [72].

**Архитектура сбора данных и расчёта коэффициента *K* в операторской сети**

Для автоматического сбора метрик был разработан программный агент **ppp_k_probe** на Python 3.7 с использованием библиотек *pysnmp* и *fastapi* (Приложение Д) [68], [72]. Агент развёрнут на выделенном сервере в сегменте NOC, что обеспечивает управляемость и предсказуемость условий эксплуатации (доступность, резервирование, контроль обновлений). Опрос BRAS выполняется по протоколу SNMPv2c, который по-прежнему широко используется в операторских сетях для задач мониторинга и совместим с большинством сетевых платформ [2], [3]. Каждые 5 минут производится сбор двух MIB-метрик: *jnxPppoePadtSent* (общее число отправленных PADT-пакетов) и *jnxPppoeActiveSessions* (текущее число активных PPPoE-сессий) [68], [72]. Выбор именно этих параметров обусловлен тем, что они напрямую отражают количество аварийных завершений сессий (через PADT) и текущую «базу нормировки» (через число активных сессий), что позволяет формировать безразмерный показатель, пригодный для сравнения между разными временными интервалами и сегментами сети [68], [72].

В агенте реализована обработка и агрегация данных: на каждом пятиминутном интервале вычисляется разность значения счётчика PADT и усреднённое значение активных сессий за интервал. Коэффициент нестабильности *K* определяется как отношение этой разности к среднему числу сессий, что обеспечивает корректную нормировку на текущий масштаб абонентской нагрузки [68], [72]. Такой подход принципиально важен в операторских сетях, поскольку абсолютные значения счётчиков могут существенно различаться между узлами и периодами, тогда как нормированный показатель сохраняет сопоставимость. Итоговые значения сохраняются в базе данных PostgreSQL 11 и экспортируются в систему мониторинга Prometheus (версии 2.8–2.10) [74]. Для визуализации используются интерактивные панели Grafana (версии 6.x), что обеспечивает удобное восприятие трендов и аномалий в динамике. Оповещения о превышении порога на трёх подряд интервалах формируются в Alertmanager, а уведомления доставляются инженерам через Telegram-бот и электронную почту [74]. Таким образом, коэффициент *K* встроен в стандартный контур эксплуатационного мониторинга и обрабатывается так же, как и любые другие KPI/алерты, что снижает порог внедрения и повышает вероятность практического применения метода в разных подразделениях оператора [68], [72].

Современные сетевые архитектуры IPoE/DHCP и GPON предлагают альтернативы классическому PPPoE-подключению. DSL Forum допускает использование IPoE/DHCP вместо PPPoE, при этом IPoE через DHCP обеспечивает сопоставимые возможности аутентификации и адресации абонентов [75]. В GPON-сетях терминалы ONU/ONT нередко поддерживают как PPPoE, так и IPoE/DHCP-режимы доступа [76], а на практике часть провайдеров продолжает использовать PPPoE по историческим причинам (биллинг, привычные процессы эксплуатации, отработанные схемы учёта). В данном контексте важно отметить, что предложенная методика расчёта *K* [68], [72] не «привязана» исключительно к PPPoE как технологии, а опирается на более общий принцип: анализ служебных событий сеансового уровня, отражающих стабильность соединений. Следовательно, при адаптации методики возможно сопоставление PPPoE-событий с аналогичными по смыслу событиями в DHCP-сценариях (DISCOVER/OFFER/REQUEST/ACK, параметры аренды и её обновления), а также с событиями регистрации/перерегистрации в GPON-контуре. При корректном подборе аналогичных метрик коэффициент *K* может быть интерпретирован как индикатор нестабильности сеансовой активности и в альтернативных архитектурах, что соответствует общей тенденции замещения PPPoE подходами IPoE/DHCP [75], [77]. В практическом плане это означает возможность масштабирования методики на новые технологические платформы без отказа от базовой идеи косвенного мониторинга [68], [72].

Дополнительно разработанный метод мониторинга качества соединений прошёл практическую проверку в производственной среде РГП на ПХВ «Казахстанский центр межбанковских расчётов Национального Банка Республики Казахстан» (КЦМР НБРК). На его основе была внедрена система мониторинга внешних подключений клиентов (Out Connections Monitoring System, OCMS), предназначенная для контроля стабильности и доступности внешних каналов связи, реализованных на основе клиентских IPSec-туннелей [72]. Система имитирует работу клиентских подключений к сервисам и позволяет оперативно выявлять сбои либо ухудшение качества связи, обеспечивая практический эффект в части сокращения времени реакции и повышения управляемости сетевой инфраструктуры [72]. Следует подчеркнуть, что данная апробация демонстрирует универсальность подхода: несмотря на различие протоколов (PPPoE vs IPSec), общая логика мониторинга строится вокруг признаков нестабильности сеансов и их динамики во времени [68], [72]

Внедрение разработки обеспечило сокращение времени реагирования при возникновении внештатных ситуаций, повысило качество локализации проблемных участков и улучшило общую эффективность эксплуатации телекоммуникационной инфраструктуры. Факт промышленного внедрения подтверждён Актом о внедрении результатов научной разработки (Приложение Е), подписанным Генеральным директором КЦМР НБРК Б.М. Жаленовым. Ввиду ограничений, связанных с банковской тайной, в настоящей

диссертации не раскрываются конфигурационные детали инфраструктуры и параметры тестирования, а также не приводятся значения ряда внутренних идентификаторов и адресных параметров. Тем не менее, даже при неполном раскрытии конкретных настроек данный опыт демонстрирует практический факт применимости методики: предложенный подход и коэффициент нестабильности *K* могут использоваться не только для анализа PPPoE-сессий, но и для оценки надёжности и стабильности сеансовых соединений в системах класса VPN и IPSec. Это важно с точки зрения переносимости результатов, поскольку в реальных организациях часто требуется унифицированный принцип мониторинга, работающий в разных сетевых сегментах и для разных типов туннелей.

## 4.2 Результаты мониторинга коэффициента нестабильности *K* в реальной сети

В течение 8 недель непрерывного мониторинга было собрано и проанализировано более 16 000 пятиминутных интервалов наблюдения, что обеспечило свыше 130 000 точек данных для каждого BRAS [68], [72]. Данный объём выборки является достаточным для оценки как фонового поведения коэффициента *K*, так и редких аномальных событий, включая аварии и массовые разрывы сессий. Кроме того, длительность наблюдений позволяет охватить разные режимы эксплуатации сети: рабочие дни и выходные, часы пиковой нагрузки и ночные периоды, а также интервалы проведения плановых работ. В совокупности это повышает устойчивость выводов и снижает вероятность того, что эффект будет обусловлен единичным нетипичным эпизодом [68], [72].

**Общая динамика коэффициента *K* и зоны стабильности**

По совокупным данным 81,2% интервалов находились в «зелёной» зоне, соответствующей фоновым разрывам сессий при добровольных отключениях абонентов, штатных переподключениях и типовых технологических изменениях [68], [72]. В «жёлтой» зоне оказалось 13,9% интервалов, причём данная зона преимущественно проявлялась в часы повышенной вечерней нагрузки и после профилактических работ, когда сеть может испытывать кратковременные переходные процессы [68], [72]. «Оранжевая» зона охватила 3,4% наблюдений, что уже можно интерпретировать как состояние повышенного риска и потенциального ухудшения пользовательского опыта, тогда как критическая «красная» зона составила 1,5% интервалов; именно в эти периоды фиксировались массовые разрывы PPPoE-сессий и активные жалобы абонентов [68], [72]. Таким образом, градация по зонам позволяет не только «видеть» аномалии, но и классифицировать их по степени критичности, что удобно для практической эксплуатации, когда ресурсы аварийных команд и инженеров ограничены [68], [72].

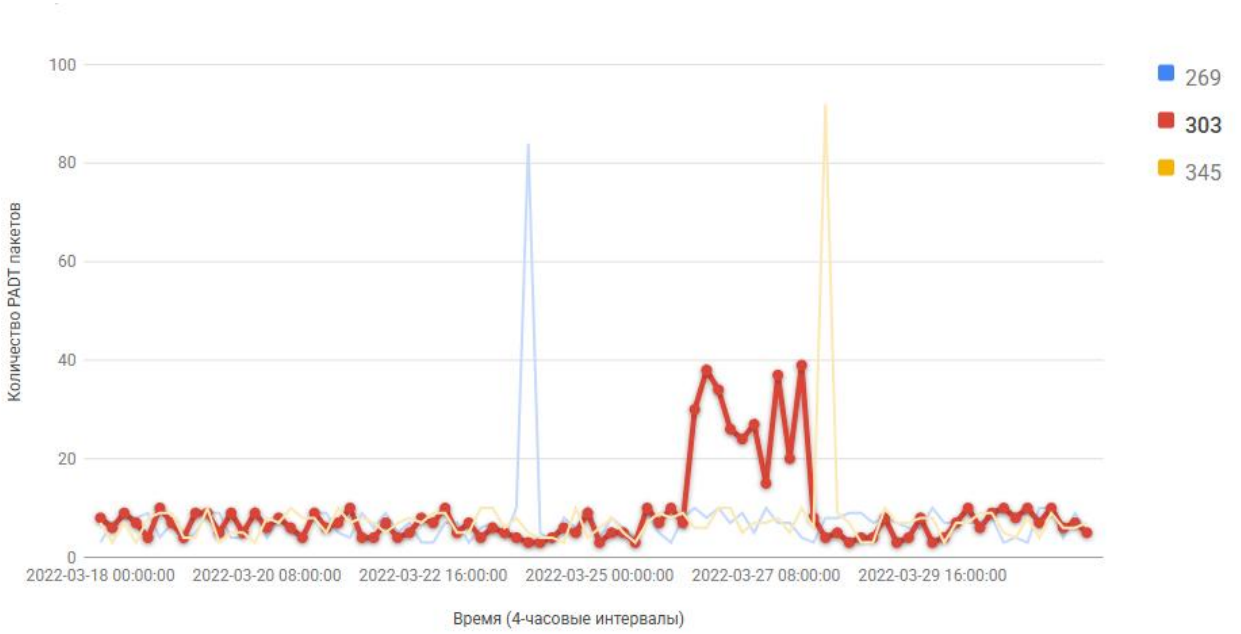


Рисунок 4.2 — демонстрирует временной ряд *K* по BRAS-1 2 за 2 недели [27]

Рисунок 4.2 наглядно демонстрирует временную динамику коэффициента *K* и подтверждает, что показатель обладает выраженной чувствительностью к изменениям состояния сети. При нормальном функционировании *K* остаётся вблизи фоновых значений и характеризуется предсказуемыми суточными колебаниями. При возникновении аварийных ситуаций наблюдаются резкие всплески, отличающиеся по амплитуде и длительности. Практическая ценность такой картины заключается в том, что даже без сложной обработки данных инженер может визуально выделять участки нестабильности и сопоставлять их с временными метками инцидентов и жалоб, то есть использовать *K* как оперативный индикатор для первичного анализа. На рисунке 4.2 видны регулярные вечерние пики и редкие резкие всплески ночью, совпадающие с аварийными событиями. Следует отметить, что вечерние пики являются ожидаемыми, поскольку в это время возрастает интенсивность пользовательской активности и нагрузка на элементы доступа, что может приводить к увеличению числа переподключений и переходных состояний [4], [68], [72]. Ночные всплески, напротив, являются менее типичными и потому более диагностически значимыми: они часто связаны либо с технологическими переключениями, либо с физическими отказами (питание, узлы доступа, деградация линии) [4], [72]. Таким образом, сама форма временного ряда несёт полезную информацию: она позволяет различать «нагрузочные» эффекты и «аварийные» эффекты, даже не прибегая к дополнительным источникам данных [68], [72].

Среднее значение *K* составило примерно 0,011 в рабочие дни и около 0,012 — в выходные; ночные значения *K* были ниже (медиана порядка 0,005), однако и в ночные часы фиксировались аварийные события, связанные с переключениями питания или отказами DSLAM [4], [68], [72]. Разница между дневным и ночным режимами закономерна: днём и вечером сеть функционирует в условиях повышенной нагрузки, и любые нестабильности

проявляются более явно, тогда как ночью интенсивность активности снижается и фоновые значения уменьшаются. При этом наличие ночных аварийных всплесков подчёркивает необходимость круглосуточного мониторинга: даже при низкой нагрузке технические сбои могут приводить к массовым разрывам сессий и, как следствие, к ухудшению доступности сервисов для части абонентов [4], [72].

**Корреляция *K* с внешними метриками**

Для проверки достоверности мониторинга выполнено сопоставление коэффициента нестабильности *K* с независимыми эксплуатационными индикаторами качества канала. В качестве эталонного QoS-показателя использовалась потеря пакетов *Loss(t)*, измеряемая активными пробниками с шагом 5 минут (см. табл. А.2 и методику расчёта в разд. 2.2), а также число обращений абонентов в систему Service Desk (тикеты) [4], [68], [72].

В качестве меры статистической связи применялся коэффициент корреляции Пирсона, поскольку коэффициент *K* построен как нормированная динамическая величина, измерения приведены к единому временному шагу и агрегированы, а ожидаемая зависимость между ростом аварийных разрывов сессий и ухудшением качества передачи (ростом *Loss(t)*) носит преимущественно линейный характер на рассматриваемых интервалах [68], [72]. С практической точки зрения данный анализ принципиально важен, поскольку коэффициент *K* сопоставляется не с производными от него величинами, а с независимыми индикаторами качества, используемыми эксплуатационными подразделениями оператора связи. Результаты корреляционного анализа приведены в таблице 4.1.

Таблица 4.1 — Матрица корреляций с независимыми индикаторами качества

| Пара метрик | Коэф. Пирсона *r* | *p* |
|---|---|---|
| *K* — *Loss(t)* | 0,960 | < 0,001 |
| *K* — тикеты абонентов | 0,615 | < 0,001 |
| *Loss(t)* — тикеты абонентов | 0,583 | < 0,001 |

Полученные значения корреляций указывают на высокую согласованность коэффициента нестабильности *K* с независимыми эксплуатационными сигналами. Сильная связь между *K* и *Loss(t)* свидетельствует о том, что рост доли аварийно завершённых PPPoE-сессий сопровождается ухудшением транспортных характеристик канала и увеличением потерь пакетов, что типично для деградации линий связи, перегрузочных режимов и частичных отказов элементов доступа [4], [68], [72]. Связь *K* с числом обращений абонентов в Service Desk подтверждает

практическую значимость показателя, поскольку тикеты отражают воспринимаемое ухудшение качества обслуживания [4], [72]. Низкие значения $p$ подтверждают статистическую значимость выявленных зависимостей и позволяют отвергнуть гипотезу об их случайном характере [68], [72].

**Анализ по BRAS и VLAN**

Анализ распределения коэффициента нестабильности $K$ по различным BRAS выявил схожие профили его поведения: медианные значения, а также 95-й перцентиль оказались сопоставимыми для BRAS-1 и BRAS-2. Существенных статистически значимых различий между шлюзами не выявлено, что указывает на однородные условия их работы и корректную эксплуатационную конфигурацию BRAS-кластера в режиме active-active [4], [68]. С практической точки зрения это означает, что наблюдаемые всплески $K$ не являются артефактом или особенностью одного конкретного устройства, а отражают реальные изменения состояния сети доступа в целом. Тем самым исключается влияние локальных аппаратных или программных факторов отдельного BRAS и подтверждается репрезентативность коэффициента нестабильности как интегрального показателя качества [72]. В то же время детальный анализ в разрезе VLAN позволил выявить отдельные проблемные сегменты, требующие углублённой диагностики. Такой результат является ожидаемым, поскольку агрегированный показатель на уровне BRAS способен сглаживать локальные нарушения, тогда как декомпозиция по VLAN позволяет локализовать аномалию до уровня конкретного транспортного сегмента, узла доступа или технологической цепочки [4], [68], [72].

В качестве характерного примера далее рассматривается VLAN 303 (см. раздел 4.3).

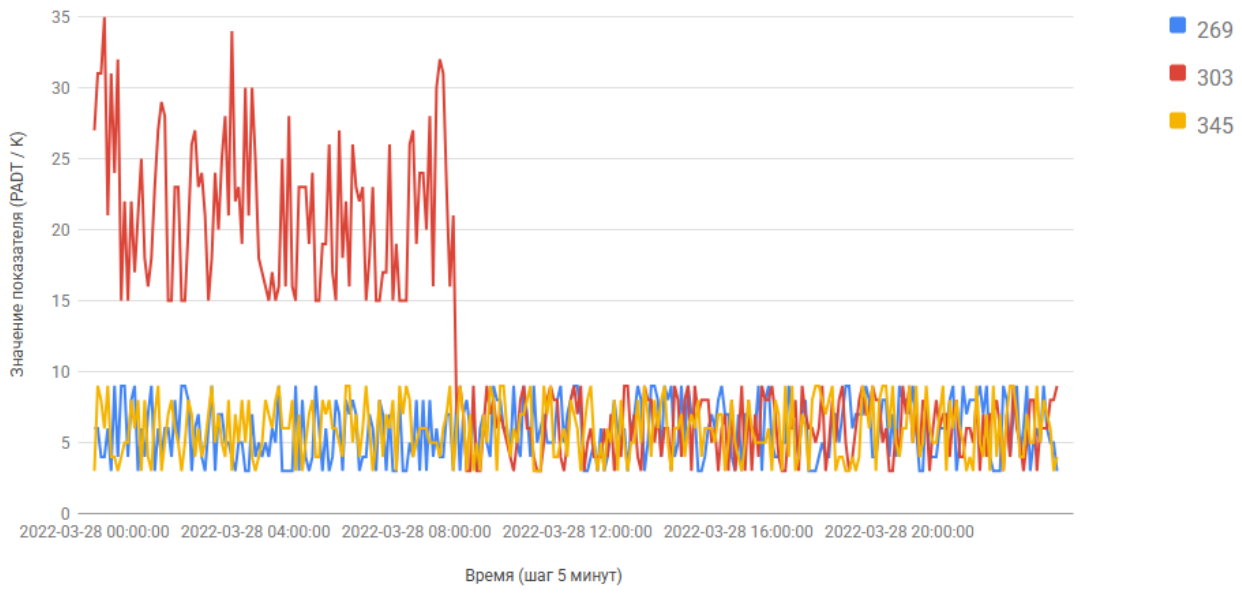


Рисунок 4.3 — Суточные профили $K$ по проблемным VLAN [27]

Рисунок 4.3 иллюстрирует суточные профили коэффициента *K* по проблемным VLAN и демонстрирует, что нестабильность может иметь как регулярный (нагрузочный), так и эпизодический (аварийный) характер. В часы пиковой нагрузки наблюдаются выраженные всплески, которые могут быть связаны с ростом числа переподключений и повышенной чувствительностью линии к шумам и перегрузкам. После аварийных ремонтов и технологических переключений также возможны временные повышения *K*, отражающие переходные процессы и восстановление нормального режима работы сети. Важно подчеркнуть, что визуальное выделение «аномального» профиля VLAN упрощает дальнейшую работу инженера: метод *K* не заменяет диагностику, но позволяет значительно сузить область поиска и быстрее перейти к анализу первопричины.

### 4.3 Пример выявления и устранения деградаций качества

#### Хронология и анализ инцидента

Был выбран период времени, охватывающий максимально широкий спектр эксплуатационных сценариев. В период с 18 марта 2022 года по 31 марта 2022 года присутствовали обычные рабочие дни, выходные дни, праздники республиканского значения, а также каникулы в школах и высших учебных заведениях; тем самым обеспечивалось разнообразие профиля пользовательской активности и нагрузки на сеть. В течение суток были охвачены как часы наибольшей нагрузки (вечерние пики), так и интервалы минимальной активности (ночные часы), что позволило сравнить поведение показателей в разных режимах. Как видно из графика на рисунке 4.3, по некоторым направлениям наблюдались единичные всплески, продолжавшиеся короткий промежуток времени, что может быть связано со штатными переключениями или локальными кратковременными изменениями качества. Однако по VLAN 303 был зафиксирован период длительностью около двух суток, когда значение коэффициента отброшенных пакетов превышало 8%. Одновременно значение *K* выросло с типичных фоновых значений порядка 9 до 39–40, то есть более чем в четыре раза. Система Prometheus зафиксировала три последовательных превышения порога, после чего Alertmanager автоматически сформировал оповещение инженеру. Параллельно в течение дня наблюдалось поступление большого количества обращений в Service Desk (около 30) от абонентов VLAN 303 с жалобами «нет доступа», «обрывы», «низкая скорость». Совпадение роста *K* и роста числа тикетов формирует характерный признак нештатного события, отличающийся от обычного процесса аутентификации или плановых работ. Дальнейший анализ показал, что наблюдаемая картина характерна для нештатных разрывов соединений, обусловленных физическими причинами на транспортной инфраструктуре. В результате было выявлено наличие проблемы с искажённым сигналом на участке радиорелейной линии связи по трассе прохождения VLAN 303. После устранения проблемы значение

коэффициента отброшенных пакетов для VLAN 303 снизилось до штатного уровня, что подтвердило корректность диагностического вывода.

**Сравнительный анализ MTTR: до и после внедрения метода**

Сравнение аналогичных аварийных инцидентов до и после внедрения косвенного метода мониторинга на основе коэффициента нестабильности *K* показало значимое сокращение времени реакции эксплуатационных подразделений. В базовый период (2021 г.) среднее время восстановления сервиса (MTTR) составляло порядка 2 ч 15 мин, при этом существенная доля задержки приходилась на этап обнаружения и первичного подтверждения аварийной ситуации.

После внедрения мониторинга по коэффициенту *K* среднее значение MTTR снизилось приблизительно до 1 ч 32 мин. Наиболее существенным фактором стало сокращение времени обнаружения аварии: оно уменьшилось до 35 мин. Это позволило оперативно инициировать процедуры диагностики и локализации неисправностей, минимизируя длительность простоя сервисов.

Достигнутое улучшение оказывает влияние не только на технические, но и на организационные показатели эксплуатации: снижается нагрузка на дежурные смены, уменьшается количество эскалаций, повышается предсказуемость процессов восстановления. В совокупности это приводит к снижению как прямых, так и косвенных издержек, включая затраты на выездные работы, перераспределение ресурсов и компенсационные выплаты абонентам.

Таблица 4.2 — Сравнение MTTR до и после внедрения метода

| Период | Средний MTTR | Время обнаружения | Повторные обращения |
|---|---|---|---|
| 2021 (до *K*) | 2 час 15 минут | 1 час | 56 |
| 2022 (после *K*) | 1 час 32 минуты | 35 минут | 21 |

Приведённые в таблице 4.2 значения показывают, что достигнутый эффект обусловлен не только ускорением восстановительных работ, но прежде всего более ранним обнаружением инцидентов и более точной локализацией источника проблемы. Сокращение числа повторных обращений абонентов дополнительно подтверждает улучшение качества реакции эксплуатационных подразделений: при длительной деградации сервисов абоненты, как правило, формируют несколько обращений, тогда как при оперативном выявлении и устранении неисправности потребность в повторных обращениях существенно снижается. Согласно информационному письму АО «Казахтелеком» (Приложение Ж), приведённые статистические данные подтверждаются материалами опытной эксплуатации, что позволяет рассматривать полученные результаты как достоверные и репрезентативные для практического применения метода в операторской сети.

### 4.4 Оценка экономической эффективности предложенного метода

Экономическая эффективность внедрения метода мониторинга на основе коэффициента нестабильности *K* оценивалась с учётом совокупных затрат и выгод, возникающих при интеграции решения в эксплуатационную среду оператора связи. В рамках анализа рассматривались как прямые расходы, связанные с развёртыванием и сопровождением системы, так и экономические эффекты, проявляющиеся в снижении потерь от аварийных ситуаций и уменьшении нагрузки на эксплуатационные и сервисные подразделения.

В оценке были выделены две основные категории затрат: капитальные вложения (CAPEX), включающие использование вычислительных ресурсов, разработку и внедрение программного обеспечения, настройку панелей визуализации и интеграцию с существующими OSS/NMS; а также операционные расходы (OPEX), охватывающие сопровождение решения, эксплуатационные трудозатраты, возможные компенсационные выплаты по SLA и нагрузку на службу технической поддержки [82]. Такой подход позволяет рассматривать предложенный метод не только как техническую инновацию, но и как инструмент оптимизации эксплуатационных затрат, что имеет принципиальное значение для операторских сетей, функционирующих в условиях жёстких экономических ограничений.

К прямым экономическим выгодам относятся снижение штрафных санкций по соглашениям об уровне обслуживания (SLA), сокращение трудозатрат аварийных бригад и уменьшение количества повторных обращений абонентов. Существенно, что снижение среднего времени восстановления (MTTR) и ускорение обнаружения аварий обладают кумулятивным эффектом: уменьшается длительность деградации сервиса, сокращается число затронутых абонентов и, как следствие, снижается нагрузка на контакт-центр.

Косвенные экономические эффекты включают рост лояльности абонентов, снижение уровня оттока и повышение индекса удовлетворённости клиентов (NPS). Несмотря на то, что подобные эффекты сложнее формализовать количественно, в практике операторов связи они рассматриваются как значимый фактор, поскольку удержание абонентской базы и снижение репутационных рисков напрямую влияют на устойчивость выручки и долгосрочные финансовые показатели компании.

**Расчёт капитальных и операционных затрат**

CAPEX (капитальные вложения) включают приобретение и/или выделение серверных ресурсов для мониторинга (например, сервер мониторинга с конфигурацией 2 AMD EPYC 16C/64G RAM и RAID для надёжного хранения данных), а также затраты на разработку скриптов и создание дашбордов Grafana/Prometheus. Дополнительно учитываются работы по интеграции системы оповещений (Telegram, email), настройке ролей доступа и обеспечению устойчивости мониторингового контура. Итоговый размер

CAPEX определяется политикой оператора: в ряде случаев возможно использование уже имеющейся инфраструктуры NOC, что снижает начальные вложения. В любом случае следует отметить, что основная ценность решения заключается в его программной составляющей и интеграции в существующий стек, а не в дорогом специализированном оборудовании.

OPEX (годовые операционные расходы) включают поддержку скриптов и системы мониторинга, администрирование баз данных и серверов, а также трудозатраты на эксплуатацию и поддержку пользователей мониторинга (инженеров NOC и смежных подразделений). По сравнению с капитальными затратами ежегодные расходы обычно являются умеренными, поскольку после первичной настройки решение функционирует в автоматическом режиме, а типовые работы сводятся к обновлениям, контролю корректности опроса и периодической корректировке пороговых значений. Таким образом, экономическая модель внедрения характеризуется относительно небольшими регулярными расходами при наличии измеримого эффекта от сокращения времени реакции на инциденты.

CAPEX (капитальные вложения):

- Сервер мониторинга (2 AMD EPYC 16C/64G RAM, RAID);
- Разработка скриптов и дашбордов Grafana/Prometheus;
- Интеграция системы оповещений (Telegram, email);

Итого CAPEX: первоначальные инвестиции, зависящие от степени использования существующей инфраструктуры оператора.

OPEX (годовые операционные расходы):

- Поддержка скриптов и системы мониторинга;
- Администрирование баз данных и серверов;
- Техническая поддержка пользователей мониторинга;

Итого OPEX: сравнительно небольшие ежегодные расходы.

**Экономия и снижение затрат после внедрения метода**

Оптимизация труда аварийных бригад. Сокращение MTTR позволило снизить трудозатраты аварийных бригад за счёт уменьшения длительности поисковых работ и более раннего запуска диагностики. В практическом плане это означает уменьшение количества выездов «вслепую» и снижение времени, затрачиваемого на подтверждение факта аварии. По данным информационного письма АО «Казахтелеком» (Приложение Ж), это обеспечило экономию значительного числа человеко-часов, а также позволило более эффективно распределять ресурсы аварийных команд между параллельными инцидентами.

Снижение нагрузки на кол-центр. Количество повторных обращений абонентов существенно уменьшилось, что привело к снижению нагрузки на кол-центр и Service Desk. В условиях массовых аварий именно контакт-центр становится одним из «узких места», поскольку рост обращений приводит к очередям, снижению качества обслуживания и дополнительным

репутационным потерям. Уменьшение числа повторных тикетов отражает не только улучшение фактического качества сервиса, но и повышение удовлетворённости абонентов, поскольку пользователи получают восстановление услуги в более короткие сроки и реже вынуждены обращаться повторно.

Число жалоб на массовые обрывы было существенно снижено, что отражается как в статистике Service Desk, так и в субъективной оценке пользовательского опыта. Отмечался также рост индекса удовлетворённости клиентов (NPS/CSI), что является характерным следствием уменьшения длительности деградаций и повышения предсказуемости работы сети. Метод легко масштабируется на другие регионы: для этого достаточно добавить новые BRAS в список опроса и адаптировать параметры визуализации в Grafana без дополнительных капитальных затрат. Существенным преимуществом является совместимость подхода с оборудованием различных вендоров (Juniper, Cisco, Huawei), поскольку метод основан на сборе стандартной телеметрии и не требует вмешательства в пользовательский трафик. Таким образом, внедрение может выполняться поэтапно и без масштабной перестройки архитектуры OSS/NMS, что повышает реализуемость решения на уровне оператора.

**Выводы по главе 4**

**Метод интегрирован в эксплуатационную практику.**

Разработанный косвенный метод мониторинга качества сервисов связи на основе коэффициента нестабильности $K$ успешно внедрён и апробирован на реальном сегменте сети АО «Казахтелеком». Использование стандартных протоколов управления (SNMPv2c), открытого программного обеспечения (Prometheus, Grafana) и скриптовой реализации на языке Python позволило встроить вычисление показателя $K$ в существующий контур OSS/NMS без вмешательства в пользовательский трафик и без приобретения специализированных лицензий. Это обеспечило технологическую совместимость решения с действующей инфраструктурой и минимальный порог внедрения. В условиях мультисервисных каналов (широкополосный доступ, IPTV, телефония) коэффициент $K$ отражает интегральное состояние сессионной подсистемы и может рассматриваться как универсальный индикатор деградации качества транспортного ресурса, общего для нескольких сервисов.

**Достоверность и чувствительность метода подтверждены эксплуатационными данными.**

В ходе опытной эксплуатации установлена устойчивая связь между ростом коэффициента $K$ и ухудшением независимых эксплуатационных показателей, включая увеличение потерь $Loss(t)$ и рост числа обращений абонентов в службу поддержки. Наличие согласованности с внешними индикаторами подтверждает, что коэффициент $K$ отражает реальные изменения состояния сети и пользовательского опыта, а не случайные

флуктуации счётчиков. Таким образом, показатель может использоваться как надёжный инструмент раннего обнаружения деградаций и как входной сигнал для процессов эскалации и последующей технической диагностики.

**Сокращено время реакции на инциденты (MTTR).**

Практическое внедрение метода привело к существенному снижению среднего времени устранения массовых аварий: MTTR уменьшился значительно, а время обнаружения инцидента сократилось до трёх-четырех интервалов мониторинга. Это позволило запускать диагностику на ранней стадии и более точно локализовать проблемные сегменты сети. В результате повысилась управляемость аварийных ситуаций, снизилась длительность простоев сервисов и уменьшилось негативное влияние инцидентов на абонентов.

**Получен измеримый экономический эффект.**

Согласно данным опытной эксплуатации и информационному письму АО «Казахтелеком» (Приложение Ж), внедрение мониторинга по коэффициенту *K* позволило сократить прямые и косвенные эксплуатационные издержки. Отмечено снижение штрафов по SLA, уменьшение нагрузки на службу поддержки и сокращение числа повторных обращений абонентов. Даже при консервативной оценке экономический эффект выражается в уменьшении времени простоя и более рациональном использовании ресурсов аварийных бригад, что имеет прямое значение для операционных расходов оператора.

**Метод обладает высокой масштабируемостью и переносимостью.**

Предложенный подход не привязан к конкретным вендорам оборудования или транспортным технологиям и может быть расширен на новые регионы и сегменты сети без доработки аппаратной части. Использование стандартных механизмов сбора телеметрии и типового мониторингового стека позволяет реализовывать внедрение поэтапно, подключая новые узлы по мере необходимости и зрелости эксплуатационных процессов. Это делает метод удобным для тиражирования в масштабах крупного оператора связи.

**Заложены перспективы дальнейшего развития.**

Результаты главы 4 создают основу для интеграции коэффициента *K* с методами машинного обучения и перехода к проактивной модели управления инцидентами, включая прогнозирование деградаций и автоматизацию сценариев реагирования в NOC. Кроме того, предложенный подход может быть адаптирован для использования в магистральных и городских сегментах IP/MPLS-сетей, где также актуальны задачи мониторинга устойчивости и качества обслуживания. В целом практическая апробация подтверждает, что коэффициент нестабильности *K* является применимым и эффективным инструментом мониторинга качества сервисов связи в реальной эксплуатации

**Заключение**

В диссертационной работе решена актуальная научно-практическая задача мониторинга качества сервисов в современных телекоммуникационных сетях, функционирующих в условиях высокой динамичности, мультисервисности и распределённости архитектур. Проведённый анализ существующих методов мониторинга показал, что традиционные подходы, основанные на прямых измерениях параметров качества обслуживания и периодическом опросе сетевого оборудования, обладают ограниченной чувствительностью к скрытым и кратковременным деградациям качества, особенно в сегментах доступа и агрегации.

В ходе исследования предложен и обоснован новый подход к мониторингу качества сервисов, основанный на анализе косвенных статистических данных пользовательских сессий протокола PPPoE. Впервые введён безразмерный коэффициент нестабильности $K$, отражающий долю аварийных завершений PPPoE-сессий за заданный интервал наблюдения. Показано, что использование данного коэффициента позволяет выявлять деградации качества обслуживания без генерации тестового трафика и без внедрения дополнительного измерительного оборудования, что обеспечивает косвенный характер мониторинга и минимальное влияние на функционирование сети.

Аналитическое обоснование коэффициента $K$ выполнено с использованием методов теории вероятностей и математической статистики, что позволило установить его статистическую состоятельность и устойчивость к случайным флуктуациям трафика. Корректность аналитических выводов подтверждена результатами имитационного моделирования в среде MATLAB, в рамках которого воспроизведены характерные режимы функционирования сети доступа и показана чувствительность коэффициента $K$ к изменениям условий обслуживания.

Для оценки прогностических возможностей предложенного показателя разработан комплекс моделей машинного обучения, включающий линейные, регуляризационные, ансамблевые и символьные методы регрессии. Проведён сравнительный анализ моделей показал, что коэффициент $K$ обладает высокой предсказуемостью на основе статистических признаков PPPoE-сессий, а применение методов машинного обучения позволяет не только повысить точность прогнозирования, но и выявить интерпретируемые зависимости между параметрами сети и качеством предоставляемых сервисов.

Практическая значимость работы подтверждена результатами опытной эксплуатации разработанного метода в реальной операторской сети. Показано, что применение коэффициента нестабильности $K$ позволяет оперативно выявлять проблемные участки сети, локализовать причины деградации качества и сокращать время реагирования на инциденты. Экономическая

оценка эффективности продемонстрировала целесообразность внедрения предложенного метода в системы мониторинга операторского уровня за счёт снижения эксплуатационных затрат и повышения устойчивости предоставления услуг.

Таким образом, совокупность полученных теоретических, экспериментальных и практических результатов подтверждает научную новизну и прикладную ценность диссертационной работы. Предложенный метод мониторинга качества сервисов на основе коэффициента нестабильности $K$ расширяет существующие подходы к обеспечению наблюдаемости телекоммуникационных сетей и может быть использован при построении и модернизации систем мониторинга в сетях операторов связи.

**Перспективы дальнейших исследований.**

Перспективы дальнейших исследований, вытекающие из результатов диссертационной работы, связаны с развитием и расширением предложенного метода мониторинга качества сервисов, а также с его адаптацией к новым архитектурам и условиям функционирования телекоммуникационных сетей.

Одним из направлений дальнейших исследований является расширение набора анализируемых косвенных статистических признаков за счёт использования данных других протоколов сетевого доступа и агрегации, включая IPoE/DHCP, RADIUS и элементы сигнализации транспортных и сервисных уровней. Интеграция статистики различных протоколов позволит повысить универсальность предложенного подхода и обеспечить его применение в сетях с гетерогенной технологической базой.

Перспективным направлением является адаптация метода мониторинга на основе коэффициента нестабильности $K$ к виртуализированным и облачным архитектурам, включая SDN/NFV и cloud-native среды. Это предполагает исследование влияния динамического масштабирования виртуальных сетевых функций, миграции контейнеров и изменения топологии overlay-сетей на статистику пользовательских сессий и поведение коэффициента $K$.

Отдельный интерес представляет развитие прогностической части метода за счёт применения более сложных моделей анализа временных рядов и графовых моделей, учитывающих топологическую структуру сети и взаимосвязь между её элементами. Использование графовых нейронных сетей и моделей пространственно-временной корреляции может повысить точность раннего обнаружения деградаций качества и улучшить локализацию их первопричин.

В дальнейшем целесообразно исследовать возможность интеграции коэффициента нестабильности $K$ с показателями качества пользовательского

восприятия (QoE), включая MOS и сервисно-ориентированные метрики. Формирование гибридных моделей, объединяющих сетевые статистические признаки и оценки пользовательского опыта, позволит повысить интерпретируемость результатов мониторинга и приблизить их к реальным условиям эксплуатации сервисов.

Практически значимым направлением является разработка автоматизированных механизмов принятия решений на основе значений и прогнозов коэффициента *K*, включая динамическую настройку параметров сети, приоритизацию трафика и инициирование превентивных действий в системах управления и оркестрации. Это создаёт предпосылки для применения предложенного метода в рамках концепций self-healing и self-optimizing networks.

Таким образом, результаты диссертационной работы формируют научный и практический задел для дальнейших исследований в области интеллектуального мониторинга качества сервисов и могут служить основой для разработки новых методов анализа и управления телекоммуникационными сетями.

**Список использованной литературы**

1. Kilkki K. Quality of Experience in Communications Ecosystem. Journal of Universal Computer Science, 2008, Vol. 14, No. 5, pp. 615–624. DOI: 10.3217/JUCS-014-05-0615.

2. RFC 1157. Simple Network Management Protocol (SNMP). IETF, 1990.

3. RFC 3411–3418. Simple Network Management Protocol (SNMPv3) Management Framework and MIB. IETF, 2002.

4. Biswas I., Abu-Tair M., Morrow P., McClean S., Bryan S., Parr G. A Dynamic Approach to MIB Polling for Software Defined Monitoring. Journal of Computer and Communications, 2017, Vol. 5, No. 5, pp. 24–1. DOI: 10.4236/jcc.2017.55003.

5. Hofstede R., Čeleda P., Trammell B., Drago I., Sadre R., Sperotto A., Pras A. Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX. IEEE Communications Surveys & Tutorials, 2014, Vol. 16, No. 4, pp. 2037–2064. DOI: 10.1109/COMST.2014.2321898.

6. RFC 3954. Cisco Systems NetFlow Services Export Version 9. IETF, 2004.

7. Li B., Springer J., Bebis G., Hadi Gunes M. A Survey of Network Flow Applications. Journal of Network and Computer Applications, 2013, Vol. 36, No. 2, pp. 567–581. DOI: 10.1016/j.jnca.2012.12.020.

8. Irino H., Katayama M., Chaki S. Flow-based Network Measurement—NetFlow & IPFIX. NTT Technical Review, 2007, Vol. 5, No. 12, pp. 76–81.

9. RFC 7011–7015. IP Flow Information Export (IPFIX): Protocol Specification, Information Model and Flow Aggregation. IETF, 2013.

10. Brownlee N. Flow-Based Measurement: IPFIX Development and Deployment. IEICE Transactions on Communications, 2011, Vol. E94-B, No. 8, pp. 2190–2197.

11. RFC 5982. IP Flow Information Export (IPFIX) Mediation: Problem Statement. IETF, 2010.

12. Phaal P., Lavine M. sFlow Version 5. sFlow.org, 2004. Available at: https://sflow.org.

13. Phaal P., Byford J., Peterson M. Network-Wide Visibility with Linux Networking and sFlow. Proc. netdev 0x15, 2019.

14. Zseby T., Hirsch T., Claise B. Packet Sampling for Flow Accounting: Challenges and Limitations. In: Passive and Active Network Measurement (PAM 2008), LNCS, Vol. 4979. Springer, 2008, pp. 61–71.

15. Brauckhoff D., Tellenbach B., Wagner A., Lakhina A., May M. Impact of Packet Sampling on Anomaly Detection Metrics. Proc. ACM IMC 2006, 2006, pp. 159–170.

16. Wang Y., Nguyen T., Dinh T., Hu Y.C., Chiu D. TeleScope: Flow-Level Video Telemetry using SDN. Proc. IEEE EWSDN, 2016.

17. RFC 8641. Subscription to YANG Notifications for Datastore Updates. IETF, 2019.

18. OpenConfig Working Group. gRPC Network Management Interface (gNMI) Specification. Internet-Draft, 2016–.

19. Kentik. How to Maximize the Value of Streaming Telemetry for Network Monitoring. Kentik Technical Blog, 2019.

20. Open Networking Foundation. SDN Architecture. ONF TR-521, Issue 1.1, 2016.

21. Tan L., Su W., Zhang W., Lv J., Zhang Z., Miao J., Liu X., Li N. In-band Network Telemetry: A Survey. Computer Networks, 2021, Vol. 186, Art. 107763. DOI: 10.1016/j.comnet.2020.107763.

22. Lorenz P. QoS and QoE in the Next Generation Networks and Wireless Networks. In: Obaidat M., Filipe J. (eds.) E-Business and Telecommunications. CCIS, Vol. 456. Springer, 2014, pp. 3–16. DOI: 10.1007/978-3-662-44788-8_1.

23. Ibarrola E., Davis M., Voisin C., Close C., Cristobo L. QoE Enhancement in Next Generation Wireless Ecosystems: A Machine Learning Approach. IEEE Communications Standards Magazine, 2019, Vol. 3, No. 3, pp. 63–70. DOI: 10.1109/MCOMSTD.001.1900001.

24. ITU-T Recommendation Y.1541. Network Performance Objectives for IP-based Services. ITU-T, Geneva, 2011.

25. Romano S.P., et al. IMT-2020 Requirements and Realization. Proc. IEEE MOCAST, 2017, pp. 1–6.

26. Atxutegi E., Liberal F., Saiz E., Ibarrola E. Toward Standardized Internet Speed Measurements for End Users. IEEE Communications Magazine, 2016, Vol. 54, No. 9, pp. 50–57.

27. Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing, 2004, Vol. 13, No. 4, pp. 600–612. DOI: 10.1109/TIP.2003.819861.

28. Rassool R. VMAF Reproducibility: Validating a Perceptual Practical Video Quality Metric. Proc. IEEE BMSB, 2017, pp. 1–2. DOI: 10.1109/BMSB.2017.7986143.

29. ITU-R Report M.2410-0. Minimum Requirements Related to Technical Performance for IMT-2020 Radio Interface(s). ITU-R, Geneva, 2017.

30. Bega D., Gramaglia M., Taleb T., Fiore M., Banchs A. QoS-based Network Slicing for 5G Systems. Proc. IEEE CSCN, 2017, pp. 1–7.

31. Chowdhury S.R., Salahuddin M.A., Limam N., Boutaba R. Re-Architecting NFV Ecosystem with Microservices. IEEE Network, 2019, Vol. 33, No. 3, pp. 168–176. DOI: 10.1109/MNET.2019.1800082.

32. Marques G., Garcia N., Pombo N. A Survey on IoT: Architectures, Elements, Applications, QoS, Platforms and Security Concepts. Springer, 2017, pp. 115–130.

33. Profanter S., Tekat A., Dorofeev K., Rickert M., Knoll A. OPC UA versus ROS, DDS, and MQTT. Proc. IEEE Industry 4.0 Workshop, 2019.

34. Baraković S., Skorin-Kapov L. Survey and Challenges of QoE Management Issues in Wireless Networks. The Scientific World Journal, 2013, Art. ID 165146.

35. Andrulionis P., Andrulionis V., et al. Multi-Layer QoE Learning System Implemented by Fiberhost. Applied Sciences, 2023, Vol. 13, No. 4, 2300.

36. Boutaba R., Salahuddin M.A., Limam N., et al. A Comprehensive Survey on Machine Learning for Networking. Journal of Internet Services and Applications, 2018, Vol. 9, No. 1, Art. 16.

37. Tan K.H., Sani N.S., Goh A., et al. Modelling and Predicting QoE of Online Video Services. International Journal of Technology, 2022, Vol. 13, No. 3, pp. 567–577.

38. Garcia-Teodoro P., Díaz-Verdejo J., et al. Autoencoders for Anomaly Detection. Sensors, 2020, Vol. 20, No. 18, 5173.

39. Malhotra P., Ramakrishnan A., Anand G., et al. LSTM-based Encoder-Decoder for Multi-Sensor Anomaly Detection. arXiv:1607.00148, 2016.

40. Schölkopf B., Platt J.C., Shawe-Taylor J., Smola A.J., Williamson R.C. Estimating the Support of a High-Dimensional Distribution. Neural Computation, 2001, Vol. 13, No. 7, pp. 1443–1471.

41. Zhang Y., et al. SA2E-AD. ACM TKDD, 2024, Vol. 18, No. 3, Art. 45.

42. Lam K.N., Wong K.I. Traffic Forecasting Using LSTM, RF and XGBoost. Proc. ICPRAM, 2024, pp. 745–752.

43. Shao Y., Zhou C., Wang H., Zhou C. Hybrid TCN–LSTM for Network Traffic Prediction. IEEE TASE, 2022, Vol. 19, No. 3, pp. 1869–1882.

44. Jiang W., Luo Z., Chan H.C.B. Graph Neural Network for Traffic Forecasting: A Survey. ISPRS IJGI, 2023, Vol. 12, No. 3, Art. 100.

45. Riesel M. Root Cause Analysis Using Bayesian Networks for IPTV Service Quality Degradation. M.Sc. Thesis. KTH, Stockholm, 2019.

46. Wehner C., Thamsen L., Bermbach D. Interactive and Intelligent Root Cause Analysis. Proc. IEEE CCGrid, 2024, pp. 381–392.

47. Luo Y., Shao Y., Wang H. Spatio-Temporal Ensemble Network with Attention. Neural Networks, 2024, Vol. 172, pp. 106–120.

48. Geiginger L.-M. Classification of Encrypted QUIC Network Traffic. M.Sc. Thesis. TU Wien, 2021.

49. Sharma A., Ghita B., Furnell S. Survey on Encrypted Network Traffic. Computer Networks, 2024, Vol. 254, Art. 110984.

50. Li H., Guo Y., et al. ConvLSTMTransNet. arXiv:2409.13179, 2024.

51. Ahmed S.F., Zhang Y., et al. Traffic Forecasting with GNNs and Data Fusion. Neurocomputing, 2024, Vol. 566, Art. 127108.

52. Salahuddin M.A., Pradhan R., Limam N., Boutaba R. Machine Learning for Anomaly Detection and RCA in SDN/NFV. Springer, 2021, pp. 145–170.

53. D'Angelo G., et al. ML for Cloud-Native Observability. Proc. IEEE/ACM Workshop, 2023.

54. Cisco Systems. Applying Machine Learning and AI in Cisco DNA Center. Cisco Live, 2022.

55. Krishnan S., Ryabinin E., et al. ML for SRE. Proc. USENIX SREcon, 2020.

56. Chen Y., et al. AI-Driven Network Slicing Management. Proc. IEEE NetSoft, 2021.

57. Bega D., Gramaglia M., Banchs A., et al. AI-Driven Traffic Engineering in SD-WAN. Proc. IEEE CSCN, 2022.

58. Cardoso R., Monteiro E., Sargento S. Observability in Cloud-Native and SDN/NFV Networks. IEEE Communications Surveys & Tutorials, 2023, Vol. 25, No. 4, pp. 2501–2535.

59. RFC 9232. Network Telemetry Framework. IETF, 2022.

60. ETSI GS NFV 006 V4.4.1. NFV Management and Orchestration; Architectural Framework. ETSI, 2022.

61. Van Rossem S., Tavernier W., Hesmans B., et al. Monitoring and Debugging NFV Service Graphs. Proc. IFIP/IEEE IM, 2017.

62. Eichelberger R.A., Scholler M., Hollick M. SFC Path Tracer. Proc. IFIP/IEEE IM, 2017.

63. Tan L., Su J. In-band Network Telemetry: A Survey. Computer Networks, 2021, Vol. 190, Art. 107969.

64. Parniewicz D., et al. In-Band Network Telemetry Tests in NREN Networks. GÉANT White Paper, 2021.

65. Joshi M. Implementation and Evaluation of In-Band Network Telemetry in P4. M.Sc. Thesis. KTH, 2021.

66. Jiang W., Wang X., Zhang Y. Graph Neural Networks for Routing Optimization. Sustainability, 2024, Vol. 16, No. 21, 9239.

67. Rusek K., Suárez-Varela J., Mestres A., et al. Graph Neural Networks for Network Modeling and Optimization. IEEE/ACM Transactions on Networking, 2020, Vol. 28, No. 6, pp. 3269–3282.

68. Zhunussov A., Baikenov A., Manankova O., et al. Statistical and ML Analysis of PPPoE Sessions. Engineering, Technology & Applied Science Research, 2025, Vol. 15, No. 1, pp. 1234–1242.

69. RFC 2516. A Method for Transmitting PPP over Ethernet (PPPoE). IETF, 1999.

70. RFC 1661. The Point-to-Point Protocol (PPP). IETF, 1994.

71. RFC 1334. PPP Authentication Protocols. IETF, 1992.

72. Zhunussov A., Baikenov A., Ilieva D. Monitoring QoS by PPPoE Packet Statistics. Proc. IEEE EE&AE 2020, 2020.

73. Ross S.M. Stochastic Processes. 2nd ed. Wiley, New York, 1996.

74. Kleinrock L. Queueing Systems. Volume I: Theory. New York, 1975.

75. Montgomery D.C., Peck E.A., Vining G.G. Introduction to Linear Regression Analysis. 5th ed. Wiley, 2012.

76. Seber G.A.F., Lee A.J. Linear Regression Analysis. 2nd ed. Wiley, 2003.

77. Belsley D.A., Kuh E., Welsch R.E. Regression Diagnostics. Wiley, 1980.

78. Tikhonov A.N., Arsenin V.Y. Solutions of Ill-posed Problems. Winston & Sons, 1977.

79. Hoerl A.E., Kennard R.W. Ridge Regression. Technometrics, 1970, Vol. 12, No. 1, pp. 55–67.

80. Tibshirani R. Regression Shrinkage and Selection via the Lasso. JRSS-B, 1996, Vol. 58, No. 1, pp. 267–288.

81. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. 2nd ed. Springer, 2009.

82. Ho T.K. Random Decision Forests. Proc. ICDAR, IEEE, 1995.

83. Biau G., Scornet E. A Random Forest Guided Tour. TEST, 2016, Vol. 25, No. 2, pp. 197–227.

84. Louppe G. Understanding Random Forests. PhD Thesis. University of Liège, 2014.

85. Probst P., Wright M.N., Boulesteix A.-L. Hyperparameters and Tuning Strategies for Random Forest. WIREs DMKD, 2019.

86. Friedman J.H. Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics, 2001, Vol. 29, No. 5, pp. 1189–1232.

87. Chen T., Guestrin C. XGBoost. Proc. ACM SIGKDD, 2016.

88. Ke G., Meng Q., Finley T., et al. LightGBM. Proc. NeurIPS, 2017.

89. Koza J.R. Genetic Programming. MIT Press, 1992.

90. Schmidt M., Lipson H. Distilling Free-form Natural Laws from Experimental Data. Science, 2009, Vol. 324, No. 5923, pp. 81–85.

91. Cranmer M., Sanchez-Gonzalez A., Battaglia P., et al. Discovering Symbolic Models. Proc. NeurIPS, 2020.

92. Udrescu S.-M., Tegmark M. AI Feynman. Science Advances, 2020, Vol. 6, No. 16, eaay2631.

93. Box G.E.P., Jenkins G.M., Reinsel G.C., Ljung G.M. Time Series Analysis. 5th ed. Wiley, 2015.

94. Brockwell P.J., Davis R.A. Introduction to Time Series and Forecasting. 3rd ed. Springer, 2016.

95. Hyndman R.J., Athanasopoulos G. Forecasting: Principles and Practice. 3rd ed. OTexts, 2021.

# ПРИЛОЖЕНИЕ А

Таблицы с исходными данными.

В таблицах представлены исходные статистические данные PPPoE, собранные с граничного маршрутизатора оператора связи, и результаты активных замеров потери пакетов с интервалом 5 минут.

Таблица Б.1 — Исходные статистические данные PPPoE-сессий (Составлено автором по результатам собственных вычислительных экспериментов)

| | xe-0/1/0.3221700764 | | | | | |
|---|---|---|---|---|---|---|
| Время измерения | Общее количество сессий | PADI | PADO | PADR | PADS | PADT |
| 2022-05-23 14:54:20 | 1487 | 0 | 1444008 | 0 | 561354 | 540058 |
| 2022-05-23 14:59:31 | 1487 | 0 | 1444123 | 0 | 561360 | 540064 |
| 2022-05-23 15:04:47 | 1486 | 0 | 1444240 | 0 | 561369 | 540074 |
| 2022-05-23 15:10:03 | 1487 | 0 | 1444366 | 0 | 561387 | 540091 |
| 2022-05-23 15:15:19 | 1200 | 0 | 1444503 | 0 | 561413 | 540404 |
| 2022-05-23 15:20:35 | 1203 | 0 | 1444628 | 0 | 561427 | 540415 |
| 2022-05-23 15:25:51 | 1206 | 0 | 1444956 | 0 | 561626 | 540608 |
| 2022-05-23 15:31:07 | 1204 | 0 | 1445071 | 0 | 561633 | 540617 |
| 2022-05-23 15:36:23 | 1206 | 0 | 1445210 | 0 | 561643 | 540625 |
| 2022-05-23 15:41:39 | 1207 | 0 | 1445330 | 0 | 561650 | 540631 |
| 2022-05-23 15:46:55 | 1246 | 0 | 1445488 | 0 | 561700 | 540642 |
| 2022-05-23 15:52:11 | 1221 | 0 | 1445615 | 0 | 561719 | 540686 |
| 2022-05-23 15:57:27 | 1253 | 0 | 1445766 | 0 | 561759 | 540694 |
| 2022-05-23 16:02:43 | 1258 | 0 | 1445893 | 0 | 561775 | 540705 |
| 2022-05-23 16:07:59 | 1259 | 0 | 1446036 | 0 | 561796 | 540725 |
| 2022-05-23 16:13:15 | 1257 | 0 | 1446228 | 0 | 561807 | 540738 |
| 2022-05-23 16:18:31 | 1261 | 0 | 1446349 | 0 | 561820 | 540747 |
| 2022-05-23 16:23:47 | 1257 | 0 | 1446478 | 0 | 561834 | 540765 |
| 2022-05-23 16:29:03 | 1301 | 0 | 1446653 | 0 | 561892 | 540778 |
| 2022-05-23 16:34:19 | 1261 | 0 | 1446798 | 0 | 561923 | 540849 |
| 2022-05-23 16:39:35 | 1263 | 0 | 1446934 | 0 | 561938 | 540862 |
| 2022-05-23 16:44:51 | 1222 | 0 | 1447073 | 0 | 561954 | 540919 |
| 2022-05-23 16:50:07 | 1274 | 0 | 1447245 | 0 | 562017 | 540930 |
| 2022-05-23 16:55:23 | 1271 | 0 | 1447360 | 0 | 562026 | 540942 |
| 2022-05-23 17:00:39 | 1474 | 0 | 1447731 | 0 | 562262 | 540970 |
| 2022-05-23 17:05:55 | 1481 | 0 | 1447869 | 0 | 562289 | 540989 |
| 2022-05-23 17:11:11 | 1487 | 0 | 1448026 | 0 | 562308 | 541002 |
| 2022-05-23 17:16:27 | 1495 | 0 | 1448167 | 0 | 562339 | 541025 |
| 2022-05-23 17:21:43 | 1494 | 0 | 1448306 | 0 | 562353 | 541040 |
| 2022-05-23 17:26:59 | 1497 | 0 | 1448439 | 0 | 562376 | 541059 |
| 2022-05-23 17:32:15 | 1502 | 0 | 1448563 | 0 | 562393 | 541071 |
| 2022-05-23 17:37:31 | 1500 | 0 | 1448699 | 0 | 562421 | 541101 |
| 2022-05-23 17:42:47 | 1503 | 0 | 1448835 | 0 | 562444 | 541121 |

Продолжение таблицы Б.1

| Время измерения | Общее количество сессий | PADI | PADO | PADR | PADS | PADT |
|---|---|---|---|---|---|---|
| 2022-05-23 17:48:03 | 1504 | 0 | 1448973 | 0 | 562457 | 541133 |
| 2022-05-23 17:53:19 | 1508 | 0 | 1449106 | 0 | 562472 | 541144 |
| 2022-05-23 17:58:35 | 1508 | 0 | 1449248 | 0 | 562488 | 541160 |
| 2022-05-23 18:03:51 | 1503 | 0 | 1449390 | 0 | 562507 | 541182 |
| 2022-05-23 18:09:07 | 1502 | 0 | 1449571 | 0 | 562525 | 541200 |
| 2022-05-23 18:14:23 | 1504 | 0 | 1449735 | 0 | 562544 | 541217 |
| 2022-05-23 18:19:39 | 1506 | 0 | 1449901 | 0 | 562564 | 541235 |
| 2022-05-23 18:24:55 | 1508 | 0 | 1449993 | 0 | 562582 | 541251 |
| 2022-05-23 18:30:11 | 1510 | 0 | 1450046 | 0 | 562598 | 541264 |
| 2022-05-23 18:35:27 | 1511 | 0 | 1450101 | 0 | 562616 | 541281 |
| 2022-05-23 18:40:43 | 1511 | 0 | 1450185 | 0 | 562636 | 541301 |
| 2022-05-23 18:45:59 | 1512 | 0 | 1450246 | 0 | 562650 | 541314 |
| 2022-05-23 18:51:15 | 1512 | 0 | 1450282 | 0 | 562669 | 541333 |
| 2022-05-23 18:56:31 | 1513 | 0 | 1450308 | 0 | 562685 | 541347 |
| 2022-05-23 19:01:47 | 1513 | 0 | 1450351 | 0 | 562701 | 541363 |
| 2022-05-23 19:07:03 | 1510 | 0 | 1450451 | 0 | 562719 | 541383 |
| 2022-05-23 19:50:03 | 1513 | 0 | 1450910 | 0 | 562865 | 541523 |
| 2022-05-23 19:55:15 | 1516 | 0 | 1450999 | 0 | 562888 | 541543 |
| 2022-05-23 20:00:31 | 1519 | 0 | 1451037 | 0 | 562908 | 541559 |
| 2022-05-23 20:05:47 | 1520 | 0 | 1451060 | 0 | 562925 | 541575 |
| 2022-05-23 20:11:03 | 1520 | 0 | 1451087 | 0 | 562947 | 541597 |
| 2022-05-23 20:16:19 | 1518 | 0 | 1451114 | 0 | 562972 | 541622 |
| 2022-05-23 20:21:35 | 1515 | 0 | 1451178 | 0 | 563027 | 541678 |
| 2022-05-23 20:26:51 | 1506 | 0 | 1451206 | 0 | 563053 | 541713 |
| 2022-05-23 20:32:07 | 1517 | 0 | 1451268 | 0 | 563088 | 541734 |
| 2022-05-23 20:37:23 | 1520 | 0 | 1451296 | 0 | 563106 | 541748 |
| 2022-05-23 20:42:39 | 1521 | 0 | 1451328 | 0 | 563131 | 541771 |
| 2022-05-23 20:47:55 | 1520 | 0 | 1451373 | 0 | 563151 | 541792 |
| 2022-05-23 20:53:11 | 1520 | 0 | 1451401 | 0 | 563171 | 541812 |
| 2022-05-23 20:58:27 | 1518 | 0 | 1451451 | 0 | 563201 | 541844 |
| 2022-05-23 21:03:43 | 1520 | 0 | 1451515 | 0 | 563233 | 541872 |
| 2022-05-23 21:08:59 | 1517 | 0 | 1451601 | 0 | 563266 | 541906 |
| 2022-05-23 21:14:15 | 1518 | 0 | 1451712 | 0 | 563300 | 541937 |
| 2022-05-23 21:19:31 | 1518 | 0 | 1451835 | 0 | 563337 | 541973 |
| 2022-05-23 21:24:47 | 1514 | 0 | 1451942 | 0 | 563365 | 542004 |
| 2022-05-23 21:30:03 | 1516 | 0 | 1452053 | 0 | 563398 | 542033 |
| 2022-05-23 21:35:19 | 1517 | 0 | 1452151 | 0 | 563435 | 542066 |
| 2022-05-23 21:40:35 | 1520 | 0 | 1452264 | 0 | 563467 | 542094 |
| 2022-05-23 21:45:51 | 1517 | 0 | 1452346 | 0 | 563500 | 542128 |
| 2022-05-23 21:51:07 | 1513 | 0 | 1452405 | 0 | 563536 | 542167 |
| 2022-05-23 21:56:23 | 1514 | 0 | 1452464 | 0 | 563572 | 542201 |
| 2022-05-23 22:01:39 | 1520 | 0 | 1452508 | 0 | 563602 | 542224 |
| 2022-05-23 22:06:55 | 1517 | 0 | 1452536 | 0 | 563627 | 542251 |

Продолжение таблицы Б.1

| Время измерения | Общее количество сессий | PADI | PADO | PADR | PADS | PADT |
|---|---|---|---|---|---|---|
| 2022-05-23 22:12:11 | 1515 | 0 | 1452563 | 0 | 563650 | 542276 |
| 2022-05-23 22:17:27 | 1514 | 0 | 1452616 | 0 | 563684 | 542310 |
| 2022-05-23 22:22:43 | 1515 | 0 | 1452652 | 0 | 563707 | 542331 |
| 2022-05-23 22:27:59 | 1514 | 0 | 1452686 | 0 | 563736 | 542360 |
| 2022-05-23 22:33:15 | 1512 | 0 | 1452720 | 0 | 563762 | 542388 |
| 2022-05-23 22:38:31 | 1510 | 0 | 1452745 | 0 | 563781 | 542407 |
| 2022-05-23 22:43:47 | 1508 | 0 | 1452821 | 0 | 563802 | 542428 |
| 2022-05-23 22:49:03 | 1508 | 0 | 1452888 | 0 | 563831 | 542456 |
| 2022-05-23 22:54:19 | 1508 | 0 | 1452950 | 0 | 563849 | 542474 |
| 2022-05-23 22:59:35 | 1508 | 0 | 1452982 | 0 | 563876 | 542498 |
| 2022-05-23 23:04:51 | 1505 | 0 | 1453054 | 0 | 563894 | 542519 |
| 2022-05-23 23:10:07 | 1500 | 0 | 1453092 | 0 | 563919 | 542548 |
| 2022-05-23 23:15:23 | 1498 | 0 | 1453119 | 0 | 563940 | 542571 |
| 2022-05-23 23:20:39 | 1501 | 0 | 1453212 | 0 | 563971 | 542596 |
| 2022-05-23 23:25:55 | 1503 | 0 | 1453274 | 0 | 563992 | 542614 |
| 2022-05-23 23:31:11 | 1497 | 0 | 1453339 | 0 | 564007 | 542633 |
| 2022-05-23 23:36:27 | 1500 | 0 | 1453402 | 0 | 564027 | 542650 |
| 2022-05-23 23:41:43 | 1499 | 0 | 1453456 | 0 | 564046 | 542669 |
| 2022-05-23 23:46:59 | 1497 | 0 | 1453491 | 0 | 564060 | 542685 |
| 2022-05-23 23:52:15 | 1493 | 0 | 1453525 | 0 | 564081 | 542707 |
| 2022-05-23 23:57:31 | 1492 | 0 | 1453545 | 0 | 564098 | 542724 |
| 2022-05-24 00:02:47 | 1495 | 0 | 1453562 | 0 | 564113 | 542736 |

Таблица Б.2 — Результаты активных измерений потерь пакетов (составлено автором по результатам собственных вычислительных экспериментов)

| Время измерения | Количество отправленных пакетов | Количество потерянных пакетов | Потери пакетов, % |
|---|---|---|---|
| 23.05.2022 14:54 | 300 | 0 | 0 |
| 23.05.2022 14:59 | 300 | 1 | 0,33 |
| 23.05.2022 15:04 | 300 | 1 | 0,33 |
| 23.05.2022 15:10 | 300 | 2 | 0,67 |
| 23.05.2022 15:15 | 300 | 8 | 2,67 |
| 23.05.2022 15:20 | 300 | 1 | 0,33 |
| 23.05.2022 15:25 | 300 | 7 | 2,33 |
| 23.05.2022 15:31 | 300 | 1 | 0,33 |
| 23.05.2022 15:36 | 300 | 1 | 0,33 |
| 23.05.2022 15:41 | 300 | 0 | 0 |
| 23.05.2022 15:46 | 300 | 0 | 0 |
| 23.05.2022 15:52 | 300 | 3 | 1 |
| 23.05.2022 15:57 | 300 | 0 | 0 |
| 23.05.2022 16:02 | 300 | 1 | 0,33 |
| 23.05.2022 16:07 | 300 | 2 | 0,67 |
| 23.05.2022 16:13 | 300 | 1 | 0,33 |

Продолжение таблицы Б.2

| Время измерения | Количество отправленных пакетов | Количество потерянных пакетов | Потери пакетов, % |
|---|---|---|---|
| 23.05.2022 16:18 | 300 | 0 | 0 |
| 23.05.2022 16:23 | 300 | 2 | 0,67 |
| 23.05.2022 16:29 | 300 | 1 | 0,33 |
| 23.05.2022 16:34 | 300 | 5 | 1,67 |
| 23.05.2022 16:39 | 300 | 1 | 0,33 |
| 23.05.2022 16:44 | 300 | 4 | 1,33 |
| 23.05.2022 16:50 | 300 | 1 | 0,33 |
| 23.05.2022 16:55 | 300 | 1 | 0,33 |
| 23.05.2022 17:00 | 300 | 2 | 0,67 |
| 23.05.2022 17:05 | 300 | 1 | 0,33 |
| 23.05.2022 17:11 | 300 | 0 | 0 |
| 23.05.2022 17:16 | 300 | 2 | 0,67 |
| 23.05.2022 17:21 | 300 | 1 | 0,33 |
| 23.05.2022 17:26 | 300 | 2 | 0,67 |
| 23.05.2022 17:32 | 300 | 1 | 0,33 |
| 23.05.2022 17:37 | 300 | 3 | 1 |
| 23.05.2022 17:42 | 300 | 2 | 0,67 |
| 23.05.2022 17:48 | 300 | 1 | 0,33 |
| 23.05.2022 17:53 | 300 | 0 | 0 |
| 23.05.2022 17:58 | 300 | 1 | 0,33 |
| 23.05.2022 18:03 | 300 | 2 | 0,67 |
| 23.05.2022 18:09 | 300 | 1 | 0,33 |
| 23.05.2022 18:14 | 300 | 1 | 0,33 |
| 23.05.2022 18:19 | 300 | 1 | 0,33 |
| 23.05.2022 18:24 | 300 | 1 | 0,33 |
| 23.05.2022 18:30 | 300 | 1 | 0,33 |
| 23.05.2022 18:35 | 300 | 1 | 0,33 |
| 23.05.2022 18:40 | 300 | 2 | 0,67 |
| 23.05.2022 18:45 | 300 | 1 | 0,33 |
| 23.05.2022 18:51 | 300 | 2 | 0,67 |
| 23.05.2022 18:56 | 300 | 1 | 0,33 |
| 23.05.2022 19:01 | 300 | 1 | 0,33 |
| 23.05.2022 19:07 | 300 | 2 | 0,67 |
| 23.05.2022 19:50 | 300 | 1 | 0,33 |
| 23.05.2022 19:55 | 300 | 1 | 0,33 |
| 23.05.2022 20:00 | 300 | 1 | 0,33 |
| 23.05.2022 20:05 | 300 | 1 | 0,33 |
| 23.05.2022 20:11 | 300 | 2 | 0,67 |
| 23.05.2022 20:16 | 300 | 2 | 0,67 |
| 23.05.2022 20:21 | 300 | 4 | 1,33 |
| 23.05.2022 20:26 | 300 | 3 | 1 |
| 23.05.2022 20:32 | 300 | 1 | 0,33 |
| 23.05.2022 20:37 | 300 | 0 | 0 |
| 23.05.2022 20:42 | 300 | 2 | 0,67 |
| 23.05.2022 20:47 | 300 | 2 | 0,67 |

Продолжение таблицы Б.2

| Время измерения | Количество отправленных пакетов | Количество потерянных пакетов | Потери пакетов, % |
|---|---|---|---|
| 23.05.2022 20:53 | 300 | 2 | 0,67 |
| 23.05.2022 20:58 | 300 | 2 | 0,67 |
| 23.05.2022 21:03 | 300 | 2 | 0,67 |
| 23.05.2022 21:08 | 300 | 3 | 1 |
| 23.05.2022 21:14 | 300 | 3 | 1 |
| 23.05.2022 21:19 | 300 | 3 | 1 |
| 23.05.2022 21:24 | 300 | 2 | 0,67 |
| 23.05.2022 21:30 | 300 | 2 | 0,67 |
| 23.05.2022 21:35 | 300 | 3 | 1 |
| 23.05.2022 21:40 | 300 | 2 | 0,67 |
| 23.05.2022 21:45 | 300 | 3 | 1 |
| 23.05.2022 21:51 | 300 | 4 | 1,33 |
| 23.05.2022 21:56 | 300 | 3 | 1 |
| 23.05.2022 22:01 | 300 | 2 | 0,67 |
| 23.05.2022 22:06 | 300 | 2 | 0,67 |
| 23.05.2022 22:12 | 300 | 2 | 0,67 |
| 23.05.2022 22:17 | 300 | 3 | 1 |
| 23.05.2022 22:22 | 300 | 2 | 0,67 |
| 23.05.2022 22:27 | 300 | 3 | 1 |
| 23.05.2022 22:33 | 300 | 3 | 1 |
| 23.05.2022 22:38 | 300 | 2 | 0,67 |
| 23.05.2022 22:43 | 300 | 3 | 1 |
| 23.05.2022 22:49 | 300 | 3 | 1 |
| 23.05.2022 22:54 | 300 | 3 | 1 |
| 23.05.2022 22:59 | 300 | 3 | 1 |
| 23.05.2022 23:04 | 300 | 3 | 1 |
| 23.05.2022 23:10 | 300 | 3 | 1 |
| 23.05.2022 23:15 | 300 | 3 | 1 |
| 23.05.2022 23:20 | 300 | 3 | 1 |
| 23.05.2022 23:25 | 300 | 2 | 0,67 |
| 23.05.2022 23:31 | 300 | 3 | 1 |
| 23.05.2022 23:36 | 300 | 2 | 0,67 |
| 23.05.2022 23:41 | 300 | 1 | 0,33 |
| 23.05.2022 23:46 | 300 | 1 | 0,33 |
| 23.05.2022 23:52 | 300 | 2 | 0,67 |
| 23.05.2022 23:57 | 300 | 1 | 0,33 |
| 24.05.2022 0:02 | 300 | 1 | 0,33 |

# ПРИЛОЖЕНИЕ Б

Исходный код лабораторного стенда, разработанного в среде MATLAB.

Приведённый код иллюстрирует принцип работы имитационной модели и используется для подтверждения статистической состоятельности предложенного коэффициента нестабильности. Результаты моделирования представлены в третьей главе.

Имитационная модель для проверки формулы коэффициента нестабильности K. Модель предназначена для статистической проверки справедливости выражения: $K$ = (PADT2 - PADT1) / S, где S — число активных PPPoE-сессий, а PADT1 и PADT2 — счётчики завершений.

```
% 1. Параметры эксперимента
% Раздел 1: фиксирование входных параметров.
S = 1000;      % число активных сессий (фиксированное S для всех прогонов)
ps = [0.01 0.02 0.05 0.1 0.2];          % истинные вероятности разрыва p (множество тестовых значений)
N  = 1e4;  % число интервалов наблюдения (размер выборки для каждого p)
mean_K  = zeros(size(ps));        % вектор средних оценок K по каждому p
ci_low  = zeros(size(ps));          % нижняя граница 95% доверительного интервала для K
ci_high = zeros(size(ps));          % верхняя граница 95% доверительного интервала для K
% (рекоменд.) Для воспроизводимости можно зафиксировать генератор:
% rng(42,'twister'); тогда результаты CI будут идентичны при повторном запуске.

% 2. Основной цикл моделирования
% Раздел 2: по каждому p проводится N испытаний.
for idx = 1:length(ps) % цикл по наборам истинных вероятностей p
  p = ps(idx); % текущее значение p

% Генерация N реализаций числа отброшенных сессий (биномиальная модель) dPADT ~ Binomial(S, p)
  dPADT = binornd(S, p, [N, 1]);        % в каждом интервале: сколько из S сессий разорвались с вероятностью p

% Расчёт оценок коэффициента нестабильности в каждом интервале
% Оценка K = ΔPADT / S
```

```
  K_est = dPADT / S;      % вектор длины N: безразмерная доля разрывов на интервал

% Оценка среднего значения и доверительного интервала
% Статистика по K_est
  mean_K(idx)  = mean(K_est);  % несмещённая оценка матожидания K (для большого N)
  se = std(K_est) / sqrt(N);          % стандартная ошибка среднего (std с делителем N-1 по умолчанию)
  ci_low(idx) = mean_K(idx) - 1.96 * se; % нижняя граница нормального 95% ДИ (аппроксимация)
  ci_high(idx) = mean_K(idx) + 1.96 * se;          % верхняя граница нормального 95% ДИ (аппроксимация)
end

% 3. Визуализация результатов
% Раздел 3: графическое сравнение с теорией.
figure; hold on; grid on;                % подготовка области построения
plot(ps, mean_K, 'bo-', 'LineWidth', 1.5);          % точки/ломаная: эмпирические средние <K>
plot(ps, ps, 'r--', 'LineWidth', 1.5);       % линия y=x: аналитическое ожидание K = p
for idx = 1:length(ps)                % покомпонентное рисование 95% ДИ
  plot([ps(idx) ps(idx)], [ci_low(idx) ci_high(idx)], 'k-',
'LineWidth', 1);                                % вертикальные отрезки CI
end
xlabel('Истинная вероятность p');                % подпись оси X
ylabel('Средняя оценка \langle K \rangle');          % подпись оси Y (с математическим средним)
legend('Имитация','Аналитическая зависимость y = x',...       % легенда графика
       '95% доверительный интервал','Location','NorthWest');
title('Проверка справедливости оценки K = \DeltaPADT / S'); % заголовок рисунка
hold off;                                % освобождение удержания графика
```

## ПРИЛОЖЕНИЕ В

Исходный код скрипта для линейных и ансамблевых регрессий

Данный программный модуль реализует процедуру машинного обучения для верификации корректности коэффициента нестабильности *K*, вычисляемого на основе статистики PPPoE-сообщений. Скрипт осуществляет загрузку и предобработку данных из Excel-файла, расчёт целевой переменной *K*, формирование обучающей выборки, обучение регрессионных моделей с перекрёстной проверкой (кросс-валидацией) и визуализацию результатов.

```
# Назначение: Верификация формулы коэффициента нестабильности K
# Среда выполнения: Python 3.10+, библиотеки pandas, numpy, scikit-learn, matplotlib

# --- Импорт библиотек ---
import pandas as pd                 # Работа с табличными данными (DataFrame)
import numpy as np                  # Математические операции и работа с массивами
import matplotlib.pyplot as plt     # Визуализация результатов

# Инструменты машинного обучения
from sklearn.model_selection import KFold, GridSearchCV, cross_validate
                                    # Кросс-валидация и подбор параметров
from sklearn.linear_model import LinearRegression, Ridge, Lasso  # Линейные модели (OLS, Ridge, Lasso)
from sklearn.ensemble import RandomForestRegressor      # Ансамблевая модель — случайный лес
from sklearn.svm import SVR         # Метод опорных векторов (регрессия)
from sklearn.metrics import mean_squared_error, r2_score, make_scorer
                                    # Метрики качества (MSE и R²)

def load_and_prepare_multicol_data(excel_path):     # Загрузка Excel-файла с многоуровневыми заголовками
  df_raw = pd.read_excel(excel_path, header=[0, 1], engine="openpyxl")
# Объединение уровней заголовков в один с разделителем «|» (пример: "xe-0/0/1 | PADT")
  df_raw.columns = [' | '.join([str(c) for c in col]).strip() for col in df_raw.columns]  # Поиск колонки с временем (могут быть разные варианты названия)
  time_col = [col for col in df_raw.columns if 'Time' in col or 'Unnamed' in col][0]       # Приведение времени к формату datetime и очищаем суффикс «ALMT»
```

```
  df_raw['Time'] =
pd.to_datetime(df_raw[time_col].astype(str).str.replace("", ""),
errors='coerce')
    df_raw = df_raw.drop(columns=[time_col])          # Определение списка
интерфейсов и параметров
  columns = df_raw.columns.drop('Time')
    interfaces = sorted(set(col.split(' | ')[0] for col in columns))
                                                      # названия интерфейсов
  params = sorted(set(col.split(' | ')[1] for col in columns))  # параметры
PPPoE
  # Преобразование таблицы из «широкой» формы в «длинную» (long)
  records = [4]
    for iface in interfaces:
        subdf = df_raw[['Time'] + [f"{iface} | {p}" for p in params if
f"{iface} | {p}" in df_raw.columns]].copy()
        subdf.columns = ['Time'] + params
        subdf['Interface'] = iface
        records.append(subdf)
  df_all = pd.concat(records)
    df_all = df_all.sort_values(by=['Interface', 'Time'])

  # --- Расчёт производных параметров ---
  df_all['PADT_prev'] = df_all.groupby('Interface')['PADT'].shift(1)
                                      # значение PADT на предыдущем шаге
  df_all['PADT_diff'] = df_all['PADT'] - df_all['PADT_prev']
                                              # изменение PADT за интервал
  df_all = df_all[df_all['Summary sessions'] != 0] # исключение интервалов
без активных сессий
  df_all = df_all.dropna(subset=['PADT_prev'])

  # Расчёт коэффициента нестабильности K = (PADT₂ - PADT₁) / S
  df_all['K'] = df_all['PADT_diff'] / df_all['Summary sessions']

  # Удаление бесконечных и пропущенных значений
  df_all = df_all.replace([np.inf, -np.inf], np.nan).dropna()

  # Определение признаков (входные параметры) и целевой переменной
  features = ['PADI', 'PADO', 'PADR', 'PADS', 'PADT_diff', 'Summary
sessions']
  X = df_all[features].copy()                         # матрица признаков
  y = df_all['K'].copy()              # вектор отклика (целевая переменная)

  return X, y, df_all
def evaluate_models_cv(X, y, cv_folds=5):       # Набор базовых моделей для
сравнения
  models = {
```

```
        'LinearRegression': LinearRegression(),# обычная линейная регрессия
        'Ridge': Ridge(alpha=1.0),                    # Ridge (L2-регуляризация)
    'Lasso': Lasso(alpha=0.1),                    # Lasso (L1-регуляризация)
  'RandomForest': RandomForestRegressor(n_estimators=100,         # число
деревьев в ансамбле (чем больше, тем устойчивее модель)
        random_state=42           #фиксирование seed для воспроизводимости
результатов
  )
}
# Список для накопления результатов оценки моделей
results = [4]
# Задавание набора метрик качества:
# MSE — среднеквадратичная ошибка,
# R²  — коэффициент детерминации
scoring = {
    'MSE': make_scorer(mean_squared_error),
    'R2': make_scorer(r2_score)
}
# Настройка k-fold кросс-валидации:
# случайное перемешивание и разбивка выборки на cv_folds подвыборок
kf = KFold(n_splits=cv_folds,                    # число фолдов кросс-валидации
  shuffle=True,                  # перемешивание данных перед разбиением
  random_state=42                # фиксирование генератора случайных чисел
)

# Последовательное обучение и оценка каждой модели из словаря models
for name, model in models.items():

  # Выполнение кросс-валидации:
  # обучение и тестирование модели на каждом фолде,
  # возвращение значения метрик и обученных экземпляров моделей
  scores = cross_validate(
    model,                              # текущая модель машинного обучения
    X,                                                  # матрица признаков
    y,                                                  # целевая переменная
    cv=kf,                                              # схема кросс-валидации
    scoring=scoring,                                    # набор вычисляемых метрик
    return_estimator=True      # сохранение обученных моделей каждого
фолда
  )
  # Вычисление среднего значения MSE по всем фолдам
  avg_mse = np.mean(scores['test_MSE'])
  # Вычисление среднего значения коэффициента детерминации R²
```

```
  avg_r2 = np.mean(scores['test_R2'])
  # Сохранение результатов для текущей модели
  results.append({
    'Model': name,                                    # название модели
    'Avg_MSE': avg_mse,                               # средняя MSE
    'Avg_R2': avg_r2,                                 # средний R²
    'BestEstimator': scores['estimator'][             # лучшая модель
      np.argmax(scores['test_R2'])                    # фолд с максимальным R²
       ]
   })
# Возвращение сводной таблицы результатов по всем моделям
return results
def tune_and_evaluate_svr(X, y, cv_folds=5):
    """
 # Функция подбора гиперпараметров и оценки модели опорных векторов для регрессии (SVR).
  """    # Сетка параметров для поиска по GridSearchCV
  param_grid = {
    'C': [0.1, 1, 10],                    # штраф за ошибку (регуляризация)
    'epsilon': [0.01, 0.1, 0.5],          # ширина ε-зоны нечувствительности
    'gamma': ['scale', 0.1, 0.01],                    # параметр ядра (для RBF)
    'kernel': ['rbf']
    }
    svr = SVR()
    grid = GridSearchCV(svr, param_grid, cv=cv_folds, scoring='r2', n_jobs=-
1)
    grid.fit(X, y)
    best_model = grid.best_estimator_
    y_pred = best_model.predict(X)
    mse = mean_squared_error(y, y_pred)
    r2 = r2_score(y, y_pred)
    return {
        'Model': 'SVR (Tuned)',
        'Avg_MSE': mse,
        'Avg_R2': r2,
        'BestEstimator': best_model
    } def plot_comparison(results):
    model_names = [r['Model'] for r in results]
    mses = [r['Avg_MSE'] for r in results]
    r2s = [r['Avg_R2'] for r in results]
    plt.figure(figsize=(10, 4))
  # График среднеквадратичной ошибки
  plt.subplot(1, 2, 1)
    plt.bar(model_names, mses)
    plt.title('Mean Squared Error (MSE)')
    plt.xticks(rotation=45)
  # График коэффициента детерминации
  plt.subplot(1, 2, 2)
```

```
    plt.bar(model_names, r2s)
    plt.title('R² Score')
    plt.xticks(rotation=45)
  plt.tight_layout()
    plt.show()
def main():
    excel_file_path = "PPPoE данные.xlsx"        # путь к файлу с исходными данными
    print("Загрузка и обработка данных с многоуровневыми заголовками...")
    X, y, df = load_and_prepare_multicol_data(excel_file_path)
    print("Обучение моделей с кросс-валидацией (k=5)...")
    results = evaluate_models_cv(X, y, cv_folds=5)
    print("Подбор гиперпараметров для SVR...")
    svr_result = tune_and_evaluate_svr(X, y, cv_folds=5)
    results.append(svr_result)
  # Вывод итоговых результатов
  print("\nРезультаты моделей:")
    for r in results:
        print(f"  {r['Model']}: Avg_MSE={r['Avg_MSE']:.6f}, Avg_R²={r['Avg_R2']:.4f}")
  best = sorted(results, key=lambda r: r['Avg_R2'], reverse=True)[0]
    print(f"\nЛучшая модель: {best['Model']} (Avg_R² = {best['Avg_R2']:.4f})")

  # Визуализация сравнения
    plot_comparison(results)

  # Сохранение итогов в CSV
  df_out = pd.DataFrame([{
        'Model': r['Model'],
        'Avg_MSE': r['Avg_MSE'],
        'Avg_R2': r['Avg_R2']
    } for r in results])
    df_out.to_csv("results.csv", index=False)
    print("Результаты сохранены в results.csv")

# Запуск при выполнении скрипта напрямую
if __name__ == "__main__":
    main()
```

## ПРИЛОЖЕНИЕ Г

Исходный код скрипта для символьной регрессии.

Символьная регрессия использовалась для независимого восстановления аналитической зависимости коэффициента *K* без априорного задания его формы. Это позволило проверить, является ли предложенное выражение естественным результатом оптимизации, а не следствием подгонки.

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from pysr import PySRRegressor

# 1. Пути и загрузка данных
data_path = r"DIR\pppoe_processed.csv"
results_path = r"C:\Program
Files\Software\SCRIPTS\PHD\pysr_symbolic_massrun.csv"
df = pd.read_csv(data_path)

# 2. Функция для вычисления дельт по всем признакам
def add_deltas(df, base_names):
    df = df.copy()
    for col in base_names:
        delta_col = f"delta_{col}"
        if delta_col not in df.columns and col in df.columns:
            df[delta_col] = df[col].diff().fillna(0)
    return df

# 3. Список комбинаций
feature_sets = [
    ['delta_PADI', 'PADO', 'PADR', 'PADS', 'PADT', 'Sessions'],
    ['PADI', 'delta_PADO', 'PADR', 'PADS', 'PADT', 'Sessions'],
    ['PADI', 'PADO', 'delta_PADR', 'PADS', 'PADT', 'Sessions'],
    ['PADI', 'PADO', 'PADR', 'delta_PADS', 'PADT', 'Sessions'],
    ['PADI', 'PADO', 'PADR', 'PADS', 'delta_PADT', 'Sessions'],
    ['PADI', 'PADO', 'PADR', 'PADS', 'PADT', 'delta_Sessions'],
    ['delta_PADI', 'delta_PADO', 'delta_PADR', 'delta_PADS', 'delta_PADT',
'delta_Sessions'],
]
target = 'K'

# 4. Добавление всех дельт
base_cols = ['PADI', 'PADO', 'PADR', 'PADS', 'PADT', 'Sessions']
df = add_deltas(df, base_cols)
results = [4]
for features in feature_sets:
```

```
    # Проверка, что все признаки есть в данных
    if not all(f in df.columns for f in features):
          print(f"Пропущено (нет всех колонок): {features}")
          continue

    # Отбрасывание строки с NaN
    subset = df[features + [target]].dropna()
      X = subset[features].values
      y = subset[target].values

    # Масштабирование X!
    scaler = StandardScaler()
      X_scaled = scaler.fit_transform(X)

    # PySR мощный режим!
    model = PySRRegressor(
          niterations=5000,
          populations=40,
          maxsize=30,
          binary_operators=["+", "-", "*", "/", "pow"],
          unary_operators=["sin", "cos", "exp", "log", "sqrt"],
          loss="loss(x, y) = (x - y)^2",
          progress=True,
          model_selection="best",
          temp_equation_file=True,
          turbo=True,
      )
      model.fit(X_scaled, y)

    # Извлечение лучшей формулы
    best_eq = model.equations_.iloc[0]
      formula = best_eq['equation'] if 'equation' in best_eq else
best_eq['Equation']
      r2 = best_eq['r2'] if 'r2' in best_eq else np.nan

      results.append({
          "features": ','.join(features),
          "r2": r2,
          "formula": formula,
      })

# 5. Сохранение результатов
results_df = pd.DataFrame(results)
results_df.to_csv(results_path, index=False, encoding='utf-8-sig')
print(f"Готово! Все результаты сохранены в {results_path}")
```

## ПРИЛОЖЕНИЕ Д

Скрипт сбора и обработки данных, написанный на языке программирования Python

```
# Скрипт сбора статистики PPPoE/абонентов с Juniper (Junos PyEZ) и записи в SQLite.
# Назначение: формировать базу для расчёта коэффициента нестабильности K (по PADT и количеству сессий).

from jnpr.junos import Device                    # Класс Device для установления NETCONF-сессии к Juniper.
from datetime import datetime                    # Метка времени для записей в БД.
from jnpr.junos.op.phyport import *              # (Не используется в коде) Операционные классы по портам; можно удалить для чистоты.
from sqlite3 import connect as con               # Подключение к SQLite, alias 'con' для краткости.
import time                                      # (Не используется явно) Возможны паузы/задержки, сейчас лишний импорт.
import os                                        # Работа с файлами (проверка/удаление DB_res.sqlite).
from jnpr.junos.factory.factory_loader import FactoryLoader  # Динамическая генерация таблиц/представлений PyEZ из YAML.
import yaml                                      # Разбор YAML-описаний таблиц/представлений RPC.

# Учетные данные и список хостов. В коде они маскированы.
username = '***'                                 # Имя пользователя для NETCONF (ssh).
password = '***'                                 # Пароль/ключ; в проде замените на безопасный способ (ssh-key/внешний секрет).
hosts = ("***")                                  # Кортеж хостов Juniper. # NOTE: из одной строки это будет ИТЕРИРУЕМАЯ ПО СИМВОЛАМ строка.
                                                 # Для одного хоста используйте ('host1',), для нескольких ('h1','h2',...).

# Соединение с локальной базой данных (основная БД накопления).
conn = con('DB.sqlite')                          # Если файла нет, SQLite создаст его автоматически.
cur = conn.cursor()                              # Курсор для выполнения SQL-запросов.

# Шаблон создания таблицы. Закомментирован, т.к. таблица вероятно уже создана.
#cur.executescript("""
#   BEGIN TRANSACTION;
```

```
#   Create table Scores(
#      ID INTEGER PRIMARY KEY AUTOINCREMENT,
#      port text,
#      vlan text,
#      score integer,
#      date text,
#      subs integer,
#      reg text
#   );
#   COMMIT;
#""")

# YAML-описание «операционных таблиц» для PyEZ FactoryLoader:
описывает RPC и отображение полей.
myYAML = """
---
SubscribersTable:
  rpc: get-subscribers
  args:
    detail: True
    client_type: 'vlan'
    dev_timeout: '90'
  item: subscriber
  key: access-type
  view: SubscribersView

SubscribersView:
  fields:
    demux: interface
    under: underlying-interface
    stack: stacked-vlan-id
    vlan: vlan-id

SessionsTable:
  rpc: get-pppoe-session-information
  item: pppoe-interface
  key: underlying-interface-name
  view: SessionsView

SessionsView:
  fields:
    demux: underlying-interface-name
"""
```

```
# Динамически регистрируем сгенерированные классы
(SubscribersTable/SessionsTable/…)
# WARNING: yaml.load без Loader генерирует предупреждение; используйте
SafeLoader.
globals().update(FactoryLoader().load(yaml.load(myYAML,
Loader=yaml.SafeLoader)))

# Главный бесконечный цикл сбора (polling). Останавливается внешним
сигналом.
while True:
   list1 = [4]                          # (Не используется) Остаток от прежней
логики; можно удалить.
   date = datetime.now().strftime('%B %d, %Y %X') # Метка времени
человекочитаемая (e.g., 'November 08, 2025 23:17:02').
                                        # NOTE: Для машинной обработки лучше ISO
8601 (datetime.utcnow().isoformat()).

   for host in hosts:                   # Итерация по списку устройств Juniper.
      print(f'start {host}')            # Лог: начало обработки конкретного
устройства.

      reg = host.split('-')[0]          # Регион/метка из имени хоста (до
первого '-').
                                        # Предполагается, что имя кодирует регион:
'ALM-core-01' -> 'ALM'.

      # Устанавливаем сессию NETCONF к устройству.
      dev = Device(host=host, user=username, password=password).open()

      # Инициализация рабочих словарей в рамках одного устройства:
      demux_database = {}               # demux_iface -> текущее число
PPPoE-сессий на интерфейсе.
      vlan_database = {}                # demux_iface -> 'vlan' (с учётом stacked
vlan).
      padt_database = {}                # demux_iface -> счётчик PADT sent
(разрывы).
      port_database = {}                # demux_iface -> 'host:underlying-
interface' (порт на устройстве).
      port = {}                         # (Не используется) Остаток; можно
удалить.

      # Запрос перечня абонентов/подключений с деталями (клиентский тип
'vlan').
```

```
    data = SubscribersTable(dev).get()        # Возвращает итерацию по
'subscriber' с полями view.

    # Обрабатываем каждого «абонента» (subscriber) — по сути,
демультиплексированные интерфейсы (demux).
    for line in data:
        st_vlan = ''                    # Инициализация суффикса для stacked
VLAN (QinQ).
        demux = line.demux                # Имя демультиплексированного
интерфейса (demux).
        demux_database[demux] = 0         # Начальное значение числа сессий
на этом demux.

        try:
            # Формируем отображение demux -> «host:underlying-interface»
(порт устройства).
            port_database[demux] = host + ':' + line.under
        except Exception as err:
            # Если поле отсутствует/ошибка доступа к атрибуту — фиксируем в
логе.
            print(err, demux, line.under, 2)

        # Если присутствует stacked VLAN (outer tag, например 0x88a8.1234),
записываем его в скобках.
        if line.stack:
            st_vlan = '(' + line.stack.replace('0x88a8.', '') + ')'

        # Основной VLAN: удаляем префикс 0x8100. и добавляем (при
наличии) stacked VLAN.
        vlan_database[demux] = line.vlan.replace('0x8100.', '') + st_vlan

        # Запрашиваем PPPoE статистику по конкретному demux (RPC-метод
вне FactoryLoader).
        data2 = dev.rpc.get_pppoe_statistics_information(
            underlying_interface_name=demux
        )

        try:
            # Ищем секцию 'pppoe-counters' и в ней тег 'padt-sent' — счётчик
отправленных PADT.
            for pppoe_statistics_information in data2.getchildren():
                if (pppoe_statistics_information.tag == 'pppoe-counters'):
                    for pppoe_counters in pppoe_statistics_information.getchildren():
                        if pppoe_counters.tag == 'padt-sent':
```

```
                    padt_database[demux] = pppoe_counters.text  # Строка ->
позже приведём к int.
        except Exception as err:
            print(err, demux, 3)            # Лог ошибки парсинга/структуры RPC-
ответа.

 #Второй проход: собираем количество активных PPPoE-сессий по
демультиплексам.
    data2 = SessionsTable(dev).get()        # Таблица по RPC 'get-pppoe-
session-information'.
    for line in data2:
        try:
            demux_database[line.demux] += 1    # Инкремент счётчика сессий для
соответствующего demux.
        except Exception as err:
            print(err, line.demux, 4)       # Если демультиплекс не был
инициализирован выше — логируем.
    # Сведение и запись в БД: для каждого demux, присутствующего во
vlan_database, обновляем/вставляем запись.
    for demux in vlan_database:
        flag = True                         # Флаг «нужно вставить новую запись»
(если не было обновления).
        # Полный перебор таблицы Scores (неэффективно на больших БД).
        # FIXME: замените на SELECT ... WHERE port=? AND vlan=? AND
date=? с параметрами и индексами.
        for line in cur.execute("""select * from  Scores"""):
            # line => (ID, port, vlan, score, date, subs, reg)
            if line[1] == port_database[demux] and line[2] == vlan_database[demux]
and line[4] == date:
                flag = False                # Нашли запись за текущую
дату/порт/VLAN — будем обновлять.
                try:
                    # Обновление накопительных полей: score (PADT) и subs (кол-
во PPPoE сессий).
                    # ВАЖНО: в SQL UPDATE список полей разделяется запятой,
а не 'AND'.
                    # FIXME: 'SET score = ? and subs = ?' — синтаксическая
ошибка. Должно быть 'SET score = ?, subs = ?'.
                    cur.execute(
                        """UPDATE Scores SET score = ? and subs = ? where port = ?
and vlan = ? and date = ?""",
                        (int(padt_database[demux]) + line[3],
                         line[5] + demux_database[demux],
                         line[1], line[2], line[4])
```

```
                    )
                    conn.commit()
                except Exception as err:
                    print(err, vlan_database[demux], 5)  # Лог SQL-ошибки (см. FIXME выше).
            if flag:
                # Вставка новой записи, если обновление не было выполнено (т.е. запись на дату ещё не создана).
                try:
                    cur.execute(
                        """insert into Scores(port, vlan, score, date, subs, reg) VALUES (?,?,?,?,?, ?) """,
                        (port_database[demux],
                         vlan_database[demux],
                         int(padt_database.get(demux, 0)),  # NOTE: безопаснее через .get(...,0) во избежание KeyError.
                         date,
                         demux_database.get(demux, 0),
                         reg)
                    )
                    conn.commit()
                except Exception as err:
                    print(err, vlan_database[demux], 6)      # Лог ошибок вставки (в т.ч. типизация/NULL).

        dev.close()                                 # Закрываем NETCONF-сессию с устройством.
        print(f'end {host}')                        # Лог окончания обработки устройства.

    # Экспорт/репликация основной БД в «публичную» копию DB_res.sqlite (атомарный обмен).
    if os.path.exists('DB_res.sqlite'):
        os.remove('DB_res.sqlite')                  # Удаляем предыдущую копию, если существует.

    conn2 = con('DB_res.sqlite')                    # Создаём новое соединение к результирующей БД.
    query = "".join(line for line in conn.iterdump())       # Дамп всей схемы и данных (SQL-скрипт).
    conn2.executescript(query)                      # Воспроизводим дамп в копии.
    conn2.close()                                   # Закрываем копию.

conn.close()
```

# ПРИЛОЖЕНИЕ Е

«Қазақстан Республикасы
Ұлттық Банкінің Қазақстан
банкаралық есеп айырысу орталығы»
шаруашылық жүргізу құқығы бар
республикалық мемлекеттік
кәсіпорны

БСН 960440000151,
А15С9Т5, ҚР, Алматы, Көктем-3, 21-үй;
тел.: (727) 250 67 22; факс: (727) 250 66 11
email: info@kisc.kz

Республиканское
государственное предприятие
на праве хозяйственного ведения
«Казахстанский центр межбанковских
расчетов Национального Банка
Республики Казахстан»

БИН 960440000151
А15С9Т5, РК, Алматы, Коктем-3, д. 21;
тел.: (727) 250 67 22; факс: (727) 250 66 11
email: info@kisc.kz

_____ №_____

_____

**АКТ О ВНЕДРЕНИИ**
**Результатов научной разработки**

Настоящим подтверждаем, что результаты диссертационного исследования Жунусова Аяна Радияновича, «Разработка метода мониторинга качества сервисов в телекоммуникационной сети» актуальны, предоставляют практический интерес и были применены для мониторинга внешних подключений клиентов РГП на ПХВ «Казахстанский центр межбанковских расчетов Национального Банка Республики Казахстан».

Применение системы мониторинга внешних подключений (Out connections monitoring system – OCMS), оформленной в виде сервера, имитирующего клиентские подключения ко всем предоставляемым сервисам компании, позволило сократить сроки реагирования при возникновении внештатных ситуаций.

**Внедрение разработки обеспечило:**

- значительную экономию времени реагирования;
- возможность отслеживания и выявления проблемных участков сети;
- проведение тестирований для оптимизации телекоммуникационной сети;
- отладку и улучшение качества предоставляемых сервисов.

Генеральный Директор Б.М. Жаленов

№000037АА

# ПРИЛОЖЕНИЕ Ж

**Информационное письмо**

Настоящим подтверждаем, что результаты диссертационного исследования Жунусова Аяна Радияновича, на тему «Разработка метода мониторинга качества сервисов в телекоммуникационной сети» были использованы в 2019-2022 гг. в деятельности АО «Казахтелеком» в форме системы мониторинга PPPoE-сессий, представляющей собой сервер для сбора и анализа данных пакетов PPPoE на маршрутизаторах с последующей визуализацией информации в виде графиков и текстовых оповещение по корпоративной почте. Применение указанной системы показало положительные результаты и позволило:

- оперативно выявлять и отслеживать проблемные участки сети;
- сократить время реагирования при возникновении внештатных ситуаций;
- проводить тестирования, направленные на оптимизацию телекоммуникационной сети;
- повысить качество предоставляемых сервисов.

В виду изменения архитектуры сети и дальнейшего развития сервисов, вышеуказанная система в АО «Казахтелеком» более не используется.

Данное письмо выдано по запросу Жунусова Аяна Радияновича для предоставления в диссертационный совет при защите диссертации на соискание степени PhD.

Директор по управлению ресурсами сети
Объединение "Дивизион "Сеть"-филиал АО "Казахтелеком"

А.Г. Кабылбеков

Дата: «9» сентября 2025 г.