Scientific publishing house

Professional Bulletin

Information Technology and Security

Issue №1/2025

A scientific journal for the best specialists in the industry. Inside: original works on AI, data analysis, cloud technologies and IT innovations.

INDEXATIONS

Google Scholar · International Scientific Indexing · Academic Resource Index ResearchBib · Научная электронная библиотека eLIBRARY.RU · CyberLeninka · CiteFactor Academic Scientific Journals

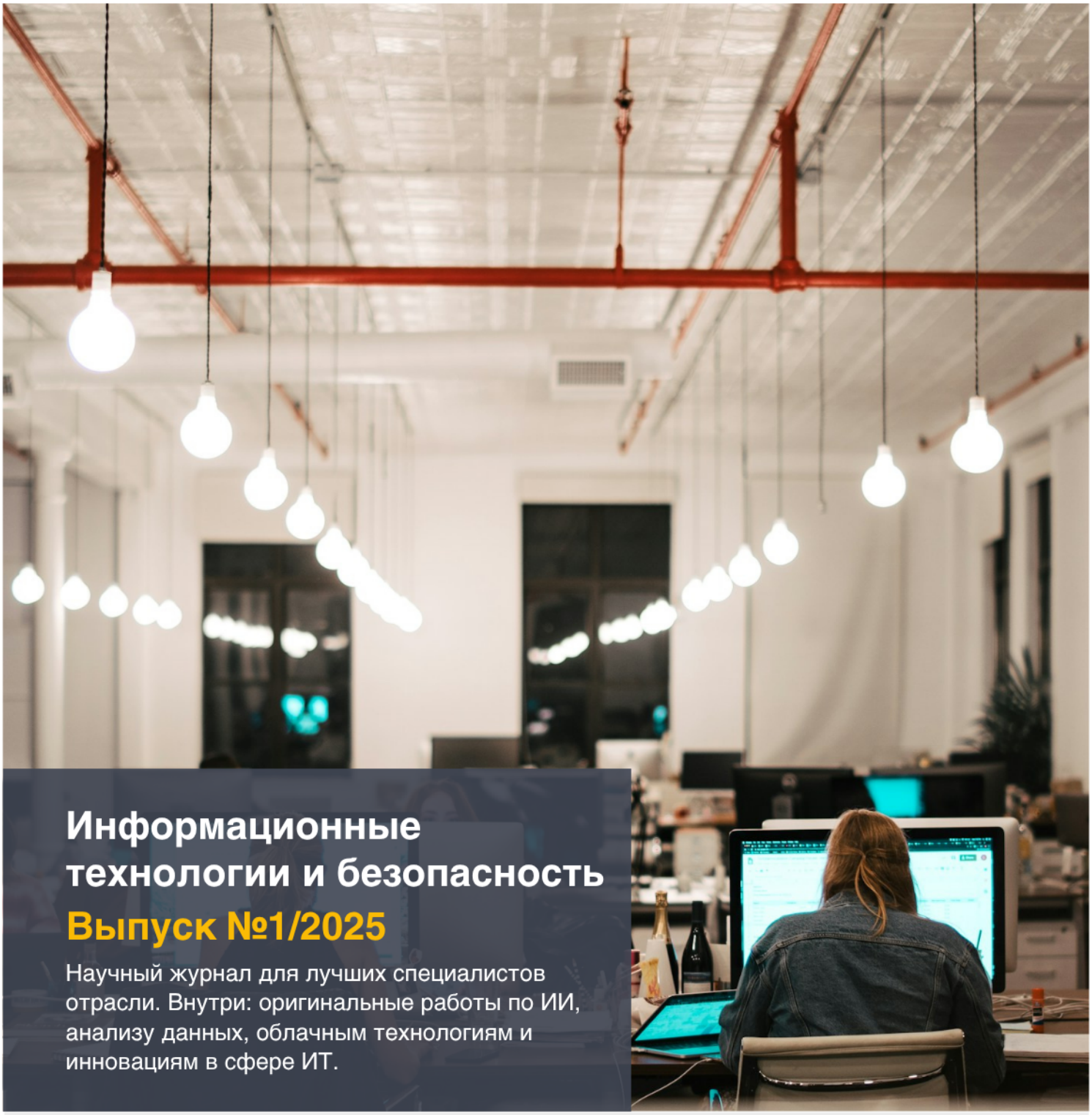
ISSN 3100-444X

support@professionalbulletinpublisher.com

professionalbulletinpublisher.com/

Brasov, Sat Sanpetru, Comuna Sanpetru, Str. Sfintii Constantin si Elena, nr. 6

Научное издательство
Профессиональный Вестник



Информационные технологии и безопасность

Выпуск №1/2025

Научный журнал для лучших специалистов отрасли. Внутри: оригинальные работы по ИИ, анализу данных, облачным технологиям и инновациям в сфере ИТ.

ИНДЕКСАЦИИ ЖУРНАЛА

Google Scholar · International Scientific Indexing · Academic Resource Index ResearchBib · Научная электронная библиотека eLIBRARY.RU · CyberLeninka · CiteFactor Academic Scientific Journals

ISSN 3100-444X

support@professionalbulletinpublisher.com
professionalbulletinpublisher.com/

Brasov, Sat Sanpetru, Comuna Sanpetru, Str. Sfintii Constantin si Elena, nr. 6

The scientific publishing house «Professional Bulletin»
Journal «Professional Bulletin. Information Technology and Security»

Professional Bulletin. Information Technology and Security is a professional scientific journal. The publication in it is recommended to practitioners and researchers who seek to find solutions to real-world problems and share their experiences with the professional community. The publication in journal is suitable for those specialists who work and actively develop advanced IT solutions, such as AI, blockchain, big data technologies and others.

The journal reviews all incoming materials. The review is double-blind, carried out by internal and external reviewers of the publishing house. Articles are indexed in a variety of international scientific databases, and access to the journal's database is open to any reader. Publication in the journal takes place 4 times a year.

Publisher's website: https://www.professionalbulletinpublisher.com/

Issue № 1/2025
Brasov County, Romania

Научное издательство «Профессиональный вестник»

Журнал «Профессиональный вестник. Информационные технологии и безопасность»

Профессиональный вестник. Информационные технологии и безопасность – профессиональное научное издание. Публикация в нем рекомендована практикам и исследователям, которые стремятся найти решения для реальных задач и поделиться своим опытом с профессиональным сообществом. Публикация в журнале подходит для тех специалистов, кто работает и активно развивает передовые ИТ-решения, такие как технологии ИИ, блокчейна, больших данных и другие.

Журнал рецензирует все входящие материалы. Рецензирование – двойное слепое, осуществляется внутренними и внешними рецензентами издательства. Статьи индексируются во множестве международных научных баз, доступ к базе данных журнала открыт для любого читателя. Публикация журнала происходит 4 раза в год.

Сайт издательства: https://www.professionalbulletinpublisher.com/

Выпуск № 1/2025

Жудец Брашов, Румыния

Contents

Loskutova V.S.
SECURITY THREAT ANALYSIS IN 6G NETWORKS USING MACHINE LEARNING MODELS 3

Dementyev N.V.
ADAPTIVE QUANTUM-RESISTANT ENCRYPTION FOR ARTIFICIAL INTELLIGENCE-BASED INFORMATION SYSTEMS 10

Bargsyan A.A.
SECURE MULTI-PARTY COMPUTATION METHODS FOR CONFIDENTIAL BIG DATA ANALYTICS 18

Norkusheva D.M.
INTERACTION MODELS OF INTELLIGENT SENSOR NETWORKS IN INDUSTRIAL INTERNET OF THINGS 25

Goryunova E.T., Krestov S.A.
COMPARATIVE EFFICIENCY OF DATA SHARDING STRATEGIES IN DISTRIBUTED LEDGER SYSTEMS 33

Grigoryan S.N., Zarutyunyan T.V.
LOAD FORECASTING SYSTEMS FOR CLOUD PLATFORMS USING HYBRID ALGORITHMS 40

Kholmatov F.A.
BLOCKCHAIN-BASED DIGITAL IDENTITY MANAGEMENT SYSTEMS FOR CROSS-BORDER INTERACTIONS 48

Gvilava N.T.
AUTONOMOUS INTELLIGENT AGENTS IN DECISION SUPPORT SYSTEMS FOR CRITICAL INFRASTRUCTURE 55

Zabelin R.T., Levshits V.E.
OPTIMIZATION OF RESOURCE ALLOCATION IN EDGE COMPUTING SYSTEMS FOR REAL TIME APPLICATIONS 62

Aitkalieva A.M., Zhumabaev O.E.
APPLICATION OF BIG DATA AND DEEP LEARNING FOR FAILURE PREDICTION IN POWER GRIDS 69

Содержание выпуска

Лоскутова В.С.
АНАЛИЗ УГРОЗ БЕЗОПАСНОСТИ В СЕТЯХ 6G С ПРИМЕНЕНИЕМ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ 3

Дементьев Н.В.
АДАПТИВНОЕ КВАНТОВО-УСТОЙЧИВОЕ ШИФРОВАНИЕ ДЛЯ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА 10

Bargsyan A.A.
SECURE MULTI-PARTY COMPUTATION METHODS FOR CONFIDENTIAL BIG DATA ANALYTICS 18

Norkusheva D.M.
INTERACTION MODELS OF INTELLIGENT SENSOR NETWORKS IN INDUSTRIAL INTERNET OF THINGS 25

Goryunova E.T., Krestov S.A.
COMPARATIVE EFFICIENCY OF DATA SHARDING STRATEGIES IN DISTRIBUTED LEDGER SYSTEMS 33

Grigoryan S.N., Zarutyunyan T.V.
LOAD FORECASTING SYSTEMS FOR CLOUD PLATFORMS USING HYBRID ALGORITHMS 40

Kholmatov F.A.
BLOCKCHAIN-BASED DIGITAL IDENTITY MANAGEMENT SYSTEMS FOR CROSS-BORDER INTERACTIONS 48

Gvilava N.T.
AUTONOMOUS INTELLIGENT AGENTS IN DECISION SUPPORT SYSTEMS FOR CRITICAL INFRASTRUCTURE 55

Забелин Р.Т., Левшиц В.Э.
ОПТИМИЗАЦИЯ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В СИСТЕМАХ ПЕРИФЕРИЙНЫХ ВЫЧИСЛЕНИЙ ДЛЯ ПРИЛОЖЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ 62

Aitkalieva A.M., Zhumabaev O.E.
APPLICATION OF BIG DATA AND DEEP LEARNING FOR FAILURE PREDICTION IN POWER GRIDS 69

УДК 004.056.5

АНАЛИЗ УГРОЗ БЕЗОПАСНОСТИ В СЕТЯХ 6G С ПРИМЕНЕНИЕМ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

Лоскутова В.С.
специалист, Кубанский государственный университет (Краснодар, Россия)

Аннотация

В статье рассматриваются актуальные угрозы безопасности, возникающие в сетях шестого поколения (6G), и анализируется потенциал применения моделей машинного обучения (МО) для их обнаружения и предотвращения. Представлена классификация типовых атак, выявлены соответствующие методы МО, обоснована необходимость дифференцированного подхода к выбору алгоритмов. Особое внимание уделено вопросам архитектуры интеграции аналитических моделей в распределённую инфраструктуру 6G, а также оценке их эффективности по ряду технических и эксплуатационных метрик. Сделан вывод о необходимости построения адаптивных, объяснимых и энергоэффективных систем безопасности с возможностью масштабирования и локального анализа.

Ключевые слова: сети 6G, безопасность, машинное обучение, обнаружение угроз, аномалия, интерпретируемость, распределённая архитектура.

SECURITY THREAT ANALYSIS IN 6G NETWORKS USING MACHINE LEARNING MODELS

Loskutova V.S.
specialist degree, Kuban state university (Krasnodar, Russia)

Abstract

The article explores emerging security threats in sixth-generation (6G) networks and evaluates the applicability of machine learning (ML) models for threat detection and prevention. It provides a typology of common attack vectors, matches them with relevant ML techniques, and emphasizes the importance of selecting context-specific algorithms. The paper further discusses architectural considerations for integrating ML into the distributed infrastructure of 6G and presents a comprehensive evaluation framework based on technical and operational metrics. The study concludes that adaptive, interpretable, and energy-efficient security systems are essential for maintaining resilience and enabling localized analysis in next-generation networks.

Keywords: 6G networks, security, machine learning, threat detection, anomaly, interpretability, distributed architecture.

Введение

С переходом от пятого поколения мобильной связи к шестому значительно расширяются функциональные возможности беспроводных сетей, включая ультранизкую задержку, экстремально высокую пропускную способность и интеграцию искусственного интеллекта в архитектуру сети. Однако наряду с этим усиливается сложность её структуры, что порождает новые векторы атак и требует принципиально иного подхода к обеспечению безопасности. Современные традиционные методы реагирования и анализа инцидентов становятся недостаточными в условиях высокой динамичности 6G-среды и усложнения угроз. Особенностью сетей 6G является тесное взаимодействие с распределёнными вычислениями, киберфизическими системами и автономными устройствами. Это создаёт предпосылки для

возникновения мультиуровневых атак, включая манипуляции с управлением доступом, внедрение вредоносных моделей искусственного интеллекта и целевые атаки на архитектурные компоненты. Учитывая необходимость обнаружения таких угроз в реальном времени, особую актуальность приобретает внедрение методов машинного обучения, способных анализировать аномалии, выявлять нетипичное поведение и адаптироваться к новым сценариям угроз. Целью данной работы является исследование эффективности применения моделей МО в задачах анализа и выявления угроз безопасности в сетях 6G. В статье рассматриваются типология атак, характеристики сетевой телеметрии, применимые алгоритмы, а также обсуждаются архитектурные и вычислительные аспекты их внедрения. Анализ проводится с учётом требований к масштабируемости, точности, интерпретируемости и скорости реагирования в условиях будущей инфраструктуры шестого поколения связи.

Основная часть

Сети шестого поколения представляют собой следующую ступень эволюции мобильной связи, в рамках которой реализуются не только экстремальные технические параметры, но и глубокая интеграция с когнитивными вычислениями, распределённым интеллектом и цифровыми двойниками. Однако с расширением архитектурных и функциональных возможностей увеличивается поверхность потенциальных атак, а характер угроз становится всё более динамичным и трудно предсказуемым. Основными целями злоумышленников в подобных средах являются перехват или подделка управляющих сообщений, атаки на целостность сетевого взаимодействия, внедрение фальсифицированных устройств и вмешательство в процессы принятия решений на уровне ИИ-модулей. В условиях масштабируемых, гетерогенных и адаптивных сетей 6G традиционные механизмы обеспечения безопасности - такие как сигнатурный анализ, статическая фильтрация или централизованный контроль доступа - оказываются неэффективными или требуют чрезмерных ресурсов. В отличие от них, методы машинного обучения способны выявлять ранее неизвестные аномалии, прогнозировать вредоносную активность и адаптироваться к новым шаблонам поведения. Наиболее перспективными направлениями применения МО являются обнаружение вторжений, классификация сетевых угроз, сегментация вредоносного трафика и выявление атак на протоколы связи в режиме реального времени [1]. Особое внимание в контексте 6G привлекают распределённые модели МО, способные функционировать на периферийных узлах сети (edge-компонентах), минимизируя задержки и уменьшая нагрузку на центральные вычислительные ресурсы. Использование таких подходов позволяет реализовать детекцию атак на месте их возникновения, не прибегая к передаче больших объёмов чувствительных данных. Кроме того, комбинация различных методов обучения - как с учителем, так и без него - обеспечивает более гибкую реакцию на сложные и быстро изменяющиеся угрозы, характерные для мультиагентных сетей нового поколения.

Одной из ключевых задач при анализе угроз в сетях 6G является обработка и интерпретация высокообъёмных потоков телеметрических данных, включающих параметры трафика, временные метки, сигнальные характеристики, сетевые события и поведенческие шаблоны абонентов. В условиях постоянного роста числа подключённых устройств и высокой скорости передачи данных возникает необходимость в использовании алгоритмов, способных не только работать с неструктурированными и разреженными наборами, но и масштабироваться без потери точности [2]. Алгоритмы МО, такие как случайные леса, градиентный бустинг, сверточные нейронные сети и модели, основанные на рекуррентных связях, демонстрируют высокую эффективность при обработке таких данных, позволяя дифференцировать типы активности и выявлять потенциальные угрозы с учётом временных зависимостей. Наряду с точностью классификации и скоростью обработки важную роль играет интерпретируемость моделей, особенно в критически важных инфраструктурах, где требуется объяснение причин принятия решений. Современные подходы в области объяснимого машинного обучения (XAI) позволяют анализировать вклад отдельных признаков в итоговую оценку угрозы, формировать отчёты для аудиторов и повышать уровень доверия со стороны операторов. В сетях 6G, где решения по безопасности могут приниматься

автономными агентами, возможность трактовать результаты классификации и детекции становится неотъемлемым требованием для сертификации и соответствия нормативным требованиям различных государств.

Особое внимание уделяется использованию методов обучения без учителя для обнаружения ранее неизвестных или нестандартных угроз, характерных для гибридных и мультидоменных атак в сетях 6G. Такие методы, как кластеризация, понижение размерности и вероятностные графовые модели, позволяют строить представление об обычном поведении системы и идентифицировать отклонения, не требуя заранее размеченных наборов данных. Это особенно актуально в условиях ограниченной доступности достоверных выборок и постоянно изменяющегося ландшафта атак [3]. Кроме того, сочетание этих подходов с методами полуобучения и активного обучения позволяет итеративно улучшать качество моделей, постепенно расширяя границы известных угроз. Важной задачей в рамках интеграции МО в инфраструктуру 6G является обеспечение защищённости самих моделей от целевых атак, таких как отравление данных (data poisoning), генерация противостоящих примеров (adversarial examples) и подделка телеметрии. Такие атаки могут приводить к ложным срабатываниям, усыплению системы или даже управляемому игнорированию критически опасной активности. Для противодействия этим рискам исследуются подходы по защите жизненного цикла моделей, включая мониторинг обучающих выборок, внедрение механизмов верификации источников данных, а также регулярную переоценку параметров модели с учётом метрик устойчивости и доверия.

Сравнительный анализ моделей случайный лес и глубокая нейронная сеть по параметрам эффективности

Для оценки применимости различных моделей машинного обучения в задачах обнаружения угроз в сетях 6G проведено сравнение двух алгоритмов: случайный лес и глубокая нейронная сеть [4]. Выбор обусловлен их распространённостью в практике анализа сетевого трафика и способностью работать с высокоразмерными, нерегулярными данными. Модели тестировались на наборе симулированных сетевых событий, отражающих аномальное поведение, включая инъекции трафика, сканирование, подмену сигнала и сбои в протоколах. Оценка проводилась по пяти основным критериям: точность обнаружения угроз, устойчивость к ложноположительным срабатываниям, время отклика (инференс), интерпретируемость результатов и вычислительная нагрузка. Каждый показатель нормирован и визуализирован на диаграмме ниже. Это позволяет комплексно оценить пригодность моделей в условиях высоконагруженной, изменчивой среды сетей шестого поколения.

Рисунок 1 демонстрирует сравнительную эффективность двух подходов по ключевым параметрам, включая точность обнаружения, устойчивость к ложным срабатываниям, интерпретируемость, скорость отклика и вычислительную нагрузку, что позволяет комплексно оценить пригодность моделей для различных сценариев применения в сетях 6G.

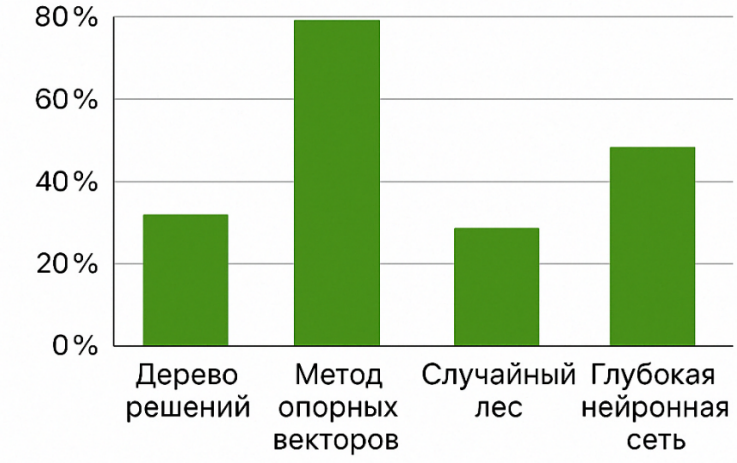


Рисунок 1. Сравнительный анализ моделей случайный лес и глубокая нейронная сеть по эффективности в задачах обнаружения угроз в сетях 6G

Модель случайный лес показала высокую интерпретируемость, устойчивость к ложноположительным срабатываниям и стабильность результатов при умеренной вычислительной нагрузке. Благодаря возможности визуального представления дерева решений и анализа важности признаков, данный алгоритм является предпочтительным для задач, где требуется объяснимость результатов и прозрачность принятия решений, в том числе в регламентированных средах или при необходимости оперативной отчётности перед аудиторами. Кроме того, невысокие требования к ресурсам позволяют внедрять модель на граничных вычислительных узлах с ограниченной производительностью [5].

Глубокая нейронная сеть, напротив, продемонстрировала более высокую точность в обнаружении сложных и неявных аномалий, а также способность адаптироваться к изменяющимся паттернам сетевого трафика. Однако её применение сопряжено с рядом ограничений: значительное время отклика при инференсе, высокая чувствительность к настройке гиперпараметров и затруднённая интерпретация внутренних представлений модели. Эти факторы могут осложнять интеграцию в системы реального времени и требуют дополнительного обеспечения устойчивости к целевым атакам на саму модель.

Таким образом, выбор алгоритма для анализа угроз в сетях 6G должен осуществляться с учётом конкретных условий эксплуатации: критичности к задержкам, доступности вычислительных ресурсов, потребности в объяснимости и типе обрабатываемых угроз. В отдельных сценариях может быть оправдано использование гибридных подходов, объединяющих преимущества обеих моделей.

Сопоставление моделей машинного обучения с типами угроз в сетях 6G

Разнообразие угроз, с которыми сталкиваются сети шестого поколения, требует системного подхода к выбору аналитических инструментов. Учитывая распределённую природу инфраструктуры 6G, высокую динамичность трафика и отсутствие жёстких границ между подсетями, классические универсальные средства реагирования теряют эффективность [6]. В такой среде именно выбор специализированной модели машинного обучения, ориентированной на конкретную категорию угроз, может существенно повысить точность обнаружения, сократить задержку реагирования и снизить уровень ложных срабатываний.

Каждый тип атаки характеризуется определённым набором признаков и векторов воздействия, что обуславливает необходимость дифференцированного подхода к обработке сетевых данных. Некоторые угрозы, такие как инъекция трафика, требуют детального анализа последовательностей пакетов, тогда как для атак на модели обнаружения - важны поведенческие шаблоны и сложные отклонения от нормального функционирования. Кроме того, целесообразность применения конкретного алгоритма зависит от факторов вычислительной нагрузки, интерпретируемости результата, устойчивости к зашумлённым данным и способности адаптироваться к новым сценариям атак [7].

Таблица 1 содержит расширенное сопоставление между типами угроз, их описанием, соответствующими методами машинного обучения, задачами, которые они решают, и преимуществами при их использовании в системах безопасности 6G.

Таблица 1

Расширенное сопоставление угроз и методов машинного обучения в сетях 6G

Тип угрозы	**Характеристика угрозы**	**Применяемый метод МО**	**Цель применения**	**Преимущества подхода**
Инъекция трафика	Изменение или вставка пакетов в поток с целью нарушения логики передачи	Случайный лес	Выявление аномалий на уровне пакетов и сессий	Высокая точность и интерпретируемость при умеренных затратах ресурсов
Сканирование порта	Активный сбор информации об открытых портах и сервисах	Метод опорных векторов	Обнаружение повторяющихся и систематических сканирований	Хорошо работает при малом объёме данных и высокой линейности

Тип угрозы	Характеристика угрозы	Применяемый метод МО	Цель применения	Преимущества подхода
Подделка управляющих сигналов	Фальсификация сигналов управления для манипулирования поведением компонентов сети	Глубокая нейронная сеть	Классификация отклонений в передаче управляющих команд	Глубокое представление сложных зависимостей и паттернов
Атака типа «отказ в обслуживании»	Перегрузка узлов или каналов связи с целью нарушения их доступности	Градиентный бустинг	Быстрое реагирование на нестабильность и высокие нагрузки	Быстрая сходимость и адаптация к изменению трафика
Внедрение вредоносного узла	Имитация легитимных устройств с целью внедрения в доверенные зоны	Кластеризация (без учителя)	Определение нетипичных устройств и поведения в сети	Не требует разметки данных, работает с новыми аномалиями
Атака на модели машинного обучения	Влияние на обучающую выборку или входные данные с целью подрыва надёжности модели	Автоэнкодер + фильтрация аномалий	Защита от атак на алгоритмы обнаружения и классификации	Выявляет скрытые манипуляции и адаптивные обходы моделей

Результаты анализа показывают, что универсального решения не существует: каждая категория угроз требует своего подхода и модели, способной учитывать специфику воздействия. Системы безопасности 6G должны быть гибкими, модульными и опираться на ансамбли алгоритмов, интегрированных в единую аналитическую платформу, способную адаптироваться к меняющейся обстановке и эволюции атакующих стратегий.

Архитектурные особенности интеграции моделей машинного обучения в инфраструктуру 6G

Интеграция методов машинного обучения в инфраструктуру сетей шестого поколения требует учёта специфических особенностей архитектуры 6G, включая высокую плотность распределённых узлов, динамическую маршрутизацию и поддержку вычислений на периферии [8]. В отличие от централизованных решений, применяемых в сетях предыдущих поколений, архитектура 6G предполагает тесную интеграцию средств анализа вблизи источника данных. Это обуславливает необходимость разработки лёгких, ресурсоэффективных моделей, способных функционировать в условиях ограниченной пропускной способности и энергоёмкости.

Одним из ключевых направлений является использование гибридных архитектур, в которых предварительная обработка и первичная фильтрация угроз осуществляются на уровне граничных устройств, а глубокий анализ - в облачной или координирующей части сети. Такой подход обеспечивает баланс между скоростью реагирования и глубиной анализа, позволяя эффективно распределять ресурсы в зависимости от текущей нагрузки [9]. Дополнительно используется подход федеративного обучения, позволяющий обучать модели на локальных данных без их передачи в центральный узел, что повышает конфиденциальность и снижает сетевые издержки.

Не менее важной задачей является обеспечение устойчивости моделей к динамике топологии сети и изменяющимся характеристикам трафика. Для этого применяются адаптивные механизмы, включающие онлайн-обновление весов моделей, использование буферов краткосрочной памяти и переключение между режимами обработки в зависимости от контекста. Такие меры позволяют моделям сохранять релевантность в условиях дрейфа данных и постоянного появления новых типов взаимодействий.

Также важно учитывать аспекты взаимодействия между отдельными интеллектуальными агентами в мультидоменных сетях 6G. В рамках распределённых систем обеспечения безопасности осуществляется координация между локальными детекторами, обмен метаинформацией о выявленных инцидентах и согласование реакций на угрозы. В этой связи актуально применение механизмов коллективного обучения и консенсуса, обеспечивающих сходимость анализа даже при наличии асинхронности и частичной потери информации.

Наконец, важным направлением является обеспечение совместимости внедряемых систем с нормативными требованиями и стандартами будущих телекоммуникационных систем. Реализация гибких политик доступа, логирования и аудита, а также интеграция с платформами доверенного исполнения и защищённой обработки становятся критически важными условиями для успешного применения МО в задачах анализа угроз в сетях 6G.

Метрики оценки эффективности моделей при обнаружении угроз в сетях 6G

Выбор и внедрение моделей машинного обучения для обеспечения сетевой безопасности невозможны без объективной оценки их эффективности. В контексте 6G, где высока нагрузка на каналы связи и критично значение времени реакции, использование стандартных метрик должно дополняться характеристиками, отражающими специфику распределённых систем и вариативность атакующих воздействий. Кроме того, необходимо учитывать не только точность классификации, но и способность модели выявлять редкие и новые типы угроз, а также устойчивость к зашумлённым или искажённым данным.

Классические метрики, такие как точность (accuracy), полнота (recall), специфичность (specificity) и F-мера, по-прежнему остаются основой для оценки качества бинарной или многоклассовой классификации. Однако в условиях сетей 6G особое значение приобретает время отклика модели, измеряемое как средняя задержка между поступлением данных и выдачей результата. Эта характеристика определяет применимость модели в системах реального времени и напрямую влияет на способность предотвращать распространение атаки.

Другим важным параметром является устойчивость модели к концептуальному дрейфу, то есть способность адаптироваться к изменениям в поведении пользователей, новых протоколах или изменяющихся схемах взаимодействия. В этом контексте измеряются показатели деградации производительности с течением времени и скорость восстановления после обновления модели. Дополнительно оценивается интерпретируемость решений, особенно в случае использования глубинных нейросетевых структур [10]. Для этих целей применяются методы SHAP, LIME и визуализация значимости признаков.

Также важно учитывать энергетическую эффективность и вычислительную нагрузку модели. Поскольку в 6G-платформах всё чаще используются периферийные узлы и устройства с ограниченными ресурсами, критично снижать энергопотребление при инференсе и уменьшать потребление памяти. Такие показатели, как FLOPS (число операций с плавающей точкой) и потребление ОЗУ, становятся частью комплексной оценки применимости модели в распределённой среде.

Комплексный подход к оценке, включающий сочетание технических, поведенческих и эксплуатационных метрик, позволяет более точно выбирать и адаптировать алгоритмы под конкретные задачи обнаружения угроз. Это особенно важно в многоуровневых системах безопасности 6G, где эффективность должна обеспечиваться не только за счёт высокой точности, но и за счёт способности модели функционировать в условиях ограничений, неопределённости и атакующих воздействий.

Заключение

Развитие сетей шестого поколения сопровождается не только технологическим прогрессом, но и возрастанием сложности киберугроз, что требует пересмотра подходов к обеспечению безопасности. В данной работе рассмотрены ключевые типы атак, характерные для среды 6G, и обоснована необходимость применения моделей машинного обучения как основы для создания адаптивных, масштабируемых и интеллектуальных систем обнаружения угроз. Проведённый анализ показал, что использование МО позволяет выявлять сложные и скрытые аномалии, адаптироваться к изменяющимся шаблонам трафика, а также обеспечивать

обнаружение атак в реальном времени. В статье представлены подходы к сопоставлению методов МО с конкретными типами угроз, даны рекомендации по архитектуре интеграции аналитических моделей, а также рассмотрены ключевые метрики эффективности, включая точность, устойчивость, интерпретируемость и вычислительную нагрузку.

Эффективное применение МО в сетях 6G требует комплексного подхода, включающего координацию вычислений между центром и периферией, защиту самих моделей от целевых атак, обеспечение совместимости с нормативными требованиями и формирование гибкой системы оценки. Дальнейшие исследования могут быть сосредоточены на разработке защищённых архитектур коллективного обучения, оптимизации энергетических затрат и создании стандартов тестирования алгоритмов в условиях сетей нового поколения.

Список литературы

1. Saeed M.M., Saeed R.A., Abdelhaq M., Alsaqour R., Hasan M.K., Mokhtar R.A. Anomaly detection in 6G networks using machine learning methods // Electronics. 2023. Vol. 12. No. 15. P. 3300.
2. Rahman M.A., Hossain M.S. A deep learning assisted software defined security architecture for 6G wireless networks: IIoT perspective // IEEE Wireless Communications. 2022. Vol. 29. No. 2. P. 52-59.
3. Gkonis P.K., Nomikos N., Trakadas P., Sarakis L., Xylouris G., Masip-Bruin X., Martrat J. Leveraging network data analytics function and machine learning for data collection, resource optimization, security and privacy in 6G networks // IEEE access. 2024. Vol. 12. P. 21320-21336.
4. Suomalainen J., Ahmad I., Shajan A., Savunen T. Cybersecurity for tactical 6G networks: Threats, architecture, and intelligence // Future Generation Computer Systems. 2025. Vol. 162. P. 107500.
5. Sakthi U., Alasmari A., Girija S.P., Senthil P., Qamar S., Hariharasitaraman S. Smart Healthcare Based Cyber Physical System Modeling by Block Chain with Cloud 6G Network and Machine Learning Techniques // Wireless Personal Communications. 2024. P. 1-25.
6. Tan Q. Deep Learning-Driven Network Security Situation Awareness Method in 6G Environment // Internet Technology Letters. 2025. Vol. 8. No. 2. P. e70006.
7. Siriwardhana Y., Porambage P., Liyanage M., Ylianttila M. AI and 6G security: Opportunities and challenges //2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit). IEEE. 2021. P. 616-621.
8. Kaur N., Gupta L. An approach to enhance iot security in 6G networks through explainable ai // arXiv preprint arXiv:2410.05310. 2024.
9. Khalid M., Ali J., Mohsin A.R., Roh B.H., Alenazi M.J. Deep learning techniques for enhanced security and privacy in 6G terrestrial–nonterrestrial network architecture // The Journal of Supercomputing. 2025. Vol. 81. No. 4. P. 631.
10. Sirohi D., Kumar N., Rana P.S., Tanwar S., Iqbal R., Hiji M. Federated learning for 6G-enabled secure communication systems: a comprehensive survey // Artificial Intelligence Review. 2023. Vol. 56. No. 10. P. 11297-11389.

УДК 004.056.55

АДАПТИВНОЕ КВАНТОВО-УСТОЙЧИВОЕ ШИФРОВАНИЕ ДЛЯ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Дементьев Н.В.
специалист, Воронежский государственный университет (Воронеж, Россия)

Аннотация

В статье рассматривается концепция адаптивного квантово-устойчивого шифрования (АКУШ), ориентированного на защиту данных в интеллектуальных информационных системах, подверженных угрозам квантовых вычислений. Описаны основные принципы реализации АКУШ, его роль в различных сферах применения, таких как автономные транспортные системы, медицинские платформы и финансовые сети. Рассматривается взаимодействие ключевых компонентов системы, таких как модули оценки контекста, предсказания угроз, выбора шифрования, криптоконтейнера и модуля обратной связи. Приводится анализ эффективности адаптивного шифрования в реальных сценариях и обсуждаются перспективы развития технологий, включая интеграцию с методами машинного обучения и гибридными вычислительными архитектурами. В статье также подчеркивается значимость адаптивного подхода для обеспечения безопасности данных в условиях постоянно меняющихся угроз.

Ключевые слова: адаптивное квантово-устойчивое шифрование, постквантовая криптография, интеллектуальные информационные системы, машинное обучение, квантовые вычисления, защита данных, распределённые системы, финансовые сети, медицинские платформы, автономные транспортные системы.

ADAPTIVE QUANTUM-RESISTANT ENCRYPTION FOR ARTIFICIAL INTELLIGENCE-BASED INFORMATION SYSTEMS

Dementyev N.V.
specialist degree, Voronezh state university (Voronezh, Russia)

Abstract

This article discusses the concept of adaptive quantum-resistant encryption (AQRE), designed to protect data in intelligent information systems vulnerable to quantum computing threats. The main principles of implementing AQRE are described, along with its role in various application fields such as autonomous transportation systems, medical platforms, and financial networks. The interaction of key system components, such as the context evaluation, threat prediction, encryption selection modules, cryptographic container, and feedback module, is also explored. The article analyzes the effectiveness of adaptive encryption in real-world scenarios and discusses the future development of technologies, including integration with machine learning methods and hybrid computational architectures. The importance of an adaptive approach for ensuring data security in the face of constantly evolving threats is emphasized.

Keywords: adaptive quantum-resistant encryption, post-quantum cryptography, intelligent information systems, machine learning, quantum computing, data security, distributed systems, financial networks, medical platforms, autonomous transportation systems.

Введение

В условиях стремительного развития квантовых вычислений традиционные криптографические методы подвергаются риску утраты устойчивости. Квантовые алгоритмы, такие как алгоритм Шора, способны за полиномиальное время взламывать широко используемые схемы асимметричного шифрования, основанные на факторизации или логарифмических задачах. Это ставит под угрозу безопасность не только государственных и корпоративных информационных систем, но и архитектур искусственного интеллекта (ИИ), оперирующих чувствительными данными. Современные информационные системы, функционирующие на базе ИИ, требуют высокоадаптивных механизмов защиты, способных учитывать динамику угроз, а также особенности самообучающихся алгоритмов. Применение статичных схем шифрования в подобных средах ведёт к значительным уязвимостям при переходе на новые вычислительные парадигмы. В связи с этим необходима разработка алгоритмов, обладающих квантовой устойчивостью, способных к адаптивному функционированию в средах с переменной архитектурой и загрузкой. Цель настоящего исследования - разработка концепции адаптивного квантово-устойчивого шифрования, предназначенного для применения в интеллектуальных информационных системах. Предполагается интеграция механизмов постквантовой криптографии с элементами машинного обучения для динамической адаптации параметров шифрования в реальном времени.

Основная часть

Развитие квантовых вычислений коренным образом меняет парадигму обеспечения информационной безопасности. Наиболее уязвимыми оказываются классические криптографические схемы, основанные на факторизации и дискретных логарифмах, чья стойкость теряет актуальность в условиях появления квантовых алгоритмов, таких как Шора и Гровера. Угроза становится особенно значимой для интеллектуальных информационных систем, использующих самонастраивающиеся модели и обрабатывающих критические данные в режиме реального времени. Такие системы нуждаются в шифровании, способном не только обеспечивать высокий уровень стойкости, но и адаптироваться к изменяющимся условиям эксплуатации - включая тип данных, контекст использования, нагрузку на систему и характер угроз. Постквантовая криптография (PQC), развивающаяся в рамках инициативы NIST по стандартизации устойчивых алгоритмов, предоставляет математически обоснованные решения, стойкие к квантовому взлому. Однако большинство схем PQC остаются статичными по своей природе: они предполагают фиксированные параметры, не учитывающие специфику ИИ-сред. Напротив, адаптивное квантово-устойчивое шифрование предполагает внедрение механизмов самонастройки криптографических протоколов в зависимости от контекста. Это может включать динамическое переключение между алгоритмами, масштабирование параметров в ответ на изменение угроз, а также интеграцию методов машинного обучения для прогнозирования атакующих воздействий и оптимизации конфигурации защиты [1].

Ниже представлен фрагмент кода на языке python, реализующий концепцию адаптивного выбора алгоритма PQC в зависимости от типа данных и уровня угрозы, определяемого предварительно обученной моделью. Архитектура построена с расчётом на встраивание в распределённую ИИ-среду с динамическим изменением условий:

```
class AdaptiveEncryptor:
    def __init__(self, threat_model):
        self.threat_model = threat_model  # ML-модель оценки угроз

    def select_scheme(self, data_type, context_features):
        risk_score = self.threat_model.predict(context_features)

        if data_type == "stream" and risk_score < 0.4:
            return self.use_NTRU()
        elif data_type == "archive" or risk_score >= 0.7:
```

```
            return self.use_Kyber()
        else:
            return self.use_SABER()

    def use_NTRU(self):
        print("Используется схема NTRU-HRSS")
        # Загрузка и вызов реализаций шифрования
        return "NTRU"

    def use_Kyber(self):
        print("Используется схема Kyber-1024")
        return "Kyber"

    def use_SABER(self):
        print("Используется схема SABER")
        return "SABER"

# Пример использования
# Предположим, модель возвращает уровень угрозы на основе контекста (0 - низкий, 1 - критический)
mock_threat_model = lambda features: 0.65  # Подставная модель для демонстрации

encryptor = AdaptiveEncryptor(threat_model=mock_threat_model)
chosen_scheme = encryptor.select_scheme(data_type="archive", context_features={"region": "cloud", "load": "high"})
```

Представленный подход позволяет обеспечить не только гибкость криптографической защиты, но и её соответствие текущему уровню угрозы и типу данных, обрабатываемых ИИ-системой. Ключевым преимуществом такой архитектуры является снижение вероятности избыточных вычислительных затрат в ситуациях с пониженной угрозой при сохранении полной стойкости в критических условиях.

Эффективное внедрение адаптивного квантово-устойчивого шифрования в интеллектуальные информационные системы требует четко структурированной архитектуры, обеспечивающей постоянный обмен данными между модулями анализа, шифрования и принятия решений. Архитектура должна учитывать специфику распределённых сред, в которых ИИ-модули и криптографические компоненты могут находиться на разных узлах вычислительной сети, в том числе в облаке или на периферийных устройствах. Ключевыми элементами данной архитектуры выступают: Модуль оценки контекста - осуществляет сбор телеметрии, включая характеристики трафика, нагрузку на систему, тип обрабатываемых данных, и результаты анализа угроз. Модуль предсказания угроз - на основе машинного обучения формирует прогноз уровня угрозы, включая вероятность целенаправленных атак, перебора или попыток внедрения зловредных пакетов. Модуль выбора шифрования - адаптивно выбирает и активирует одну из заранее определённых схем PQC в зависимости от входных параметров, минимизируя затраты на шифрование и одновременно соблюдая необходимый уровень безопасности. Криптоконтейнер - выполняет шифрование и дешифрование в соответствии с выбранной схемой, обеспечивает журналирование операций и защиту ключей. Модуль обратной связи - возвращает в цикл обучения информацию о результатах работы выбранного алгоритма, что позволяет корректировать модели угроз и адаптировать поведение системы в долгосрочной перспективе.

Эта архитектура обеспечивает не только гибкость и адаптивность, но и способность к самообучению и оптимизации на основе накопленных данных. Важным аспектом является способность системы к динамическому реагированию на изменения в угрозах и условиях эксплуатации. Каждый модуль взаимодействует с другими через четко определённые

интерфейсы, что позволяет поддерживать устойчивость системы даже в условиях сбоя или возникновения новых угроз.

Система адаптивного квантово-устойчивого шифрования должна быть интегрирована с существующими механизмами защиты данных, такими как средства обнаружения вторжений (IDS), а также с протоколами защиты в сетях передачи данных, что обеспечивает комплексную защиту на всех уровнях архитектуры [2]. Например, модуль выбора шифрования может использовать результаты работы системы IDS для уточнения выбора шифра в зависимости от текущего уровня угроз. Кроме того, архитектура должна учитывать требования к производительности. Модуль обратной связи играет важную роль в оптимизации работы системы. На основе информации о производительности алгоритмов шифрования и их применимости в разных сценариях, система может адаптировать параметры для улучшения скорости обработки без ущерба для безопасности. Это особенно важно в реальных сценариях использования, таких как беспилотные транспортные средства или автоматизированные медицинские системы, где задержка в обработке данных может привести к критическим последствиям.

Технологии машинного обучения, интегрируемые в систему, могут также использоваться для предсказания появления новых типов атак, что даёт возможность заранее подготовиться к их возможному возникновению. Например, использование алгоритмов, способных идентифицировать аномальные паттерны в трафике и действиях пользователей, помогает минимизировать риски, связанные с неизвестными или ранее не предусмотренными уязвимостями. Таким образом, успешная реализация адаптивного квантово-устойчивого шифрования в интеллектуальных системах требует комплексного подхода к проектированию, который включает не только выбор устойчивых к квантовому взлому алгоритмов, но и обеспечение их гибкости и способности адаптироваться к изменениям угроз и условий эксплуатации.

Оценка эффективности применения адаптивного квантово-устойчивого шифрования в интеллектуальных системах: анализ производительности и устойчивости к угрозам

Эффективность адаптивного квантово-устойчивого шифрования в интеллектуальных информационных системах определяется рядом факторов, среди которых основными являются криптографическая стойкость, производительность системы и её способность адаптироваться к изменениям угроз. Эти параметры играют ключевую роль при выборе конкретных алгоритмов шифрования для различных типов данных и уровней угроз. Важно отметить, что в интеллектуальных системах, использующих ИИ и машинное обучение, требуется быстрое и динамичное шифрование, которое не нарушает процессов принятия решений в реальном времени [3].

Для объективной оценки различных алгоритмов PQC и их применения в реальных условиях, важно провести сравнение производительности, криптостойкости и адаптивности. Таблица 1 содержит сравнительный анализ нескольких популярных постквантовых алгоритмов с учётом их производительности в реальных системах с переменной нагрузкой и угрозами. Рассматриваются такие показатели, как время шифрования, потребление памяти, устойчивость к квантовым атакам и возможность адаптации в условиях изменяющихся данных и угроз.

Таблица 1

Сравнительный анализ постквантовых алгоритмов шифрования по ключевым параметрам

Алгоритм	**Время шифрования (мс)**	**Память (МБ)**	**Квантовая устойчивость**	**Адаптивность**	**Пригодность для реального времени**
Kyber-1024	12.4	3.5	Высокая	Средняя	Частичная
NTRU-HRSS-701	10.1	2.8	Средняя	Высокая	Высокая

SABER	15.9	4.2	Высокая	Средняя	Средняя
BIKE	22.3	5.0	Средняя	Высокая	Частичная

Сравнительный анализ в таблице 1 показывает, что алгоритм NTRU-HRSS-701 обладает наилучшим балансом между производительностью, адаптивностью и устойчивостью к угрозам. Этот алгоритм является оптимальным выбором для динамически изменяющихся ИИ-сред, где важно минимизировать время шифрования и эффективно использовать вычислительные ресурсы. Kyber-1024 и SABER, несмотря на более высокую криптостойкость, показывают более высокое потребление памяти и времени, что ограничивает их использование в системах реального времени с высокой нагрузкой. BIKE имеет высокую устойчивость к угрозам, однако из-за высокой вычислительной нагрузки его применение ограничено в реальных приложениях.

Реализация адаптивного квантово-устойчивого шифрования в распределённых ИИ-системах: вызовы и решения

Интеллектуальные системы, использующие распределённую архитектуру, требуют продвинутых решений для защиты данных, поскольку их компоненты могут находиться на различных узлах вычислительной сети [4]. В таких системах важны не только эффективное шифрование, но и способность адаптироваться к изменяющимся условиям с минимальными задержками. Адаптивное квантово-устойчивое шифрование идеально подходит для таких систем, поскольку оно позволяет динамически менять параметры шифрования в зависимости от типа данных и уровня угроз.

Ключевыми элементами этой системы являются следующие модули:

- **Модуль оценки контекста** - собирает данные о трафике и нагрузке, а также анализирует угрозы.
- **Модуль предсказания угроз** - на основе машинного обучения прогнозирует уровень угроз и идентифицирует потенциальные атаки.
- **Модуль выбора шифрования** - адаптивно выбирает подходящую схему PQC, оптимизируя баланс между безопасностью и производительностью.
- **Криптоконтейнер** - выполняет шифрование и дешифрование в соответствии с выбранной схемой, управляет ключами.
- **Модуль обратной связи** - отправляет данные о результатах работы системы в цикл обучения для корректировки алгоритмов.

Внедрение АКУШ в распределённые ИИ-системы сталкивается с несколькими вызовами, среди которых необходимость обеспечения независимости узлов при сохранении целостности данных и минимизации времени отклика [5]. Модули шифрования и дешифрования должны быть синхронизированы для обеспечения непрерывной работы всей системы (рис. 1). Использование интерфейсов программирования приложений (API) помогает гибко интегрировать и адаптировать алгоритмы шифрования на разных уровнях системы, обеспечивая их быструю настройку в зависимости от текущих угроз.

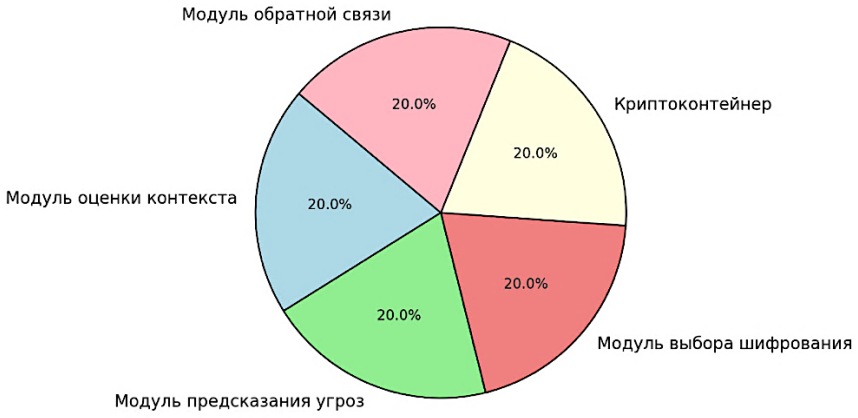


Рисунок 1. Архитектура реализации адаптивного квантово-устойчивого шифрования в распределённой ИИ-системе

Реализация адаптивного квантово-устойчивого шифрования в распределённых ИИ-системах обеспечивает высокую степень безопасности данных при минимизации задержек и вычислительных затрат. Интеграция таких компонентов, как модуль оценки контекста, модуль предсказания угроз, модуль выбора шифрования, криптоконтейнер и модуль обратной связи, позволяет гибко и динамично адаптировать криптографическую защиту в зависимости от текущих угроз и условий работы системы [6]. Это обеспечивает как высокую устойчивость к квантовым атакам, так и эффективное функционирование в реальном времени. Использование интерфейсов программирования приложений позволяет эффективно интегрировать шифрование на разных уровнях распределённой системы, обеспечивая её безопасность и производительность без потери гибкости.

Применение адаптивного квантово-устойчивого шифрования в реальных сценариях: защита данных в автономных системах, медицинских и финансовых приложениях

В условиях роста применения интеллектуальных систем, таких как автономные транспортные средства, медицинские платформы и финансовые сети, безопасность данных становится важнейшим приоритетом. Адаптивное квантово-устойчивое шифрование предоставляет решения для защиты данных, которые должны быть не только защищены от квантовых атак, но и обработаны в реальном времени с минимальной задержкой.

В автономных транспортных системах, где данные обрабатываются с высокой скоростью и в условиях динамически изменяющейся среды, важно быстро реагировать на потенциальные угрозы. Использование АКУШ позволяет динамически изменять параметры шифрования, обеспечивая защиту данных без замедления работы системы [7].

Медицинские платформы, которые обрабатывают персональные данные пациентов, требуют высокой степени защиты от возможных атак, а также соблюдения стандартов конфиденциальности. Здесь также необходимо учитывать низкие задержки в обработке данных, что делает АКУШ идеальным решением для таких приложений.

Финансовые сети сталкиваются с огромным количеством транзакций, которые требуют высокой скорости обработки и одновременно должны быть защищены от различных угроз. Адаптивное шифрование позволяет минимизировать риски, связанные с кражей данных или манипуляциями с транзакциями, при этом не нарушая быстродействие системы.

На рисунке 2 представлена диаграмма, иллюстрирующая, как адаптивное квантово-устойчивое шифрование может быть интегрировано в эти реальные сценарии для защиты данных.

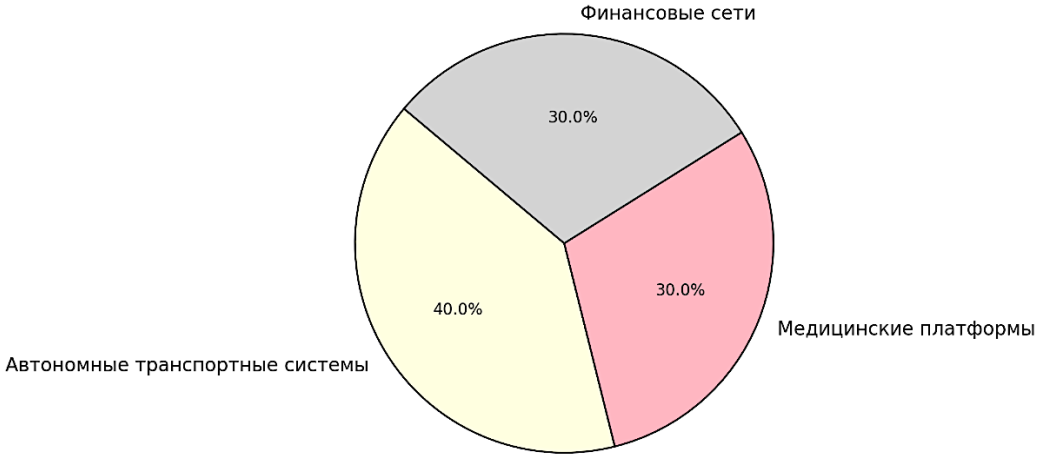


Рисунок 2. Применение адаптивного квантово-устойчивого шифрования в реальных сценариях

Диаграмма показывает распределение применения адаптивного квантово-устойчивого шифрования в ключевых сферах, таких как автономные транспортные системы, медицинские платформы и финансовые сети. Наибольшую долю занимает защита данных в автономных транспортных системах, что подчёркивает важность быстродействующего шифрования в динамически изменяющихся условиях. За этим следуют медицинские платформы и

финансовые сети, для которых также требуется высокая степень защиты данных [8]. Применение АКУШ в этих сферах помогает эффективно балансировать между безопасностью данных и производительностью системы, минимизируя риски при обработке и передаче критичных данных в реальном времени.

Перспективы развития адаптивного квантово-устойчивого шифрования в интеллектуальных системах: будущее и инновации

С развитием квантовых вычислений и усилением угроз, связанных с возможностью взлома традиционных криптографических алгоритмов, адаптивное квантово-устойчивое шифрование будет играть всё более важную роль в обеспечении безопасности интеллектуальных информационных систем. В ближайшие годы ожидается дальнейшее развитие как теоретических основ, так и практических методов реализации постквантовых криптографических решений, которые смогут поддерживать защиту данных в условиях квантовых вычислений.

Одной из перспективных тенденций является развитие квантово-устойчивых алгоритмов с учётом машинного обучения. Эти алгоритмы будут не только защищать от квантовых атак, но и адаптироваться к новым типам угроз, прогнозируемым на основе анализа больших данных. Внедрение машинного обучения в процесс выбора и настройки алгоритмов шифрования откроет новые возможности для повышения эффективности защиты данных в реальном времени.

Другим важным направлением является интеграция адаптивного шифрования в гибридные системы, использующие как традиционные вычисления, так и квантовые технологии. Разработка гибридных архитектур, где элементы ИТ-систем будут использовать как классические, так и квантово-устойчивые алгоритмы шифрования в зависимости от ситуации, обеспечит максимальную безопасность данных без значительных потерь в производительности [9].

Особое внимание следует уделить интероперабельности между различными криптографическими модулями и стандартами. С ростом числа различных приложений и систем, использование унифицированных интерфейсов для адаптивных криптографических решений позволит улучшить взаимодействие между различными компонентами инфраструктуры и обеспечить более гибкую настройку в условиях быстро меняющихся угроз.

Не менее важным направлением является повышение устойчивости алгоритмов к новым типам атак, включая атаки на сам процесс шифрования, а также атаки на инфраструктуру хранения и распространения ключей. Совершенствование методов защиты ключей, в том числе использование многоуровневых систем защиты, станет важной частью будущих технологий в области адаптивного шифрования.

Таким образом, будущее адаптивного квантово-устойчивого шифрования состоит в его интеграции в интеллектуальные системы с учётом новых технологий, таких как квантовые вычисления и машинное обучение, и в разработке решений, которые обеспечат высокий уровень безопасности данных в условиях постоянно изменяющихся угроз и технологических изменений.

Заключение

Адаптивное квантово-устойчивое шифрование представляет собой важный шаг в обеспечении безопасности данных в условиях стремительного развития квантовых вычислений. Его способность динамически адаптироваться к изменяющимся условиям и угрозам, а также эффективно взаимодействовать с интеллектуальными системами, делает его ключевым элементом защиты данных в современных распределённых системах. Применение АКУШ в таких критичных сферах, как автономные транспортные системы, медицинские платформы и финансовые сети, подчеркивает его важность для защиты данных в реальном времени при минимальных задержках.

Постоянное развитие постквантовой криптографии и её интеграция с методами машинного обучения и гибридными вычислительными архитектурами откроют новые возможности для повышения безопасности данных. Важно отметить, что адаптивное

шифрование не только защищает от квантовых атак, но и позволяет оперативно реагировать на изменения угроз, что особенно актуально для динамично развивающихся ИТ-систем.

В будущем ожидается дальнейшее совершенствование алгоритмов и технологий, направленных на обеспечение устойчивости к новым типам атак и улучшение процесса управления ключами. Интеграция АКУШ в интеллектуальные системы станет неотъемлемой частью обеспечения комплексной безопасности данных, предоставляя новые возможности для защиты критической информации в условиях быстро меняющегося технологического ландшафта.

Из этого следует, что адаптивное квантово-устойчивое шифрование станет основой для создания более защищённых, устойчивых и эффективных информационных систем, способных справляться с вызовами будущего.

Список литературы

1. Singh S., Kumar D. Enhancing cyber security using quantum computing and artificial intelligence: A review // algorithms. 2024. Vol. 4. No. 3.
2. Arumugam S.K., Kumari S., Tiwari S., Tyagi A.K. Quantum Computing for Next-Generation Artificial Intelligence-Based Blockchain // Quantum Computing. Auerbach Publications. 2025. P. 251-267.
3. Taherdoost H., Le T.V., Slimani K. Cryptographic Techniques in Artificial Intelligence Security: A Bibliometric Review // Cryptography. 2025. Vol. 9. No. 1. P. 17.
4. Thirupathi L., Akshaya B., Reddy P.C., Harsha S.S., Reddy E.S. Integration of AI and Quantum Computing in Cyber Security // Integration of AI, Quantum Computing, and Semiconductor Technology. IGI Global. 2025. P. 29-56.
5. Qadri S., Malik J.A., Shah H., Raza M.A., Alsanoosy T., Saleem M. Innovating with Quantum Computing Approaches in Block-Chain for Enhanced Security and Data Privacy in Agricultural IoT Systems // Computational Intelligence in Internet of Agricultural Things. Cham: Springer Nature Switzerland. 2024. P. 339-370.
6. Khatoon A., Riaz R. Quantum Computing Impacts on Smart City Cybersecurity Through Resilient Defense Framework: Quantum Computing Impacts on Resilient Cybersecurity Frameworks for Smart Cities // Ubiquitous Technology Journal. 2025. Vol. 1. No. 1. P. 23-31.
7. Nair M.M., Deshmukh A., Tyagi A.K. Artificial intelligence for cyber security: Current trends and future challenges // Automated Secure Computing for Next-Generation Systems. 2024. P. 83-114.
8. Hasan K.F., Simpson L., Baee M.A.R., Islam C., Rahman Z., Armstrong W., McKague M.A framework for migrating to post-quantum cryptography: Security dependency analysis and case studies // IEEE Access. 2024. Vol. 12. P. 23427-23450.
9. Olutimehin A.T. Advancing cloud security in digital finance: AI-driven threat detection, cryptographic solutions, and privacy challenges // Cryptographic Solutions, and Privacy Challenges. 2025.

UDC 004.056

SECURE MULTI-PARTY COMPUTATION METHODS FOR CONFIDENTIAL BIG DATA ANALYTICS

Bargsyan A.A.
master's degree, State engineering university of Armenia
(Yerevan, Armenia)

МЕТОДЫ ЗАЩИЩЁННЫХ МНОГОПОЛЬЗОВАТЕЛЬСКИХ ВЫЧИСЛЕНИЙ ДЛЯ КОНФИДЕНЦИАЛЬНОЙ АНАЛИТИКИ БОЛЬШИХ ДАННЫХ

Баргсян А.А.
магистр, Государственный инженерный университет Армении
(Ереван, Армения)

Abstract

This article explores secure multi-party computation (SMPC) as a foundational cryptographic approach for performing collaborative analytics on sensitive big data without compromising privacy. The study analyzes the architectural components, protocol mechanisms, and practical considerations for integrating SMPC into large-scale analytical systems. Key focus areas include data representation, secure aggregation, performance optimization, and interoperability with machine learning workflows. Through illustrative examples and technical evaluation, the paper highlights current limitations and emerging solutions for scalable, privacy-preserving computation. The findings offer insights into designing secure analytics pipelines suitable for real-world deployment across regulated and distributed environments.

Keywords: secure multi-party computation, confidential data, privacy-preserving analytics, Big Data, distributed protocols, secure aggregation, machine learning, data protection.

Аннотация

В статье рассматриваются методы безопасных многопартийных вычислений (SMPC) как криптографическая основа для конфиденциальной обработки больших данных в условиях распределённой аналитики. Исследуются архитектурные принципы, механизмы протоколов и практические аспекты внедрения SMPC в масштабируемые аналитические платформы. Особое внимание уделяется представлению данных, безопасной агрегации, стратегиям оптимизации производительности и интеграции с рабочими процессами машинного обучения. Приведённые примеры и технический анализ демонстрируют существующие ограничения и потенциальные решения в области защищённой вычислительной аналитики. Представленные результаты способствуют формированию надёжной и масштабируемой среды для анализа чувствительных данных в различных отраслях.

Ключевые слова: безопасные многопартийные вычисления, конфиденциальные данные, приватная аналитика, большие данные, распределённые протоколы, защищённая агрегация, машинное обучение, защита данных.

Introduction

The growing reliance on large-scale data-driven systems has intensified concerns regarding data confidentiality, particularly in contexts where multiple stakeholders must jointly analyze sensitive datasets. Traditional approaches to secure analytics often require data centralization, which introduces

significant privacy risks and regulatory challenges. As industries increasingly adopt distributed data processing across organizational or jurisdictional boundaries, ensuring privacy without compromising analytical capabilities becomes a critical objective. This tension is especially evident in sectors such as healthcare, finance, and public governance, where regulatory frameworks prohibit raw data sharing while demanding collaborative insights.

Secure multi-party computation offers a cryptographic paradigm that enables joint computation over private inputs without revealing individual data to participating entities. By allowing mutually distrusting parties to collaborate on data processing while preserving input confidentiality, SMPC provides a foundation for privacy-preserving analytics across decentralized infrastructures. Recent advances in cryptographic protocols, including secret sharing, homomorphic encryption, and oblivious transfer, have significantly improved the efficiency and scalability of such systems, making them more viable for integration with Big Data technologies.

This paper aims to examine the methodological foundations and practical implementation aspects of SMPC in the context of confidential analytics on large-scale data. The study explores protocol designs, system architectures, and deployment considerations necessary for real-world applications. It also analyzes performance trade-offs, compatibility with existing data infrastructures, and potential for integration with machine learning pipelines. The research contributes to the development of secure computational environments where privacy, accuracy, and scalability can coexist without the need for centralized trust.

Main part

Core protocol structure of secure multi-party computation

Secure multi-party computation protocols are designed to allow multiple participants to jointly compute a function over their private inputs while ensuring that no party gains access to the inputs of others. At the heart of these protocols lies the definition of a shared computational goal expressed as a function, typically decomposed into basic arithmetic or logical operations. Each party encrypts or encodes its data in a way that permits manipulation without exposing the raw values, enabling cooperative execution of the overall computation [1]. Such protocols are underpinned by foundational cryptographic mechanisms such as additive secret sharing, where data is split into fragments distributed among participants.

A typical computation proceeds in synchronized rounds, with each round consisting of local computation, message exchange, and reconstruction. The communication topology and trust assumptions determine whether protocols follow a semi-honest or malicious threat model. In semi-honest scenarios, parties follow the protocol but may try to infer hidden information; in malicious models, active deviation is anticipated and must be mitigated with verifiable computation steps. Protocols must also be resilient to latency and data loss in distributed networks, which is particularly important in big data settings where computation spans heterogeneous and geographically dispersed systems [2].

The pseudocode below presents a simplified implementation of an additive secret sharing protocol used to compute the sum of inputs from multiple parties without revealing their individual values.

```
# Simplified additive secret sharing for sum computation

import random

def share_secret(secret, n):
    """Split secret into n shares."""
    shares = [random.randint(0, 1000) for _ in range(n - 1)]
    final_share = secret - sum(shares)
    shares.append(final_share)
    return shares

def reconstruct(shares_list):
```

```
    """Reconstruct original value from all shares."""
    return sum(shares_list)

# Example: three-party computation
secret_inputs = [30, 50, 40]  # confidential inputs from three parties
n = len(secret_inputs)

# Each party shares their input
all_shares = [share_secret(secret, n) for secret in secret_inputs]

# Each party sums received shares for local partial result
partial_sums = [sum(party_shares) for party_shares in zip(*all_shares)]

# Reconstruct final result
result = reconstruct(partial_sums)
print("Final computed sum without revealing inputs:", result)
```

This example demonstrates the essential mechanics of additive secret sharing, where private inputs are decomposed into distributed shares and securely aggregated without disclosure. The protocol is lightweight and suitable for summation tasks or input averaging in collaborative environments [3]. While simplistic, it forms the foundation for more complex secure computations involving matrix operations, machine learning inference, or statistical analysis in privacy-sensitive domains.

Data representation and encoding techniques in SMPC workflows

The effectiveness of SMPC protocols in large-scale analytical systems heavily depends on how data is represented and encoded prior to computation. Unlike conventional processing, where raw values are directly accessible, SMPC systems require that inputs be transformed into protected formats that preserve both structure and operational compatibility. Data encoding must support modular arithmetic and be resilient to truncation, overflow, and rounding errors, especially in floating-point domains. This becomes particularly relevant in privacy-preserving statistical computations, where accuracy must be retained across decentralized operations [4].

Integer-based encoding schemes, such as fixed-point representation, are widely adopted due to their compatibility with arithmetic sharing mechanisms. These formats enable efficient addition and multiplication over finite fields or rings, allowing protocols to operate on encrypted or secret-shared data without the need for costly cryptographic conversions. Moreover, batch encoding strategies have emerged to improve throughput in high-dimensional datasets, enabling parallel computation over matrix-shaped data. Careful selection of modulus size and base precision is crucial to maintain both correctness and performance.

In addition to numeric representations, categorical and structured data pose specific challenges [5]. Common preprocessing steps-such as one-hot encoding or binary transformation-must be adapted to privacy-preserving settings, where neither input labels nor encoded vectors can be exposed. These operations must be embedded into the protocol logic without leaking structural information through intermediate states or memory access patterns. As a result, data representation becomes a design constraint as much as an implementation detail, influencing the feasibility and scalability of SMPC in real-world analytics pipelines.

Secure aggregation mechanism for federated analytics

Secure aggregation plays a central role in federated analytics settings, where data contributors independently compute local updates that are later combined into a global result. In privacy-sensitive scenarios, this aggregation must occur without revealing individual updates to any party, including the orchestrator. SMPC-based aggregation schemes address this by enabling participants to mask their local outputs with randomly generated values that cancel out upon summation [6]. These protocols allow analytics such as mean, weighted sums, or even gradient accumulation to be performed across multiple parties, with formal privacy guarantees.

One of the core advantages of this approach is its compatibility with asynchronous or partially connected systems. Participants can operate independently and submit masked results when ready, without requiring synchronized rounds or persistent connectivity. Moreover, masking techniques can be combined with cryptographic commitments or integrity checks to ensure that submitted values are structurally correct and free from tampering [7]. This makes the scheme suitable for deployment in environments such as mobile networks, industrial sensor arrays, or inter-institutional research collaborations.

The following example illustrates a simplified implementation of secure aggregation using additive masking. Each participant adds a random noise vector to their local data and distributes corresponding canceling masks to others, ensuring that individual updates remain private but the global sum remains correct.

```
import numpy as np

def generate_masks(num_parties, vector_size):
    """Generate a set of canceling masks for secure aggregation."""
    masks = np.random.randint(-10, 10, (num_parties, vector_size))
    total_mask = np.sum(masks, axis=0)
    masks[-1] -= total_mask  # Ensure masks cancel out in aggregation
    return masks

def secure_aggregate(local_updates, masks):
    """Apply masks to local updates and sum masked values."""
    masked_updates = [u + m for u, m in zip(local_updates, masks)]
    aggregate = np.sum(masked_updates, axis=0)
    return aggregate

# Simulation: three clients compute local updates
num_clients = 3
vector_dim = 5
local_updates = [np.random.randint(0, 5, vector_dim) for _ in range(num_clients)]
masks = generate_masks(num_clients, vector_dim)

# Aggregator receives masked updates
aggregated_result = secure_aggregate(local_updates, masks)
print("Securely aggregated result:", aggregated_result)
```

This example demonstrates how additive masking can enable secure aggregation in federated systems without disclosing individual data contributions [8]. The use of canceling random vectors ensures that intermediate values remain private while allowing the correct global result to be recovered. Such techniques form the foundation of many privacy-preserving analytics protocols used in cross-device, cross-organization, or cross-border data collaborations.

Performance constraints and optimization strategies in SMPC systems

The deployment of SMPC protocols in big data contexts introduces substantial computational and communication overhead compared to conventional processing pipelines. These constraints stem from the cryptographic nature of secure computation, which often requires multiple rounds of interaction, modular arithmetic over finite fields, and the exchange of intermediate masked values. In large-scale analytics scenarios, where datasets contain millions of records or high-dimensional features, these costs can quickly render naïve implementations impractical [9]. Therefore, achieving acceptable performance in real-world applications demands both protocol-level optimizations and system-level adaptations.

One of the key performance bottlenecks lies in the network. Since many SMPC schemes rely on interactive operations between parties, latency and bandwidth become critical factors [10]. Protocols must be designed to minimize the number of communication rounds, reduce the size of

transmitted payloads, and support asynchronous execution. Techniques such as precomputation, where parties compute cryptographic shares or randomness in advance, can significantly reduce online latency. In some settings, hybrid approaches that combine secure computation with differential privacy or trusted execution environments are adopted to offload expensive operations while retaining security guarantees.

At the computational level, the complexity of arithmetic operations-particularly multiplication and comparison-is another limiting factor. Secure multiplication protocols often involve multiple communication rounds or require pre-shared multiplication triples, which may not scale efficiently with the dataset size. To address this, modern SMPC frameworks implement batch processing, parallel evaluation, and optimized circuits that reduce the gate complexity of common analytical functions [11]. Additionally, approximate computation techniques, such as fixed-point arithmetic and reduced-precision encoding, are employed to strike a balance between computational efficiency and result fidelity, especially in iterative algorithms such as training machine learning models.

Resource management is also a crucial concern. SMPC implementations must be tailored to the hardware and software constraints of deployment environments, whether in cloud clusters, on-premises servers, or edge devices. Memory usage, threading, and garbage collection behavior must be optimized to prevent system stalls during execution. Moreover, adaptive scheduling mechanisms that allocate computation tasks dynamically across available nodes help improve throughput and fault tolerance. These optimizations collectively enhance the feasibility of SMPC integration into production-scale data analytics systems, ensuring that confidentiality does not come at the expense of scalability and responsiveness.

Practical optimization strategies for scalable secure computation

As secure multi-party computation becomes more prevalent in large-scale data analysis, practical optimization strategies play a crucial role in bridging the gap between theoretical protocols and deployable systems [12]. These techniques are aimed at mitigating specific bottlenecks inherent to secure computation, such as latency, arithmetic overhead, and limited concurrency. In real-world applications, selecting the right combination of optimizations determines not only the speed of computation but also the scalability and fault resilience of the entire system.

The following table 1 presents a summary of widely adopted optimization techniques in SMPC implementations, along with the corresponding system-level bottlenecks they address and their impact on computational performance.

Table 1

Performance optimization strategies in SMPC systems

Optimization technique	Targeted bottleneck	Effect on performance
Precomputation of randomness	Online latency	Reduces wait time during live execution
Batch processing of secure operations	Per-operation overhead	Improves throughput for repeated computations
Use of fixed-point arithmetic	Arithmetic complexity	Decreases cost of numeric operations
Reduced communication rounds	Network delay	Minimizes inter-party synchronization delay
Parallel execution of local steps	Processing time	Accelerates computation across nodes
Adaptive task scheduling	Load balancing and throughput	Enhances scalability in heterogeneous systems

The optimization techniques outlined above demonstrate how targeted improvements at both the algorithmic and infrastructural levels can significantly enhance the efficiency of SMPC-based analytics. While each method addresses a distinct performance bottleneck, their combined application enables secure computation to scale toward production-level workloads without compromising privacy guarantees. Selecting appropriate strategies requires careful evaluation of system constraints, workload characteristics, and resource availability, highlighting the need for flexible and modular SMPC frameworks tailored to real-world deployment environments.

Integration with machine learning workflows in confidential analytics

Integrating SMPC protocols into machine learning (ML) pipelines introduces a new layer of complexity, driven by the need to balance computational privacy with model accuracy, training efficiency, and workflow automation. In many privacy-sensitive domains-such as healthcare diagnostics, financial risk modeling, and population-level behavior prediction-ML tasks must be executed collaboratively without revealing proprietary datasets. Secure computation frameworks provide a mechanism for such privacy-preserving collaboration, allowing distributed model training or inference without centralizing data.

One of the principal challenges in SMPC-ML integration is adapting iterative optimization algorithms, such as stochastic gradient descent, to function within a secure setting. These algorithms typically require repeated computation of gradients, updates to model parameters, and aggregation of local contributions across participants. When executed under secure protocols, each of these steps becomes significantly more resource-intensive, both in terms of communication and computation. In response, research efforts have focused on optimizing sub-protocols for secure matrix operations, developing quantization-aware training methods, and reducing the depth of arithmetic circuits used in model evaluation.

Another consideration involves supporting diverse ML models, from linear classifiers to deep neural networks. While simple models can often be implemented using fixed-point arithmetic and shallow circuits, more complex architectures require approximate activation functions, layer-wise encryption, or hybrid execution models where sensitive layers are computed securely while others operate in plaintext [13]. Additionally, data preprocessing tasks-such as normalization, encoding, and feature selection-must be securely embedded into the pipeline, ensuring that privacy is maintained end-to-end.

From a systems perspective, successful integration also depends on compatibility with existing ML frameworks and infrastructure. SMPC implementations must offer clean APIs, model conversion tools, and parallelizable backends to work alongside platforms like TensorFlow, PyTorch, or federated learning engines. Scalability, fault tolerance, and reproducibility become critical, particularly in multi-organizational settings where compute environments and data schemas differ. These requirements highlight the need for robust middleware that bridges secure computation engines with modern ML ecosystems, enabling confidential analytics to transition from experimental prototypes to reliable components of production data science workflows.

Conclusion

The evolution of secure multi-party computation has transformed the landscape of privacy-preserving analytics, offering practical tools for collaborative computation without compromising data confidentiality. By enabling distributed entities to jointly process sensitive information while retaining control over their private inputs, SMPC addresses a critical need in data-driven sectors governed by stringent regulatory and ethical constraints. Its applicability extends beyond theoretical models, reaching real-world systems that require both analytical insight and rigorous privacy protection.

This study has examined the structural principles, protocol designs, and implementation strategies that underpin the application of SMPC in big data environments. Through practical examples, code-level demonstrations, and architectural considerations, the analysis highlights both the capabilities and limitations of current approaches. Attention was given to optimization strategies, performance bottlenecks, and integration pathways with machine learning workflows-factors that define the viability of SMPC in production-scale deployments.

As the demand for confidential analytics continues to grow, future development of SMPC systems must focus on improving computational efficiency, reducing communication overhead, and enhancing interoperability with existing data science infrastructure. The design of modular, scalable, and developer-accessible SMPC frameworks will be central to this progress, paving the way for secure and collaborative data analysis at scale.

References

1. Alghamdi W., Salama R., Sirija M., Abbas A.R., Dilnoza K. Secure multi-party computation for collaborative data analysis // E3S Web of Conferences. EDP Sciences. 2023. Vol. 399. P. 04034.
2. Pappa C.K. Zero-Trust Cryptographic Protocols and Differential Privacy Techniques for Scalable Secure Multi-Party Computation in Big Data Analytics // J. Electrical Systems. 2024. Vol. 20. No. 5s. P. 2114-2123.
3. Sahinbas K., Catak F.O. Secure multi-party computation-based privacy-preserving data analysis in healthcare IoT systems // Interpretable Cognitive Internet of Things for Healthcare. Cham: Springer International Publishing. 2023. P. 57-72.
4. Nookala G. Secure Multiparty Computation (SMC) for Privacy-Preserving Data Analysis // Journal of Big Data and Smart Systems. 2023. Vol. 4. No. 1.
5. Salako A.O., Adesokan-Imran T.O., Tiwo O.J., Metibemu O.C., Onyenaucheya O.S., Olaniyi O.O. Securing Confidentiality in Distributed Ledger Systems with Secure Multi-Party Computation for Financial Data Protection // Journal of Engineering Research and Reports. 2025. Vol. 27. No. 3. P. 352-373.
6. Olusegun J., Holland M., Brightwood S., Jerry H., Frank E. Distributed Secure Multi-Party Computation (SMPC) for Cloud-Based Big Data Analytics. 2024.
7. Liu T. Research on Privacy Techniques Based on Multi-Party Secure Computation // 2024 3rd International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS). IEEE. 2024. P. 912-917.
8. Liu D., Yu G., Zhong Z., Song Y. Secure multi-party computation with secret sharing for real-time data aggregation in IIoT // Computer Communications. 2024. Vol. 224. P. 159-168.
9. Ahammed M.F., Labu M.R. Privacy-preserving data sharing in healthcare: advances in secure multiparty computation // Journal of Medical and Health Studies. 2024. Vol. 5. No. 2. P. 37-47.
10. Joshi D., Sanghi A., Agarwal G., Joshi B. Techniques for Protecting Privacy in Big Data Security: A Comprehensive Review // 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT). IEEE. 2024. Vol. 1. P. 1-7.
11. Becker S., Bösch C., Hettwer B., Hoeren T., Rombach M., Trieflinger S., Yalame H. Multi-Party Computation in Corporate Data Processing: Legal and Technical Insights // Cryptology ePrint Archive. 2025.
12. Dangi D., Santhi G. Secured multi-party data release on cloud for big data privacy-preserving using fusion learning // Turkish Journal of Computer and Mathematics Education. 2021. Vol. 12. No. 3. P. 4716-4725.
13. Yogi M.K., Mundru Y. Genomic data analysis with variant of secure multi-party computation technique // Journal of Trends in Computer Science and Smart Technology. 2024. Vol. 5. No. 4. P. 450-470.

UDC 004.722

INTERACTION MODELS OF INTELLIGENT SENSOR NETWORKS IN INDUSTRIAL INTERNET OF THINGS

Norkusheva D.M.
bachelor's degree, L.N. Gumilyov Eurasian national university (Astana, Kazakhstan)

МОДЕЛИ ВЗАИМОДЕЙСТВИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СЕНСОРНЫХ СЕТЕЙ В ПРОМЫШЛЕННОМ ИНТЕРНЕТЕ ВЕЩЕЙ

Норкушева Д.М.
бакалавр, Евразийский национальный университет имени Л.Н. Гумилёва (Астана, Казахстан)

Abstract

Interaction models in intelligent sensor networks (ISNs) form the basis for autonomous communication and coordination in industrial internet of things (IIoT) systems. The analysis focuses on topological structures, hierarchical communication layers, synchronization strategies, and decentralized behavior control. Core challenges related to interoperability, temporal consistency, and field-level integration are discussed alongside technical patterns for achieving scalable and resilient performance. The findings contribute to the development of robust ISN infrastructures capable of operating under the complexity of modern industrial environments.

Keywords: intelligent sensor networks, IIoT, decentralized coordination, interaction models, synchronization, interoperability, industrial protocols, distributed sensing.

Аннотация

Модели взаимодействия интеллектуальных сенсорных сетей (ИСН) служат основой для автономной коммуникации и согласованных действий в системах промышленного Интернета вещей (IIoT). Представлены ключевые подходы к построению топологий, организации многоуровневой передачи данных, синхронизации во времени и децентрализованному управлению поведением узлов. Особое внимание уделено совместимости, устойчивости к сбоям и особенностям применения в условиях реального промышленного производства. Предложенные выводы ориентированы на создание масштабируемых и надёжных ИСН-решений для сложных распределённых систем.

Ключевые слова: интеллектуальные сенсорные сети, IIoT, децентрализованная координация, модели взаимодействия, синхронизация, совместимость, промышленные протоколы, распределённые сенсоры.

Introduction

The rapid development of industrial automation and cyber-physical infrastructure has intensified the deployment of distributed sensing systems across manufacturing, energy, logistics, and resource management sectors. These systems rely on sensor networks capable of capturing environmental parameters, process states, and operational metrics in real time. As the scale and complexity of such networks increase, the emphasis shifts from simple data acquisition toward intelligent behavior and coordinated interaction among sensing units. This transition marks the evolution toward intelligent sensor networks, where autonomous sensing, processing, and communication capabilities enable adaptive, context-aware functionality.

The integration of ISNs within the framework of the industrial internet of things brings new challenges and opportunities. IIoT, defined as a distributed ecosystem of devices, communication protocols, and edge-computing systems applied in industrial settings, demands robust interaction models to ensure efficient, scalable, and resilient operations. In this context, interaction refers not only to data exchange but also to collaborative sensing, event-driven decision-making, and cooperative task execution. The behavior of ISNs is governed by a combination of decentralized logic, embedded intelligence, and networked coordination mechanisms that operate across heterogeneous physical and digital infrastructures.

This paper aims to investigate the conceptual and practical models of interaction among ISNs in IIoT environments. The focus lies on structural principles, communication topologies, decision protocols, and the role of local autonomy in global system behavior. Special attention is given to the design patterns that support synchronization, fault tolerance, and scalability, as well as to the constraints imposed by resource-limited devices and dynamic industrial environments. The objective is to provide a systematic overview of interaction models that form the basis for developing resilient, intelligent, and interoperable sensor networks in industrial domains.

Main part

Interaction within intelligent sensor networks deployed in IIoT environments extends far beyond traditional data polling or broadcasting. Modern systems are expected to exhibit autonomous behavior, which includes local decision-making, adaptive sampling, and collaborative filtering. These features are implemented through decentralized algorithms that operate under constrained power, memory, and computational capabilities [1]. The interaction paradigm must therefore accommodate heterogeneous communication requirements, varying temporal constraints, and real-time responsiveness.

In practice, the behavior of ISNs is shaped by the underlying interaction models-whether they are event-triggered, schedule-based, or opportunistic. Each model defines the conditions under which nodes exchange information, synchronize states, or delegate computation. Event-triggered interactions are often employed in anomaly detection, where a sensor activates a transmission only when predefined thresholds are breached. Scheduled models rely on predefined communication intervals, useful in energy-sensitive applications with predictable patterns. Opportunistic strategies, on the other hand, enable information exchange based on proximity or channel availability, which is particularly relevant in mobile or dynamically changing industrial layouts [2].

Furthermore, the choice of interaction model has a direct impact on the network's ability to self-organize and adapt. In industrial contexts characterized by noise, interference, and component failures, systems must maintain operational coherence without centralized control. This necessitates peer-to-peer negotiation protocols, dynamic topology discovery, and local rule execution. The balance between global coordination and local autonomy becomes a key factor in designing ISNs that can function reliably under stress, while still supporting complex industrial processes such as condition-based maintenance, decentralized control, and distributed diagnostics.

The structure of communication in ISNs is often layered to separate concerns such as sensing, aggregation, decision-making, and actuation. This layered interaction allows for modular system design, where each node can specialize in one or more functional roles depending on its position in the network [3]. For example, edge-layer nodes may primarily perform data collection and preliminary filtering, while intermediary units focus on local aggregation and consensus building. This architectural modularity supports scalability and fault isolation, which are essential in IIoT scenarios that span multiple physical zones and operational domains.

A critical aspect of interaction design is the selection of communication protocols tailored to the constraints of industrial environments. Factors such as electromagnetic interference, spatial coverage gaps, and real-time delivery requirements necessitate the use of robust, lightweight protocols. Popular choices include time-slotted channel hopping (TSCH), WirelessHART, and deterministic Ethernet variants. These protocols are designed to support synchronized multi-hop communication, minimize packet collisions, and ensure deterministic message delivery-properties

that are especially relevant in safety-critical systems such as automated assembly lines or energy grid control units [4].

Moreover, effective interaction relies not only on communication fidelity but also on contextual awareness and task alignment among nodes. Sensors must understand not only when and how to communicate but also what information is relevant to share under specific operational circumstances. This is achieved through embedded rule engines, dynamic priority queues, and semantic data models that allow nodes to make informed decisions about data relevance, urgency, and destination. Such context-driven interaction mechanisms reduce unnecessary traffic, preserve bandwidth, and enhance the overall responsiveness of the IIoT system.

Topological configurations for sensor interaction in industrial environments

The physical and logical topology of sensor networks plays a central role in shaping how interaction unfolds across an IIoT system. Common topological configurations include star, mesh, cluster-tree, and hybrid arrangements, each offering distinct trade-offs in terms of resilience, scalability, and latency [5]. In industrial settings, where equipment layout, electromagnetic interference, and reliability requirements vary significantly, the topology must be chosen or adapted dynamically based on operational constraints.

Star topologies, while easy to manage, suffer from single-point failure vulnerabilities and limited scalability. Mesh configurations, by contrast, support robust multi-path communication and can self-heal by rerouting data around failed nodes, but they impose higher protocol complexity and require careful synchronization. Cluster-tree topologies represent a structured compromise, enabling localized interaction within clusters while maintaining global coordination through a hierarchical backbone [6]. Hybrid models increasingly combine these features to exploit spatial hierarchies and optimize traffic flows in real-time.

Figure 1 illustrates typical topological interaction models employed in ISNs within industrial facilities. The figure highlights the structural roles of nodes, interaction flows, and communication dependencies that define each topology.

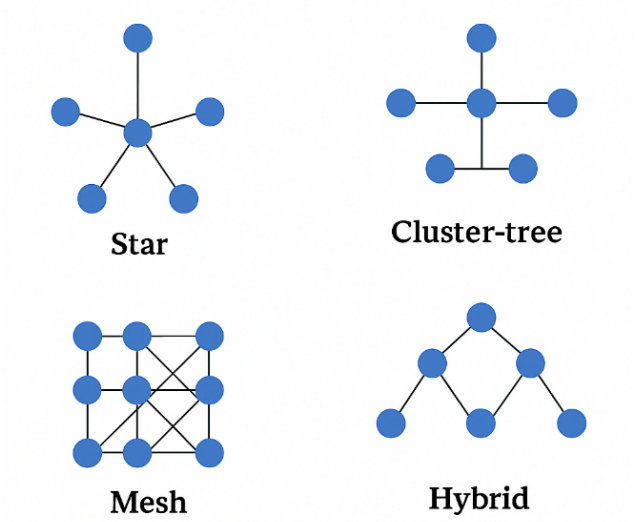


Figure 1. Topological interaction models for intelligent sensor networks in IIoT environments

The figure highlights how different topological configurations influence the interaction capabilities of intelligent sensor networks in industrial systems. While star structures provide centralized simplicity, they lack robustness under node failure. Mesh and cluster-tree (with lowercase «tree») arrangements offer enhanced resilience and dynamic adaptability, crucial for high-availability scenarios [7]. Hybrid models, integrating multiple topologies, present a balanced approach that supports both localized autonomy and system-wide coordination. These configurations serve as the structural basis for designing interaction models capable of maintaining communication integrity and operational continuity in IIoT environments.

Layered communication frameworks for coordinated sensor behavior

In complex industrial environments, sensor networks must operate across multiple abstraction levels to ensure both local responsiveness and system-wide coherence [8]. Layered communication

frameworks are employed to decompose responsibilities into functional strata-typically including physical, data link, network, coordination, and application layers. This stratification facilitates modularity, simplifies integration, and supports heterogeneity in hardware and protocols. Each layer is responsible for a specific aspect of interaction: for example, the coordination layer handles consensus and synchronization, while the application layer interprets semantic content for decision support systems.

Such frameworks are particularly beneficial in large-scale deployments, where diverse sensor types and roles coexist. For instance, condition monitoring sensors may interact within their local layer to detect anomalies, while simultaneously reporting summaries upward to a supervisory control system [9]. Meanwhile, actuators respond to coordinated commands derived from aggregated sensor input. This vertical communication model is complemented by horizontal interactions between peer nodes, enabling fault isolation, redundancy, and local optimization. The overall framework ensures that data flow, control signals, and analytical feedback propagate across the network in a structured, traceable manner.

Figure 2 presents a generalized schematic of a layered communication framework for ISNs in industrial systems, illustrating the flow of messages and interaction logic across hierarchical layers.

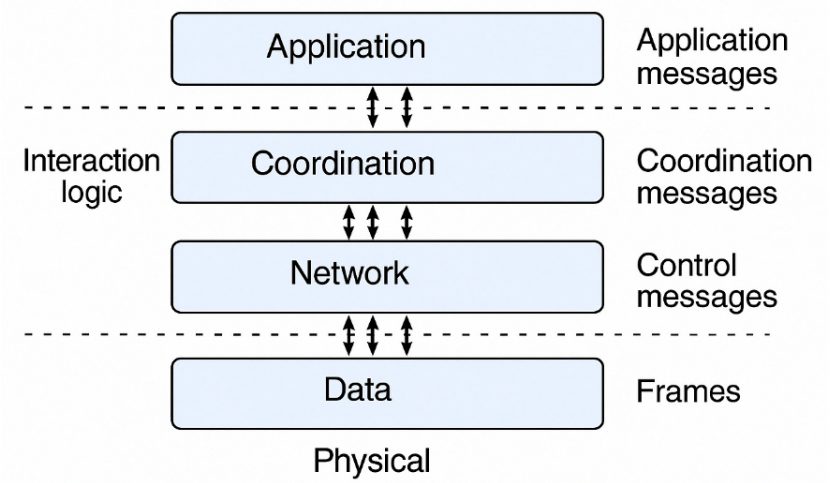


Figure 2. Layered communication framework for intelligent sensor networks in industrial systems

The figure illustrates a modular architecture in which each communication layer manages specific tasks-from raw data handling to semantic interpretation-enabling structured and scalable interaction among intelligent sensors. This layered approach enhances system maintainability, promotes interoperability across platforms, and ensures that data exchange aligns with both operational constraints and application-level objectives.

Behavior coordination strategies for decentralized sensor clusters

In large-scale IIoT deployments, sensor nodes are often grouped into autonomous clusters that must coordinate behavior without central supervision. These decentralized sensor clusters are expected to perform tasks such as fault detection, load balancing, or collaborative event classification in real time. Coordination within such groups relies on local information exchange, probabilistic consensus, and shared behavioral rules. The challenge lies in enabling consistency of group behavior despite variable connectivity, partial observability, and asynchronous communication patterns [10].

Several strategies have been proposed to support coordination in sensor clusters, including leader election, behavior imitation, and reputation-based mechanisms. In leader-based models, a representative node temporarily orchestrates communication and decision flow, while in imitation-based systems, nodes replicate the behavior of more reliable or better-performing neighbors. Reputation-based strategies add a layer of trust scoring, allowing nodes to weigh received information based on the credibility of the sender. Each method introduces different trade-offs between convergence speed, resilience to adversarial behavior, and energy efficiency.

Figure 3 illustrates typical coordination patterns in decentralized ISN clusters. It shows how local rules and neighborhood awareness lead to emergent system behavior, enabling reliable operation without centralized logic.

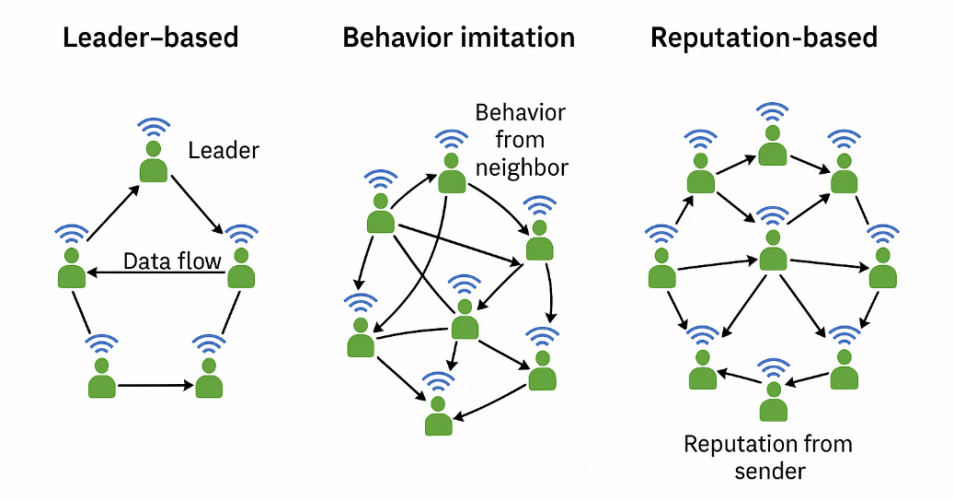


Figure 3. Coordination strategies in decentralized sensor clusters

This figure illustrates three primary coordination approaches in ISNs: leader-based control, behavior imitation among peers, and reputation-weighted consensus. Each method supports decentralized decision-making by leveraging local context and interaction history, enabling autonomous cluster operation without reliance on a central authority. The visual distinctions in structure and data flow demonstrate how different strategies achieve balance between autonomy, trust, and synchronization.

Temporal synchronization and consistency maintenance in dynamic sensor environments

Maintaining temporal coherence across distributed sensor nodes is a fundamental requirement for ensuring accurate and reliable operation in IIoT systems. Time-sensitive applications-such as event detection, process control, or energy optimization-depend heavily on the ability of individual nodes to interpret events within a consistent temporal frame of reference. In decentralized architectures, where sensor nodes operate asynchronously and may experience delays, jitters, or local clock drift, achieving network-wide synchronization poses significant technical challenges.

Synchronization protocols for ISNs are broadly categorized into clock synchronization schemes and event-driven synchronization. Clock synchronization protocols aim to align internal clocks of nodes using message exchanges and statistical correction techniques. Examples include protocols such as precision time protocol (PTP), flooding time synchronization protocol (FTSP), and reference broadcast synchronization (RBS). These schemes use techniques like averaging timestamps, minimizing skew, and recursive adjustments to maintain coherence. However, their efficiency degrades in noisy or mobile environments where packet loss and topology changes are frequent. In such contexts, event-driven synchronization is often favored, where coordination is based on shared sensing events rather than time alignment, reducing overhead but limiting global temporal accuracy.

In dynamic industrial settings, maintaining consistency is further complicated by partial visibility, inconsistent sensing intervals, and transient node failures. To address this, modern ISN architectures implement consistency maintenance layers that include buffer alignment, timestamp reconciliation, and data version control [11]. These mechanisms ensure that even in the presence of network fragmentation or reconfiguration, critical data remains temporally valid and usable for downstream analysis. Additionally, distributed consensus algorithms such as Paxos and Raft have been adapted for use in sensor environments to synchronize state and ensure that updates are propagated reliably, even under failure-prone conditions.

Ultimately, the design of synchronization mechanisms must strike a careful balance between precision, communication overhead, and fault tolerance. For low-power devices, the need to minimize energy consumption may preclude frequent synchronization messages, leading to the adoption of hybrid models that combine loose synchronization with local event correction. Meanwhile, latency-sensitive applications demand strict guarantees, pushing for high-frequency synchronization at the cost of increased resource usage. As IIoT systems continue to evolve, adaptive synchronization strategies that dynamically adjust behavior based on system load, node density, and operational

criticality will become increasingly important for sustaining reliable sensor interaction across temporal and spatial scales.

Interoperability and standardization in heterogeneous sensor ecosystems

One of the defining characteristics of industrial IIoT environments is the coexistence of heterogeneous devices originating from multiple vendors, generations, and technological paradigms. Intelligent sensor networks deployed in such settings must therefore support interoperability not only at the hardware and protocol level, but also across data semantics, control logic, and system objectives [12]. Achieving reliable interaction among diverse nodes requires adherence to common standards, modular integration architectures, and adaptive abstraction mechanisms that can bridge technological gaps without sacrificing performance or security.

Interoperability challenges arise in several layers of the interaction stack. At the communication level, differences in wireless technologies (e.g., Zigbee, LoRa, Wi-Fi, 6LoWPAN) and transport protocols (e.g., MQTT, CoAP, OPC UA) can hinder seamless integration, especially when low-power devices operate alongside high-bandwidth systems. Middleware solutions and protocol translation gateways are often introduced to mediate between incompatible stacks, but they introduce latency and additional points of failure [13]. As a response, standardization bodies have promoted cross-compatible specifications, including IEEE 1451 for smart transducer interfaces and ISO/IEC 30141 for IoT reference architectures, aiming to reduce fragmentation and promote plug-and-play functionality across industrial platforms.

At the data level, semantic interoperability becomes a major concern. Sensor outputs must not only conform to shared formats but also carry consistent meaning across applications and analytic modules. Ontology-driven frameworks and semantic annotation techniques are increasingly applied to enrich sensor data streams with machine-readable metadata, facilitating automated processing, integration, and reasoning. These methods enhance the discoverability and composability of sensor services while supporting advanced use cases such as federated learning, decentralized control, and adaptive system configuration.

On the application side, interaction logic must remain robust under variation in node behavior, functional roles, and domain-specific constraints. This requires flexible software architectures built on modular microservices, service-oriented messaging, and event-driven orchestration. Standard APIs and interface contracts enable the replacement or upgrading of individual components without compromising overall system integrity. Furthermore, conformance testing, certification programs, and simulation environments help validate interaction compatibility before deployment, reducing integration risk and ensuring smooth operation across the entire IIoT landscape.

Practical deployment considerations and field-level constraints

Despite significant progress in the theoretical modeling of intelligent sensor interaction, the deployment of such systems in operational industrial environments presents a range of practical challenges. Field-level conditions-such as harsh physical environments, electromagnetic interference, limited accessibility, and safety-critical constraints-impose additional demands on the robustness and adaptability of sensor network interaction models [14]. In such settings, even minor inconsistencies in communication or coordination can have outsized consequences, including system shutdowns, product defects, or compromised worker safety.

Hardware-level reliability is a fundamental concern. Sensor nodes must endure mechanical vibrations, temperature fluctuations, dust, humidity, and other stressors that can degrade performance over time. Redundancy mechanisms, including node duplication and failover routing, are often implemented to mitigate single-point vulnerabilities. However, these add complexity to the interaction model, particularly when synchronizing redundant streams or merging conflicting observations. Energy harvesting methods and ultra-low-power design patterns are also integrated into interaction logic to extend operational life without compromising functionality.

Another critical factor is deployment topology. In dense industrial spaces, signal interference and multipath propagation affect wireless communication quality, necessitating adaptive power control and dynamic channel management. Sensor placement strategies must account not only for coverage and accessibility, but also for maintainability and compliance with regulatory zoning

requirements. Interaction models are thus informed by physical layout constraints, necessitating localization awareness, signal quality estimation, and fallback mechanisms in the case of signal degradation [15].

Finally, integration with legacy infrastructure remains a common barrier. Many industrial systems were not designed with IIoT in mind, relying on proprietary protocols or closed-loop control schemes. Bridging these systems with modern ISNs requires careful consideration of timing compatibility, protocol adaptation, and security hardening. Gateways and protocol bridges are often introduced to mediate between legacy systems and modern sensor clusters, but these components themselves must conform to the overall interaction logic to prevent bottlenecks or inconsistencies. Successful deployment therefore depends not only on the strength of the models but also on their ability to adapt to heterogeneous and constrained operational environments.

Conclusion

The interaction between intelligent sensor nodes in industrial Internet of things environments is shaped by a complex interplay of communication protocols, coordination strategies, and system-level constraints. As IIoT systems evolve toward greater autonomy and scalability, the underlying models of interaction must support decentralized decision-making, dynamic topology management, and consistent temporal behavior. These requirements necessitate robust, adaptive, and resource-aware designs capable of maintaining performance across diverse operational contexts.

This study has provided a comprehensive examination of structural and functional principles that guide the behavior of intelligent sensor networks in industrial settings. Through analysis of topological configurations, layered communication frameworks, synchronization mechanisms, and coordination strategies, the paper highlights key factors influencing the reliability and efficiency of distributed sensing infrastructures. Emphasis was also placed on practical considerations such as interoperability, standardization, and field deployment challenges, all of which play a critical role in the transition from prototype systems to production-grade deployments.

In light of emerging industrial demands, future research should focus on the integration of learning-based interaction policies, self-healing coordination mechanisms, and secure interoperability protocols. These developments will be essential in building intelligent sensor networks that are not only functionally effective but also resilient, context-aware, and aligned with the operational realities of next-generation industrial systems.

References

1. Gupta D., de Albuquerque V.H.C., Khanna A., Mehta P.L. Smart Sensors for Industrial Internet of Things // Springer International Publishing. 2021. Vol. 10. P. 978-3.
2. Djenouri Y., Belhadi A., Srivastava G., Houssein E.H., Lin J.C.W. Sensor data fusion for the industrial artificial intelligence of things // Expert Systems. 2022. Vol. 39. No. 5. P. e12875.
3. Smoliarchuk V. Methods and techniques for improving the efficiency of business processes in manufacturing companies // Cold Science. 2025. No. 13. P. 53-60.
4. Majid M., Habib S., Javed A.R., Rizwan M., Srivastava G., Gadekallu T.R., Lin J.C.W. Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review // Sensors. 2022. Vol. 22. No. 6. P. 2087.
5. Li J., Dai J., Issakhov A., Almojil S.F., Souri A. Towards decision support systems for energy management in the smart industry and Internet of Things // Computers & Industrial Engineering. 2021. Vol. 161. P. 107671.
6. Sharma S., Verma V.K. An integrated exploration on internet of things and wireless sensor networks // Wireless Personal Communications. 2022. Vol. 124. No. 3. P. 2735-2770.
7. Jiang B., Li J., Yue G., Song H. Differential privacy for industrial internet of things: Opportunities, applications, and challenges // IEEE Internet of Things Journal. 2021. Vol. 8. No. 13. P. 10430-10451.
8. Seng K.P., Ang L.M., Ngharamike E. Artificial intelligence Internet of Things: A new paradigm of distributed sensor networks // International Journal of Distributed Sensor Networks. 2022. Vol. 18. No. 3. P. 15501477211062835.

9. Garifullin R. Application of RxJS and NgRx for reactive programming in industrial web development: methods for managing asynchronous data streams and application state // International Journal of Professional Science. 2024. No. 12-2. P. 42-47.
10. Xu H., Wu J., Pan Q., Guan X., Guizani M. A survey on digital twin for industrial internet of things: Applications, technologies and tools // IEEE Communications Surveys & Tutorials. 2023. Vol. 25. No. 4. P. 2569-2598.
11. Tharewal S., Ashfaque M.W., Banu S.S., Uma P., Hassen S.M., Shabaz M. Intrusion detection system for industrial Internet of Things based on deep reinforcement learning // Wireless Communications and Mobile Computing. 2022. Vol. 2022. No. 1. P. 9023719.
12. Ahmed S.F., Alam M.S.B., Hoque M., Lameesa A., Afrin S., Farah T., Muyeen S.M. Industrial Internet of Things enabled technologies, challenges, and future directions // Computers and Electrical Engineering. 2023. Vol. 110. P. 108847.
13. Nourildean S.W., Hassib M.D., Mohammed Y.A. Internet of things based wireless sensor network: a review // Indones. J. Electr. Eng. Comput. Sci. 2022. Vol. 27. No. 1. P. 246-261.
14. Peter O., Pradhan A., Mbohwa C. Industrial internet of things (IIoT): opportunities, challenges, and requirements in manufacturing businesses in emerging economies // Procedia Computer Science. 2023. Vol. 217. P. 856-865.
15. Lăzăroiu G., Kliestik T., Novak A. Internet of things smart devices, industrial artificial intelligence, and real-time sensor networks in sustainable cyber-physical production systems // Journal of Self-Governance and Management Economics. 2021. Vol. 9. No. 1. P. 20-30.

UDC 004.7

COMPARATIVE EFFICIENCY OF DATA SHARDING STRATEGIES IN DISTRIBUTED LEDGER SYSTEMS

Goryunova E.T.
specialist degree, Novosibirsk state university (Novosibirsk, Russia)

Krestov S.A.
specialist degree, Novosibirsk state university (Novosibirsk, Russia)

СРАВНИТЕЛЬНАЯ ЭФФЕКТИВНОСТЬ СТРАТЕГИЙ ШАРДИНГА ДАННЫХ В РАСПРЕДЕЛЁННЫХ РЕЕСТРОВЫХ СИСТЕМАХ

Горюнова Е.Т.
специалист, Новосибирский государственный университет (Новосибирск, Россия)

Крестов С.А.
специалист, Новосибирский государственный университет (Новосибирск, Россия)

Abstract

This paper presents a comparative evaluation of data sharding strategies used in distributed ledger systems. The analysis explores partitioning methods, cross-shard transaction protocols, storage architectures, and security implications associated with fragmenting ledger state. Particular attention is given to performance trade-offs, consistency management, and resistance to targeted attacks in partitioned networks. The findings offer practical insight into how different sharding approaches influence system scalability, responsiveness, and reliability. The study serves as a foundation for future design choices in high-performance ledger infrastructures.

Keywords: data sharding, distributed ledger, partitioned systems, cross-shard transactions, ledger architecture, scalability, performance, blockchain security.

Аннотация

В работе проведён сравнительный анализ стратегий шардинга данных в распределённых реестровых системах. Рассматриваются методы разделения состояния, протоколы обработки межшардовых транзакций, архитектурные решения хранения и аспекты безопасности, возникающие при фрагментации реестра. Отдельное внимание уделено балансу между производительностью, управлением согласованностью и устойчивостью к целевым атакам в условиях раздельных подсетей. Представленные результаты дают практическое понимание влияния различных подходов к шардингу на масштабируемость, отклик и надёжность систем. Исследование формирует основу для выбора архитектурных решений в высоконагруженных реестровых платформах.

Ключевые слова: шардинг данных, распределённый реестр, фрагментированные системы, межшардовые транзакции, архитектура реестра, масштабируемость, производительность, безопасность блокчейна.

Introduction

The rapid expansion of distributed ledger technologies (DLTs) in recent years has been driven by the increasing demand for secure, transparent, and decentralized data management. However, as

transaction volumes grow and participation scales globally, the limitations of monolithic ledger architectures become more pronounced. Bottlenecks in throughput, rising latency, and inefficiencies in state replication challenge the practical deployment of DLT-based platforms in real-world, high-frequency environments. To address these constraints, architectural paradigms based on data partitioning have emerged as a viable approach to enhance performance without sacrificing decentralization.

Data sharding-an approach rooted in distributed database design-has gained attention as a scalable solution for distributing ledger state across multiple parallel processing units or network nodes. Sharding allows segments of the ledger to be processed independently, reducing the computational and communication burden on individual participants. Various strategies have been proposed, differing in how they allocate data, route transactions, and handle cross-shard communication. These differences directly affect throughput, fault tolerance, and consistency models, making it essential to evaluate sharding methods through both theoretical analysis and empirical validation.

This study provides a structured examination of the efficiency of multiple data sharding strategies within the context of DLT platforms. The investigation encompasses static and dynamic partitioning schemes, load-balancing techniques, and transaction routing models. Special focus is placed on how these strategies perform under heterogeneous conditions, including variable network topologies and workload distributions. By comparing the practical implications of different approaches, the paper contributes to the development of performance-aware design principles for scalable and reliable distributed ledger infrastructures.

Main part

Structural classification of sharding techniques in distributed ledgers

Sharding strategies in distributed ledger systems are developed to address the problem of limited scalability by distributing storage and processing responsibilities among multiple nodes or sub-networks. These strategies differ significantly in how data is partitioned and accessed, how inter-shard communication is managed, and what consistency guarantees can be provided across partitions. In practice, the design of a sharding model must balance simplicity of implementation, efficiency of cross-shard operations, and the ability to adapt to varying workloads or deployment conditions.

To provide a comparative overview, several representative approaches to data sharding in DLTs have been analyzed and summarized [1]. These include static key-based partitioning, dynamic schemes that adjust based on observed activity, region-aware placement strategies, hash-based random distribution, and explicit routing mechanisms for inter-shard coordination. The following table 1 presents a structural comparison of these methods, focusing on five critical characteristics: data distribution logic, scalability potential, inter-shard communication overhead, and the complexity of consistency enforcement.

Table 1

Comparison of data sharding strategies in distributed ledger technologies

Sharding strategy	**Data distribution method**	**Scalability**	**Cross-shard communication overhead**	**Consistency complexity**
Static key-range partitioning	Fixed value ranges assigned to shards	Moderate; requires pre-analysis	Low	Simple within fixed ranges
Dynamic workload-aware sharding	Partitions adjusted based on access patterns	High; adapts to usage load	Moderate	Complex due to dynamic changes
Geographic region-based sharding	Allocation by physical or network location	Contextual; depends on topology	Low to moderate	Moderate; regionally consistent

Sharding strategy	Data distribution method	Scalability	Cross-shard communication overhead	Consistency complexity
Random hash-based partitioning	Keys mapped using hash functions	High; evenly balanced in theory	High	Requires global coordination
Cross-shard transaction routing	Routing layer manages inter-shard transfers	Variable; limited by routing overhead	High	High; requires transaction-level tracking

The comparative analysis shows that while static and geographically-aware sharding offer lower communication overhead, their adaptability to dynamic load is limited. In contrast, dynamic and hash-based approaches provide improved scalability at the cost of increased coordination complexity. Cross-shard routing introduces further overhead, particularly in systems where transactional atomicity must be preserved. Selecting a sharding strategy thus requires a careful trade-off between performance, network structure, and operational guarantees [2].

Beyond structural classification, the practical implications of each sharding strategy vary depending on deployment context and system objectives. For example, static key-range partitioning may suit systems with predictable access patterns, such as supply chain tracking or digital identity registries, but lacks flexibility when transaction distribution shifts over time. Conversely, dynamic workload-aware sharding introduces adaptability but demands real-time monitoring, rebalancing logic, and robust metadata tracking, increasing the overall operational overhead.

Geographic or network-based sharding introduces physical locality into data placement, which can significantly improve performance in latency-sensitive environments. However, this advantage may diminish in cloud-based or virtualized networks where physical proximity does not guarantee consistent communication quality [3]. Randomized hash-based strategies are often appealing for their simplicity and load balancing properties but struggle with maintaining atomicity and consistency during inter-shard transactions, especially in public blockchain environments.

As illustrated in figure 1, inter-shard routing mechanisms-while introduced to support operations across partitioned ledger spaces-are accompanied by significant architectural challenges. These include ambiguity in transaction ordering, difficulties in rollback handling during failure scenarios, and elevated risks of double-spending or inconsistencies in partial states. Therefore, the selection and implementation of such strategies require careful evaluation not only of their theoretical soundness but also of their operational resilience, edge-case handling, and integration with the broader governance and consensus structures of the distributed ledger environment.

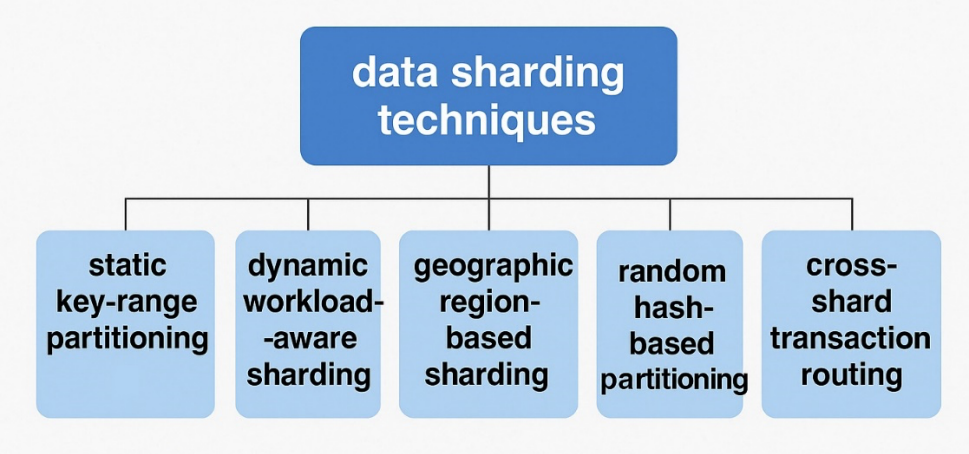


Figure 1. Structural classification of data sharding techniques in distributed ledgers

The figure provides a visual overview of the main data sharding strategies examined in this section. It highlights the logical distinctions and structural placement of each method, offering a clear reference for comparative analysis. This schematic reinforces the classification framework used to analyze performance trade-offs across varying ledger architectures.

Throughput and latency considerations in partitioned ledger systems

Performance metrics such as throughput and latency serve as primary indicators of the operational efficiency of distributed ledger systems employing sharding [4]. Throughput is generally defined as the number of transactions successfully processed per unit time, while latency refers to the delay between transaction submission and its final confirmation. In sharded environments, these metrics are influenced by a variety of factors, including the shard assignment algorithm, inter-shard message propagation delay, and the complexity of consensus synchronization across partitions.

In systems with static sharding, throughput tends to scale linearly with the number of shards, assuming an even distribution of workload and minimal inter-shard interaction. However, real-world usage often introduces workload imbalance and state contention, resulting in bottlenecks that offset the expected gains [5]. Dynamic sharding mechanisms attempt to address this by reallocating data or workload based on observed metrics, yet such adaptivity introduces overhead that may reduce net throughput, particularly in networks with frequent reconfiguration cycles or unstable connectivity.

Latency is particularly sensitive to the coordination model. Systems that require strong consistency guarantees-especially during cross-shard transactions-must perform multiple rounds of verification and commit procedures, increasing the time to finality. To mitigate this, some architectures adopt relaxed consistency models or delayed finality, sacrificing determinism for performance. Ultimately, the selection of a sharding strategy must be aligned with application-level tolerances: real-time systems demand minimal latency, while archival or batch-processing applications may prioritize throughput and state integrity.

Cross-shard transaction processing and consistency management

One of the defining challenges in sharded distributed ledger architectures is the processing of transactions that span multiple shards. Unlike single-shard operations, cross-shard transactions require coordinated execution across independent partitions, each maintaining its own subset of the global state [6]. This coordination must ensure atomicity, consistency, and isolation despite the absence of centralized control. Achieving these properties necessitates the design of robust communication protocols, transaction staging mechanisms, and conflict resolution strategies.

Several models have been proposed to manage cross-shard operations, including two-phase commit (2PC), optimistic concurrency control, and asynchronous messaging with eventual reconciliation. The 2PC approach ensures strong consistency by having all involved shards prepare and confirm the transaction before committing it globally. While effective, this method introduces significant latency and is vulnerable to deadlock in the presence of node failure. Optimistic concurrency models, by contrast, allow tentative execution followed by validation, reducing latency but risking rollback when conflicts arise. Asynchronous designs prioritize throughput and scalability by deferring coordination, accepting the risk of temporary inconsistencies.

In practice, the choice of model depends on the application's tolerance to temporary divergence and its need for fast finality. Financial ledgers, for instance, often require strict consistency and cannot afford conflicting transaction states, necessitating stronger coordination. In contrast, supply chain traceability systems may tolerate short-term inconsistencies in favor of performance. Additional mechanisms such as versioned state tracking, deterministic ordering, and cryptographic proofs (e.g., Merkle proofs for inter-shard state inclusion) are integrated to support secure reconciliation and verification across partitions.

Maintaining consistency across shards is further complicated by network dynamics, such as variable message delays and asynchronous node participation. To address this, some systems implement consistency layers that monitor state divergence and trigger reconciliation cycles or fallback consensus procedures. Others integrate coordination metadata directly into the transaction payloads, enabling context-aware validation at the shard level. These architectural choices influence not only correctness but also the resource efficiency and resilience of the overall system.

Ultimately, the effective processing of cross-shard transactions is a trade-off between protocol complexity, consistency guarantees, and system responsiveness. As applications demand both scalability and correctness, the future of sharded ledgers will likely depend on hybrid models that

dynamically adjust the degree of coordination based on transaction type, system load, and risk profile [7].

Models of cross-shard transaction processing in distributed ledger environments

The ability of a distributed ledger system to reliably process transactions that span multiple shards is a crucial factor in its practical viability. As data and users are divided across independent partitions, transactions affecting multiple shards must be executed in a coordinated and consistent manner. Without proper synchronization, inconsistencies in ledger state may emerge, leading to conflicting records, invalid balances, or security vulnerabilities. Therefore, designing robust models for cross-shard transaction handling is essential for preserving system integrity in sharded architectures.

Multiple approaches to cross-shard transaction processing have been proposed, each offering a different balance between consistency, performance, and failure tolerance. Traditional methods such as two-phase commit ensure atomicity but are susceptible to delays and coordination deadlocks. More modern techniques, including optimistic concurrency control and asynchronous reconciliation, aim to reduce overhead but introduce risks of temporary divergence. In addition, cryptographic methods like Merkle proof-based validation provide lightweight, secure ways to verify state inclusion across shards. These strategies differ not only in their implementation complexity but also in their resilience to failures and impact on transaction finality time.

The following table 2 summarizes and compares five common models of cross-shard transaction processing, focusing on consistency level, latency implications, and fault sensitivity.

Table 2

Comparison of cross-shard transaction processing models

Transaction model	Consistency guarantee	Latency impact	Failure sensitivity
Two-phase commit (2PC)	Strong (atomic and durable)	High	High (vulnerable to node stalls)
Optimistic concurrency control	Eventual (with possible rollback)	Low to moderate	Medium (depends on conflict rate)
Asynchronous reconciliation	Weak (requires post-verification)	Low	Low (tolerates delays)
Versioned state tracking	Moderate (conflict-aware updates)	Moderate	Medium (requires revalidation)
Merkle proof-based validation	High (cryptographic verification)	Moderate	Low (state inclusion verifiable)

This comparative overview highlights that there is no one-size-fits-all solution to cross-shard transaction processing. Systems prioritizing strong consistency and transactional determinism may opt for protocols like 2PC, despite their higher latency. Applications tolerant of temporary divergence can benefit from optimistic or asynchronous approaches, improving responsiveness. Cryptographic verification offers an efficient compromise, enhancing trust in inter-shard data without complex coordination. Ultimately, the choice of model should be informed by the operational profile of the ledger system, the nature of its workload, and the criticality of real-time consistency [8].

Storage optimization techniques in partitioned ledger environments

Efficient data storage is a fundamental requirement for scalable sharded ledger systems. As the number of partitions grows and historical data accumulates, the choice of storage strategies directly influences system responsiveness, fault tolerance, and operational cost. Each shard must manage not only transactional state but also indices, metadata, and recovery checkpoints. Consequently, designers face trade-offs between redundancy, retrieval speed, and storage economy that must be resolved based on workload type and access frequency.

Several architectural models have emerged to address these concerns. Some partitions employ full data replication to ensure high availability and quick recovery, especially in safety-critical systems. Others apply erasure coding techniques to balance fault tolerance with reduced storage footprint by splitting data into fragments with mathematical parity. In certain implementations, only

indexed summaries or state deltas are retained at the shard level, with full data archived externally or offloaded to cold storage layers. These approaches differ in their retrieval complexity, failure recovery mechanisms, and consistency implications.

Figure 2 presents a schematic overview of common storage strategies employed across independent partitions in sharded ledger systems.

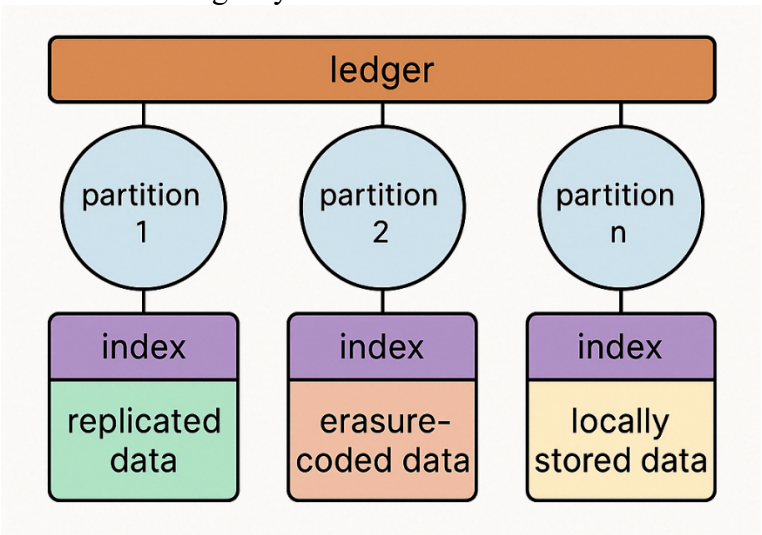


Figure 2. Storage strategies for sharded ledger systems

The figure illustrates how distinct shards may adopt different storage paradigms: full replication for resilience, erasure coding for storage efficiency, and local indexing for minimal operational load. The visual differentiation reinforces the idea that no single approach is universally optimal; rather, hybrid storage layering across partitions may offer the best trade-off between durability, space optimization, and retrieval latency in large-scale distributed ledger infrastructures.

Security implications of sharding in distributed ledgers

While sharding significantly enhances the scalability of distributed ledger systems, it also introduces unique security considerations that must be carefully addressed during protocol design and deployment. The very act of partitioning data and computation across independent subnets alters the traditional trust and threat models inherent to monolithic architectures. Each shard becomes a potential point of vulnerability, where local compromise may impact the integrity or availability of a segment of the global ledger.

One of the most critical security concerns is the risk of shard takeover attacks, in which an adversary gains control of a majority of the validating nodes within a single shard. Unlike global consensus mechanisms that rely on distributed quorum, individual shards often operate under reduced participant diversity, making them more susceptible to targeted collusion or sybil attacks. To mitigate this, some systems implement periodic re-shuffling of shard membership, use randomized assignment of nodes, or require cross-shard notarization before a transaction is finalized. However, these techniques introduce operational overhead and must be balanced against performance and complexity.

Another dimension of risk involves cross-shard transaction manipulation. As transactions span multiple partitions, the opportunity arises for adversaries to intercept, delay, or reorder messages to create inconsistent state transitions or double-spending scenarios. Ensuring secure coordination across shards requires cryptographic proofs, verifiable delay functions, and secure messaging protocols that are resistant to tampering or timing attacks [9]. Moreover, safeguarding transaction integrity depends on strict atomicity and rollback mechanisms, which must operate effectively even under partial network failure or adversarial interference.

Finally, data privacy and leakage become nuanced issues in sharded environments. While segmentation may isolate data access to specific shards, it can also reveal patterns in data distribution, transaction frequency, or network topology that adversaries could exploit for inference attacks. Zero-knowledge proofs and homomorphic encryption are being explored as privacy-preserving enhancements, though their integration with sharded architectures remains an open research

challenge. Additionally, careful attention must be paid to metadata exposure in inter-shard routing and consensus logs, where auxiliary information may inadvertently disclose sensitive context.

In summary, the decentralization benefits provided by sharding must be weighed against the complexity of maintaining consistent security guarantees across a fragmented system. A secure sharded ledger must not only defend each partition independently but also ensure that the collective behavior of the system preserves confidentiality, integrity, and availability at scale. The design of such architectures demands a multidisciplinary approach, integrating distributed systems theory, cryptography, and real-world operational insight.

Conclusion

Data sharding has emerged as a foundational technique for enhancing the scalability and responsiveness of distributed ledger systems. By segmenting state and workload across independent partitions, sharding enables parallelism, reduces transaction congestion, and aligns computational responsibilities with network topology. However, this architectural evolution introduces new challenges that span coordination, consistency, storage, and security domains.

This study has provided a comparative analysis of diverse sharding strategies, including static and dynamic partitioning, cross-shard transaction models, storage optimization schemes, and security mechanisms. Through structured examination of design trade-offs and performance characteristics, the paper highlights that the choice of sharding approach must be carefully aligned with system objectives, including fault tolerance, real-time responsiveness, and consistency requirements. No single model offers universal superiority; instead, adaptive and hybrid architectures often present the most viable path forward.

As distributed ledger technologies continue to evolve toward broader adoption in finance, supply chain, identity, and beyond, future research must focus on the refinement of sharding protocols that combine efficiency with verifiable trust guarantees. The advancement of formal verification, cryptographic coordination, and context-aware orchestration will play a critical role in shaping the next generation of scalable, secure, and interoperable ledger systems.

References

1. Quan B.L.Y., Wahab N.H.A., Al-Dhaqm A., Alshammari A., Aqarni A., Abd Razak S., Wei K.T. Recent Advances in Sharding Techniques for Scalable Blockchain Networks: A Review // IEEE Access. 2024.
2. Blazhkovskii A. Collecting metrics for continuous platform monitoring // Universum: technical sciences : electronic scientific journal. 2025. No. 3(132). P. 10-15.
3. Dhulavvagol P.M., Totad S.G. Performance enhancement of distributed system using HDFS federation and sharding // Procedia Computer Science. 2023. Vol. 218. P. 2830-2841.
4. Wu J., Yuan L., Xie T., Dai H. A sharding blockchain protocol for enhanced scalability and performance optimization through account transaction reconfiguration // Journal of King Saud University-Computer and Information Sciences. 2024. Vol. 36. No. 8. P. 102184.
5. Terletska K. Low-level memory management in scalable distributed architectures: Approaches to improving reliability and performance of digital services // International Journal of Research Publication and Reviews. 2025. Vol. 6(4). P. 5078-5081.
6. Aslam A., Postolache O., Oliveira S., Pereira J.D. Securing IoT Sensors Using Sharding-Based Blockchain Network Technology Integration: A Systematic Review // Sensors (Basel, Switzerland). 2025. Vol. 25. No. 3. P. 807.
7. Heo J.W., Ramachandran G.S., Dorri A., Jurdak R. Blockchain data storage optimisations: a comprehensive survey // ACM Computing Surveys. 2024. Vol. 56. No. 7. P. 1-27.
8. Yang H., Zhang X., Wu Z., Wang L., Chen X., Liu L. Co-sharding: a sharding scheme for large-scale internet of things application // Distributed Ledger Technologies: Research and Practice. 2024. Vol. 3. No. 1. P. 1-16.
9. Xu G., Zhou Z., Song X., Huang Y. Research on transaction allocation strategy in blockchain state sharding // Future Generation Computer Systems. 2025. P. 107756.

UDC 004.7:004.8:004.021

LOAD FORECASTING SYSTEMS FOR CLOUD PLATFORMS USING HYBRID ALGORITHMS

Grigoryan S.N.
postgraduate student, Azerbaijan state oil and industry university (Baku, Azerbaijan)

Zarutyunyan T.V.
postgraduate student, Azerbaijan state oil and industry university (Baku, Azerbaijan)

СИСТЕМЫ ПРОГНОЗИРОВАНИЯ НАГРУЗКИ ДЛЯ ОБЛАЧНЫХ ПЛАТФОРМ С ИСПОЛЬЗОВАНИЕМ ГИБРИДНЫХ АЛГОРИТМОВ

Григорян С.Н.
аспирант, Азербайджанский государственный университет нефти и промышленности (Баку, Азербайджан)

Зарутюнян Т.В.
аспирант, Азербайджанский государственный университет нефти и промышленности (Баку, Азербайджан)

Abstract

Hybrid forecasting systems have become essential for anticipating dynamic resource demands in cloud computing. By integrating machine learning, time-series modeling, and adaptive mechanisms, these systems enable accurate load predictions across heterogeneous workloads and fluctuating usage patterns. The study explores the design and evaluation of such models, highlighting architectural considerations, empirical trade-offs, and real-time deployment strategies. Results from comparative experiments demonstrate the effectiveness of hybrid approaches in reducing forecasting error and improving provisioning efficiency. Emphasis is placed on system responsiveness, model adaptability, and performance under operational constraints.

Keywords: load forecasting, cloud computing, hybrid models, adaptive learning, time-series prediction, resource allocation, performance evaluation.

Аннотация

Гибридные системы прогнозирования позволяют точно оценивать нагрузку в условиях изменяющихся облачных рабочих процессов. Их архитектура основана на сочетании методов машинного обучения, анализа временных рядов и адаптивных алгоритмов, что обеспечивает устойчивость к нерегулярности данных и дрейфу концепции. В работе представлены ключевые компоненты построения таких моделей, даны сравнительные характеристики эффективности и проанализированы сценарии их внедрения в реальном времени. По результатам тестирования показано, что комбинированные алгоритмы обеспечивают снижение ошибок прогноза и способствуют более эффективному управлению ресурсами.

Ключевые слова: прогнозирование нагрузки, облачные вычисления, гибридные модели, адаптивное обучение, временные ряды, распределение ресурсов, оценка эффективности.

Introduction

The rapid growth of digital infrastructure and cloud computing has led to a significant increase in resource variability, making accurate load forecasting a critical component for maintaining efficiency, scalability, and service reliability. Cloud platforms must continuously adapt to fluctuating user demands, dynamic application workloads, and shifting network conditions. This operational complexity requires advanced predictive systems capable of anticipating resource utilization patterns with high precision. Traditional statistical methods, while effective in stable environments, often fall short in capturing non-linear, multi-source dependencies characteristic of modern cloud infrastructures.

To address these limitations, hybrid algorithms have gained prominence by combining the strengths of machine learning techniques, time-series models, and heuristic optimization approaches. These composite methods offer a more flexible framework for capturing short-term spikes, long-term trends, and contextual anomalies in workload behavior. For instance, the integration of artificial neural networks with autoregressive models or evolutionary algorithms has demonstrated improved performance in forecast accuracy, adaptability, and generalization. Moreover, hybridization enables the incorporation of external variables such as seasonal patterns, service-level agreements, and user mobility into the prediction model.

The aim of this study is to examine the design, implementation, and comparative performance of hybrid load forecasting systems tailored for cloud environments. The research focuses on evaluating different algorithmic combinations, architectural frameworks, and deployment strategies that optimize forecasting accuracy while preserving computational efficiency. In doing so, the paper seeks to establish practical guidelines for selecting, tuning, and integrating hybrid forecasting models into cloud orchestration workflows, ultimately supporting proactive resource management and cost optimization.

Main part

Efficient load forecasting in cloud computing environments requires models that can account for diverse workload characteristics, system heterogeneity, and time-varying patterns. Unlike conventional server infrastructures, cloud platforms are elastic by design, dynamically allocating resources across distributed virtual machines, containers, and microservices [1]. This operational fluidity necessitates predictive systems that go beyond static modeling to incorporate dynamic behavioral cues and real-time signals from infrastructure and application layers.

A typical cloud workload exhibits multidimensional temporal dependencies. For instance, diurnal usage cycles, seasonal load surges, and unpredictable bursts due to marketing campaigns or external events introduce layers of complexity that challenge simple linear models. Furthermore, workload distributions may shift due to changes in user behavior, application updates, or migration between data centers. Capturing these dynamics requires models capable of adapting to concept drift and non-stationary data while maintaining real-time inference performance.

To achieve these goals, forecasting systems increasingly adopt hybrid approaches that integrate multiple algorithmic components. These systems typically combine data-driven learning models-such as long short-term memory (LSTM) networks or gradient boosting regressors-with signal decomposition methods and statistical filters. Hybrid architectures allow for parallel processing of trend, seasonality, and residual components, with the outputs merged through ensemble strategies. This modularity not only improves accuracy but also enables scalability and modular deployment, allowing each component to be independently tuned or updated.

The effectiveness of a forecasting system is largely determined by its ability to generalize across different cloud service models-infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Each model exhibits distinct workload signatures, driven by varying levels of abstraction, user interaction, and orchestration granularity. For example, IaaS workloads often reflect direct user-initiated provisioning events, whereas SaaS platforms experience aggregated and highly variable demand patterns influenced by application logic and multitenancy [2].

Hybrid forecasting architectures must accommodate these differences by incorporating feature extraction mechanisms that can adapt to domain-specific indicators. These may include CPU

utilization metrics, memory pressure, network throughput, disk I/O rates, and service response times, which collectively inform the system about workload stress and resource saturation points. Preprocessing techniques such as normalization, dimensionality reduction, and frequency filtering are applied to ensure that input data remains interpretable and noise-resilient across time.

Additionally, model interpretability and computational efficiency are critical in operational deployments. Hybrid models that combine black-box neural components with interpretable statistical elements-such as exponential smoothing or regression trees-can offer both high accuracy and traceability of decisions [3]. This is particularly important for cloud providers aiming to maintain transparency in autoscaling logic and meet regulatory or contractual requirements. The design of such forecasting systems must therefore reflect not only predictive performance goals but also architectural and governance constraints specific to the cloud context.

The training and evaluation of hybrid forecasting models require careful dataset selection and validation procedures that reflect real-world cloud dynamics. Historical traces of cloud workloads, obtained from production logs or public repositories such as google cluster data or azure VM traces, serve as the foundation for building and testing models. However, these datasets often suffer from class imbalance, missing values, and irregular sampling intervals. To address these issues, preprocessing pipelines are designed to align time steps, interpolate gaps, and filter out anomalous behavior not representative of typical system usage.

Evaluation metrics for forecast performance extend beyond conventional measures such as mean absolute error (MAE) and root mean squared error (RMSE). In cloud environments, forecasting quality must also consider the consequences of over- or under-provisioning. An overestimation may result in unnecessary resource allocation and cost inefficiency, while underestimation can lead to service-level agreement (SLA) violations and degraded user experience. As such, cost-aware loss functions, percentile-based error analysis, and capacity violation tracking are increasingly incorporated into model assessment protocols.

Moreover, retraining and model adaptation mechanisms are essential in the face of evolving cloud workloads. Rather than deploying static models, hybrid systems may operate under online learning frameworks or periodic batch updates, depending on latency tolerance and model complexity. In mission-critical environments, model refresh cycles are orchestrated to avoid service disruption, often leveraging rolling windows, staged deployment, or shadow evaluation. These lifecycle considerations are a key part of ensuring that forecasting systems remain aligned with the operational realities of the cloud.

Hybrid architecture for multivariate load prediction in cloud platforms

Accurate load prediction in cloud environments requires models capable of analyzing multiple correlated indicators simultaneously. These indicators often include CPU usage, memory allocation, disk throughput, and network latency, each of which may reflect distinct yet interdependent load patterns. Hybrid architectures designed for this purpose typically combine machine learning techniques with time-series analysis tools to capture both temporal dependencies and cross-metric interactions [4].

A common approach is to preprocess the multivariate time series using signal decomposition or feature scaling, and then route the transformed data through separate predictive blocks. For example, long short-term memory networks can capture temporal correlations, while gradient boosting machines (GBMs) or random forests can identify nonlinear relationships among input features. The outputs of these blocks are often fused via weighted ensembles or meta-learners, enhancing overall accuracy and robustness.

Below is a simplified implementation of such a hybrid model pipeline using python and keras/scikit-learn. The code demonstrates how LSTM can be combined with gradient boosting for enhanced multivariate forecasting.

```
import numpy as np
import pandas as pd
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.preprocessing import StandardScaler
```

```
from keras.models import Sequential
from keras.layers import LSTM, Dense

# Load multivariate cloud metrics (e.g., CPU, memory, network)
data = pd.read_csv('cloud_metrics.csv')
features = data[['cpu_usage', 'memory_alloc', 'net_traffic']].values
targets = data['future_load'].values

# Scale features
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Prepare data for LSTM
X_seq = []
y_seq = []
window_size = 10
for i in range(len(scaled_features) - window_size):
    X_seq.append(scaled_features[i:i+window_size])
    y_seq.append(targets[i+window_size])
X_seq = np.array(X_seq)
y_seq = np.array(y_seq)

# Train LSTM model
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(window_size, X_seq.shape[2])))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
model.fit(X_seq, y_seq, epochs=20, verbose=0)

# Extract LSTM predictions for GBM input
lstm_output = model.predict(X_seq)

# Concatenate original features with LSTM output
gbm_input = np.hstack((scaled_features[window_size:], lstm_output))

# Train GBM on extended features
gbm = GradientBoostingRegressor()
gbm.fit(gbm_input, targets[window_size:])
```

The implementation of hybrid architectures combining LSTM networks with gradient boosting models demonstrates a promising approach to multivariate load forecasting in cloud environments. By leveraging the strengths of sequence modeling and feature-based regression, such systems can more effectively capture both temporal trends and nonlinear relationships among operational metrics. This layered strategy not only improves forecast accuracy but also enhances model flexibility, allowing the system to adapt to diverse workload conditions and heterogeneous input patterns. The integration of neural and tree-based learners provides a balanced trade-off between interpretability, scalability, and predictive performance, which is essential for real-time decision-making in dynamic cloud platforms [5].

Model integration workflow for real-time cloud resource prediction

Deploying hybrid forecasting systems within operational cloud platforms requires the integration of multiple components into a cohesive workflow. This includes data ingestion modules, real-time feature preprocessing pipelines, parallel prediction engines, and orchestration logic that governs model selection and decision application. The architecture must support modular deployment, fault isolation, and asynchronous updates to ensure system reliability and scalability.

In most implementations, forecasting models are encapsulated as services that interact through message queues or API calls. Feature extraction runs continuously on incoming metrics, with

buffering and windowing applied to synchronize input streams. Prediction results are passed to the orchestration layer, which evaluates threshold conditions, scaling policies, or scheduling directives [6]. This pipeline can be enhanced with feedback loops that capture actual load outcomes, enabling online learning or error correction modules to refine future forecasts.

Figure 1 illustrates a generalized workflow for integrating hybrid forecasting models into cloud environments. The diagram highlights the interactions among modules, real-time data pathways, and system feedback loops that support adaptive decision-making.

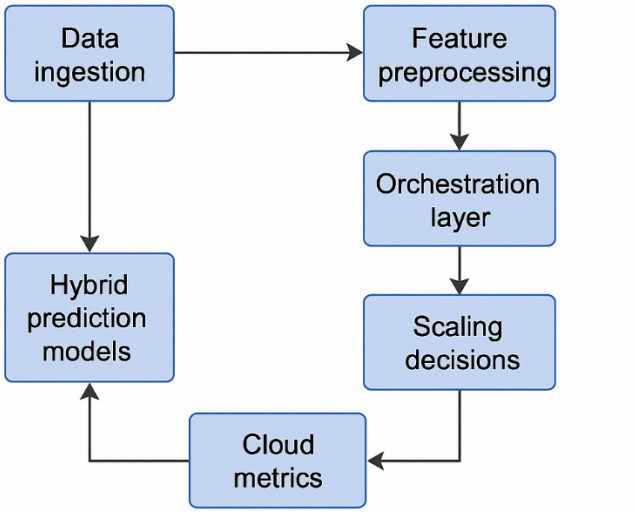


Figure 1. Model integration workflow for real-time cloud resource prediction

The figure demonstrates the modular pipeline used for real-time forecasting and scaling in cloud environments. Each component-ranging from data ingestion to decision orchestration-is placed in an adaptive loop, reinforcing the system's capacity for continuous feedback, model retraining, and operational optimization. The integration of prediction models within a reactive architecture supports proactive resource planning and reinforces system resilience under dynamic workloads.

Evaluation protocols and performance metrics for hybrid forecasting models

Evaluating the effectiveness of hybrid forecasting systems in cloud environments requires comprehensive protocols that reflect both statistical precision and operational impact. While conventional performance metrics-such as mean absolute percentage error (MAPE), mean squared error (MSE), and R^2 score-remain valuable, they provide an incomplete picture when isolated from real-world deployment implications. In predictive systems supporting cloud orchestration, forecast accuracy must be interpreted in the context of infrastructure efficiency, SLA compliance, and cost overhead introduced by resource misallocation [7].

Modern evaluation pipelines include multi-dimensional analysis frameworks that align forecast error metrics with system-level consequences. For instance, underestimation of load can result in service degradation or scaling delays, whereas overestimation leads to resource idleness and increased operational expenditure. To reflect these realities, hybrid models are increasingly assessed using asymmetric loss functions, cost-aware scoring, and metrics such as the resource provisioning deviation index (RPDI), which quantifies the degree of deviation between forecasted and actual resource allocations. These advanced metrics allow for a more nuanced comparison of models under varying workload patterns and tolerance thresholds.

Furthermore, temporal sensitivity and stability over time are key considerations. Forecasting systems deployed in production must perform consistently across different time intervals, usage spikes, and structural changes in cloud traffic. Therefore, rolling-window validation, online testing with delayed labels, and backtesting across historical workload segments are incorporated into the evaluation process. These techniques reveal model robustness under distributional shifts, helping avoid overfitting to specific event patterns or static seasonal cycles.

In addition to prediction quality, computational efficiency and model responsiveness are integral to real-time applicability. Hybrid systems must operate within strict inference time budgets

to avoid introducing latency into decision workflows. Evaluation protocols may therefore include timing benchmarks, memory usage profiling, and container-level latency tracking. Models that cannot meet real-time constraints-despite offering high accuracy-may be unsuitable for deployment in latency-sensitive environments, such as autoscaling controllers or edge-cloud hybrid nodes.

Finally, explainability and diagnostic capabilities are emerging as critical dimensions of performance evaluation [8]. As hybrid models become increasingly complex, cloud operators must be able to understand, audit, and trust the system's outputs. This has led to the integration of explainable AI (XAI) components into the evaluation stack, such as SHAP values for feature contribution analysis or attention maps in recurrent models. These tools support transparent forecasting logic and facilitate model debugging, tuning, and compliance with transparency mandates in regulated cloud services.

Empirical performance comparison of hybrid forecasting models

To assess the practical efficiency of various forecasting strategies in real-world cloud scenarios, a comparative evaluation was conducted using a set of hybrid and baseline models. The goal was to identify trade-offs between forecast accuracy, interpretability, and runtime performance. Each model was tested against a common multivariate workload dataset, with standard preprocessing and aligned training-validation protocols to ensure fairness. Key metrics included mean absolute percentage error, resource provisioning deviation index, average inference latency, and qualitative interpretability ranking.

Table 1 summarizes the comparative performance of the models tested, ranging from single-algorithm baselines to more complex hybrid compositions. Inference latency was measured under a standardized cloud container environment, and interpretability was ranked based on model transparency and feature attribution availability.

Table 1

Extended comparative evaluation of forecasting models

Model	MAPE (%)	RMSE	RPDI (↓)	Latency (ms)	Training time (s)	Memory usage (MB)	Interpretability
LSTM only	13.2	22.5	0.32	52	240	180	Low
GBM only	11.8	20.1	0.29	35	95	110	Medium
LSTM + GBM (hybrid)	8.5	14.7	0.17	67	410	260	Medium
ARIMA + GBM	9.7	16.3	0.22	60	360	230	Medium
LSTM + XGBoost + kalman filter	7.9	13.2	0.15	74	510	300	Low

The extended analysis reveals that hybrid architectures, particularly those integrating deep learning with ensemble and filtering methods, offer superior predictive accuracy and provisioning precision. However, these benefits are accompanied by higher training time, memory usage, and system latency, which may limit their applicability in resource-constrained or real-time scenarios. Simpler models like GBM deliver moderate accuracy with better efficiency, suggesting a favorable trade-off for certain deployment environments. The results underscore the importance of context-aware model selection, balancing predictive strength with infrastructure limitations and explainability needs.

Adaptive learning and model updating in dynamic cloud environments

In operational cloud platforms, workload characteristics evolve continuously due to changing user behavior, software updates, and seasonal demand fluctuations. Static forecasting models, trained once and deployed indefinitely, often degrade over time in accuracy and responsiveness. To mitigate this, modern forecasting architectures incorporate adaptive learning mechanisms that enable models

to retrain, fine-tune, or recalibrate based on new observations. This allows the forecasting system to remain aligned with the current statistical properties of the workload [9].

Adaptive updating strategies include incremental learning, where models are refined using streaming data; periodic batch retraining, triggered by predefined time intervals; and concept drift detection, which initiates retraining when data distribution shifts are detected. These approaches can be combined with model versioning and rollback mechanisms to ensure that degraded models are identified and replaced without service interruption. In critical systems, shadow testing is often applied, allowing new model versions to run in parallel with production forecasts for comparison before deployment.

The following python code demonstrates a lightweight adaptive learning loop using a gradient boosting model retrained periodically based on accumulated error. The mechanism checks forecast residuals against a rolling threshold and triggers model refresh when performance drops below a defined level.

```
import numpy as np
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error

# Initialize historical training data
X_train, y_train = get_initial_dataset()
model = GradientBoostingRegressor()
model.fit(X_train, y_train)

# Monitoring loop for adaptive update
residual_threshold = 15  # RMSE threshold
sliding_window = []

for batch in data_stream():  # Simulated incoming data
    X_batch, y_batch = batch
    y_pred = model.predict(X_batch)
    error = mean_squared_error(y_batch, y_pred, squared=False)  # RMSE
    sliding_window.append(error)

    # Keep last 5 RMSE scores
    if len(sliding_window) > 5:
        sliding_window.pop(0)

    avg_error = np.mean(sliding_window)

    if avg_error > residual_threshold:
        # Trigger model update
        print("Updating model...")
        X_new, y_new = fetch_updated_data()
        model.fit(X_new, y_new)
        sliding_window.clear()
```

The integration of adaptive learning into cloud-based forecasting systems enables models to remain effective amid shifting workload patterns and operational dynamics. By monitoring performance in real time and triggering targeted retraining, these systems reduce long-term drift and maintain forecast reliability without continuous manual intervention [10]. While adaptive mechanisms introduce additional complexity in model lifecycle management, they are essential for ensuring sustained accuracy in environments characterized by variability, user heterogeneity, and frequent application changes. The example implementation highlights how lightweight, threshold-based updating can be embedded within forecasting pipelines to support responsive and resilient prediction services.

Conclusion

Accurate load forecasting is a critical enabler of efficiency and scalability in modern cloud platforms. As resource usage becomes increasingly volatile and application demands more dynamic, forecasting systems must evolve from static, monolithic models to adaptive, hybrid architectures capable of real-time operation and continuous refinement. This study has examined the structure, integration, and empirical performance of various forecasting approaches, with a focus on the application of hybrid algorithms that combine the strengths of deep learning, statistical modeling, and heuristic adaptation.

The findings demonstrate that hybrid models offer substantial improvements in prediction accuracy and provisioning reliability when compared to single-method baselines. However, these gains are accompanied by increased system complexity, higher latency, and greater computational overhead, necessitating a careful trade-off analysis in practical deployments. The integration of adaptive learning mechanisms further enhances model longevity and responsiveness, ensuring robustness under workload evolution and concept drift.

Future developments in load forecasting systems are likely to center around explainable hybrid architectures, real-time retraining under resource constraints, and cross-platform interoperability. By embedding forecasting capabilities into the core of cloud orchestration workflows, providers can achieve proactive resource management, reduce operational costs, and sustain service-level objectives in increasingly dynamic digital environments.

References

1. Peng H., Wen W.S., Tseng M.L., Li L.L. A cloud load forecasting model with nonlinear changes using whale optimization algorithm hybrid strategy // Soft Computing. 2021. Vol. 25. No. 15. P. 10205-10220.
2. Rotib H.W., Nappu M.B., Tahir Z., Arief A., Shiddiq M.Y. Electric load forecasting for Internet of Things smart home using hybrid PCA and ARIMA algorithm // International Journal of Electrical and Electronic Engineering & Telecommunications. 2021. Vol. 10. No. 6. P. 369-376.
3. Simaiya S., Lilhore U.K., Sharma Y.K., Rao K.B., Maheswara Rao V.V.R., Baliyan A., Alroobaea R. A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques // Scientific Reports. 2024. Vol. 14. No. 1. P. 1337.
4. Patel E., Kushwaha D.S. A hybrid CNN-LSTM model for predicting server load in cloud computing // The Journal of Supercomputing. 2022. Vol. 78. No. 8. P. 1-30.
5. Devi K.L., Valli S. Time series-based workload prediction using the statistical hybrid model for the cloud environment // Computing. 2023. Vol. 105. No. 2. P. 353-374.
6. Anupama K.C., Shivakumar B.R., Nagaraja R. Resource utilization prediction in cloud computing using hybrid model // International Journal of Advanced Computer Science and Applications. 2021. Vol. 12. No. 4.
7. Bacanin N., Simic V., Zivkovic M., Alrasheedi M., Petrovic A. Cloud computing load prediction by decomposition reinforced attention long short-term memory network optimized by modified particle swarm optimization algorithm // Annals of Operations Research. 2023. P. 1-34.
8. Hu Y., Li J., Hong M., Ren J., Man Y. Industrial artificial intelligence based energy management system: Integrated framework for electricity load forecasting and fault prediction // Energy. 2022. Vol. 244. P. 123195.
9. Asiri M.M., Aldehim G., Alotaibi F.A., Alnfiai M.M., Assiri M., Mahmud A. Short-term load forecasting in smart grids using hybrid deep learning // IEEE Access. 2024. Vol. 12. P. 23504-23513.
10. Toumi H., Brahmi Z., Gammoudi M.M. RTSLPS: Real time server load prediction system for the ever-changing cloud computing environment // Journal of King Saud University-Computer and Information Sciences. 2022. Vol. 34. No. 2. P. 342-353.

UDC 004.75

BLOCKCHAIN-BASED DIGITAL IDENTITY MANAGEMENT SYSTEMS FOR CROSS-BORDER INTERACTIONS

Kholmatov F.A.
master's degree, Moscow institute of physics and technology
(Moscow, Russia)

СИСТЕМЫ УПРАВЛЕНИЯ ЦИФРОВОЙ ИДЕНТИЧНОСТЬЮ НА ОСНОВЕ БЛОКЧЕЙНА ДЛЯ ТРАНСГРАНИЧНОГО ВЗАИМОДЕЙСТВИЯ

Холматов Ф.А.
магистр, Московский физико-технический институт
(Москва, Россия)

Abstract

Blockchain-based digital identity systems are reshaping how individuals and institutions manage identity credentials across jurisdictions. By leveraging decentralized identifiers, verifiable credentials, and distributed trust models, these systems enhance privacy, data control, and interoperability. This paper investigates the architectural foundations, governance mechanisms, and institutional challenges associated with cross-border deployment. Key emphasis is placed on privacy-preserving strategies, regulatory alignment, and multistakeholder trust frameworks. The study highlights that while technical standards provide a solid base, scalable adoption depends on legal harmonization and institutional integration.

Keywords: decentralized identity, blockchain, verifiable credentials, cross-border interoperability, trust frameworks, privacy, digital governance.

Аннотация

Цифровые системы идентификации на основе блокчейна формируют новое представление о способах управления идентификационными данными в трансграничной среде. Использование децентрализованных идентификаторов, верифицируемых аттестатов и распределённых моделей доверия обеспечивает повышение конфиденциальности, контроль над персональными данными и совместимость между юрисдикциями. В статье рассматриваются архитектурные принципы, механизмы управления и институциональные барьеры, сопровождающие внедрение таких систем. Особое внимание уделено вопросам защиты данных, нормативного соответствия и координации участников. Показано, что устойчивое масштабирование возможно только при условии правового согласования и межинституционального взаимодействия.

Ключевые слова: децентрализованная идентичность, блокчейн, верифицируемые удостоверения, трансграничное взаимодействие, модели доверия, конфиденциальность, цифровое управление.

Introduction

In the digital era, identity verification has become a cornerstone of secure access to services, particularly in international contexts where regulatory frameworks, trust boundaries, and technological infrastructure vary significantly across jurisdictions. Traditional identity management systems-often centralized, fragmented, and non-interoperable-struggle to provide users and

institutions with reliable cross-border verification capabilities. These limitations hamper seamless digital interaction between states, impede regulatory compliance, and expose sensitive identity data to increased risk of compromise.

Blockchain technology has emerged as a transformative foundation for rethinking digital identity systems. Its decentralized structure, cryptographic security, and immutable recordkeeping make it a promising candidate for building trustless, interoperable identity solutions. Blockchain-based identity management systems enable individuals to control their personal data while facilitating secure, verifiable interactions across institutional and national boundaries. The concept of self-sovereign identity (SSI), supported by distributed ledger technology, empowers users to selectively disclose information, revoke permissions, and engage in authentication processes without relying on a single centralized authority.

This study explores the architecture, implementation challenges, and cross-border applicability of blockchain-based digital identity systems. Special attention is given to their potential for enabling interoperability among heterogeneous legal systems, enhancing privacy through cryptographic credential management, and supporting real-time authentication in digital ecosystems spanning multiple states. Through comparative analysis and architectural synthesis, the paper seeks to define best practices for deploying blockchain-enabled identity frameworks that meet both user-centric and institutional requirements in international digital interactions.

Main part. Architectural foundations of blockchain-based identity systems

Designing a digital identity system based on blockchain requires a careful balance between decentralization, data privacy, verifiability, and compliance with legal requirements. Unlike conventional identity infrastructures, which rely on centralized authorities to issue, store, and verify credentials, blockchain-based architectures distribute trust across a network of nodes, each maintaining a synchronized ledger of identity-related events. This paradigm shift supports the implementation of self-sovereign identity, wherein individuals possess full control over their digital credentials and disclose only what is necessary for each transaction.

Core components of such systems include identity issuers, holders, and verifiers, often coordinated through smart contracts that govern credential lifecycle, access permissions, and revocation logic [1]. The verifiable credential model, standardized by the W3C, serves as a foundation for cryptographically signed attestations that can be stored off-chain while being anchored to a blockchain for integrity verification. Interactions between actors are facilitated via decentralized identifiers (DIDs), which function as resolvable, non-correlatable references to identity data. These identifiers are key to preserving user privacy while ensuring traceable, auditable interactions in cross-border contexts.

Figure 1 illustrates a generalized architecture of a blockchain-based identity management system, highlighting the roles of each actor, credential flows, and the interaction between on-chain and off-chain components.

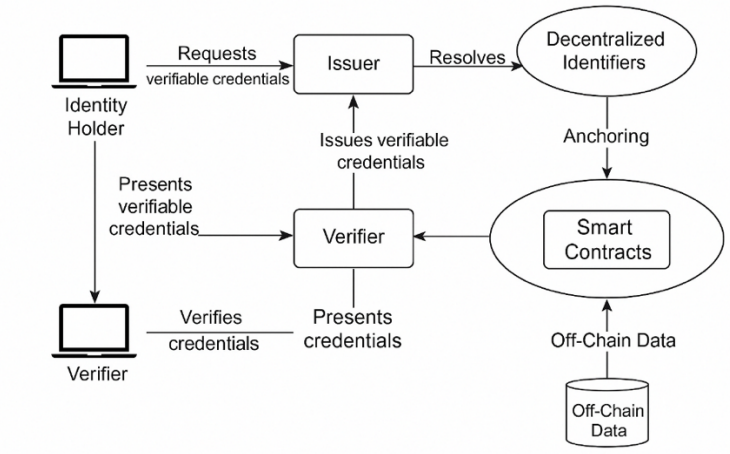


Figure 1. Blockchain-based identity management system architecture

The figure outlines the core components of a decentralized identity ecosystem built on blockchain. It shows how the identity holder interacts with issuers and verifiers through verifiable

credentials and decentralized identifiers, while off-chain data and smart contracts support credential validation and trust management. This visual structure highlights the modularity and autonomy of actors, underscoring the system's capacity to function across borders without relying on centralized control [2].

To ensure interoperability and trust across jurisdictions, blockchain-based identity architectures must also integrate governance mechanisms that define the roles, responsibilities, and compliance standards for participating entities. These governance frameworks may be implemented through consortia or alliances of organizations from different countries, each operating nodes and collectively managing rules through consensus protocols. Such arrangements allow for scalable federation while maintaining transparent audit trails and consistent identity resolution logic.

A critical design challenge lies in balancing data minimization with identity verifiability. To reduce exposure of personally identifiable information (PII), modern architectures rely on zero-knowledge proofs (ZKPs), selective disclosure mechanisms, and off-chain data vaults. Credential metadata or hash digests may be recorded on-chain, while sensitive attributes remain encrypted and accessible only to authorized verifiers. This separation enables strong privacy guarantees while preserving cryptographic verifiability. Additionally, revocation registries and expiration controls are embedded within the system to prevent misuse of outdated or compromised credentials.

Another essential aspect is the portability of identities across national and technological boundaries [3]. To this end, systems must support common standards such as W3C Verifiable Credentials and DID specifications, as well as interoperability protocols like DIDComm or OpenID for verifiable presentations (OID4VP). These layers allow identity holders to use the same credential in multiple domains-such as travel, healthcare, banking, and education-without duplication or re-verification. In cross-border scenarios, this leads to reduced onboarding time, lower administrative costs, and improved user experience while maintaining institutional assurance levels.

Legal and regulatory considerations in cross-border identity frameworks

The deployment of blockchain-based identity systems across national borders introduces complex legal challenges related to jurisdiction, data sovereignty, and regulatory compliance. Traditional identity verification processes are deeply embedded within national legal frameworks, often requiring adherence to local data protection laws, institutional mandates, and technical certification protocols. In contrast, blockchain architectures operate beyond centralized oversight, raising concerns about accountability, liability, and enforceability in the absence of a single legal anchor [4].

A central issue lies in the classification of digital identity data under regional regulations such as the General Data Protection Regulation (GDPR) in the European Union or the Personal Data Protection Law (PDPL) in jurisdictions across the Middle East and Asia. While blockchain promotes transparency and immutability, these attributes may conflict with legal principles such as the right to be forgotten or data rectification. To resolve this tension, emerging identity systems incorporate privacy-preserving technologies and adopt «off-chain first» design strategies, wherein only non-sensitive references or cryptographic hashes are stored on-chain.

Cross-border use cases further complicate regulatory alignment due to disparities in legal definitions of identity, trust frameworks, and electronic signature recognition. For example, an identity credential recognized in one country may not meet the legal evidentiary standards in another, particularly when the underlying issuer is not part of an approved trust list [5]. To address this fragmentation, interoperability frameworks and legal harmonization efforts-such as the eIDAS 2.0 regulation and the UN model laws on electronic commerce-seek to establish common ground for cross-recognition of decentralized identities. However, their adoption remains uneven and often lags behind technological innovation.

Interoperability and standardization challenges in decentralized identity ecosystems

Interoperability is a fundamental requirement for blockchain-based identity systems, especially in cross-border contexts where infrastructures, legal frameworks, and trust models differ widely. Without shared standards for credential exchange and validation, identity systems remain siloed and fail to deliver on the promise of user-controlled, portable, and verifiable digital identity. Achieving

meaningful interoperability requires consistent support for credential lifecycle management, trust resolution, and technical compatibility across diverse platforms and institutions.

Most implementations rely on established standards that define identity object formats and cryptographic validation mechanisms [6]. These standards enable credentials issued in one system to be accepted and verified in another, even when underlying technologies differ. However, practical interoperability is often hindered by mismatched implementations, inconsistent governance models, and incompatible methods for credential anchoring and resolution across distributed ledgers.

Technical challenges include variation in DID resolution across networks, discrepancies in credential schemas, and fragmented support for secure credential presentation. While VCs may follow the same structural specification, differences in how issuers handle metadata, revocation, and assurance levels lead to verification uncertainty. In addition, trust frameworks that govern identity ecosystems are often locally administered, resulting in divergent credential acceptance policies that complicate transnational recognition.

To mitigate these issues, several initiatives have introduced interoperability profiles, cross-ledger resolution tools, and conformance frameworks. Projects such as DIF, the Trust over IP Foundation, and EBSI contribute to establishing bridges between identity networks and improving semantic alignment [7]. Real-time credential exchange is increasingly facilitated using standardized communication protocols like DIDComm and OID4VP, allowing identity holders and verifiers to interact reliably across system boundaries.

Achieving true interoperability, however, requires more than protocol compliance. Trust interoperability must also be established through legally recognized credential policies, shared governance procedures, and cross-jurisdictional assurance frameworks. Only by combining technical, institutional, and regulatory alignment can blockchain-based identity systems support secure, seamless digital interactions on a global scale.

Privacy and data governance in blockchain identity systems

The protection of personal data in decentralized digital identity ecosystems involves resolving the inherent tension between transparency and confidentiality. This challenge is particularly pronounced in cross-border applications, where jurisdictions impose divergent data protection rules and expectations. Immutable data trails, while beneficial for audit and verification, may contradict privacy principles such as data erasure and minimal disclosure.

To address these concerns, modern identity platforms deploy layered governance and technical mechanisms. These include zero-knowledge proofs, selective disclosure techniques, and off-chain data vaults, which together allow users to prove statements about their identity without disclosing raw data. At the same time, permissioned access rules and decentralized control models ensure that information flows remain traceable but limited in exposure [8].

Figure 2 illustrates the combined application of governance layers and privacy technologies. The flowchart highlights how consent-based access control, encrypted storage, and modular identity components work together to create a privacy-aware, cross-compatible system. By distributing control across users, issuers, and verifiers, and separating data layers from verification logic, these architectures offer scalable privacy protection aligned with global regulatory diversity.

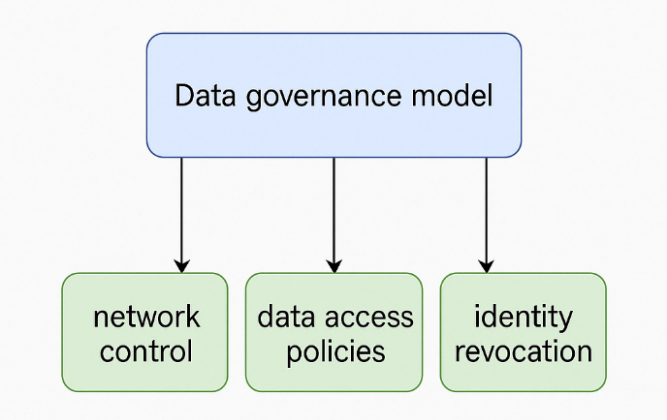


Figure 2. Layered architecture for privacy and data governance in blockchain identity systems

A conceptual figure illustrating the interaction between technical privacy enablers and governance components in decentralized identity platforms. The diagram emphasizes the modular separation of data storage, access control, and verification logic, enabling compliance with diverse privacy regulations through selective disclosure, encrypted off-chain storage, and user-centric consent models [9].

Trust frameworks and institutional adoption barriers

The adoption of blockchain-based digital identity systems at a national and cross-border scale is contingent upon the establishment of formal trust frameworks that define roles, responsibilities, and assurance levels among participating institutions. These frameworks serve as the backbone for evaluating credential validity, ensuring interoperability, and providing mechanisms for dispute resolution. Without such agreements, decentralized identity networks remain fragmented, unable to achieve the consistency and reliability required for high-stakes transactions such as immigration, finance, or cross-border e-health services.

Institutional actors-such as governments, banks, and regulatory authorities-are often cautious in adopting decentralized solutions due to concerns over accountability, legal enforceability, and technological maturity [10]. Traditional identity systems are deeply integrated with legacy databases and centralized verification mechanisms, which makes integration with blockchain-based platforms both technically and administratively complex. Furthermore, questions arise regarding governance authority in decentralized networks: Who defines trust rules? Who manages revocation lists? Who ensures compliance across jurisdictions?

Trust frameworks address these concerns by establishing a layered structure of trust anchors, assurance policies, and credential exchange protocols. These elements allow entities to map their internal trust requirements onto external systems, provided they share aligned verification standards. In cross-border contexts, trust must be both legally recognized and cryptographically verifiable, which presents challenges in aligning national identity laws, digital signature regimes, and data protection policies. While initiatives such as eIDAS 2.0 and the Pan-Canadian Trust Framework offer blueprints, global consensus remains limited.

An additional barrier to institutional adoption is the lack of unified certification standards for decentralized identity components. Many emerging platforms rely on custom smart contracts, proprietary DID registries, and non-audited wallet implementations, which undermine institutional confidence and risk regulatory non-compliance. Without trusted certification bodies, it becomes difficult to assess the security, stability, and auditability of these systems. This leads to slow procurement cycles, pilot stagnation, and siloed deployments that lack scalability.

To overcome these challenges, efforts must focus on multilateral agreements, technical conformance testing, and open governance. Institutions require not only the technological tools to interface with decentralized identity networks but also legal clarity and operational guidance. Pilot programs involving public-private partnerships, sandbox environments, and cross-border trials are essential to build institutional trust, validate models, and support iterative policy formation. In doing so, blockchain-based digital identity systems may evolve from experimental technology into a reliable infrastructure for global identity assurance.

Institutional stakeholders and trust orchestration in decentralized identity systems

The successful deployment of blockchain-based identity infrastructures depends not only on technological soundness but also on the coordinated involvement of diverse institutional stakeholders. These include government agencies, regulatory bodies, financial institutions, educational authorities, and technology providers-all of which play distinct yet interdependent roles in the construction and validation of digital trust ecosystems [11]. Without institutional alignment, decentralized identity networks risk fragmentation, non-recognition, and legal uncertainty.

Each stakeholder brings a unique set of responsibilities to the ecosystem. Governments are often responsible for anchoring foundational identities (such as passports or national IDs), establishing legal trust frameworks, and enforcing data protection standards. Financial institutions and telecom providers frequently act as attribute issuers, validating user identities for Know Your Customer (KYC) and authentication purposes. Regulators define compliance boundaries and technical

standards, while technology vendors provide infrastructure, credential wallets, and integration interfaces. Effective trust orchestration requires formal mechanisms to coordinate these roles and ensure shared adherence to credential assurance policies.

Figure 3 depicts a layered model of institutional participation in blockchain-based identity networks. It illustrates how trust anchors, credential issuers, verifiers, and governance bodies interact through formalized protocols and interoperability agreements. The flow of credentials and assurance metadata is shown as a dynamic exchange, mediated by policy enforcement components and governed through modular trust registries.

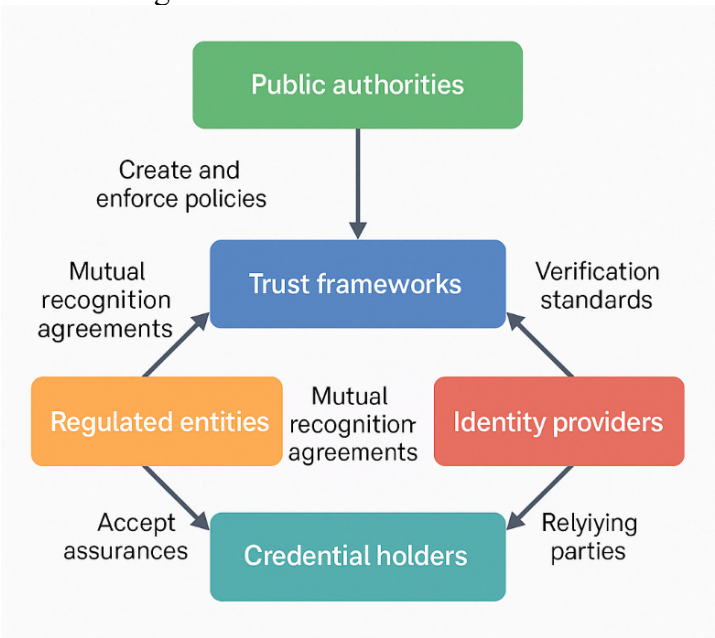


Figure 3. Institutional stakeholders and trust orchestration layers

This structural model highlights the modular and multilateral nature of trust in decentralized environments. Rather than relying on a single central authority, legitimacy is derived from overlapping validations across a federated trust mesh. Stakeholders may operate under different jurisdictions and assurance levels, but by adhering to shared credential formats, policy vocabularies, and certification mechanisms, they can collaborate without sacrificing sovereignty or security. This makes the model particularly suitable for cross-border digital ecosystems, where decentralization is necessary to reconcile diverse institutional mandates while enabling seamless, user-centric identity experiences.

A schematic representation of institutional roles and governance coordination in decentralized digital identity ecosystems. The diagram visualizes how credential issuers, verifiers, regulators, and trust anchors interact through policy-driven exchanges, federated governance structures, and dynamic trust registries to support scalable, cross-border identity assurance.

Conclusion

Blockchain-based digital identity systems offer a transformative alternative to traditional centralized models, particularly in contexts requiring secure, verifiable, and portable identity credentials across national borders. Through decentralized trust mechanisms, cryptographic verification, and user-centric control, these systems promise to resolve long-standing challenges related to data fragmentation, privacy risks, and cross-jurisdictional interoperability.

This study has examined the architectural components, governance principles, and institutional dynamics that underpin the deployment of decentralized identity networks. Particular attention was given to privacy-preserving technologies, legal alignment, interoperability standards, and multilateral trust frameworks. The analysis shows that while technical standards such as DIDs and verifiable credentials provide a solid foundation, the success of cross-border adoption ultimately depends on regulatory convergence, institutional collaboration, and robust certification ecosystems.

Looking forward, the maturation of governance models, integration with public sector infrastructure, and participation of international consortia will be essential for scaling blockchain-

based identity solutions globally. By aligning technology, law, and policy, decentralized identity systems can become a core enabler of trusted digital interaction in an increasingly interconnected world.

References

1. Supangkat S.H., Firmansyah H.S., Rizkia I., Kinanda R. Challenges in Implementing Cross-Border Digital Identity Systems for Global Public Infrastructure: A Comprehensive Analysis // IEEE Access. 2025.
2. El Haddouti S., Ouaguid A., Ech-Cherif E.l., Kettani M.D. Fedidchain: An innovative blockchain-enabled framework for cross-border interoperability and trust management in identity federation systems // Journal of Network and Systems Management. 2023. Vol. 31. No. 2. P. 42.
3. Kulothungan V. A blockchain-enabled approach to cross-border compliance and trust // 2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA). IEEE. 2024. P. 446-454.
4. Zhou F., Liu Y. Blockchain-enabled cross-border e-commerce supply chain management: A bibliometric systematic review // Sustainability. 2022. Vol. 14. No. 23. P. 15918.
5. Broshka E., Jahankhani H. Evaluating the Importance of SSI-Blockchain Digital Identity // Navigating the Intersection of Artificial Intelligence, Security, and Ethical Governance. 2024. P. 87.
6. Chen J., Lu F., Liu Y., Peng S., Cai Z., Mo F. Cross trust: A decentralized MA-ABE mechanism for cross-border identity authentication // International Journal of Critical Infrastructure Protection. 2024. Vol. 44. P. 100661.
7. Wang F., Gai Y., Zhang H. Blockchain user digital identity big data and information security process protection based on network trust // Journal of King Saud University-Computer and Information Sciences. 2024. Vol. 36. No. 4. P. 102031.
8. Buttar A.M., Shahid M.A., Arshad M.N., Akbar M.A. Decentralized Identity Management Using Blockchain Technology: Challenges and Solutions // Blockchain Transformations: Navigating the Decentralized Protocols Era. Cham: Springer Nature Switzerland. 2024. P. 131-166.
9. Eyo-Udo N.L., Agho M.O., Onukwulu E.C., Sule A.K., Azubuike C., Nigeria L., Nigeria P. Advances in Blockchain Solutions for Secure and Efficient Cross-Border Payment Systems // International Journal of Research and Innovation in Applied Science. 2024. Vol. 9. No. 12. P. 536-563.
10. Dudak A., Israfilov A. Application of blockchain in IT infrastructure management: new opportunities for security assurance // German International Journal of Modern Science. 2024. № 92. P. 103-107.
11. Saranya S., Manikandan K., Nagaraju J., Nagendiran S., Geetha B.T. Blockchain-Based Identity Management: Enhancing Privacy and Security in Digital Identity System // 2024 7th International Conference on Contemporary Computing and Informatics (IC3I). IEEE. 2024. Vol. 7. P. 1620-1625.

UDC 004.89

AUTONOMOUS INTELLIGENT AGENTS IN DECISION SUPPORT SYSTEMS FOR CRITICAL INFRASTRUCTURE

Gvilava N.T.
postgraduate student, Ilia State University (Tbilisi, Georgia)

АВТОНОМНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ АГЕНТЫ В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ КРИТИЧЕСКИ ВАЖНОЙ ИНФРАСТРУКТУРЫ

Гвилава Н.Т.
аспирант, Илийский государственный университет (Тбилиси, Грузия)

Abstract

The integration of autonomous intelligent agents into decision support systems enhances the capacity of critical infrastructure to operate reliably in dynamic and uncertain environments. These agents provide essential functions such as real-time monitoring, adaptive response, distributed coordination, and learning-based optimization. The article examines the functional architecture, communication patterns, and implementation strategies of autonomous agents across different infrastructure domains. Through architectural modeling, decision logic representation, and analysis of scalability and fault tolerance, the study demonstrates how agents support resilient, decentralized decision-making. Particular attention is given to layered integration, enabling agents to function effectively at sensing, control, coordination, and strategic levels. The findings contribute to the development of intelligent, explainable, and adaptable decision support frameworks for infrastructure resilience.

Keywords: autonomous agents, critical infrastructure, decision support systems, adaptive control, distributed coordination, fault tolerance, intelligent monitoring.

Аннотация

Интеграция автономных интеллектуальных агентов в системы поддержки принятия решений повышает устойчивость критической инфраструктуры к неопределённости и внешним воздействиям. Такие агенты выполняют ключевые функции: мониторинг в реальном времени, адаптивное реагирование, распределённую координацию и оптимизацию на основе обучения. В статье рассматриваются функциональная архитектура, коммуникационные схемы и стратегии внедрения агентов в различных инфраструктурных контекстах. Путём анализа архитектурных моделей, логики принятия решений и механизмов масштабируемости и отказоустойчивости показано, как агентные системы обеспечивают децентрализованное и устойчивое управление. Особое внимание уделено многоуровневой интеграции агентов - от уровня сенсоров и контроля до координации и стратегического планирования. Представленные результаты способствуют формированию интеллектуальных и адаптивных платформ для повышения надёжности инфраструктурных систем.

Ключевые слова: автономные агенты, критическая инфраструктура, системы поддержки принятия решений, адаптивное управление, распределённая координация, отказоустойчивость, интеллектуальный мониторинг.

Introduction

Ensuring the operational stability and security of critical infrastructure requires the implementation of advanced control systems capable of autonomous decision-making under high uncertainty. Traditional decision support systems (DSS) often rely on static algorithms and predefined scenarios, which significantly limits their adaptability to complex, dynamic environments. The increasing complexity of modern infrastructure-encompassing energy, transport, healthcare, and communication sectors-demands the integration of intelligent components capable of real-time data processing, context-aware reasoning, and proactive response generation. In this context, the use of autonomous intelligent agents (AIA) emerges as a promising paradigm for enhancing the responsiveness, resilience, and adaptability of DSS in high-stakes operational domains.

Autonomous intelligent agents represent a class of software entities equipped with autonomous behavior, learning mechanisms, and decision-making capabilities based on artificial intelligence (AI) algorithms. These agents are designed to perceive environmental changes, evaluate potential outcomes, and execute actions with minimal or no human intervention. When integrated into DSS for critical infrastructure, AIA can facilitate continuous system monitoring, predictive analysis, and adaptive control in response to emerging threats or system deviations. Their distributed nature also allows for scalable and decentralized decision-making, which is essential in large, interconnected infrastructures where centralized control becomes a bottleneck or a point of vulnerability.

The objective of this study is to analyze the functional role, architectural models, and implementation challenges of autonomous intelligent agents in decision support systems serving critical infrastructure. The article explores key design principles, compares existing implementations, and evaluates their performance in terms of adaptability, fault tolerance, and real-time responsiveness. The study is grounded in recent advancements in multi-agent systems, machine learning, and cyber-physical infrastructure management, aiming to contribute to the development of resilient, self-organizing, and intelligent control frameworks capable of operating reliably under uncertainty and stress conditions.

Main part

Functional architecture of autonomous intelligent agents in decision support systems

The integration of autonomous intelligent agents into decision support systems requires a well-defined architectural framework that supports autonomy, communication, adaptability, and system-wide coordination. The architecture must be capable of processing heterogeneous data flows, interpreting complex operational contexts, and initiating timely and optimal actions without direct human input. Unlike traditional centralized models, which are limited in scalability and responsiveness, modern AIA-based DSS rely on modular, distributed, and often hybrid architectures combining rule-based reasoning with machine learning (ML) components.

A typical functional architecture of an AIA in a DSS environment includes several interrelated layers: the perception layer, responsible for data acquisition and preprocessing; the reasoning and inference layer, where contextual analysis and decision-making occur; the learning layer, enabling adaptation based on historical performance and environmental feedback; and the actuation layer, which interfaces with external systems to execute actions. Each agent in the system operates semi-independently, yet remains synchronized through a shared knowledge base and communication protocols. This architecture facilitates scalability and robustness, enabling critical infrastructure systems to handle both routine operations and unexpected disruptions [1].

Figure 1 presents a generalized architectural model of autonomous intelligent agents embedded in a DSS for critical infrastructure. The diagram illustrates the flow of information between layers, the interaction between agents, and the feedback mechanisms necessary for learning and adaptation.

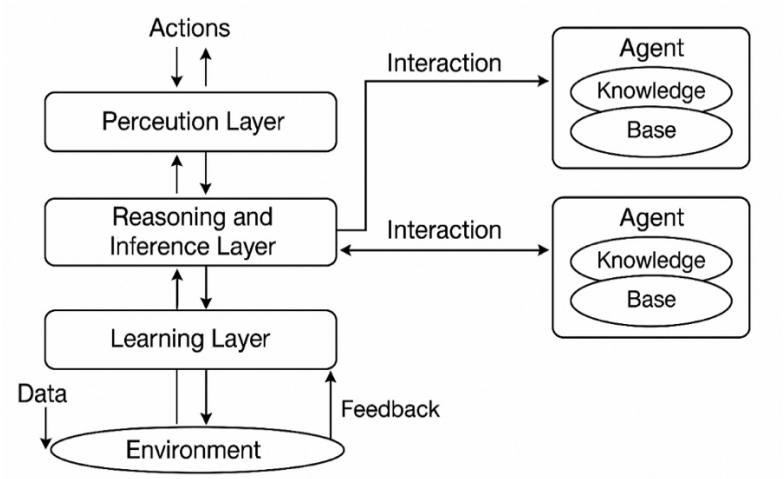


Figure 1. Functional architecture of autonomous intelligent agents in a decision support system

Figure 1 clearly outlines the hierarchical structure and data flow mechanisms that underpin agent-based decision support systems. By enabling autonomous agents to continuously interact with their environment and adjust their behavior through feedback and internal learning, the architecture supports scalable, resilient, and adaptive operations-critical for maintaining functional integrity in complex infrastructure systems.

Key functional capabilities of autonomous agents in infrastructure-level decision processes

The implementation of autonomous agents in infrastructure-focused DSS relies on their ability to execute a specific set of functional capabilities critical for real-time system resilience. Among these capabilities, situational awareness holds central importance, enabling agents to interpret sensor inputs, detect deviations, and correlate internal events with broader operational contexts [2]. Through continuous perception and analysis, agents contribute to early anomaly detection and dynamic adaptation in high-risk environments. Their ability to autonomously manage this complexity makes them suitable for infrastructures where human response times may be insufficient or error-prone under stress.

Another vital capability is decentralized coordination, which allows agents to operate cooperatively while remaining partially independent. This is particularly relevant in domains such as smart grids, water distribution networks, and transport control systems, where information latency or single-point failures can have cascading consequences. Agents exchange status updates, local forecasts, and decision justifications, enabling the system as a whole to maintain coherence and redundancy. Such coordination requires robust communication protocols, distributed consensus mechanisms, and role-based agent design to prevent conflicts and ensure synchronization across subsystems.

Additionally, adaptive learning is fundamental to improving long-term performance. Agents must go beyond rule-based responses by integrating supervised, unsupervised, or reinforcement learning approaches depending on the scenario. This allows them to refine their decision logic over time, adapting to evolving environmental conditions and threat models. In infrastructure contexts, this capability supports predictive maintenance, traffic flow optimization, load balancing, and other efficiency-driven goals. Ultimately, the integration of these functions allows autonomous agents to extend the intelligence of DSS beyond static models, enabling continuous system improvement and real-time operational assurance.

The effective deployment of these functional capabilities also depends on agents' capacity to manage uncertainty and incomplete information, a common challenge in real-world infrastructure environments. Critical systems frequently operate under conditions where data may be noisy, delayed, or partially unavailable due to sensor malfunctions, network congestion, or cyber incidents. Autonomous agents must therefore employ probabilistic reasoning, fuzzy logic, or belief models to infer system states and support decisions under ambiguity. This uncertainty management is crucial for maintaining safety and functionality when deterministic models prove inadequate [3].

Moreover, the resilience of infrastructure supported by agents depends on their fault tolerance and capacity for graceful degradation. Agents must detect internal failures, isolate compromised nodes, and reallocate tasks among healthy components to preserve core functionality. This is particularly relevant in cyber-physical systems where hardware faults, cyberattacks, or cascading outages can disrupt coordination. Embedded self-checking mechanisms and agent-level redundancy are essential for localizing and mitigating the impact of such disruptions. These features ensure that critical infrastructure can continue operating in a degraded but controlled state until full recovery is possible.

Finally, the integration of agents with human operators must be designed to facilitate trust, transparency, and controllability. While autonomy is a central feature, critical infrastructure still demands human oversight, particularly in cases involving ethical trade-offs or emergency override. Agents must be able to explain their decisions, communicate alerts effectively, and accept operator input when necessary. Human-in-the-loop mechanisms, explainable AI components, and supervisory control interfaces help bridge the gap between automated intelligence and operational responsibility. These interfaces are indispensable in sectors where regulatory compliance and public accountability are paramount.

Agent-based decision logic and example implementation in infrastructure context

Autonomous agents must operate on flexible decision logic capable of reacting to changes in the environment, system state, and interaction with other agents. This logic can be encoded through rule-based structures, decision trees, or learning-enhanced control flows. In infrastructure systems, such logic governs actions like fault isolation, resource reallocation, emergency response, and risk prioritization [4]. To illustrate the basic implementation of this logic, a simplified pseudocode representation is provided below. It models an agent that monitors critical metrics (e.g., temperature, pressure, load) and responds based on a threshold- and state-aware decision framework.

```
class Infrastructureagent:
    def __init__(self, id, threshold_map, critical_zones):
        self.id = id
        self.threshold_map = threshold_map
        self.critical_zones = critical_zones
        self.status = "normal"

    def sense_environment(self, sensor_data):
        self.metrics = sensor_data
        self.analyze_status()

    def analyze_status(self):
        for metric, value in self.metrics.items():
            threshold = self.threshold_map.get(metric, None)
            if threshold and value > threshold:
                self.status = "alert"
                self.respond(metric, value)

    def respond(self, metric, value):
        if metric in self.critical_zones:
            self.status = "critical"
            self.trigger_emergency_protocol(metric, value)
        else:
            self.status = "warning"
            self.issue_warning(metric, value)

    def trigger_emergency_protocol(self, metric, value):
        print(f"[{self.id}] CRITICAL: {metric} = {value}. Emergency protocol initiated.")

    def issue_warning(self, metric, value):
```

```
        print(f"[{self.id}] WARNING: {metric} = {value}. Monitoring closely.")

# Example usage
agent = InfrastructureAgent(
    id="Node-7",
    threshold_map={"temperature": 75, "pressure": 120},
    critical_zones=["temperature"]
)

sensor_input = {"temperature": 82, "pressure": 110}
agent.sense_environment(sensor_input)
```

The presented example demonstrates a basic agent capable of monitoring key parameters, identifying threshold breaches, and executing context-sensitive responses. In practice, such logic is extended with probabilistic inference, machine learning classifiers, and multi-agent communication layers. However, even in this simplified form, the model illustrates essential patterns: autonomous sensing, condition-based classification, and action generation. These form the foundation for scalable, adaptive control in real-time infrastructure environments [5].

Communication and coordination patterns in multi-agent infrastructure systems

The effectiveness of autonomous agents in critical infrastructure depends not only on individual capabilities but also on how agents communicate, coordinate, and make distributed decisions as a collective system. Multi-agent communication architectures must enable real-time information exchange, synchronization of shared objectives, and resolution of conflicting actions. These systems are particularly relevant in energy networks, urban mobility grids, and water distribution systems, where local conditions affect global stability. Coordination mechanisms are typically based on consensus protocols, role-based agent hierarchies, or behavior-driven negotiation models.

Agents often utilize a combination of broadcast, peer-to-peer, and hierarchical messaging depending on system topology and latency constraints. For example, in energy infrastructure, grid balancing agents exchange load information, negotiate load shedding, or reroute flows dynamically. Failure to coordinate can lead to cascading faults or inefficient resource usage. Therefore, effective agent communication must be fault-tolerant, low-latency, and bandwidth-aware, particularly in infrastructure where communication delays can compromise safety or compliance [6].

Figure 2 illustrates typical communication and coordination patterns in a distributed multi-agent system embedded within a decision support framework for critical infrastructure. The diagram highlights how agents form clusters, route messages, and align decisions through shared protocols and local autonomy.

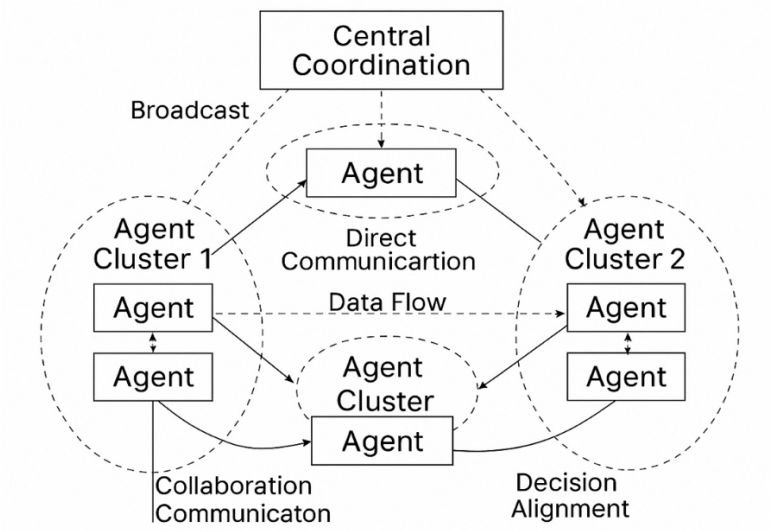


Figure 2. Communication and coordination patterns in multi-agent infrastructure systems

Figure 2 illustrates the multi-layered structure of agent communication, where clusters operate semi-independently while maintaining coordination through central broadcasting and inter-cluster

synchronization. This architecture enhances the system's ability to respond to localized disruptions while preserving global stability, making it particularly suitable for large-scale, heterogeneous infrastructure environments.

Scalability and fault tolerance in agent-based infrastructure systems

The deployment of autonomous agents in large-scale infrastructure environments necessitates architectures that are inherently scalable and fault-tolerant. As systems grow in complexity-spanning geographically distributed assets, heterogeneous technologies, and multi-domain interactions-the underlying agent framework must support seamless expansion and resilience to localized or systemic failures. Scalability in this context implies that agents can be added or reconfigured dynamically without compromising the performance or consistency of the overall decision-making process.

A key enabler of scalability is the modular design of agent clusters, each responsible for a specific functional or geographic domain [7]. These clusters can operate semi-independently while adhering to shared communication protocols and global objectives. Distributed control, as opposed to centralized orchestration, reduces bottlenecks and increases parallelism in computational processes. Moreover, agents within these clusters can autonomously negotiate role reassignments and data handovers, enabling real-time adaptation to changing operational loads or physical topology.

Fault tolerance is achieved through redundancy, replication, and local recovery mechanisms. Agents must be capable of detecting malfunctioning peers, redistributing tasks, and maintaining a degraded yet operational service state. In mission-critical systems such as transportation control or power grid management, such continuity is essential to avoid cascading disruptions. Techniques like agent health monitoring, failover procedures, and consensus-based state replication help maintain stability under adverse conditions. Importantly, these mechanisms must be lightweight enough to operate within the resource constraints typical of embedded systems and edge devices commonly used in infrastructure.

Integration levels of autonomous agents across infrastructure domains

The application of autonomous agents across infrastructure sectors requires domain-specific adaptation strategies, as the nature of decision processes, latency constraints, and safety requirements varies significantly between contexts [8]. In sectors such as power distribution, agents must operate under real-time constraints with strict reliability guarantees, whereas in logistics or urban mobility systems, responsiveness may be balanced with optimization goals. The integration of agents into existing infrastructure therefore occurs across multiple functional levels: sensing, local control, coordination, and strategic management.

Figure 3 shows the hierarchical integration of autonomous agents into four key operational layers of critical infrastructure systems. The structure enables scalable deployment of agent functionality - starting from sensing and anomaly detection, progressing through local control and coordination, and culminating in strategic management [9]. Each level addresses distinct operational challenges related to autonomy, decision latency, and data complexity, providing a framework for positioning agent roles in accordance with system-critical requirements.

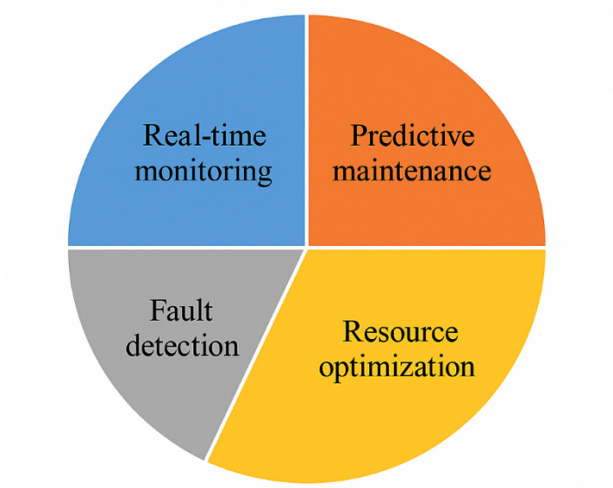


Figure 3. Integration levels of autonomous agents across infrastructure domains

The layered structure reflects the progressive deepening of agent functionality, from basic sensing and anomaly detection to strategic-level reasoning and coordination. This hierarchy ensures that autonomous agents can be effectively positioned within the appropriate operational context, allowing for both localized responsiveness and system-wide optimization.

Conclusion

The growing complexity and interdependence of critical infrastructure systems necessitate the use of decision support frameworks capable of operating reliably under uncertainty, stress, and scale. Autonomous intelligent agents, when integrated into such systems, enable real-time responsiveness, decentralized control, and adaptive learning-capabilities that are essential for ensuring operational continuity and resilience. Their layered integration, from anomaly detection to strategic coordination, allows for fine-grained deployment tailored to the specific demands of various infrastructure domains.

The analysis has demonstrated that functional architectures built around perception, reasoning, learning, and action enable agents to autonomously detect, interpret, and respond to evolving operational contexts. Scalability and fault tolerance are achieved through modular clustering, distributed coordination, and self-recovery mechanisms, while effective communication protocols support collaboration across agent groups. These features collectively contribute to a more robust and intelligent decision support environment capable of enhancing the reliability and efficiency of infrastructure systems.

The continued advancement of agent-based approaches, supported by explainable decision models, human-in-the-loop integration, and domain-specific adaptation, will be instrumental in shaping the future of infrastructure resilience. As critical systems evolve in complexity and exposure, autonomous intelligent agents offer a scalable, adaptive, and intelligent solution for managing risk, optimizing performance, and supporting informed, real-time decision-making at all operational levels.

References

1. Guo Z., Meng D., Chakraborty C., Fan X.R., Bhardwaj A., Yu K. Autonomous behavioral decision for vehicular agents based on cyber-physical social intelligence // IEEE Transactions on Computational Social Systems. 2022. Vol. 10. No. 4. P. 2111-2122.
2. Sharma S., Islam N., Singh G., Dhir A. Why do retail customers adopt artificial intelligence (AI) based autonomous decision-making systems? // IEEE Transactions on Engineering Management. 2022. Vol. 71. P. 1846-1861.
3. Cheng Y., Zhang C., Zhang Z., Meng X., Hong S., Li W., He X. Exploring large language model based intelligent agents: Definitions, methods, and prospects // arXiv preprint arXiv:2401.03428. 2024.
4. Bolgov S. Automation of business processes using integration platforms and backend technologies // International Research Journal of Modernization in Engineering Technology and Science. 2024. Vol. 6(12). P. 3847-3851.
5. Li S., Shu K., Chen C., Cao D. Planning and decision-making for connected autonomous vehicles at road intersections: A review // Chinese Journal of Mechanical Engineering. 2021. Vol. 34. P. 1-18.
6. Aksjonov A., Kyrki V. Rule-based decision-making system for autonomous vehicles at intersections with mixed traffic environment // 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE. 2021. P. 660-666.
7. Mills D., Pudney S., Pevcin P., Dvorak J. Evidence-based public policy decision-making in smart cities: Does extant theory support achievement of city sustainability objectives? // Sustainability. 2021. Vol. 14. No. 1. P. 3.
8. Caballero W.N., Rios Insua D., Banks D. Decision support issues in automated driving systems // International Transactions in Operational Research. 2023. Vol. 30. No. 3. P. 1216-1244.
9. Goriparthi R.G. Reinforcement Learning in IoT: Enhancing Smart Device Autonomy through AI // Computing. 2024. Vol. 2. P. 89-109.

УДК 004.7

ОПТИМИЗАЦИЯ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В СИСТЕМАХ ПЕРИФЕРИЙНЫХ ВЫЧИСЛЕНИЙ ДЛЯ ПРИЛОЖЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ

Забелин Р.Т.
бакалавр, Московский государственный технический университет им. Н.Э. Баумана (Москва, Россия)

Левшиц В.Э.
бакалавр, Московский государственный технический университет им. Н.Э. Баумана (Москва, Россия)

Аннотация

Оптимизация распределения ресурсов в периферийных вычислительных системах требует учёта специфики приложений реального времени, включая жёсткие ограничения по задержке, устойчивость к перегрузкам и необходимость локальной обработки данных. Представлен анализ требований к архитектурам edge-сетей, классифицированы прикладные сценарии и сформулированы критерии эффективности распределения. Выделены алгоритмические подходы к назначению задач и проанализировано влияние сетевых параметров на стратегию управления. Показана значимость контекстно-адаптивных и гибридных методов, обеспечивающих баланс между производительностью, отказоустойчивостью и масштабируемостью систем edge-компьютинга.

Ключевые слова: периферийные вычисления, распределение ресурсов, приложения реального времени, edge-сети, адаптивные алгоритмы, задержка, балансировка нагрузки.

OPTIMIZATION OF RESOURCE ALLOCATION IN EDGE COMPUTING SYSTEMS FOR REAL TIME APPLICATIONS

Zabelin R.T.
bachelor's degree, Bauman Moscow state technical university (Moscow, Russia)

Levshits V.E.
bachelor's degree, Bauman Moscow state technical university (Moscow, Russia)

Abstract

Resource allocation in edge computing systems must account for the strict timing, reliability, and locality requirements of real-time applications. This study provides a structured analysis of edge architectures, application classes, and optimization criteria for task distribution. Several algorithmic strategies are outlined, including priority-based, delay-aware, and predictive methods, with particular attention to network dynamics and heterogeneity. Emphasis is placed on the role of adaptive and hybrid models that enhance resilience and performance in dynamic computing environments.

Keywords: edge computing, resource allocation, real-time applications, task scheduling, adaptive algorithms, latency, load balancing.

Введение

Системы периферийных вычислений (edge computing) становятся ключевым элементом современной распределённой цифровой инфраструктуры, особенно в приложениях, предъявляющих жёсткие требования к задержке, надёжности и локальной обработке данных. В отличие от традиционных облачных архитектур, edge-вычисления обеспечивают выполнение вычислительных задач вблизи источников данных - на уровне сенсоров, мобильных устройств и локальных шлюзов, что значительно снижает время отклика и уменьшает нагрузку на центральные узлы. Однако эффективное функционирование таких систем требует гибких и динамически адаптируемых механизмов управления вычислительными и сетевыми ресурсами.

Особенно остро проблема распределения ресурсов встает в приложениях реального времени, таких как автономное вождение, промышленная автоматизация, дистанционное медицинское наблюдение и тактильный интернет. Здесь даже незначительная задержка или перегрузка одного из узлов может привести к деградации качества обслуживания, нарушению безопасности или полной недоступности критически важной функции. В условиях высокой изменчивости нагрузки, разнообразия вычислительных профилей и ограниченности ресурсов на периферии традиционные алгоритмы управления становятся неэффективными или избыточно затратными.

Цель настоящего исследования - разработка и обоснование подходов к оптимизации распределения ресурсов в периферийных вычислительных системах с учётом требований реального времени. В рамках работы рассматриваются существующие архитектурные модели edge-обработки, классифицируются типовые сценарии приложений, анализируются критерии качества распределения ресурсов и предлагаются алгоритмические решения, направленные на достижение баланса между вычислительной эффективностью, задержкой и устойчивостью системы в условиях ограниченного контекста и высокой динамики среды.

Основная часть

Классификация приложений реального времени и особенности нагрузки на системы edge-вычислений

Оптимизация распределения ресурсов в системах периферийных вычислений невозможна без понимания характера задач, для которых эти системы предназначены. Приложения реального времени существенно различаются по интенсивности трафика, критичности к задержкам, устойчивости к перегрузкам и необходимому уровню обработки данных «на месте». Эти параметры напрямую влияют на стратегию планирования и миграции задач, выбор архитектурных шаблонов и алгоритмов управления ресурсами [1].

В условиях сетей следующего поколения (5G/6G), где периферийные вычисления используются для обслуживания гетерогенных и пространственно распределённых устройств, становится актуальной задача классификации прикладных сценариев по критериям нагрузки и требований к качеству обслуживания (QoS). Такой подход позволяет не только унифицировать принципы архитектурного проектирования, но и формализовать параметры, которые должны учитываться при динамическом распределении вычислений, хранения и пропускной способности на уровне edge-узлов [2].

Таблица 1 содержит обобщённую классификацию типовых приложений реального времени, их характеристики и ключевые требования к системам периферийных вычислений, необходимых для их эффективной поддержки.

Таблица 1

Характеристики приложений реального времени и требования к системам периферийных вычислений

Класс приложения	Характер нагрузки	Ключевые требования
Автономное вождение	Высокочастотная обработка потоков от сенсоров	Максимально допустимая задержка < 10 мс
Класс приложения	**Характер нагрузки**	**Ключевые требования**

Промышленный контроль	Циклические сигналы управления и телеметрия	Гарантированная предсказуемость и отказоустойчивость
Дистанционная хирургия	Низкая задержка, высокая точность	Стабильность отклика, медицинская точность
Интернет вещей в умных зданиях	Нерегулярные события, высокая вариативность	Энергоэффективность и адаптивность к событиям
Видеонаблюдение в реальном времени	Потоковое видео, требующее компрессии и предобработки	Обработка больших объёмов видео в реальном времени

Показанная классификация подчёркивает необходимость контекстно-зависимого подхода к распределению ресурсов. Вычислительная архитектура должна адаптироваться не только к объёму входящих задач, но и к специфике конкретного приложения, учитывая особенности сетевого взаимодействия, предсказуемость нагрузки и допустимые границы задержек. Это позволяет проектировать отказоустойчивые, энергоэффективные и масштабируемые решения для критически важных сценариев реального времени.

Критерии оптимизации распределения ресурсов в edge-среде

Для эффективного распределения ресурсов в системах периферийных вычислений необходимо учитывать множество факторов, зависящих как от внутренних параметров системы, так и от внешних условий эксплуатации. Универсальные подходы, ориентированные лишь на один критерий (например, минимизацию задержки), не всегда обеспечивают сбалансированную работу в условиях высокой нагрузки, энергозависимости и нестабильности каналов [3]. В этой связи возникает необходимость в формализации ключевых критериев оптимизации, которые могут использоваться при проектировании систем управления ресурсами.

Критерии выбора включают как классические метрики - задержка, пропускная способность, энергоэффективность, - так и более сложные показатели, такие как адаптивность к изменяющимся условиям и устойчивость к отказам. Их значимость может существенно различаться в зависимости от типа приложения: для телемедицинской диагностики первична надёжность, для систем реального времени важна задержка, для IoT-контроллеров - энергопотребление. Это обуславливает необходимость контекстно-зависимого мультикритериального подхода к управлению ресурсами на периферии.

Таблица 2 систематизирует основные критерии оптимизации, их описание и применимость в различных классах приложений реального времени.

Таблица 2

Критерии оптимизации распределения ресурсов в edge-среде

Критерий оптимизации	**Описание**	**Применимость к приложениям**
Минимизация задержки	Сокращение времени от поступления задачи до завершения её обработки	Критически важные системы управления, AR/VR
Максимизация пропускной способности	Эффективное использование каналов связи и узлов передачи данных	Видеонаблюдение, потоковые данные
Снижение энергопотребления	Продление времени автономной работы устройств	Устройства IoT, умные здания
Обеспечение отказоустойчивости	Поддержание работоспособности при сбоях узлов и каналов	Промышленные сети, автономные машины
Балансировка вычислительной нагрузки	Равномерное распределение задач между узлами по мощности и загрузке	Сценарии с переменной интенсивностью нагрузки
Критерий оптимизации	**Описание**	**Применимость к приложениям**

Адаптация к динамическим условиям	Корректировка стратегии при изменении характеристик среды	Все классы приложений в изменяющейся среде

Представленные критерии служат основой для построения гибких и адаптивных алгоритмов управления ресурсами. В условиях высокой гетерогенности приложений и ограничений на вычислительные мощности на периферии, выбор критериев оптимизации должен быть обусловлен прикладной задачей, профилем нагрузки и требуемым уровнем отказоустойчивости [4]. Эффективные стратегии должны учитывать динамику среды, возможность приоритизации задач и необходимость поддержки согласованности между узлами.

Алгоритмы распределения задач в edge-средах и их характеристика

В системах периферийных вычислений важнейшей задачей является выбор подходящего алгоритма распределения задач между узлами, обладающими разной вычислительной мощностью, доступом к сети и ограничениями по энергии. От эффективности стратегии назначения задач зависит соблюдение временных ограничений, равномерность загрузки компонентов и устойчивость к изменениям среды [5]. Выбор конкретного алгоритма определяется профилем нагрузки, классом приложения и доступностью системных метрик в момент принятия решений.

На практике применяются как простые эвристические методы, так и сложные алгоритмы, основанные на машинном обучении и прогнозировании поведения среды [6]. Некоторые алгоритмы ориентированы на минимизацию задержек, другие - на распределение нагрузки, третьи - на адаптацию к изменениям и обучение на исторических данных. Важно, чтобы выбранный подход обеспечивал не только производительность, но и предсказуемость выполнения задач, особенно в критически важных приложениях.

Таблица 3 обобщает наиболее распространённые алгоритмы распределения задач в edge-средах, описывает их принципы работы и ключевые преимущества.

Таблица 3

Алгоритмы распределения задач в edge-средах и их характеристики

Алгоритм распределения	**Принцип работы**	**Преимущества**
На основе приоритетов (static priority)	Задачи классифицируются по фиксированному приоритету; высокие приоритеты обслуживаются первыми	Простота реализации, предсказуемость выполнения
Распределение по кругу (round-robin)	Задачи распределяются последовательно по доступным узлам без учёта состояния	Равномерность в условиях однородных задач
Минимизация задержки (delay-aware scheduling)	Учитываются задержки между узлами и выбирается оптимальный маршрут	Подходит для систем с ограничениями по времени отклика
Балансировка нагрузки (load balancing)	Распределение задач осуществляется на основе текущей загрузки узлов	Снижает перегрузки, повышает общую производительность
Распределение с предсказанием (predictive allocation)	Прогнозируется будущее состояние нагрузки и ресурсов	Обеспечивает упреждающее распределение, адаптивность к изменениям
Обучаемое распределение (RL-based scheduling)	Используются алгоритмы обучения с подкреплением для динамического выбора узлов	Гибкая адаптация, обучение на опыте, высокая эффективность в нестабильной среде

Выбор алгоритма распределения должен опираться на конкретные цели системы: минимизацию времени отклика, стабильную производительность или адаптацию к условиям. В приложениях реального времени наилучшие результаты часто достигаются при

гибридизации подходов - например, сочетании эвристики и обучаемых моделей, что позволяет достичь компромисса между предсказуемостью, скоростью и адаптивностью [7].

Контекстно-адаптивные подходы к управлению ресурсами в гетерогенных edge-сетях

Современные edge-системы представляют собой высоко гетерогенные вычислительные среды, включающие устройства с различными уровнями производительности, доступом к источникам питания, параметрами сетевого соединения и архитектурными ограничениями. В такой среде статические или универсальные подходы к управлению ресурсами оказываются неэффективными и нередко приводят к локальным перегрузкам, непредсказуемым задержкам или неоптимальному использованию каналов связи [8]. Это особенно критично в приложениях реального времени, где любые отклонения от допустимого окна выполнения могут привести к потере функциональности.

Контекстно-адаптивное управление предполагает, что стратегия распределения ресурсов должна учитывать не только внутренние параметры вычислительных узлов, но и динамику окружающей среды, поведение пользователей и специфику выполняемых задач. Это включает мониторинг текущей загрузки, состояния сети, типов поступающих данных и их критичности [9]. Использование таких параметров позволяет выстраивать более точные и эффективные алгоритмы, которые способны корректировать свои действия в режиме реального времени без вмешательства оператора.

Одним из ключевых компонентов таких подходов является использование моделей предиктивной аналитики и онтологического описания сценариев выполнения задач. Это позволяет формировать не просто реактивную, а проактивную стратегию управления ресурсами, заранее прогнозируя узкие места и адаптируя инфраструктуру под изменяющиеся условия. Дополнительно, внедрение правил приоритизации на основе профилей приложений и уровня важности задач обеспечивает гарантированное обслуживание критически значимых функций даже в условиях перегрузки системы.

Контекстная адаптация также требует распределённого согласования между узлами edge-сети. Это означает, что локальные решения должны быть скоординированы с глобальной политикой, чтобы избежать конфликтов, дублирования вычислений или деградации глобального качества обслуживания. Для реализации такой координации могут применяться децентрализованные алгоритмы консенсуса, синхронизированные политики принятия решений или доверенные агенты, контролирующие состояние соседних узлов.

Таким образом, контекстно-адаптивные подходы представляют собой важнейший вектор развития edge-инфраструктуры [10]. Их применение позволяет обеспечить гибкость, предсказуемость и устойчивость систем в условиях высокой динамичности, ограниченных ресурсов и требований к гарантированному качеству обслуживания, характерных для приложений реального времени.

Влияние параметров сети на стратегию распределения ресурсов

В периферийных вычислительных системах, работающих в реальном времени, качество и стабильность сетевых соединений играют критически важную роль. В отличие от облачных решений, где взаимодействие с инфраструктурой централизовано и относительно предсказуемо, edge-архитектуры характеризуются высокой степенью сетевой фрагментации, наличием беспроводных каналов различной надёжности и постоянными изменениями топологии. Эти факторы существенно усложняют задачу выбора оптимальных путей передачи, миграции задач и синхронизации узлов.

Одним из ключевых параметров, влияющих на распределение ресурсов, является задержка передачи данных. При планировании задач необходимо учитывать не только вычислительные характеристики узлов, но и латентность каналов, через которые происходит передача между источником, вычислительным узлом и конечной точкой. Даже высокопроизводительный узел может оказаться неэффективным, если связан с источником медленным каналом с нестабильной пропускной способностью.

Вторым важным параметром является ширина канала и уровень его загрузки. Периферийные устройства могут использовать каналы общего назначения, подверженные конкуренции за пропускную способность, что делает невозможной надёжную доставку данных при высокой плотности трафика [11]. Для таких случаев необходимо реализовывать адаптивные алгоритмы маршрутизации и перегруппировки задач, способные в реальном времени выбирать не только наименее загруженные узлы, но и узлы, обеспечивающие наилучшую сетевую связность.

Также критичным аспектом является надёжность соединений, особенно в условиях подвижных узлов или нестабильной радиосреды. В таких сценариях ресурсы должны распределяться с учётом возможных обрывов соединения, временной недоступности и необходимости локального буферизации данных. Это требует разработки стратегий с элементами предсказания и резервирования, позволяющих избегать потери данных и поддерживать непрерывность функционирования даже при частичном отказе сети.

Наконец, важным становится учёт географического положения узлов и плотности устройств, особенно в задачах, где критичен физический контекст (например, логистика, транспорт, AR/VR). В таких случаях следует внедрять геопространственные критерии в систему принятия решений, придавая приоритет узлам, находящимся ближе к источникам генерации данных и действия.

Таким образом, параметры сети являются неотъемлемой частью модели распределения ресурсов и требуют включения в оптимизационные алгоритмы наравне с вычислительными и энергетическими характеристиками. Только при комплексном учёте сетевых факторов возможно построение устойчивой, масштабируемой и адаптивной периферийной вычислительной системы, способной стабильно функционировать в реальном времени.

Заключение

В условиях стремительного развития приложений реального времени и повышения требований к скорости отклика, устойчивости и локальной обработке данных, системы периферийных вычислений становятся ключевым компонентом современной вычислительной архитектуры. Однако их высокая гетерогенность, ограниченность ресурсов и нестабильность сетевого окружения предъявляют особые требования к механизмам управления распределением задач и ресурсов.

В рамках настоящей работы проведён структурный анализ основных классов приложений реального времени, обоснованы критерии оптимизации распределения ресурсов, рассмотрены алгоритмические подходы к управлению нагрузкой, а также выделены факторы, влияющие на эффективность в условиях динамичной и неопределённой среды. Показано, что для достижения требуемого уровня качества обслуживания необходимо использовать гибридные, контекстно-адаптивные методы, включающие как эвристики, так и алгоритмы машинного обучения, работающие в тесной связке с мониторингом текущих параметров системы.

Оптимизация распределения ресурсов в edge-сетях требует комплексного подхода, включающего учёт вычислительных, энергетических и сетевых параметров, а также адаптацию к профилю приложений и изменяющимся условиям эксплуатации. Дальнейшие исследования могут быть направлены на разработку предиктивных моделей и самообучающихся систем, способных в реальном времени реагировать на изменения нагрузки и архитектурные перестройки инфраструктуры.

Список литературы

1. Куприяновский В.П., Намиот Д.Е., Покусаев О.Н. Периферийные вычисления для современной мобильности // International Journal of Open Information Technologies. 2025. Т. 13. № 2. С. 94-104.
2. Бердиназарова А., Кулиев Э., Акыев А. Вычисления на периферии: роль edge computing в современных сетях // Вестник науки. 2024. Т. 3. № 9 (78). С. 342-345.

3. Liu J., Li C., Luo Y. Efficient resource allocation for IoT applications in mobile edge computing via dynamic request scheduling optimization // Expert Systems with Applications. 2024. Vol. 255. P. 124716.
4. Alfakih T., Hassan M.M., Al-Razgan M. Multi-objective accelerated particle swarm optimization with dynamic programing technique for resource allocation in mobile edge computing // IEEE Access. 2021. Vol. 9. P. 167503-167520.
5. Haibeh L.A., Yagoub M.C.E., Jarray A. A survey on mobile edge computing infrastructure: Design, resource management, and optimization approaches // IEEE Access. 2022. Vol. 10. P. 27591-27610.
6. Baburao D., Pavankumar T., Prabhu C.S.R. Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method // Applied Nanoscience. 2023. Vol. 13. No. 2. P. 1045-1054.
7. Abouaomar A., Cherkaoui S., Mlika Z., Kobbane A. Resource provisioning in edge computing for latency-sensitive applications // IEEE Internet of Things Journal. 2021. Vol. 8. No. 14. P. 11088-11099.
8. Bahreini T., Badri H., Grosu D. Mechanisms for resource allocation and pricing in mobile edge computing systems // IEEE Transactions on Parallel and Distributed Systems. 2021. Vol. 33. No. 3. P. 667-682.
9. Smirnov A. Monitoring and logging in distributed systems: application of OpenTelemetry and the ELK stack // Universum: technical sciences : electronic scientific journal. 2025. No. 3(132). P. 30-33.
10. Djigal H., Xu J., Liu L., Zhang Y. Machine and deep learning for resource allocation in multi-access edge computing: A survey // IEEE Communications Surveys & Tutorials. 2022. Vol. 24. No. 4. P. 2449-2494.
11. Tuli S., Mirhakimi F., Pallewatta S., Zawad S., Casale G., Javadi B., Jennings N.R. AI augmented Edge and Fog computing: Trends and challenges // Journal of Network and Computer Applications. 2023. Vol. 216. P. 103648.

UDC 004.9

APPLICATION OF BIG DATA AND DEEP LEARNING FOR FAILURE PREDICTION IN POWER GRIDS

Aitkalieva A.M.
bachelor's degree, Al-Farabi Kazakh National University (Almaty, Kazakhstan)

Zhumabaev O.E.
bachelor's degree, Al-Farabi Kazakh National University (Almaty, Kazakhstan)

ПРИМЕНЕНИЕ БОЛЬШИХ ДАННЫХ И ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ СБОЕВ В ЭНЕРГОСЕТЯХ

Айткалиева А.М.
бакалавр, Казахский национальный университет имени аль-Фараби (Алматы, Казахстан)

Жумабаев О.Э.
бакалавр, Казахский национальный университет имени аль-Фараби (Алматы, Казахстан)

Abstract

Modern power grids generate large-scale, heterogeneous data that require advanced analytical approaches for effective failure prediction and risk mitigation. This article explores the integration of Big Data technologies and deep learning models to enable predictive analytics across critical power infrastructure. A detailed analysis of model architectures-including LSTM, GRU, CNN, and Transformer-is provided, with comparisons based on temporal modeling capabilities, computational efficiency, noise tolerance, and real-time applicability. The study further examines implementation scenarios, system integration challenges, and reliability considerations. Prototypes and evaluation metrics are discussed to support practical adoption. The findings highlight the importance of designing adaptive, explainable, and scalable solutions that align with the complexity and safety demands of real-world grid environments.

Keywords: failure prediction, deep learning, power grid, Big Data, LSTM, GRU, Transformer, smart grid monitoring, anomaly detection, real-time analytics.

Аннотация

Современные энергетические системы генерируют большие объёмы разнородных данных, требующих применения продвинутых аналитических методов для прогнозирования отказов и управления рисками. В статье рассматривается применение технологий обработки больших данных и моделей глубокого обучения для предсказательной аналитики в критически важной инфраструктуре. Проведён сравнительный анализ архитектур моделей (LSTM, GRU, CNN, Transformer) по критериям временного моделирования, вычислительной эффективности, устойчивости к шуму и применимости в реальном времени. Освещены сценарии внедрения, проблемы интеграции и надёжности. Представлены прототипы и метрики оценки, подчёркивающие необходимость адаптивных, интерпретируемых и масштабируемых решений в условиях высокой сложности и требований к безопасности энергосетей.

Ключевые слова: прогнозирование отказов, глубокое обучение, энергосистема, большие данные, LSTM, GRU, Transformer, мониторинг умной сети, обнаружение аномалий, аналитика в реальном времени.

Introduction

Modern power grids operate as complex, distributed systems characterized by high interconnectivity, dynamic load behavior, and sensitivity to external and internal disruptions. With the increasing penetration of renewable energy sources, demand-side variability, and aging infrastructure components, the risk of faults, blackouts, and cascading failures has become more pronounced. Conventional fault detection systems, relying on static thresholds or predefined rule sets, are often insufficient for timely and accurate failure prediction. As the consequences of system-wide outages become more severe-impacting public safety, economic activity, and national security-there is a growing need for intelligent, predictive solutions that can operate at scale and under uncertainty.

Recent advances in big data technologies and deep learning (DL) algorithms provide a promising foundation for developing data-driven models capable of forecasting failures in power grid operations. Big data systems enable real-time ingestion and processing of vast, heterogeneous datasets, including sensor measurements, maintenance logs, weather forecasts, and grid topologies. When integrated with DL models such as convolutional neural networks (CNN), recurrent neural networks (RNN), and transformer-based architectures, these platforms can uncover complex patterns and temporal correlations that traditional analytics fail to detect. The synergistic use of data volume, velocity, and variety with adaptive learning models allows for enhanced accuracy in detecting anomalies and predicting potential breakdowns.

The objective of this study is to investigate the application of big data infrastructure and deep learning models for predictive analysis of failures in power grids. The article explores the architectural components of the data pipeline, the design and training of learning models, and the performance evaluation of predictive systems. The analysis includes examples of model structures, implementation strategies, and domain-specific challenges, with an emphasis on scalability, interpretability, and integration into real-time grid monitoring systems. The study aims to contribute to the development of resilient, intelligent frameworks for power grid management that reduce failure risks and improve operational efficiency.

Main part

Power grids generate vast amounts of operational data from a multitude of sources, including phasor measurement units (PMUs), supervisory control and data acquisition (SCADA) systems, smart meters, weather sensors, and maintenance reports. The heterogeneity and high velocity of this data present both a challenge and an opportunity: while traditional analytical methods struggle to process such volume and diversity in real time, big data technologies offer scalable frameworks to handle continuous data flows, integrate disparate data types, and support advanced analytics. Distributed computing platforms such as apache hadoop and apache spark are widely used to manage and process power system data at scale, enabling the construction of end-to-end pipelines for data cleaning, transformation, feature extraction, and model deployment.

Within this data-driven ecosystem, deep learning models have demonstrated significant potential in identifying latent patterns associated with fault development and grid instabilities. Unlike shallow machine learning methods that depend on manual feature engineering, deep neural networks autonomously learn hierarchical representations from raw or minimally processed input. This capability is particularly valuable for time-series data, where recurrent neural networks and long short-term memory (LSTM) models can capture temporal dependencies, detect anomalous trends, and provide early warning signals for potential failures. In addition, convolutional neural networks have been successfully applied to structured sensor grids or transformed spectrograms, revealing spatial-temporal correlations that precede critical events [1].

The integration of big data infrastructure with deep learning systems requires careful architectural design to ensure performance, accuracy, and interpretability. Model performance depends not only on algorithmic choice but also on data quality, labeling strategy, and training

efficiency. Furthermore, challenges such as data imbalance, concept drift, and explainability must be addressed when implementing predictive solutions in real-world power systems. Despite these challenges, the convergence of high-throughput data collection, scalable processing architectures, and adaptive learning algorithms marks a transformative shift in how power grid failures can be anticipated and mitigated.

An essential step in building effective prediction models is the identification and engineering of relevant features that can serve as early indicators of system degradation or instability. While deep models are capable of autonomously extracting representations from raw input, the inclusion of domain-specific features-such as voltage sag duration, frequency deviation trends, switching patterns, or load transfer metrics-can enhance model interpretability and training convergence. In hybrid frameworks, handcrafted features are used alongside learned representations to improve prediction performance and facilitate expert validation of model behavior [2].

To address the temporal complexity of grid dynamics, models are often trained on time-segmented windows derived from historical event data, annotated with failure labels. This framing enables the detection of evolving fault signatures, which may manifest as subtle, gradually intensifying deviations in voltage, current, or phase angle. Model architectures are selected based on the type and granularity of the data: sequence-based models are preferred for long-range temporal dependencies, while spatially structured data-such as grid-level snapshots-may be better processed with convolutional or attention-based mechanisms. In both cases, prediction is framed as a supervised classification or regression task, with performance evaluated using metrics such as precision, recall, F1-score, and mean absolute error.

Deployment of predictive models in operational environments requires integration with existing monitoring and control systems [3]. Stream processing components are configured to feed real-time sensor data into the inference pipeline, allowing the system to issue alerts or trigger predefined mitigation protocols upon detection of anomalous patterns. To ensure responsiveness, models are optimized for low-latency execution, and inference results are prioritized based on severity and location of predicted failures. In mission-critical settings, explainability tools are embedded to provide transparency into model decisions, enabling operators to verify, override, or supplement automated responses. These features contribute to the practical applicability and trustworthiness of deep learning-based failure prediction systems in modern power grids.

A critical challenge in real-world implementation is the variability and imbalance of training data. Failures in power grids are relatively rare compared to normal operation, leading to strongly skewed datasets where fault instances represent a small fraction of the total. This imbalance can significantly degrade model performance, causing underestimation of fault probabilities and reducing sensitivity to early warning signals. Common strategies to address this include oversampling of failure instances, synthetic data generation, and the application of cost-sensitive loss functions during training. In addition, ensemble methods are employed to increase robustness, aggregating multiple models trained on different data subsets or failure types to improve generalization and stability.

Furthermore, the effectiveness of predictive frameworks is influenced by the system’s ability to adapt to changing operational conditions [4]. Grid topologies, load profiles, and environmental influences evolve over time, introducing concept drift that may render static models obsolete. Continuous learning mechanisms, including periodic retraining, online adaptation, or reinforcement-driven feedback loops, are essential to maintain relevance in dynamic environments. These mechanisms must be carefully balanced to prevent degradation due to overfitting on transient anomalies or incorporation of erroneous labels. Robust validation and monitoring of model drift become integral components of the full deployment lifecycle.

Another important consideration in the design of predictive systems for power grids is their ability to operate across different spatial and hierarchical levels of the infrastructure. Failures may originate at the component level-such as transformer overheating or line degradation-or emerge from broader interactions between substations, transmission zones, and external influences like weather or demand surges. Predictive models must therefore incorporate both local and system-wide indicators to generate accurate forecasts. This requirement often leads to the development of multi-scale

architectures, where input features are aggregated across spatial tiers and processed through parallel or nested learning layers, allowing the system to reason about localized anomalies within a broader grid-wide context.

Interoperability with legacy systems and regulatory compliance further shape the technical constraints of deploying predictive models. In many grid environments, centralized supervisory platforms remain the core of operations, necessitating that learning components seamlessly integrate with existing interfaces and follow data governance protocols. Furthermore, compliance with industry standards-such as IEC 61850 for communication or NERC CIP for cybersecurity-requires that predictive tools be auditable, traceable, and secured. These constraints influence architectural decisions, including data encryption, model transparency, and access control mechanisms, ensuring that innovation does not compromise the safety and accountability of the infrastructure [5].

Comparative analysis of deep learning models for failure prediction in power grids

Choosing an appropriate deep learning model for failure prediction in power grids requires careful consideration of several factors: the structure and scale of input data, the temporal and spatial complexity of target patterns, operational constraints, and the criticality of real-time responsiveness [6]. Models vary in how they process sequences, tolerate noise, scale across data volumes, and support interpretability-key aspects when deployed in high-risk, infrastructure-bound environments. As power grid data ranges from short-term signal fluctuations to long-horizon system evolution, no single architecture performs optimally across all contexts.

To facilitate informed model selection, table 1 provides a comparative summary of four commonly applied deep learning architectures. These models are assessed across multiple criteria, including their ability to model temporal dependencies, computational efficiency, robustness to data imperfections, interpretability, and applicability in real-time operational settings. The comparison highlights the trade-offs between accuracy, speed, complexity, and deployment feasibility, serving as a practical framework for aligning predictive analytics with the technical and infrastructural demands of power systems.

Table 1

Comparative characteristics of deep learning models for failure prediction in power grids

Model	Input type	Temporal dependency modeling	Computational efficiency	Noise and outlier tolerance	Interpretability	Real-time applicability
LSTM	Sequential time series from operational data	Captures long-term patterns through memory cells	High training/inference cost due to complex gate mechanisms	Moderate; may overfit on noisy sequences	Low; internal states are hard to interpret	Moderate; suitable with optimized execution pipelines
GRU	Compressed time sequences with fewer parameters	Models mid-to-long dependencies with simplified design	More efficient than LSTM; faster training and convergence	Balanced; performs well on moderately noisy datasets	Low; interpretability slightly improved vs. LSTM	High; efficient for near-real-time deployment
CNN	Structured sensor data, grid-based or spatial formats	Limited temporal capture; strong spatial pattern detection	Low complexity; fast training and inference	High resistance due to local filters and pooling layers	Moderate; feature maps can assist in interpretability	High; ideal for embedded or low-latency use cases
Model	**Input type**	**Temporal dependency modeling**	**Computational efficiency**	**Noise and outlier tolerance**	**Interpretability**	**Real-time applicability**

Transformer	Multivariate time series and event sequences	Captures global temporal correlations via self-attention	Very high computational demand; slower model convergence	Moderate; effective with normalization and regularization	Low; relies on post-hoc explanation methods	Low; less suited for real-time due to high resource usage

Each architecture presents distinct advantages and limitations depending on the application scenario [7]. LSTM and GRU models are strong choices for capturing time-dependent patterns, with GRU offering improved computational efficiency. CNNs are well suited for structured data and real-time execution, especially in scenarios emphasizing spatial pattern recognition and low-latency requirements. Transformers demonstrate superior capacity in modeling long-range dependencies but require significant computational resources, limiting their applicability in time-sensitive or resource-constrained settings. Model selection should reflect not only predictive accuracy but also integration feasibility, system criticality, and explainability requirements [8].

Evaluation of implementation scenarios and system performance in real-world deployments

The practical integration of deep learning models into power grid monitoring systems has been tested in multiple research and industry-driven pilot projects. These implementations differ in terms of data sources, system scale, prediction objectives, and latency requirements. Some deployments focus on substation-level anomaly detection, while others aim at wide-area failure forecasting across transmission lines [9]. Key performance indicators include detection accuracy, false positive rates, inference latency, and integration compatibility with existing supervisory platforms. Performance evaluation also considers model robustness under noisy or incomplete data and the effectiveness of the alerting system in initiating preventive actions.

Table 2 presents a comparative overview of selected implementation scenarios based on reported studies and field applications. The comparison includes model type, deployment scale, data volume, system latency, and observed prediction effectiveness. The diversity of conditions highlights how system architecture and model tuning must be adapted to the operational context.

Table 2

Real-world implementation scenarios for failure prediction in power grids

Scenario	Model used	Deployment scale	Data volume	Latency requirement	Prediction effectiveness
Urban substation anomaly detection	CNN	Single substation	Medium (real-time sensor streams)	Low	High precision, limited horizon
Regional load forecasting and failure prediction	GRU	Regional grid (10+ substations)	High (time series + weather data)	Moderate	Stable forecasts with 85% accuracy
Wide-area fault detection in transmission grid	LSTM	Nationwide transmission system	Very high (PMU + SCADA feeds)	Low	Accurate detection with low false positives
Smart meter network predictive analytics	Transformer	Distributed household network	Medium-high (smart meters, billing data)	Moderate	Adaptive but latency-sensitive

The table illustrates the variability of implementation contexts for deep learning models in power grid failure prediction. CNNs demonstrate strong performance in localized environments with strict latency demands, while GRU and LSTM models offer scalable solutions for regional and national applications with structured time series inputs. Transformer-based systems show promise in

handling complex, heterogeneous data but face limitations in latency-sensitive use cases. These results underscore the need to tailor model selection and system architecture to the specific operational scale, data characteristics, and performance constraints of each deployment scenario.

Prototype implementation of a predictive module for grid monitoring

To demonstrate the practical aspects of failure prediction in power grids, a prototype module was developed using a simplified architecture combining preprocessing, threshold-based classification, and integration with a predictive model [10]. While advanced models such as LSTM or GRU are typically used in production environments, initial prototypes often rely on ensemble learning or shallow classifiers to validate pipeline behavior and integration points. This modular design facilitates testing across various input types and prediction horizons, enabling rapid adaptation to different grid segments and data configurations.

The prototype includes functionality for preprocessing raw time-series data, normalizing input features, and applying a trained model to determine operational status. Based on the predicted probability of failure, the system issues alerts or maintains a normal state. This logic is embedded within a lightweight inference module, suitable for edge deployment or integration with streaming analytics platforms. The pseudocode representation below illustrates the core components of this predictive logic.

```
class Faultpredictor:
    def __init__(self, model):
        self.model = model

    def preprocess(self, raw_data):
        # Normalize and reshape input
        normalized = (raw_data - raw_data.mean()) / raw_data.std()
        return normalized.reshape(1, -1)

    def predict(self, input_data):
        processed = self.preprocess(input_data)
        prediction = self.model.predict_proba(processed)[0][1]
        return "ALERT" if prediction > 0.8 else "NORMAL"

# Example usage
from sklearn.ensemble import RandomForestClassifier
import numpy as np

# Simulated trained model (for illustration)
mock_model = RandomForestClassifier()
mock_model.fit(np.random.rand(100, 50), np.random.randint(0, 2, 100))

# Instantiate predictor and test with sample input
predictor = FaultPredictor(mock_model)
input_sample = np.random.rand(50)
status = predictor.predict(input_sample)

print(f"System status: {status}")
```

The presented prototype illustrates the fundamental structure of a predictive fault detection module designed for integration into grid monitoring systems. Despite its simplified architecture, the module encapsulates key operational stages: data preprocessing, probabilistic prediction, and decision logic. Its modular design and low computational footprint make it suitable for edge-level deployment or real-time inference pipelines in distributed grid environments [11]. While advanced architectures offer superior accuracy, prototypes such as this provide a crucial foundation for testing system integration, validating workflows, and enabling progressive refinement toward production-ready solutions.

Integration challenges and reliability considerations in real-world grid environments

The successful deployment of predictive systems based on deep learning in power grids requires more than model accuracy-it demands seamless integration into existing operational frameworks and assurance of system reliability under varying real-world conditions. One of the key challenges is infrastructure heterogeneity. Power grids often comprise a mix of legacy systems, proprietary protocols, and hardware with differing data sampling rates and communication standards. Integrating a predictive module into such an environment calls for adaptable interfaces, protocol converters, and robust data fusion mechanisms to ensure compatibility and consistency across platforms.

Another critical challenge is ensuring the reliability and safety of automated predictions. In high-stakes environments such as power transmission and distribution, false positives can lead to unnecessary system reconfigurations, while false negatives may result in undetected risks and costly outages. As a result, predictive systems must be thoroughly validated using domain-specific benchmarks, historical event data, and stress-test simulations under worst-case scenarios. Reliability engineering principles, including redundancy, fail-safe fallback logic, and continuous performance monitoring, must be embedded into the prediction pipeline to maintain trust and operational continuity.

Furthermore, human oversight remains an essential component in practical deployments. Despite advances in automation, predictive models should function as decision support tools rather than autonomous decision-makers. This requires transparent interfaces that present model outputs alongside contextual information and allow for operator intervention when necessary. Explainable AI methods, such as saliency maps or feature attribution techniques, should be integrated to justify predictions and facilitate regulatory compliance. In combination, these design considerations form the foundation for trustworthy and effective integration of deep learning-based prediction systems in operational power grid environments.

Conclusion

The convergence of big data infrastructure and deep learning techniques offers a powerful framework for failure prediction in power grid systems. By leveraging real-time, high-volume, and multi-source data, predictive models can identify early indicators of instability and support proactive interventions, reducing the likelihood of cascading failures and operational disruptions. Deep learning architectures such as LSTM, GRU, CNN, and Transformer enable the modeling of complex spatial-temporal dependencies, offering flexibility across diverse deployment scenarios.

Through comparative analysis and implementation examples, this study has demonstrated the strengths and trade-offs of each model in relation to data characteristics, computational requirements, and latency constraints. Practical modules can be embedded within monitoring pipelines or deployed at the edge, supporting rapid detection and response. However, successful integration requires addressing interoperability with legacy systems, ensuring model robustness under uncertainty, and maintaining transparency and human oversight in critical operations.

The findings emphasize the need for adaptive, explainable, and scalable predictive systems tailored to the operational realities of modern power infrastructure. As power grids continue to evolve in complexity and exposure, the role of intelligent, data-driven solutions will become increasingly central to infrastructure resilience and energy system reliability.

References

1. Koshy S., Rahul S., Sunitha R., Cheriyan E.P. Smart grid–based big data analytics using machine learning and artificial intelligence: A survey // Artif. Intell. Internet Things Renew. Energy Syst. 2021. Vol. 12. P. 241.
2. Elahe M.F., Jin M., Zeng P. Review of load data analytics using deep learning in smart grids: Open load datasets, methodologies, and application challenges // International Journal of Energy Research. 2021. Vol. 45. No. 10. P. 14274-14305.
3. Stepanov M. Adaptive control systems for optimizing electric drive operation and reducing energy consumption in challenging conditions // Original research. 2024. Vol. 14. No. 9. P. 86-92.

4. Liao W., Bak-Jensen B., Pillai J.R., Wang Y., Wang Y. A review of graph neural networks and their applications in power systems // Journal of Modern Power Systems and Clean Energy. 2021. Vol. 10. No. 2. P. 345-360.
5. Belagoune S., Bali N., Bakdi A., Baadji B., Atif K. Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems // Measurement. 2021. Vol. 177. P. 109330.
6. Boddapati V.N. Optimizing production efficiency in manufacturing using big data and AI/ML. 2025.
7. Borodin I. The impact of Building Information Modeling (BIM) technology on the quality and accuracy of design in the construction industry // Annali d'Italia. 2024. No. 62. P. 116-118.
8. Ageed Z.S., Zeebaree S.R., Sadeeq M.M., Kak S.F., Yahia H.S., Mahmood M.R., Ibrahim I.M. Comprehensive survey of big data mining approaches in cloud systems // Qubahan Academic Journal. 2021. Vol. 1. No. 2. P. 29-38.
9. Chehri A., Fofana I., Yang X. Security risk modeling in smart grid critical infrastructures in the era of big data and artificial intelligence // Sustainability. 2021. Vol. 13. No. 6. P. 3196.
10. Huang B., Wang J. Applications of physics-informed neural networks in power systems-a review // IEEE Transactions on Power Systems. 2022. Vol. 38. No. 1. P. 572-588.
11. Sircar A., Yadav K., Rayavarapu K., Bist N., Oza H. Application of machine learning and artificial intelligence in oil and gas industry // Petroleum Research. 2021. Vol. 6. No. 4. P. 379-391.