

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Санкт-Петербургский государственный университет телекоммуникаций им.
проф. М.А. Бонч-Бруевича»

На правах рукописи

Данг Ван Тханг

**РАЗРАБОТКА И ИССЛЕДОВАНИЕ МОДЕЛЕЙ И МЕТОДОВ
ИНТЕГРАЦИИ ТУМАННЫХ ВЫЧИСЛЕНИЙ И ФЕДЕРАТИВНОГО
ОБУЧЕНИЯ В СЕТЯХ СВЯЗИ ШЕСТОГО ПОКОЛЕНИЯ**

2.2.15 - Системы, сети и устройства телекоммуникаций

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель
доктор технических наук
Волков Артем Николаевич

Санкт-Петербург – 2026

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	2
ВВЕДЕНИЕ	4
ГЛАВА 1. ИССЛЕДОВАНИЕ ИНТЕГРАЦИИ ТУМАННЫХ ВЫЧИСЛЕНИЙ И ФЕДЕРАТИВНОГО ОБУЧЕНИЯ В СЕТЯХ СВЯЗИ ШЕСТОГО ПОКОЛЕНИЯ. 13	
1.1 Введение	13
1.2 Анализ сетей 6G: базовые технологии и направления развития	15
1.3 Переход к новой архитектуре 6G: сеть связи с искусственным интеллектом	25
1.4 Туманные вычисления: концепция, архитектура и роль в сетях шестого поколения	40
1.5 Федеративное обучение: принципы, архитектура и роль в сетях шестого поколения	51
1.6 Интеграция туманных вычислений и федеративного обучения в архитектуре сетей шестого поколения	54
1.7 Задачи диссертации	59
1.8 Выводы по Главе 1.....	61
ГЛАВА 2. МОДЕЛЬ ФЕДЕРАТИВНОГО МАШИННОГО ОБУЧЕНИЯ С ТУМАННЫМИ ВЫЧИСЛЕНИЯМИ И МЕТОД ВЫБОРКИ ПОЛЬЗОВАТЕЛЬСКИХ УСТРОЙСТВ НА ОСНОВЕ ДИНАМИЧЕСКОГО ПОРОГА	63
2.1 Введение	63
2.2 Обзор методов выборки устройств в федеративном машинном обучении	64
2.3 Модель федеративного машинного обучения с туманными вычислениями и метод динамического порога выбора пользовательских устройств.....	68
2.4 Теоретические основы метода выборки устройств на основе динамического порога скорости изменения градиента функции потерь.....	71
2.5 Предлагаемая модель интеграции FL и FC	75
2.6 Тестирование предложенной модели и метода	81

2.7 Выводы по Главе 2.....	88
ГЛАВА 3. ДЕЦЕНТРАЛИЗОВАННАЯ ОБЛАЧНАЯ СТРУКТУРА И МОДЕЛЬ ФЕДЕРАТИВНОГО ОБУЧЕНИЯ С ТУМАННЫМИ АЛЬЯНСАМИ ДЛЯ ГОРОДСКИХ СЕТЕВЫХ ВСЕЛЕННЫХ	89
3.1 Введение	89
3.2 Анализ существующих решений и постановка задачи	91
3.3 Предлагаемая общая структура DFL-FC-AGCRN.....	93
3.4 Моделирование и анализ результатов	107
3.5 Выводы по Главе 3.....	122
ГЛАВА 4. СТРУКТУРА И МЕТОД РАЗВЁРТЫВАНИЯ ФЕДЕРАТИВНОГО МАШИННОГО ОБУЧЕНИЯ НА ОСНОВЕ WEBASSEMBLY В ГЕТЕРОГЕННОЙ СРЕДЕ ТУМАННЫХ ВЫЧИСЛЕНИЙ.....	124
4.1 Введение	124
4.2 Анализ существующих подходов к развертыванию FL на гетерогенных узлах	125
4.3 Структура и теоретические основы	130
4.4 Предлагаемая структура на основе WebAssembly.....	132
4.5. Экспериментальная оценка.....	140
4.6 Выводы по Главе 4.....	147
ЗАКЛЮЧЕНИЕ.....	149
СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ.....	151
СПИСОК ЛИТЕРАТУРЫ	155
Приложение.....	168

ВВЕДЕНИЕ

Актуальность темы исследования. После старта внедрения сетей пятого поколения (IMT-2020/5G) и достижения ими определённых результатов мировое научное сообщество сосредоточило усилия на исследовании и формировании сетей шестого поколения. Сети шестого поколения (IMT-2030/6G) представляют собой естественное развитие предыдущих поколений с возросшими характеристиками и ключевыми особенностями, которые заключается в интеграции космического, воздушного, наземного и морского сегментов (с англ. SAGSIN - Space Air Ground Sea Integrated Networks) и широкого применения искусственного интеллекта. К характеристикам относятся повышение надёжности передачи, достижение ультрамалых задержек, а также поддержка сверхвысокой плотности устройств. Ожидается, что сети шестого поколения создадут основу для широкого распространения новых сервисов, таких как иммерсивные услуги, смешанная реальность, автономный (беспилотный) транспорт, системы телеприсутствия, и будут способствовать развитию сетевых вселенных (метавселенных).

Сети 6G формируются как качественно новая архитектурная парадигма, в которой искусственный интеллект (ИИ) перестаёт быть инструментом оптимизации и становится неотъемлемым элементом сети. В основе данного подхода лежит концепция естественной интеграции искусственного интеллекта (AI-native), согласно которой механизмы ИИ закладываются в архитектуру сети на стадии проектирования, а не добавляются как надстройка над готовой инфраструктурой. В 2023 году этот принцип был закреплён в рекомендации Международного Союза Электросвязи (МСЭ) Р М.2160 [1], где в качестве основополагающих принципов проектирования сетей связи шестого поколения являются: устойчивость, безопасность и отказоустойчивость, подключение неохваченных пользователей и повсеместный искусственный интеллект для повышения общей производительности системы.

Параллельно на уровне международной стандартизации сформировалась концепция сети связи с искусственным интеллектом, представляющая собой

телекоммуникационную сеть, в которой технологии ИИ интегрированы в качестве естественных компонентов на всех уровнях архитектуры с целью повышения эффективности, возможностей и надёжности систем и сетей связи. Резолюция 101 ВАСЭ-24 (Всемирная Ассамблея Стандартизации Электросвязи в Нью-Дели, 2024) [2] зафиксировала возрастающую актуальность ИИ и обусловленную этим потребность в надёжных технических стандартах для сетей связи с поддержкой ИИ, а также поручила МСЭ-Т продолжить работу по применению ИИ при разработке рекомендаций, связанных с эксплуатацией и управлением сетями, надёжностью, безопасностью и сетями с поддержкой ИИ. Тем самым на международном уровне была подтверждена стратегическая направленность развития телекоммуникационной отрасли в сторону глубокой интеграции ИИ в сетевую инфраструктуру.

Реализация указанных принципов в условиях реальной инфраструктуры представляет собой сложную задачу. Среди технологий, рассматриваемых как перспективная основа, всё чаще выделяют совместное использование туманных вычислений (Fog Computing, FC) и федеративного обучения (Federated Learning, FL) в сетях шестого поколения. FC позволяют перенести обработку данных ближе к их источнику: между облаком и пользовательскими устройствами формируется промежуточный вычислительный слой, который принимает на себя значительную часть нагрузки, снижает затраты на передачу данных и уменьшает задержку. FL предполагает обучение моделей ИИ непосредственно на устройствах без передачи исходных данных в основной центр обработки данных (ЦОД), что устраняет узкие места, связанные с пропускной способностью сети, и обеспечивает соответствие требованиям защиты персональных данных. Сочетание этих технологий формирует ресурсную базу для реализации моделей и методов в сетях связи с ИИ.

Таким образом, разработка и исследование моделей и методов интеграции туманных вычислений и федеративного обучения в сетях связи шестого поколения представляет собой актуальную научную задачу, решение которой способствует практической реализации принципов “AI-native” [3] и концепции сети связи с ИИ

в условиях распределённой гетерогенной граничной среды сетей связи шестого поколения.

Разработанность темы исследования. Существует множество работ, посвящённых развитию сетей IMT-2020/5G и последующих поколений, а также архитектур граничных и туманных вычислений. Значительный вклад в теоретические и экспериментальные исследования инфокоммуникационных сетей внесли такие российские учёные, как А.Е. Кучерявый, Е.А. Кучерявый, А.С.А. Мутханна, К.Е. Самуйлов, А.Н. Волков и О.А. Маркелов, А.С. Аджемов, В.М. Вишневский, Ю.В. Гайдамака, В.С. Елагин, А.И. Парамонов, С.Н. Степанов, В.Г. Карташевский, В.О. Тихвинский, Б.С. Гольдштейн, А.Б. Гольдштейн, а также ведущие зарубежные исследователи M. Z. Chowdhury, M. Latva-aho, H. Pennanen, C. Puliafito, M. Verma и A. A. Ateya.

Значительный вклад в развитие парадигмы распределённого искусственного интеллекта, в частности алгоритмов федеративного обучения и децентрализованной оптимизации сетей связи, внесли B. McMahan, P. Kairouz, G. B. Giannakis, W. Chen, T. Nishio, T. Wang, A.A. Ateya.

В области прогнозирования городского трафика с использованием моделей глубокого обучения и пространственно-временных графовых нейронных сетей важные результаты получены в трудах L. Zhao, B. Yu, L. Bai, T. Qi. Кроме того, вопросам развёртывания искусственного интеллекта на границе сети с применением современных сред выполнения, таких как WebAssembly, посвящены работы M. Garofalo, J. Bachmeier, S. Berard, P. Gackstatter.

Настоящая диссертация, в отличие от известных подходов к построению и функционированию сетей связи, направлена на решение научной задачи разработки моделей и методов интеграции туманных вычислений и федеративного обучения в условиях широкого распространения искусственного интеллекта на уровне инфраструктуры сетей связи шестого поколения.

Объект и предмет исследования. Объектом исследования являются сети связи на основе туманных вычислений и федеративного обучения. **Предметом**

исследования являются модели и методы интеграции туманных вычислений, федеративного обучения в сети связи шестого поколения.

Цель и задачи исследования. Целью диссертационной работы является исследование и разработка моделей и методов построения сетей шестого поколения на основе туманных вычислений и федеративного обучения.

Цель работы достигается путём последовательного решения следующих задач:

- анализ существующего положения в области исследования сетей шестого поколения, роли и места искусственного интеллекта, а также туманных вычислений и федеративного обучения.;

- анализ существующего положения дел в области международной стандартизации, в частности рекомендаций МСЭ-Т по искусственному интеллекту, сетям шестого поколения, туманным вычислениям и федеративному обучению;

- разработка метода выборки в раундах федеративного обучения на основе динамического порога скорости изменения градиента функции потерь с вычислением порога на уровне туманных вычислений для снижения коммуникационных издержек;

- разработка децентрализованной структуры федеративного обучения с туманными вычислениями на основе адаптивной графовой свёрточной рекуррентной нейронной сети для обеспечения автономного функционирования при нестабильном соединении с облачным уровнем на примере задачи прогнозирования городского транспортного потока в сервисах городских сетевых вселенных (CitiVerse);

- разработка структуры и метода кроссплатформенного развёртывания федеративного обучения на гетерогенных узлах туманных вычислений на основе технологии WebAssembly и среды выполнения Wasmtime;

Научная новизна. Научная новизна работы состоит в следующем:

1. Разработан метод выбора пользовательских устройств и модель федеративного обучения с туманными вычислениями для сетей шестого поколения, отличающийся от известных тем, что адаптивный порог выбора динамически

пересчитывается на уровне туманных вычислений в каждом раунде на основе статистики скоростей изменения нормы градиента, что позволяет учитывать неоднородность локальных данных и динамику процесса сходимости, а также снизить объём сетевого взаимодействия между устройствами и центральным сервером на 39% при сохранении точности модели.

2. Разработана децентрализованная структура федеративного обучения с туманными вычислениями на основе адаптивной графовой свёрточной рекуррентной сети для прогнозирования городских транспортных потоков в сервисах городских сетевых вселенных (CitiVerse), отличающаяся от известных тем, что полностью исключает центральный сервер из цикла обучения и агрегации модели искусственного интеллекта, а механизм отдельной агрегации глобальных весов и локальных векторов встраивания обеспечивает баланс между общими знаниями и региональной адаптацией модели при решении проблемы Non-IID (неоднородных) данных, а также учитывает новую структуру туманных альянсов. Предложенный подход демонстрирует высокую эффективность: снижает потребность в сетевых ресурсах на ~48 %, по сравнению с традиционными методами федеративного обучения, ускоряет сходимость модели при сохранении точности модели прогноза.

3. Разработаны структура и метод кроссплатформенного развёртывания федеративного обучения на гетерогенных узлах туманных вычислений в архитектуре «облако – туман – пользовательское устройство», отличающиеся от известных решений тем, что система байт-кода WebAssembly используется как инфраструктурный слой выполнения локального обучения моделей федеративного искусственного интеллекта на гетерогенных туманных узлах, что устраняет необходимость в перекомпиляции и переконфигурации под каждую целевую аппаратную архитектуру, в отличие от традиционного (т.н. нативного) подхода и контейнеризации, тем самым ускоряя процесс развёртывания модулей федеративного искусственного интеллекта в гетерогенной среде туманных вычислений, что существенно снижает сложность развёртывания в сетях шестого поколения.

Теоретическая значимость работы. Теоретическая значимость диссертационной работы определяется совокупностью разработанных моделей и методов интеграции туманных вычислений и федеративного машинного обучения в сетях связи шестого поколения. Полученные результаты расширяют научные представления о принципах построения распределённых систем искусственного интеллекта в условиях гетерогенной вычислительной инфраструктуры сетей связи шестого поколения и создают теоретическую основу для дальнейших исследований в данной области.

Каждый из трёх научных результатов обладает самостоятельной теоретической ценностью. Для задачи адаптивного выбора пользовательских устройств предложены формализованный метод и модель федеративного обучения с туманными вычислениями на основе скорости изменения нормы градиента функции потерь с аналитическими соотношениями для порога, пересчитываемого по статистике устройств предыдущего раунда непосредственно на уровне туманных узлов. Для решения задачи децентрализованной агрегации предложены иерархическая структура и модель федеративного обучения на основе туманных альянсов, предусматривающие кластеризацию узлов с использованием алгоритма HDBSCAN и механизм обеспечения отказоустойчивости. Теоретически обоснован подход к раздельной обработке глобальных весов модели и локальных векторов встраивания с учетом особенностей распределения данных в условиях Non-IID; для задачи кроссплатформенного развёртывания разработаны структура и метод развёртывания федеративного обучения с переносимым слоем выполнения на основе технологии WebAssembly на гетерогенных узлах туманных вычислений, устанавливающая количественные соотношения между накладными расходами среды выполнения и числом раундов обучения модели ИИ.

Практическая значимость работы. Практическая ценность работы состоит в разработке моделей и методов интеграции туманных вычислений и федеративного обучения, обеспечивающих практическую реализацию принципов естественного искусственного интеллекта в сетях связи (с англ., AI-native networks) в условиях распределённой гетерогенной сети шестого поколения.

Реализация результатов работы. Полученные в диссертационной работе результаты внедрены в Санкт-Петербургском государственном университете телекоммуникаций им. проф. М.А. Бонч-Бруевича при чтении лекций и проведении практических занятий по курсам «Искусственный интеллект в сетях и системах связи», "Современные проблемы науки в условиях перехода к сетям шестого поколения (6G)», «Системы, сети и устройства телекоммуникаций". Акт реализации диссертационных исследований представлен в Приложении к диссертации.

Методы исследования. При проведении исследований были использованы методы теоретического исследования, теории оптимизации, методы машинного обучения и глубокого обучения, включая графовые свёрточные сети и рекуррентные нейронные сети, методы пространственной кластеризации, методы имитационного и натурального моделирования.

Основные положения, выносимые на защиту.

1. Метод выбора пользовательских устройств и модель федеративного обучения с туманными вычислениями на основе динамического порога скорости изменения нормы градиента функции потерь, позволяющий одновременно учитывать неоднородность локальных данных и скорость изменения градиента на текущем этапе обучения модели искусственного интеллекта, а также снизить объём сетевого взаимодействия между пользовательскими устройствами и сервером на 39%.

2. Децентрализованная структура и модель федеративного обучения с туманными альянсами на основе адаптивной графовой свёрточной рекуррентной нейронной сети для прогнозирования городских транспортных потоков в сервисах городских сетевых вселенных (CitiVerse), обеспечивающая снижение сетевого взаимодействия до 48% по сравнению с традиционными методами федеративного обучения.

3. Структура и метод развёртывания федеративного обучения на основе WebAssembly (низкоуровневого байт-кода) в гетерогенных средах туманных вычислений, обеспечивающие кроссплатформенное выполнение единого модуля

обучения нейронной сети на узлах с различной аппаратной архитектурой, снижение потребления оперативной памяти на 69,5% по сравнению со стандартной реализацией при сохранении точности и скорости сходимости модели федеративного искусственного интеллекта.

Достоверность результатов. Степень достоверности полученных результатов подтверждается корректным применением математического аппарата теории вероятностей, теории оптимизации и методов машинного обучения, результатами имитационного моделирования на открытых реальных наборах данных (MNIST, CIFAR-10, METR-LA, PeMSD7M) и результатами натурных экспериментов на физических устройствах различных аппаратных архитектур, а также публикациями и выступлениями на российских и международных научных конференциях.

Апробация работы. Основные положения диссертационной работы обсуждались и были одобрены на следующих конференциях и семинарах: Международная конференция по современным системам вычислений и связям следующего поколения (International Conference on Advanced Computing & Next-Generation Communication, ICACNGC 2024) – Санкт-Петербург, Российская Федерация – 2024; Международная конференция по технологиям голографического телеприсутствия и голосетевселенной (International Conference on Holographic Telepresence Technologies and Holnetverse) – Санкт-Петербург, Российская Федерация – 2024; 4-я Международная конференция по инженерии и технологиям (4th SLIIT International Conference on Engineering and Technology, SICET 2025) – Шри-Ланкийский институт информационных технологий (SLIIT), Малабе, Шри-Ланка – 9–12 сентября 2025; 9-я Международная конференция по сетям будущего и распределённым системам (The 9th International Conference on Future Networks and Distributed Systems, ICFNDS '25) – Дубай, ОАЭ – 8–9 декабря 2025; 80-я научно-техническая конференция Санкт-Петербургского НТО РЭС им. А. С. Попова, посвящённая Дню радио – Санкт-Петербург, Российская Федерация – 2025; 81-я научно-техническая конференция Санкт-Петербургского НТО РЭС им. А. С. Попова, посвящённая Дню радио – Санкт-Петербург, Российская Федерация

– 2026; а также на научных семинарах кафедры сетей связи и передачи данных Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича.

Публикации. Основные результаты диссертации изложены в 10 публикациях, в том числе в 3 работах, опубликованных в журналах из перечня ВАК Министерства образования и науки Российской Федерации; 4 работах в изданиях, включённых в международные базы цитирования Scopus и Web of Science; 3 работах в других научных изданиях и материалах конференций.

Соответствие паспорту специальности. Диссертационная работа выполнена по специальности 2.2.15. Системы, сети и устройства телекоммуникаций и соответствует следующим пунктам паспорта специальности: 1, 2, 4, 7, 8, 12.

Структура и объем работы. Диссертация состоит из введения, 4 глав, заключения, списка литературы и двух приложений. Работа содержит 169 страниц машинописного текста, 36 рисунков, 24 таблиц и список литературы из 108 наименований.

Личный вклад автора. Все основные результаты диссертационной работы получены автором самостоятельно. В экспериментальных исследованиях автор принимал непосредственное участие на всех этапах.

ГЛАВА 1. ИССЛЕДОВАНИЕ ИНТЕГРАЦИИ ТУМАННЫХ ВЫЧИСЛЕНИЙ И ФЕДЕРАТИВНОГО ОБУЧЕНИЯ В СЕТЯХ СВЯЗИ ШЕСТОГО ПОКОЛЕНИЯ

1.1 Введение

С развитием телекоммуникационных сетей количество подключенных умных устройств Интернета вещей растет. Приложения нового поколения, такие как голографические коммуникации, телеприсутствие, расширенная реальность, автономные транспортные средства и умные города, выдвигают новые и более высокие требования. Именно эти вызовы побуждают международное научное сообщество сосредоточиться на исследованиях сетей следующего поколения. МСЭ официально называет данные сети IMT-2030, которые также широко известны под названием 6G.

Рекомендация МСЭ R M.2160, утвержденная сектором радиосвязи МСЭ в 2023 году, является основным руководящим документом для систем сетей 6G. Этот документ определяет шесть основных сценариев использования. Три из них унаследованы от 5G, а еще три являются совершенно новыми. К новым сценариям относятся искусственный интеллект в связи (AI in Communication), повсеместная связь (Ubiquitous Connectivity) и интегрированное зондирование и связь (Integrated Sensing and Communication). В частности, рекомендация M.2160 впервые вводит принцип проектирования естественного искусственного интеллекта в сетях связи (с англ., AI-native networks). Согласно этому принципу, искусственный интеллект должен быть интегрирован в саму архитектуру сети непосредственно на этапе проектирования, а не должен выступать просто дополнительным уровнем, который добавляется позже.

Для реализации концепции “AI-native” в условиях реальной телекоммуникационной инфраструктуры особый интерес представляют два перспективных технологических направления: федеративное обучение и туманные вычисления. Технология FC создает промежуточный вычислительный уровень между облаком и пользовательскими устройствами. Это приближает обработку

данных к источнику их генерации. Таким образом снижается задержка и уменьшается нагрузка на ядро сети. В то же время FL позволяет обучать модели искусственного интеллекта непосредственно на устройствах без передачи необработанных данных в центр. Это одновременно решает проблему пропускной способности и защищает конфиденциальность пользователей. Интеграция данных технологий в единую архитектурную среду открывает возможности для построения распределённых систем искусственного интеллекта, обеспечивающих автономное функционирование в условиях гетерогенной инфраструктуры [4, 5].

Параллельно с развитием технологий МСЭ-Т последовательно формирует нормативную базу для интеграции ИИ и распределённых вычислений в сетевую инфраструктуру. Рекомендация Y.3172 определяет архитектурную основу для встраивания машинного обучения в сети связи через распределённую модель ML-конвейера, узлы которого могут размещаться на различных уровнях инфраструктуры от облачных серверов до конечных устройств. Рекомендация Y.3173 вводит шкалу оценки интеллектуальности сети от уровня L_0 , при котором все функции выполняются оператором вручную, до уровня L_5 , соответствующего полностью автономной работе без участия человека. Рекомендация Y.3224 устанавливает архитектуру для федеративного обучения с агрегацией внутри сети, определяя три взаимодействующие функциональные роли: клиент FL, внутрисетевой агрегатор и глобальный координатор. Пара рекомендаций Y.3400 и Y.3401 формирует требования и архитектурную основу для совместной координации сетевых и вычислительных ресурсов как единого целого.

Вместе с тем сопоставление этих нормативных документов с реальными условиями развёртывания обнаруживает ряд пробелов. Прежде всего, архитектура ML-конвейера согласно Y.3172 неявно ориентирована на централизованную модель сбора и обработки данных, что перестаёт соответствовать условиям масштабирования до миллиардов конечных узлов и ужесточающимся требованиям к защите персональных данных. Именно это несоответствие обуславливает необходимость федеративного обучения, реализуемого непосредственно на уровне туманных узлов. Далее, рекомендация Y.3224 вводит роль внутрисетевого

агрегатора, однако оставляет открытым вопрос о его физическом размещении и координации в гетерогенной инфраструктуре, что делает задачу архитектурной интеграции туманных узлов в цикл FL самостоятельной и нерешённой.

Глава 1 формирует теоретическую и аналитическую основу всей диссертации. Центральным результатом главы является трёхуровневая интегрированная архитектура туманных вычислений и федеративного обучения для сетей шестого поколения, разработанная по итогам анализа современного состояния исследовательской области. В разделе 1.2 анализируются особенности развития сетей шестого поколения, включая технологии и направления развития. В разделе 1.3 рассматривается переход к новой архитектуре 6G и концепция сети связи с искусственным интеллектом. Раздел 1.4 раскрывает концепцию туманных вычислений, их архитектуру и роль в сетях шестого поколения. В разделе 1.5 рассматриваются принципы и архитектура федеративного обучения применительно к требованиям сетей шестого поколения. На основе проведённого анализа в разделе 1.6 предлагается интегрированная архитектура туманных вычислений и федеративного обучения в архитектуре сетей шестого поколения. Раздел 1.7 содержит постановку задачи диссертации, а раздел 1.8 завершает главу выводами по результатам проведённых исследований

1.2 Анализ сетей 6G: базовые технологии и направления развития

Процесс коммерциализации и стандартизации беспроводных сетей пятого поколения 5G к настоящему моменту практически завершён [6]. При этом темпы внедрения оказались значительно выше первоначальных прогнозов. Сети 5G создали надёжную основу для цифровой трансформации в мировом масштабе [7]. К наиболее важным результатам развития этого поколения относится успешная реализация трёх основных категорий услуг: eMBB, URLLC и mMTC, что представлено на рисунке 1.

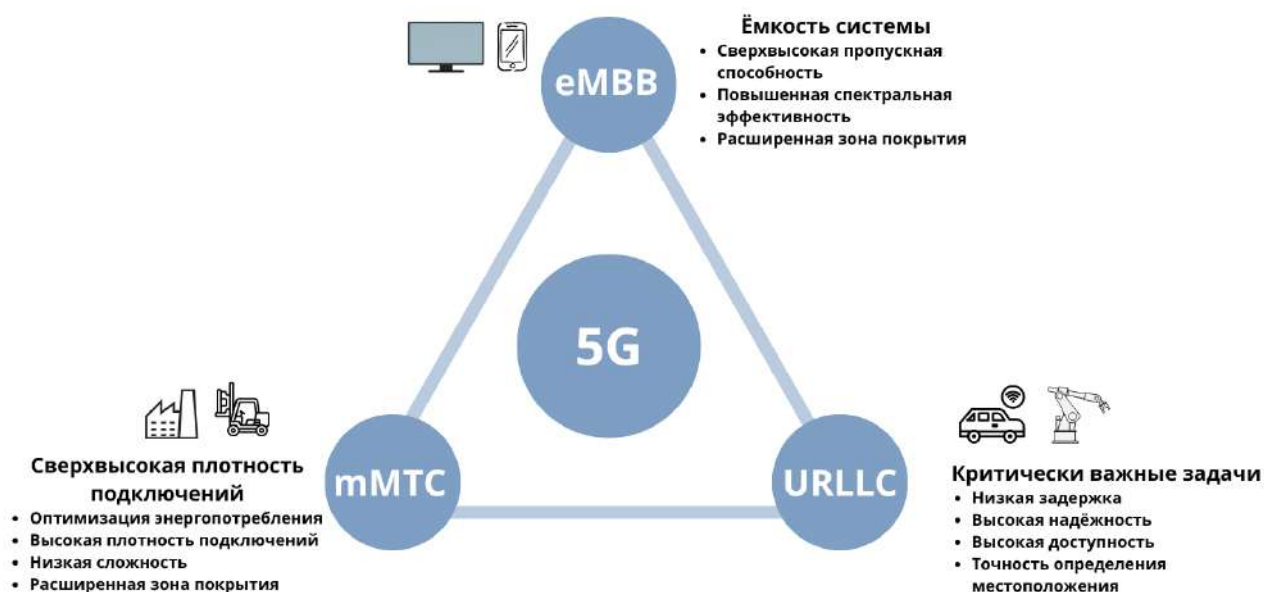


Рисунок 1 — Развитие услуг сети 5G

Тем не менее этот прогресс повлёт за собой стремительный, фактически экспоненциальный рост числа подключённых интеллектуальных устройств. Новые приложения предъявляют значительные требования к вычислительным ресурсам, что в корне меняет подход к проектированию сетевых архитектур. По прогнозам, объём данных, передаваемых в сетях, в ближайшей перспективе выйдет на уровень зеттабайт. Это потребует обработки сложных форматов, порождаемых новыми видами услуг: голографической связью, телеприсутствием и цифровыми двойниками в реальном времени [8, 9, 10].

В рамках своей долгосрочной исследовательской программы МСЭ официально определил следующее поколение сетей под названием IMT-2030 (от англ. International Mobile Telecommunications for 2030 and Beyond). В научном сообществе IMT-2030 широко известен как система связи 6G, то есть шестое поколение глобальных систем мобильной связи [12]. Появление 6G знаменует не просто эволюцию передачи данных, а коренной пересмотр самой парадигмы инфраструктуры информационных технологий и связи. По ключевым техническим показателям сети шестого поколения существенно превзойдут своих предшественников [13]. Рабочие частоты, индивидуальная и пиковая скорость передачи данных, эффективность использования спектра, поддержка мобильности,

а также задержка претерпят качественные изменения. Наглядное сравнение 5G приведено на рисунке 2.

Ожидается, что 6G устранил традиционные границы между физическим миром, цифровым пространством и человеком. Беспроводная сеть при этом рассматривается как интегрированная платформа, обеспечивающая предоставление вычислительных ресурсов, средств хранения данных и инструментов искусственного интеллекта непосредственно в рамках телекоммуникационной инфраструктуры. Ведущие исследователи характеризуют данную концепцию как «интеллектуальная сеть всего» (Intelligent Network of Everything), где сеть выступает в роли интегрирующей платформы, обеспечивающей взаимодействие, анализ данных и управление всеми объектами в обществе [14].

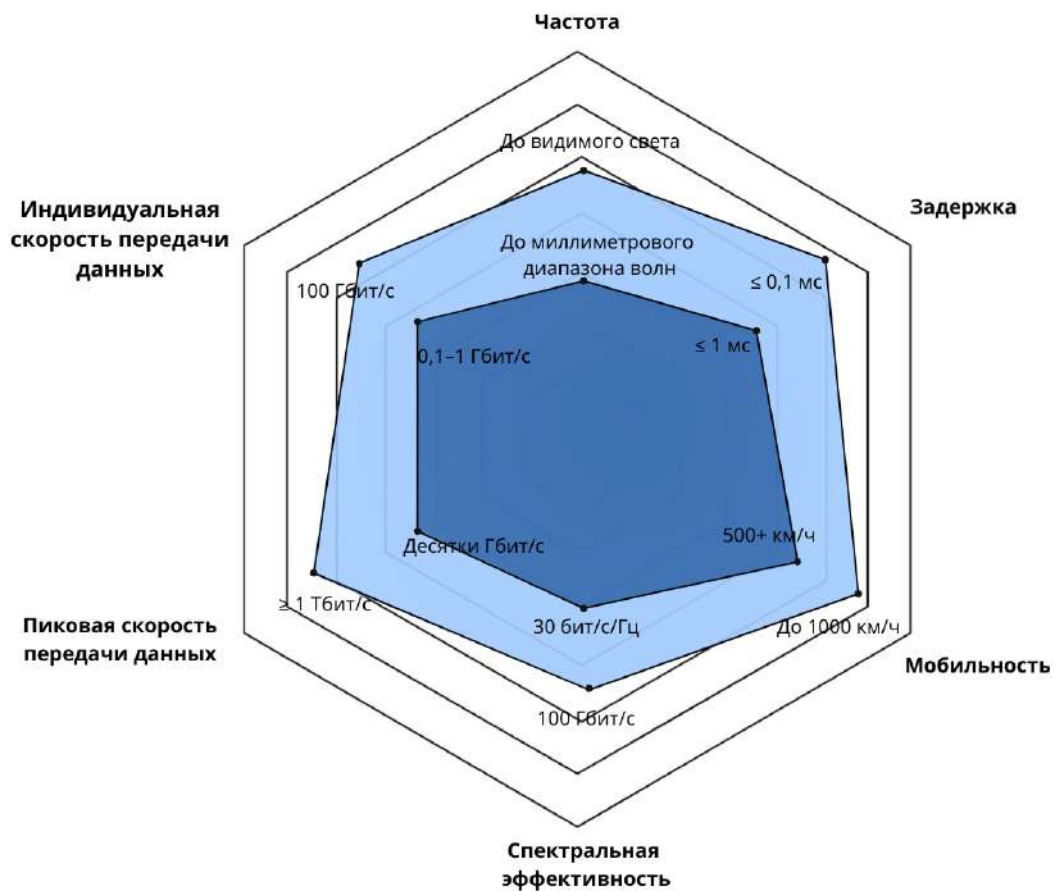


Рисунок 2 — Сравнение ключевых характеристик сетей 5G и 6G

В ноябре 2023 года на Ассамблее радиосвязи МСЭ была утверждена рекомендация МСЭ-R М.2160 под названием «Концепция и общие цели будущего развития ИМТ на 2030 год и далее» (Framework and overall objectives of the future development of IMT for 2030 and beyond) [1]. Данная рекомендация является основополагающим документом для направления 6G и задаёт ориентиры для исследований, разработок, инвестиций и стандартизации в мировой телекоммуникационной отрасли на ближайшие годы. Документ содержит технологическое видение, сценарии применения, показатели технических возможностей и стратегические направления развития экосистемы мобильной связи после 2030 года.

Главная цель 6G состоит в создании системы связи, способной обеспечить повсеместное подключение, интеллектуальность, устойчивость и многофункциональность. В рекомендации МСЭ-R М.2160 чётко формулируется набор основных принципов проектирования 6G:

– *Устойчивость*. 6G проектируется с учётом экономии ресурсов с самого начала. Цель состоит в том, чтобы не допустить ситуации, при которой рост объёмов трафика опережает улучшение энергоэффективности, что ведёт к абсолютному увеличению потребления электроэнергии даже при относительном повышении эффективности. В рекомендации вводится обязательный показатель энергоэффективности, а также требования к проектированию оборудования и сетевой инфраструктуры с учётом полного жизненного цикла для сокращения электронных отходов.

– *Всеобъемлющая связь*. Архитектура 6G должна преодолевать географические барьеры за счёт объединения наземных и неназемных сетей. Это позволит решить проблему недоступности связи в отдельных регионах, характерную для предыдущих поколений.

– *Повсеместный интеллект*. Данный принцип отражает переход к модели сети, в которой искусственный интеллект встроен непосредственно в физическую структуру системы, формируя так называемую архитектуру AI-native. В сетях 6G

алгоритмы машинного обучения внедряются на всех уровнях архитектуры от физического до уровня управления и ядра сети.

– *Безопасность и способность к восстановлению.* Эти требования относятся к базовым проектным принципам сетей 6G. Непрерывность функционирования и защищённость сети шестого поколения обеспечиваются уже на этапе её проектирования.

Архитектура сетей 6G предусматривается шесть базовых сценариев использования. Среди перечисленных сценариев часть унаследована от стандартов 5G, тогда как остальные представляют принципиально новые модели обслуживания. На рисунке 3 показаны основные сценарии использования 6G, которые напрямую связаны с приложениями 5G:

– *Массовые коммуникации (Massive Communication)* представляет собой развитие концепции mMTC. Данный сценарий охватывает взаимодействие между роботами и пользователями, между роботами, а также работу устройств Интернета вещей в контексте умных и безопасных городов.

– *Иммерсивные коммуникации (Immersive Communication)* представляет собой расширение концепции eMBB. Вместе с тем данный сценарий объединяет элементы mMTC и URLLC, что свидетельствует о конвергенции ранее обособленных технологических направлений.

– *Гипернадёжные сети связи с ультрамалой задержкой (Hyper Reliable and Low-Latency Communication)* представляет собой усовершенствованную версию URLLC. Сценарий HURLLC полностью наследует возможности URLLC, но предъявляет более строгие требования для ряда приложений в сетях 2030 года.

– *Искусственный Интеллект в связи (Artificial Intelligence in Communication)* логически продолжает концепцию интеграции ИИ в телекоммуникационные сети, заложенную ещё в 5G. В рамках 6G данное направление получило самостоятельный статус принципа AI-native, согласно которому механизмы искусственного интеллекта закладываются в архитектуру сети на этапе проектирования, формируя основу концепции сети связи с искусственным интеллектом.

– *Повсеместная/всепроникающая связь (Ubiquitous Connectivity)* воплощает концепцию STIN (Space-Terrestrial Integrated Network), предложенную МСЭ-Т для сетей 2030 года. Этот сценарий способствует достижению ряда целей устойчивого развития, в том числе сокращению цифрового разрыва в отдалённых и труднодоступных регионах [15].

– *Интегрированное зондирование и связь (Integrated Sensing and Communication)* формирует совокупность задач в области позиционирования в сетях 6G и тесно связан с другими направлениями применения. Важно отметить, что данный сценарий устанавливает требование к точности определения местоположения объектов до 10 см.

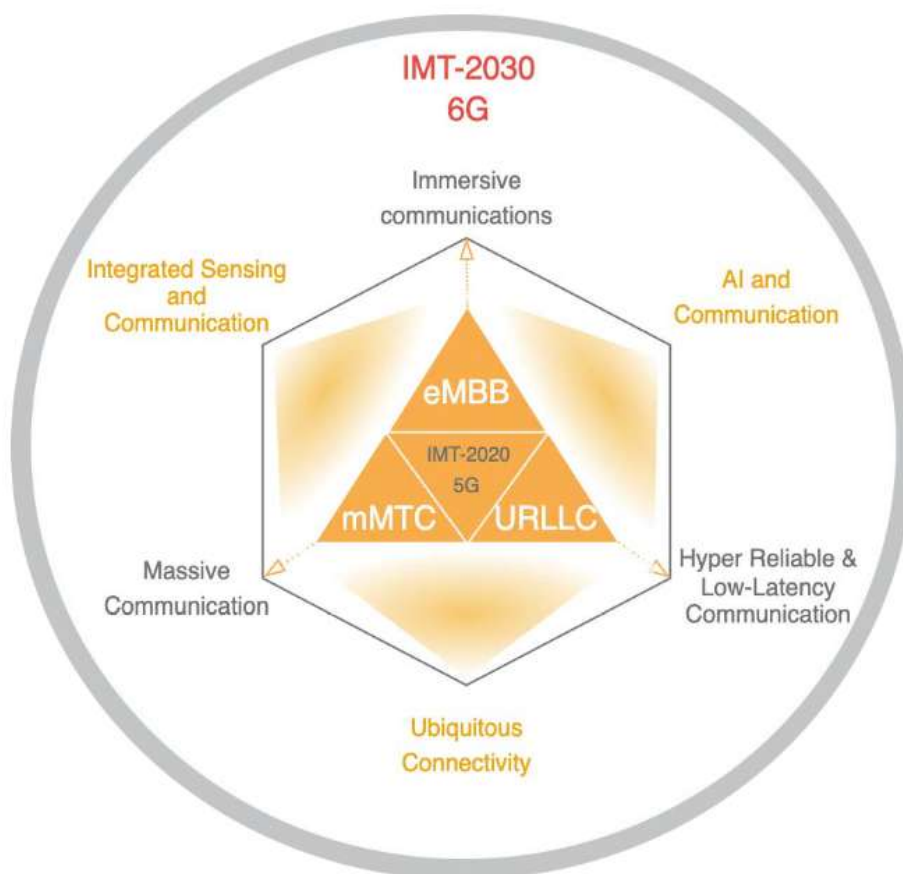


Рисунок 3 — Сценарии использования 6G (МСЭ-R M.2160)

Перечисленные сценарии показывают, что 6G выступает не просто как платформа связи, а становится интеллектуальной цифровой инфраструктурой, обслуживающей самые разные отрасли. По сравнению с 5G это качественно иной

уровень развития сетевых технологий. Для поддержки реализации описанных выше сценариев использования в рекомендации МСЭ-R М.2160 определён набор показателей технических возможностей, описывающих характеристики, которых должна достичь система 6G. Сравнительный анализ требований к техническим возможностям сетей 5G и 6G, основанный на рекомендациях МСЭ-T IMT-2020 [16] и IMT-2030 [1], представлен в таблице 1; при этом ряд показателей унаследован от предшествующего поколения, тогда как остальные являются принципиально новыми.

Таблица 1 — Показатели технических возможностей 5G и 6G

Показатели	5G	6G
Пиковая скорость передачи данных для пользователя	20 [Гбит/с]	[100-200 Гбит/с]
Скорость передачи данных пользователю - среднее значение на практике	1 [Гбит/с]	10 [Гбит/с]
Эффективность спектра	3x раза от IMT-A	1-3x раза от IMT-2020
Пропускная способность на квадрат площади (Мбит/с/м ²)	10	5-1000
Плотность подключенных устройств к сети (устройств/км ²)	10 ⁶	10 ⁶ - 10 ⁸
Мобильность устройств (км/ч)	500	500-1000
Задержка (мс)	1	0.1 - 1
Надежность	10 ⁻⁵	10 ⁻⁵ - 10 ⁻⁷
Безопасность, конфиденциальность и отказоустойчивость	—	Повышена
Сенсорные возможности	—	Новое требование/показатель
Возможности, связанные с ИИ	—	Новое требование/показатель
Устойчивость	—	Новое требование/показатель
Точность позиционирования	—	Повышена (1-10 [см])

Из приведённого сравнения видно, что 6G устанавливает значительно более высокие технические требования практически по всем показателям. Такие базовые параметры, как скорость передачи данных, задержка, надёжность и плотность подключений, существенно повышены для того, чтобы соответствовать растущим потребностям будущих приложений и сценариев использования. Помимо улучшения уже существующих характеристик, 6G вводит ряд совершенно новых требований, которые в предыдущем поколении не рассматривались: встроенные сенсорные возможности, интеграция искусственного интеллекта, устойчивость и повышенная точность позиционирования. Совокупность этих требований свидетельствует о качественном расширении концепции мобильных сетей следующего поколения. Целью становится уже не только повышение качества передачи данных, но и формирование интеллектуальной, отказоустойчивой инфраструктуры связи, способной поддерживать широкий спектр сервисов в гетерогенной среде.

Требования к производительности системы 6G включают скорость передачи данных в сотни гигабит в секунду, крайне малую задержку и возможность зондирования окружающей среды на уровне, недоступном для технологий сетей 4G и 5G. Поэтому развитие 6G невозможно строить исключительно на усовершенствовании предшествующих систем. Необходимы принципиальные прорывы в теории информации, технологиях радиосвязи и архитектуре сетей. В исследованиях МСЭ-R М.2516 [17] и МСЭ-R М.2541 [18] выделен ряд ключевых технологий, которые рассматриваются как фундамент будущей экосистемы 6G. К ним относятся освоение новых частотных диапазонов, радиоинтерфейс со встроенным искусственным интеллектом, связь, совмещённая с зондированием, а также сетевые архитектуры, объединяющие функции связи и вычислений:

– *Освоение новых частотных диапазонов (Терагерцовая связь и частоты выше 100 ГГц)*: Одна из главных проблем современных телекоммуникаций состоит в ограниченности традиционного радиочастотного спектра. Диапазоны ниже 6 ГГц используются практически полностью. Даже миллиметровые волны (mmWave), задействованные в сетях 5G, начинают сталкиваться с ограничениями по полосе и

эффективности использования спектра. Чтобы удовлетворить потребности приложений в сетях 6G в передаче больших объёмов данных, исследователи обращаются к диапазонам sub-THz (примерно 100–300 ГГц) и терагерцовым частотам (выше 300 ГГц). Как отмечается в отчёте МСЭ -R М.2541, развёртывание систем связи в этих диапазонах технически осуществимо и способно обеспечить непрерывную полосу пропускания в десятки гигагерц на один канал. Благодаря столь широкой полосе системы 6G смогут достигать пиковых скоростей от 50 до 200 Гбит/с без чрезмерной зависимости от сложных методов мультиплексирования или сильного сжатия данных. Кроме того, очень малая длина волны терагерцового сигнала позволяет создавать антенны малого размера. Это открывает возможность размещения систем со сверхмассивными антенными решётками с множественным вводом и выводом (Ultra-Massive Multiple Input Multiple Output) с сотнями и даже тысячами антенных элементов в одном устройстве, что позволяет формировать узконаправленные лучи и тем самым повышать эффективность передачи [19]. Вместе с тем терагерцовая связь сопряжена с серьёзными физическими трудностями. Среди них, значительное затухание сигнала в свободном пространстве, сильное поглощение водяным паром и молекулами газов атмосферы, а также экранирование сигнала препятствиями. Для решения этих проблем активно разрабатываются реконфигурируемые интеллектуальные поверхности (RIS, аббр. от англ. Reconfigurable Intelligent Surfaces), которые представляют собой интеллектуальные поверхности на основе метаматериалов, способные управлять фазой и амплитудой отражённых электромагнитных волн. С помощью RIS можно перенаправлять радиосигнал и создавать альтернативные пути передачи при блокировке прямого канала, что помогает поддерживать стабильное соединение в сложных условиях распространения.

– *Радиоинтерфейс с естественной архитектурой искусственного интеллекта (Artificial Intelligence Native Air Interface)*. Ещё одно важное направление развития 6G связано с проектированием радиоинтерфейса, в который искусственный интеллект интегрирован на уровне базовых принципов функционирования, так называемый AI-native-интерфейс. Данная концепция

активно обсуждается в рамках международной стандартизации и рассматривается как один из ключевых элементов перспективных систем 6G. В частности, на ВАСЭ-24 вопросам интеграции ИИ в телекоммуникационные системы было уделено значительное внимание, что дополнительно подтверждает актуальность данного направления [2]. В системах связи предыдущих поколений обработка сигналов (модуляция, кодирование и декодирование) строилась на фиксированных математических моделях, основанных на теории Шеннона. Такой подход не оставлял места для обучения и адаптации, которые сегодня обеспечивают методы машинного обучения [20]. В архитектуре 6G искусственный интеллект внедряется в радиointерфейс в рамках нового подхода к семантическим коммуникациям, который в литературе также обозначается как Post-Shannon communication [11]. В отличие от традиционных систем, ориентированных на точную передачу битов, такой подход оптимизирует передачу с учётом смысла данных и целей приложения. Интеграция ИИ в физический уровень 6G, по прогнозам, будет проходить в три этапа (таблица 2).

Таблица 2 — Три этапа интеграции ИИ в физический уровень 6G

Этап	Характеристика	Роль ИИ
Этап 1	Усиление блоков обработки сигналов	ИИ поддерживает такие функции, как оценка канала, синхронизация и декодирование
Этап 2	Интеграция нескольких блоков обработки	Модели глубокого обучения заменяют многие традиционные алгоритмы передачи
Этап 3	Проектирование физического уровня с использованием ИИ	Система передачи и приёма оптимизируется сквозным образом (end-to-end) с применением методов машинного обучения

– *Интегрированное зондирование и связь (ISAC, аббр. от англ. Integrated Sensing and Communication)*. В традиционных системах телекоммуникаций

функции связи и радиолокации проектировались как две независимые системы. Однако в 6G они объединены в общую архитектуру ISAC [21, 22]. В системе ISAC один и тот же радиосигнал одновременно используется для двух целей: передачи данных между устройствами и получения информации об окружающей среде на основе отражённых сигналов. Подход повышает эффективность использования спектра и инфраструктуры, расширяя области применения. Поиск компромисса между характеристиками радиолокации и связи является важной задачей при реализации ISAC. Радиолокация требует высокой автокорреляции сигнала, связь — высокой спектральной эффективности. Для преодоления этого противоречия разработан сигнал OTFS (Orthogonal Time Frequency Space), устойчивый к доплеровскому сдвигу при высоких скоростях движения объектов. Данный сигнал может быть использован для передачи в условиях высокой скорости движения с большим эффектом Доплера. Базовые станции 6G могут использоваться как крупномасштабные датчики окружающей среды благодаря технологии ISAC. Это позволит сети выполнять такие задачи, как создание карт окружающей среды, точное определение местоположения устройств и обнаружение и отслеживание объектов в пространстве. В ряде случаев система ISAC способна обеспечить достаточную точность позиционирования без обращения к спутниковым навигационным системам.

1.3 Переход к новой архитектуре 6G: сеть связи с искусственным интеллектом

Стремительный рост числа приложений, требующих малой задержки и обработки данных в реальном времени, стал одним из ключевых факторов трансформации архитектуры телекоммуникационных сетей. В традиционных системах функции связи и вычислений, как правило, реализовывались на отдельных инфраструктурах. Однако в рамках 6G наблюдается тенденция к их глубокой интеграции, в результате чего формируется единая платформа информационно-коммуникационных технологий (ICT, *аббр. от англ.* Information and Communication Technology), ориентированная на поддержку приложений

реального времени [23]. Концепция сети связи с искусственным интеллектом (AI-enabled telecommunications network) получила нормативное закрепление в Резолюции 101 ВАСЭ-24, которая определила её как телекоммуникационную сеть, в которой технологии ИИ интегрированы в качестве нативных компонентов на всех уровнях архитектуры с целью повышения эффективности, возможностей и надёжности систем связи. Тем самым на международном уровне был подтверждён переход от использования ИИ как вспомогательного инструмента к его роли как системообразующего элемента инфраструктуры связи.

В этом контексте сети 6G всё чаще рассматриваются как интеллектуальная среда, в которой сеть связи выполняет не только функцию передачи данных, но и выступает в роли распределённой вычислительной и интеллектуальной платформы [24, 25]. Такая концепция предполагает переход от модели «связь как услуга» к модели «сеть как платформа распределённого интеллекта»; согласно этому подходу, экосистема 6G включает три взаимосвязанных компонента:

Таблица 3 — Три основные составляющие экосистемы 6G

Компонент	Роль
Передовая беспроводная инфраструктура	Обеспечивает сверхвысокоскоростное соединение
Интегрированный искусственный интеллект	Управляет сетью и оптимизирует её
Интернет всего	Обеспечивает взаимосвязь между людьми, устройствами и окружающей средой

Сочетание указанных компонентов позволяет сетям 6G поддерживать новые классы сервисов, включая мобильный интеллект, телеприсутствие, передовые приложения цифрового здравоохранения, голографические коммуникации и сервисы, связанные с концепцией метавселенной. В рамках таких сценариев устройства и системы способны автономно собирать, анализировать и обмениваться данными в реальном времени [26, 27].

Реализация таких сервисов требует соответствующей архитектурной поддержки. Важной особенностью 6G является развитие граничных вычислений и туманных вычислений, а также распределённого хранения данных. В отличие от традиционного подхода, при котором обработка данных осуществляется преимущественно в централизованных центрах обработки данных, в системах 6G значительная часть вычислений переносится на сетевые узлы, расположенные вблизи пользователя. Применение граничных вычислений, граничного кэширования и FC позволяет снизить нагрузку на ядро сети, уменьшить задержки передачи данных и повысить качество обслуживания приложений реального времени [28, 29]. Кроме того, эта архитектура служит основой для создания цифровых двойников, то есть цифровых моделей реальных физических систем. Цифровые двойники улучшают процесс принятия решений, моделируя и прогнозируя функционирование сложных систем, таких как промышленные предприятия, города. Наиболее показательным примером практической реализации цифровых двойников в контексте 6G является концепция CitiVerse. CitiVerse представляет собой частный вид метавселенной, построенной на сети цифровых двойников городских объектов, включая дорожную инфраструктуру и городские транспортные потоки (например, автомобильный трафик), функционирование которой требует глубокой интеграции туманных вычислений и федеративного обучения.

По мере роста числа подключённых устройств и расширения спектра цифровых услуг сложность системы 6G стремительно возрастает. Традиционные методы управления сетью, основанные на участии человека или статической автоматизации по заданным правилам, уже не обеспечивают необходимой эффективности. Разнообразие сервисов, охватывающих как дистанционную хирургию с минимальной задержкой, так и промышленный IoT с огромной плотностью подключений, требует от сети способности к самостоятельной настройке и оптимизации в режиме реального времени. Именно эта потребность обусловила переход к концепции нативной интеграции искусственного интеллекта в архитектуру сети, закреплённой в Рекомендации МСЭ-Р М.2160 в качестве

сценария «Искусственный интеллект в связи» (Artificial Intelligence and Communication), являющегося одним из шести базовых сценариев использования сетей 6G. Помимо определения сценариев использования, Рекомендация МСЭ-Р М.2160 закрепила четыре основополагающих принципа проектирования ИМТ-2030, применимых ко всем сценариям: повсеместный интеллект, устойчивость, безопасность и отказоустойчивость, а также подключение неохваченных пользователей. Принцип повсеместного интеллекта непосредственно определяет архитектурную роль туманных вычислений и федеративного обучения как механизмов размещения интеллекта на всех уровнях сетевой инфраструктуры. Принципиальное отличие этого сценария от подхода, реализованного в сетях 5G, состоит в том, что ИИ перестаёт быть инструментом оптимизации отдельных функций и становится частью самой инфраструктуры. Данный сценарий предполагает реализацию принципиально новых сетевых возможностей: сбор и подготовку данных из распределённых источников, обучение моделей ИИ непосредственно на узлах сети, распределённый вывод результатов модели, обмен моделями между участниками и оркестрацию вычислительных ресурсов. Таким образом, сети связи шестого поколения рассматриваются не просто как среда передачи данных, а как активная вычислительная платформа, способная поддерживать полный жизненный цикл систем ИИ от сбора данных до развёртывания и обновления моделей.

Реализация указанных возможностей в условиях реальной инфраструктуры предъявляет особые требования к архитектуре вычислительного слоя сети. Ни полностью централизованная облачная модель, ни исключительно граничная не способны обеспечить одновременно требуемую вычислительную мощность и сверхнизкую задержку. В этих условиях туманные вычисления и федеративное обучение приобретают роль ключевых технологических инструментов: первые формируют промежуточный вычислительный слой между облаком и конечными устройствами, вторые обеспечивают децентрализованное обучение моделей ИИ без централизованной передачи данных. Совместное применение этих технологий

открывает возможность для практической реализации принципа AI-native в распределённой и гетерогенной инфраструктуре сетей шестого поколения.

МСЭ-Т Y.3172 (Y.3177) — Техническая основа для ИИ в сетях шестого поколения

МСЭ-Т Y.3172 «Architectural Framework for Machine Learning in Future Networks including IMT-2020» является первой базовой рекомендацией МСЭ-Т, устанавливающей стандартизированную архитектуру для встраивания машинного обучения в телекоммуникационную инфраструктуру [30]. Кроме того, именно этот документ содержит исходное определение машинного обучения, на которое напрямую ссылаются все последующие рекомендации серии, посвящённые применению ИИ в телекоммуникационных сетях.

Y.3172 отвечает на принципиальный технический вопрос: каким образом компоненты машинного обучения, изначально рассчитанные на среду централизованных вычислений, могут быть системно интегрированы в распределённую, неоднородную и постоянно меняющуюся сетевую инфраструктуру IMT-2020 [31]? Предложенное решение состоит в определении наложенной архитектуры с высокой степенью технологической нейтральности. Логическая архитектура строится на уровне абстракции, независимо от конкретных сетевых технологий (3GPP, MEC, EdgeX и других), что позволяет затем создавать на её основе конкретные технологические реализации.

Центральным и наиболее широко цитируемым вкладом рекомендации МСЭ-Т Y.3172 является понятие ML-конвейера (Machine Learning Pipeline), цепочки логических функциональных узлов, описывающей поток обработки от необработанных данных до управляющего воздействия на сеть. ML-конвейер состоит из семи типов функциональных узлов с чётко разграниченными ролями, представленных на рисунке 5.

Узел-источник (Source) представляет собой источник необработанных данных, например журналы маршрутизаторов, значения ключевых показателей эффективности (KPI, *от англ.* Key Performance Indicators) базовых станций или данные телеметрии сетевых компонентов. Узел-сборщик (Collector) отвечает за

сбор данных от одного или нескольких узлов-источников и может выполнять первичную фильтрацию и агрегацию. Узел предварительной обработки (Preprocessor) осуществляет предварительную обработку данных, включая очистку, нормализацию и объединение. Узел модели (Model) является ядром ML-конвейера и выполняет обучение или логический вывод модели ML. Узел политики (Policy) применяет бизнес-правила к результатам работы модели. Например, если модель предсказывает вероятность перегрузки на уровне 80%, узел политики решает, следует ли активировать перенаправление трафика, исходя из ограничений соглашения об уровне обслуживания (SLA, *от англ.* Service Level Agreement) и доступных сетевых ресурсов. Узел-распределитель (Distributor) определяет целевые узлы-приёмники и направляет выходные данные узла модели к соответствующим получателям. Узел-приёмник (Sink) является конечной точкой ML-конвейера, где действие выполняется над сетью, например отправка команды конфигурации контроллеру программно-конфигурируемой сети (Software-Defined Networking).

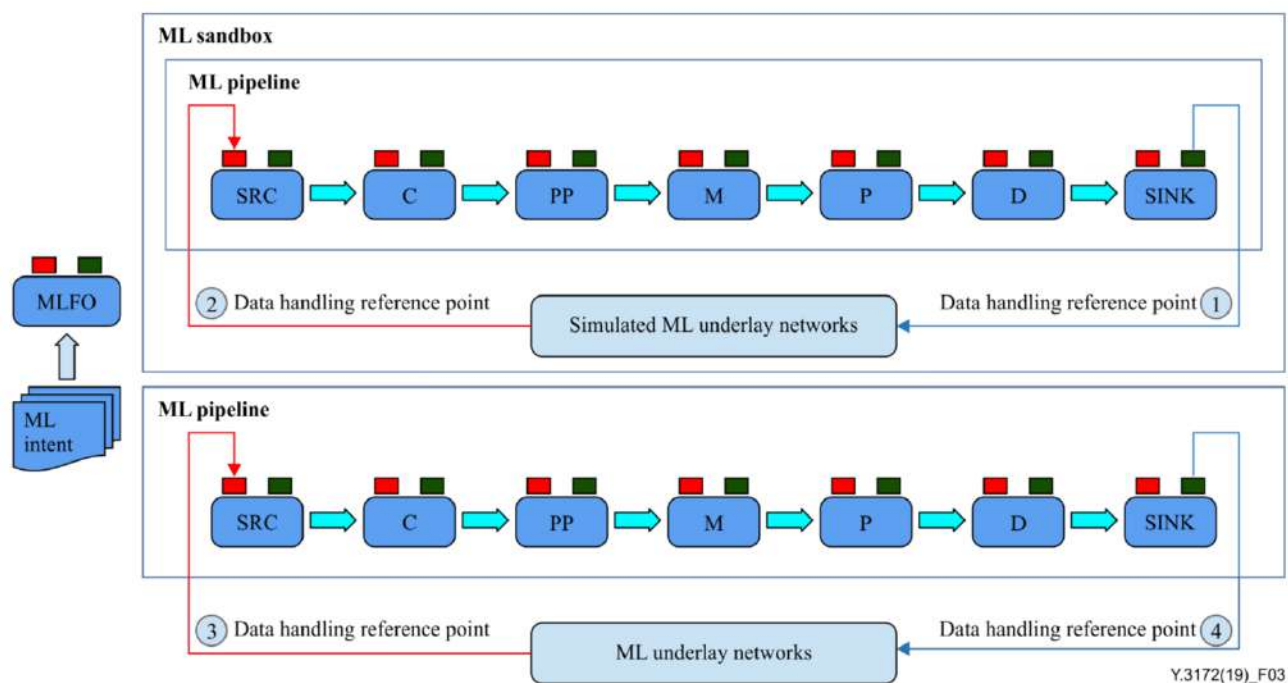


Рисунок 5 — Высокоуровневые архитектурные компоненты системы машинного обучения в рекомендациях МСЭ-Т Y.3172

Наиболее важной технической особенностью ML-конвейера в рекомендации МСЭ-Т Y.3172 является возможность декомпозиции и цепочного связывания узлов. Узлы ML-конвейера могут размещаться на различных уровнях сетевой инфраструктуры, включая оконечные устройства, граничные серверы, узлы туманных вычислений и центры обработки данных. Более того, несколько ML-конвейеров могут объединяться в цепочки, где выход одного конвейера становится входом следующего. Это позволяет строить сложные многоуровневые аналитические системы ML, что особенно важно при распределённом развёртывании по архитектуре централизованного и распределённого узлов базовой станции следующего поколения (gNodeB-CU/gNodeB-DU). Помимо ML-конвейера, рекомендация МСЭ-Т Y.3172 определяет три важных инфраструктурных компонента. Оркестратор функций ML (MLFO, аббр. от англ. ML Function Orchestrator) является функциональной сущностью системы. Он управляет полным жизненным циклом ML-конвейера. В его задачи входит развёртывание конвейера и мониторинг его производительности. При снижении точности модели ниже допустимого порога MLFO инициирует её переобучение или замену. MLFO взаимодействует с оркестратором виртуализации сетевых функций (NFVO, аббр. от англ. Network Functions Virtualization Orchestrator) и может быть реализован как встроенная функциональная подсистема в составе NFVO.

– *Песочница ML* представляет собой изолированную программную среду, предназначенную для обучения, тестирования и оценки моделей ML без какого-либо воздействия на функционирующую сеть. Согласно рекомендации МСЭ-Т Y.3172, песочница МО обеспечивает генерацию имитационных данных. Оценка устойчивости модели с помощью специализированного тестирования на робастность и планирование экспериментальных операций. В дополнение к этому, песочница ML обеспечивает контроль качества данных и выполнение проверок корректности для имитационной ML-сети, что является необходимым условием обеспечения надёжности модели перед ее развёртыванием в реальной сетевой инфраструктуре.

– *Намерение ML* позволяет администратору описывать желаемые цели на уровне бизнес-задач, не задавая конкретные параметры алгоритмов. Затем MLFO интерпретирует намерение ML и выбирает конвейер, модель и стратегию обучения, подходящие для этого. Концепция сетей на основе намерений (англ. Intent-Based Networks) из рекомендации МСЭ-Т Y.2344, которая формирует единый поток управления от бизнес-уровня до уровня исполнения ML, напрямую связана с целью машинного обучения.

Основное практическое указание рекомендации МСЭ-Т Y.3172, непосредственно относящееся к архитектуре FC, рассматриваемой далее, касается разделения функций: обучение модели, накопление знаний, обычно выполняется на облачных серверах, поскольку не имеет строгих временных ограничений, но требует большой вычислительной мощности. В то же время логический вывод, применение знаний, для которого необходима работа в реальном времени, выполняется на граничных узлах с использованием уже полученных знаний. Такое разделение функций точно соответствует вычислительной модели FC и создаёт стандартизированную основу для проектирования систем иерархического FL в будущем.

МСЭ-Т Y.3177 «Architectural Framework for Artificial Intelligence-Based Network Automation for Resource and Fault Management in Future Networks including IMT-2020», утверждённая в феврале 2021 года, выполняет роль важного технического связующего звена между общей архитектурой машинного обучения Y.3172. Область действия Y.3177 сосредоточена на двух критически важных направлениях эксплуатации: автоматизация управления сетевыми ресурсами на основе ИИ и интеллектуальное управление сбоями. Одним из заметных вкладов данной рекомендации является предложение единой логической архитектуры. Это общая структура, не привязанная к какой-либо конкретной технологии, позволяющая чётко определить понятия, терминологию и способы взаимосвязи функций ML. При применении этой структуры к конкретной технологии разработчик системы может без труда построить подходящую архитектуру развёртывания.

Рекомендация МСЭ-Т Y.3177 определяет процесс автоматического управления сбоями на основе ИИ, состоящий из трёх взаимосвязанных этапов в рамках замкнутого контура управления. Первый этап, обнаружение аномалий с использованием моделей обучения без учителя для выявления нетипичных паттернов в сетевой телеметрии. Второй этап, анализ первопричин (Root Cause Analysis), результаты которого предоставляются для контроля, но и напрямую автономным функциям для настройки NFVO. Третий этап, самовосстановление, при котором система автоматически выбирает и выполняет корректирующие действия из набора заранее изученных рабочих процессов восстановления. Результаты самовосстановления оцениваются и используются в качестве обратной связи для совершенствования последующих процессов восстановления. Рекомендация МСЭ-Т Y.3177 прямо указывает на необходимость непрерывного улучшения автоматизированных процессов восстановления после сбоев, что непосредственно соответствует принципу повсеместного интеллекта, закреплённому в Рекомендации МСЭ-Р М.2160 в качестве одного из базовых принципов проектирования сетей связи с искусственным интеллектом IMT-2030.

Помимо процессов управления сбоями, в Рекомендации МСЭ-Т Y.3172 детально описана модель многоуровневого развёртывания ML-конвейера, допускающая распределение его компонентов между несколькими уровнями инфраструктуры с возможностью цепочного связывания узлов в зависимости от требований конкретного варианта использования.

МСЭ-Т Y.3173 — Система оценки уровней интеллекта L_0 – L_5

Рекомендация МСЭ-Т Y.3173 «Концепция оценки уровней интеллекта будущих сетей, включая IMT-2020», утверждённая 6 февраля 2020 года, построена непосредственно на архитектурной базе Рекомендации МСЭ-Т Y.3172 и формулирует методологию количественной оценки, сравнения и задания целевого уровня интеллектуальности различных сетевых систем [32]. Предложенная методология основана на пяти универсальных функциональных измерениях и совместима с аналогичными отраслевыми моделями, такими как шкала Форума управления телекоммуникациями (TM Forum, от англ. TeleManagement Forum) для

интеллектуальных сетей, шкала Общества автомобильных инженеров (SAE, от англ. Society of Automotive Engineers) для автономных транспортных средств и другие системы оценки сетевого интеллекта. Согласно Рекомендации МСЭ-Т Y.3173, конечной целью интеллектуальной сетевой системы является достижение полной автоматизации на уровне L_5 , при котором сеть функционирует полностью самостоятельно без вмешательства человека, что соответствует принципу повсеместного интеллекта, закреплённому в Рекомендации МСЭ-Р М.2160. Однако переход к этому уровню не происходит одномоментно, поскольку каждый уровень обладает собственной практической ценностью и может быть развёрнут самостоятельно в реальных сценариях применения сетей связи шестого поколения.

Рекомендация включает пять измерений, основанных на распределении ответственности между людьми и системой:

– *Измерение 1, отображение потребностей*: требования к сервисам могут быть автоматически преобразованы в технические параметры системы и распределены по соответствующим сетевым компонентам без участия оператора.

– *Измерение 2, сбор данных*: система автоматически собирает данные о состоянии сети, что позволяет обеспечить необходимую частоту, точность и быстрое обнаружение отклонений.

– *Измерение 3, анализ*: система обрабатывает данные самостоятельно, обнаруживает тенденции, обнаруживает аномалии и делает выводы без помощи человека.

– *Измерение 4, принятие решений*: оценивает способность системы самостоятельно выбирать и утверждать корректирующие или оптимизационные действия на основе результатов анализа.

– *Измерение 5, выполнение действий*: оценивает, насколько система самостоятельно выполняет принятые решения без помощи человека, включая сложные и опасные ситуации.

На основе распределения ответственности между системой и человеком по пяти указанным измерениям рекомендация Y.3173 определяет шесть уровней интеллекта от L_0 до L_5 , представленных в таблице 4:

Таблица 4 — Уровни интеллекта сети по рекомендации МСЭ-Т Y.3173 и распределение ответственности между системой и человеком

Название	Отображение потребностей	Сбор данных	Анализ	Принятие решений	Исполнение
Ручное управление L_0	Человек	Человек	Человек	Человек	Человек
Поддержка L_1	Человек	Человек и система	Человек	Человек	Человек
Базовый интеллект L_2	Человек	Система	Человек и система	Человек	Человек
Средний уровень интеллекта L_3	Человек	Система	Система	Человек и система	Человек и система
Продвинутый интеллект L_4	Человек	Система	Система	Система	Система
Полная автономность L_5	Система	Система	Система	Система	Система

На уровне L_0 все пять функций выполняет человек, а система лишь предоставляет базовые средства отображения информации. На уровнях L_1 и L_2 система постепенно берёт на себя функции сбора данных и помогает в выполнении действий, однако анализ и принятие решений по-прежнему остаются за человеком. Это типичный уровень для самоорганизующихся сетей в существующих сетях 4G и 5G. На уровне L_3 система полностью автоматизирует анализ и начинает разделять с человеком ответственность за принятие решений и их выполнение. На уровне L_4 система самостоятельно управляет сетью в подавляющем большинстве сценариев, включая ранее не встречавшиеся ситуации. Человек лишь задаёт высокоуровневые цели и осуществляет общий контроль. Единственное измерение, в котором человек сохраняет ведущую роль, это отображение потребностей, то есть интерпретация

бизнес-требований в сетевые цели. На уровне L_5 система самостоятельно выполняет и функцию отображения потребностей: она сама определяет оптимальные цели на основе глобального состояния и потребностей сервисов без какого-либо участия человека. Это долгосрочное видение сетей ИМТ-2030.

МСЭ-Т Y.3400 и Y.3224 — Объединение вычислений и сетей, федеративное обучение

Рекомендация МСЭ-Т Y.3400 «Coordination of Networking and Computing in IMT-2020 Networks and Beyond - Requirements», утверждённая в декабре 2023 года, системно определяет технические требования к модели, в которой сетевые и вычислительные ресурсы управляются как единое целое, а не как две параллельные и независимые инфраструктуры, что было свойственно традиционным архитектурам [33]. В рамках такого подхода ИМТ-2030 рассматривается уже не как канал передачи данных, а как распределённая вычислительная платформа, где каждый сетевой узел одновременно является точкой завершения соединения и полноценной вычислительной единицей.

Рекомендация МСЭ-Т Y.3400 организует технические требования в четыре функциональные группы, связанные строгой иерархией.

– *Группа требований к единому измерению ресурсов* устанавливает, что система координации сети и вычислений (CNC, аббр. от англ. Computing and Networking Coordination) должна уметь измерять и представлять вычислительные, сетевые ресурсы и ресурсы хранения в рамках единой модели показателей, независимо от неоднородности физической инфраструктуры. Измеряемые показатели включают пропускную способность, задержку, джиттер, вычислительную мощность процессора и графического процессора, объём хранения и текущий уровень нагрузки.

– *Группа требований к обнаружению и идентификации ресурсов* определяет механизмы автоматического обнаружения, регистрации и отслеживания состояния ресурсов CNC в реальном времени. Сюда входят уникальная идентификация каждого ресурса, мониторинг его свойств и рабочего состояния, обнаружение

аномалий в поведении ресурсов и ведение истории производительности для поддержки планирования и управления.

– *Группа требований к совместному планированию* предписывает, что вместо отдельной оптимизации сетевых и вычислительных ресурсов, как это принято в традиционных системах, система CNC должна быть способна одновременно и согласованно распределять оба вида ресурсов от различных поставщиков. По своей сути эта задача представляет собой многокритериальную оптимизацию с ограничениями из области связи (задержка, пропускная способность, потери пакетов) и из области вычислений (время обработки, энергопотребление, доступность).

– *Группа требований к единому управлению* определяет механизмы регистрации ресурсов, мониторинга состояния системы, автоматического восстановления после сбоев и координации ресурсов между различными доменами управления. Это включает обработку нештатных ситуаций: перегрузку ресурсов, потерю связи с узлом и резкие изменения в структуре трафика.

Y.3400 дополняется рекомендацией МСЭ-Т Y.3401 (2024) «Framework for Coordination of Networking and Computing Capability in IMT-2020 Networks and Beyond», которая предоставляет соответствующую структуру возможностей [34]. Если Y.3400 определяет, что система CNC должна уметь делать, то Y.3401 определяет, каким образом она это реализует. Пара Y.3400/Y.3401 образует полную стандартную основу для проектирования и оценки любой системы, объединяющей вычисления и сеть, включая архитектуру FC для IMT-2030, предлагаемую в данной диссертации.

МСЭ-Т Y.3224 — Федеративное обучение с агрегацией в сети

Рекомендация МСЭ-Т Y.3224 «Федеративное обучение с агрегацией в сети в конвергентных фиксированных, мобильных и спутниковых сетях», утверждённая в августе 2025 года, посвящена определению архитектурной основы и требований к внутрисетевому FL с агрегацией (InNetFL, аббр. от англ. In-Network aggregated Federated Learning) в среде конвергентных фиксированных, мобильных и спутниковых сетей [35].

В обычных системах FL каждое клиентское устройство отправляет обновление локально обученной модели на центральный сервер для глобальной агрегации. Однако если этот процесс выполняется непосредственно внутри сети на промежуточных узлах, необходимость передачи каждого отдельного обновления через всю сеть значительно сокращается. Используя ресурсы периферийных и туманных вычислений, а также промежуточные узлы с вычислительными возможностями, многозвенная сеть может выполнять агрегацию моделей на каждом промежуточном узле перед передачей данных на следующий участок или на центральный сервер.

Рекомендация также устанавливает важные функциональные требования: управление смещением и справедливостью при обучении, механизмы обработки ошибок и контроля перегрузки, протоколы взаимодействия и структуру управления между участниками. Кроме того, концепция InNetFL предусматривает непрерывный мониторинг производительности системы FL для выполнения заданных требований к точности. Это существенный шаг в стандартизации развёртывания распределённых систем ИИ на телекоммуникационной инфраструктуре следующих поколений. МСЭ-Т Y.3224 также определяет архитектуру InNetFL на основе трёх чётко разграниченных функциональных ролей. Клиентом FL является окончное устройство в конвергентной фиксированной, мобильной и спутниковой сети (FMSC, аббр. от англ. Fixed, Mobile and Satellite Converged networks), например устройство IoT или мобильное устройство, которое выполняет локальное обучение модели на собственных данных и передаёт обновление модели ближайшему внутрисетевому агрегатору, а не центральному серверу. Внутрисетевой агрегатор представляет собой промежуточный узел сети, обычно граничный сервер или узел FC. Он выполняет локальную агрегацию обновлений от группы ближайших клиентов FL и передает результат на более высокий уровень. Для организации глобальной агрегации глобальный координатор объединяет результаты нескольких внутрисетевых агрегаторов и формирует глобальную модель. Затем эта модель распространяется на нижестоящие уровни.

Архитектура InNetFL с указанием трех функциональных ролей и направлений передачи обновлений модели показана на рисунке 6.

Сравнение трехуровневых архитектур InNetFL из рекомендации МСЭ Т Y.3224 и стандартной трехуровневой архитектуры FC показывает основной технический факт: узел FC является наиболее естественной физической единицей для размещения внутрисетевого агрегатора. Устройства IoT и оконечные устройства соответствуют клиентам FL. Ближайшие узлы FC относятся к внутрисетевым агрегаторам, которые выполняют локальную агрегацию. Верхние узлы FC или облако относятся к глобальному координатору, который выполняет глобальную агрегацию. Именно это обстоятельство служит основанием и сильным стимулом для того, чтобы в данной работе сосредоточить внимание на интеграции FC в архитектуру, предлагаемую в последующих разделах.

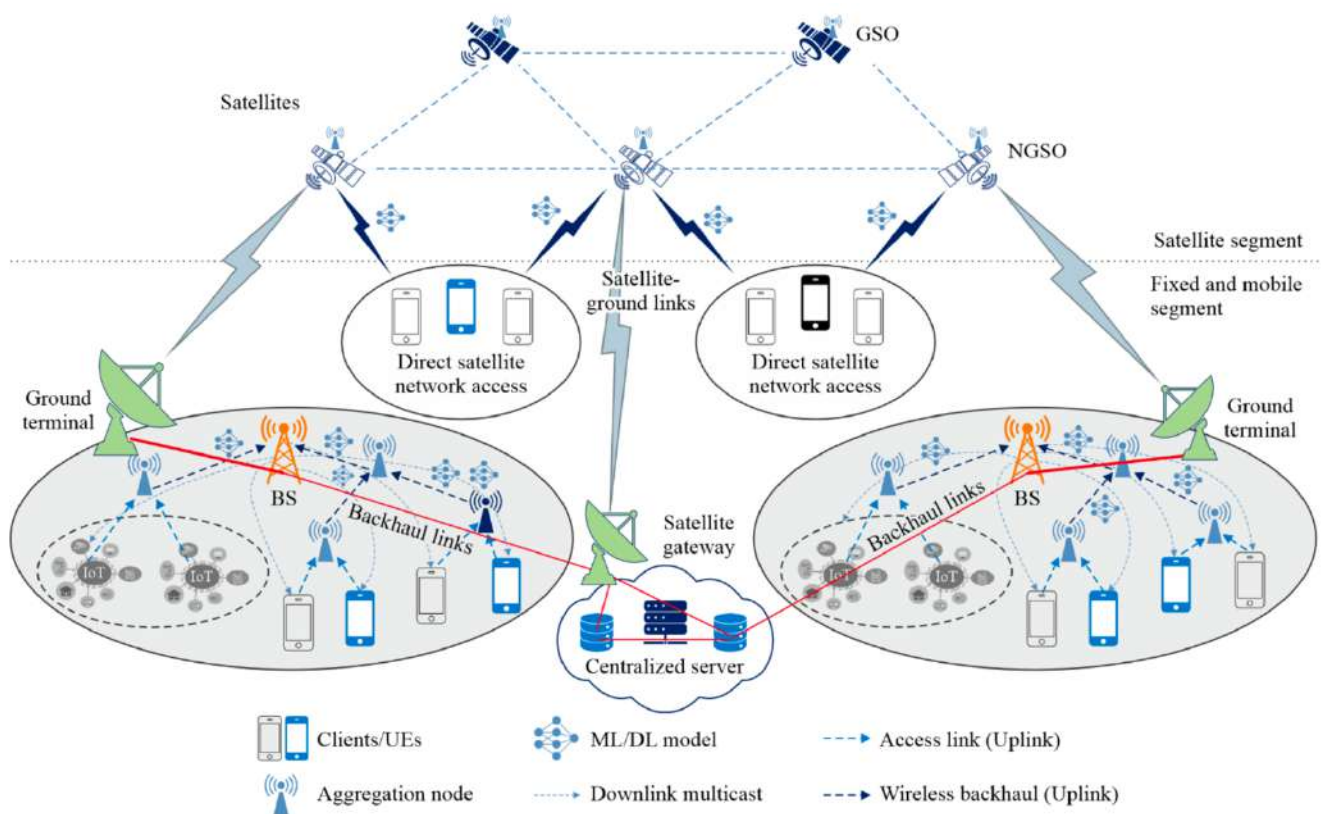


Рисунок 6 — Обзор архитектуры федеративного обучения с внутрисетевой агрегацией в FMSC

1.4 Туманные вычисления: концепция, архитектура и роль в сетях шестого поколения

До широкого распространения облачных вычислений предприятия были вынуждены самостоятельно приобретать и обслуживать собственную вычислительную инфраструктуру, системы хранения данных и программное обеспечение. Облачные вычисления представляют собой модель предоставления удобного сетевого доступа по требованию к пулу конфигурируемых вычислительных ресурсов, которые могут оперативно выделяться и освобождаться при минимальном участии поставщика услуг и пользователя [36]. Данная модель позволяет существенно снизить как первоначальные капитальные затраты, так и текущие эксплуатационные расходы по сравнению с традиционным подходом, при котором организация самостоятельно приобретает и обслуживает физические серверы, системы хранения данных и сетевое оборудование.

Вместе с тем по мере роста числа подключённых устройств и распространения приложений реального времени централизованная передача всех данных в удалённые центры обработки обнаруживает всё более серьёзные ограничения. Во-первых, задержка передачи: географическое расстояние между конечным устройством и центральным облаком является непосредственной причиной больших задержек, что критически сказывается на работе приложений, чувствительных к времени отклика [37]. Приложения нового поколения, определённые в рамках сетей 6G, включают управление беспилотными транспортными средствами, телеприсутствие, иммерсивные сервисы расширенной реальности и голографическую связь. Для таких приложений характерны жёсткие требования к сквозной задержке, которая в зависимости от сценария использования может составлять от единиц миллисекунд до значений порядка 1 мс [38]. Удовлетворение данного требования в рамках централизованной модели облачной обработки в большинстве случаев оказывается затруднительным. Во-вторых, пропускная способность: непрерывный рост объёма данных, генерируемых на периферии сети, превращает каналы магистральной сети в узкое место, поскольку

их пропускная способность не увеличивается пропорционально. В-третьих, конфиденциальность: данные, собираемые конечными устройствами, нередко носят чувствительный характер, и локальная обработка на периферии сети обеспечивает заметно более высокий уровень защиты частной информации пользователей по сравнению с передачей необработанных данных в облако.

Туманные вычисления возникли как архитектурный ответ на ограничения централизованной облачной модели. В отличие от традиционного подхода, FC переносит вычислительные ресурсы, хранение данных и сетевые функции на промежуточный уровень инфраструктуры, физически расположенный ближе к источникам данных. В результате формируется распределённая среда обработки, охватывающая континуум «облако – туманные узлы – пользовательские устройства», в которой каждый уровень выполняет задачи в соответствии со своими вычислительными возможностями и требованиями к задержке. При этом FC не заменяет облачную модель, а дополняет её: облако сохраняет роль центра стратегической координации и долгосрочного хранения, тогда как туманный уровень берёт на себя задачи, требующие оперативного отклика.

Концепция туманных вычислений была предложена исследователями компании Cisco и получила формализованное описание в научной литературе в 2012 году. В основополагающей работе Бономи и соавторы определили FC как «распределённую платформу виртуализации, обеспечивающую вычисления, хранение данных и сетевое взаимодействие между конечными устройствами и традиционными облачными центрами обработки данных» [39]. Это определение акцентирует три фундаментальных свойства: географическую распределённость, промежуточное положение в континууме «облако – пользовательские устройства» и роль дополнения к облаку. Помимо технических характеристик, FC обладают рядом значимых свойств: осведомлённостью о географическом положении (location awareness), поддержкой мобильности (mobility support), взаимодействием в режиме реального времени (real-time interaction), неоднородностью инфраструктуры (heterogeneity), межплатформенной совместимостью

(interoperability), а также интеграцией аналитики с облаком (analytics and cloud integration).

Базовая эталонная архитектура среды FC, предложенная Cisco и впоследствии стандартизированная консорциумом OpenFog, включает три уровня. При необходимости она расширяется до четырёх уровней за счёт добавления шлюзового уровня IoT (IoT gateway) между конечными устройствами и туманными узлами. Функции каждого из трёх уровней архитектуры FC разграничены следующим образом:

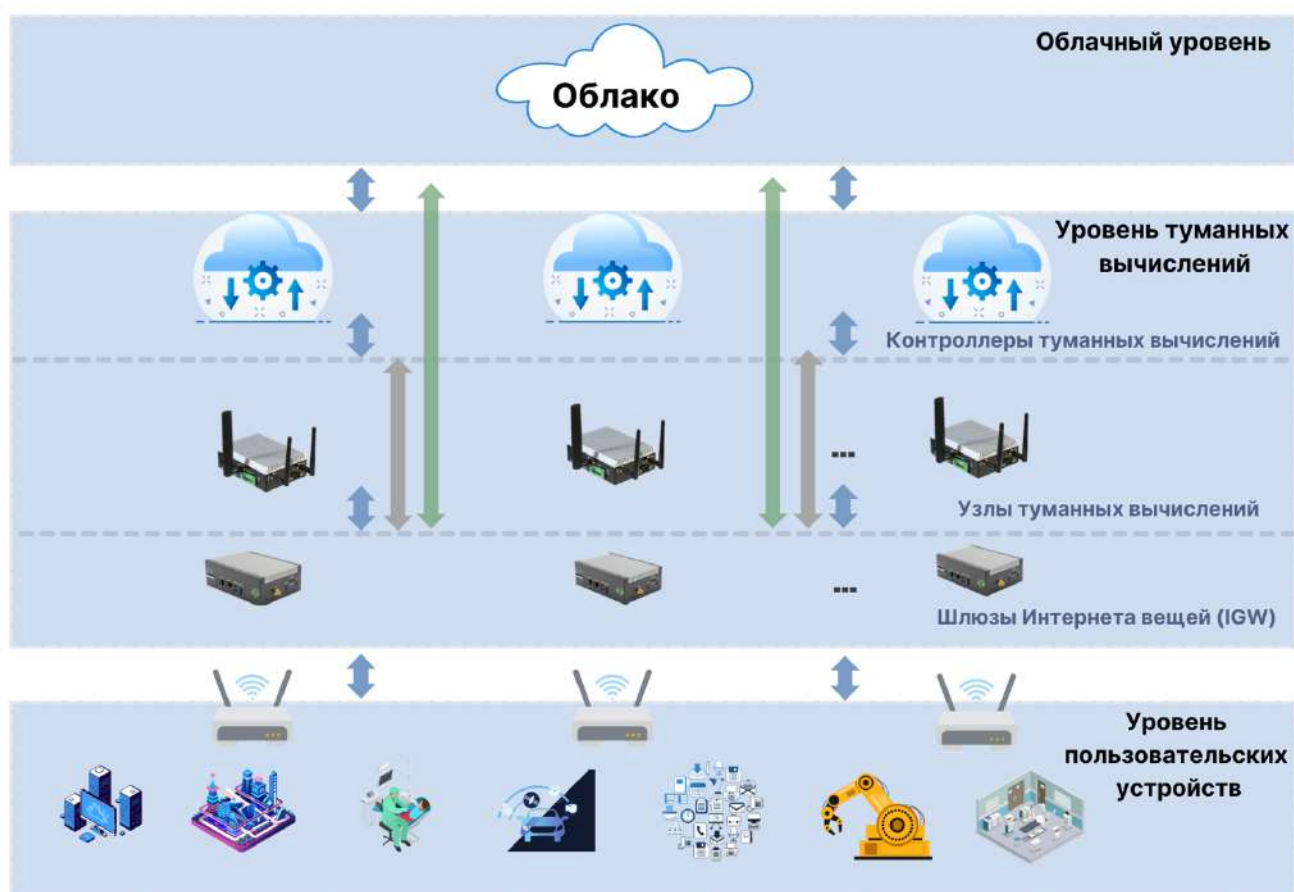


Рисунок 7 — Архитектура инфраструктуры «облако – туманные узлы – пользовательские устройства»

– *Уровень пользовательских устройств.* К этому уровню относятся устройства IoT, датчики, интеллектуальные транспортные средства, носимые гаджеты и мобильные телефоны. Здесь осуществляется сбор первичных данных, выполняется базовая локальная обработка и формируются запросы на вычисления.

Устройства данного уровня, как правило, ограничены по энергопотреблению, вычислительным ресурсам и объёму памяти.

– *Уровень туманных вычислений.* Промежуточный уровень образован туманными узлами: точками доступа Wi-Fi, маршрутизаторами, fog-серверами, коммутаторами, базовыми станциями, встраиваемыми компьютерами и аналогичными устройствами, размещёнными на периферии сети. Туманные узлы выполняют обработку, фильтрацию, агрегацию и анализ данных от нижестоящего уровня, а также служат точкой координации с вышестоящим облаком. Внутренняя организация уровня FC является многоуровневой иерархией: при обнаружении события датчиком данные передаются на туманный узел первого порядка; если локальных ресурсов недостаточно, этот узел перенаправляет данные на туманный узел второго порядка (региональный fog-сервер), и так далее по цепочке. Стандарт OpenFog RA определяет число уровней N как гибкий параметр, зависящий от масштаба конкретного развёртывания [40]. В этом состоит принципиальное отличие от ETSI MEC: MEC ориентирован на один-два уровня на периферии операторской сети, тогда как FC допускает организацию более гибкой распределённой цепочки вдоль всего континуума «облако – пользовательские устройства» [41].

– *Облачный уровень.* Этот уровень представлен высокопроизводительными серверами в крупных центрах обработки данных. На него возлагаются задачи, не чувствительные к задержке: долгосрочный стратегический анализ данных, крупномасштабное обучение моделей ИИ, хранение исторических данных и глобальная координация. Роль облачного уровня при этом не сокращается, а переосмысливается: из единственной точки обработки оно превращается в координирующий центр в непрерывной экосистеме «облако – туманные узлы».

Таким образом, FC обеспечивают обработку данных непосредственно у источника их генерации, что минимизирует задержку и снижает нагрузку на каналы передачи данных. Совокупность перечисленных свойств определяет FC как ключевой инфраструктурный компонент для реализации распределённого ИИ в сетях ИМТ-2030.

По сравнению с облачными вычислениями FC обладают рядом отличительных характеристик, которые позволяют FC непосредственно удовлетворять требованиям сетей связи с искусственным интеллектом, реализуемых в соответствии с принципом AI-native. Ниже рассмотрены наиболее показательные из этих характеристик:

– *Гибкость.* FC совместима с широким спектром приложений IoT: от экологического мониторинга до управления производством. Распределённые ресурсы хранения и обработки обеспечивают взаимодействие с разнообразными конечными устройствами.

– *Связь в реальном времени.* FC допускает одновременное взаимодействие между туманными узлами. Это необходимо для приложений IoT, требующих мгновенного отклика, в частности для систем контроля безопасности и интеллектуального управления городскими транспортными потоками/трафиком.

– *Физическое распределение.* В отличие от облака, FC распределяет приложения и сервисы по всей территории охвата. Это снижает задержку и повышает производительность для масштабных приложений IoT.

– *Малая задержка и учёт местоположения.* FC выполняет обработку вблизи конечных устройств, обеспечивая минимальное время ожидания в диапазоне 1–10 мс. Осведомлённость о географическом положении позволяет группировать устройства как по территориальной близости, так и по особенностям генерируемых ими данных.

– *Совместимость.* Модули FC способны адаптироваться и взаимодействовать с разнообразными платформами от различных поставщиков услуг. Эта возможность обеспечивается уровнем абстракции Fog Abstraction Layer в архитектуре OpenFog.

– *Обеспечение аналитики и интеграции с облаком.* FC занимает промежуточное положение между конечными устройствами и облаком. Обработка и анализ информации выполняются вблизи источника, после чего в облако передаются только отфильтрованные данные.

– *Неоднородность.* Туманные узлы выпускаются различными производителями с собственными архитектурными решениями: ARM, x86, MIPS, RISC-V. Это обеспечивает гибкость и совместимость на разнообразных аппаратных и программных платформах.

– *Обеспечение гибкости.* FC поддерживает прямое подключение мобильных устройств и допускает использование гибких технологий, в том числе протоколов разделения идентификатора и местоположения. Такая возможность необходима для сетей связи транспортных средств.

В настоящее время существует несколько моделей вычислений с различной степенью распределённости: облачные вычисления, MEC, периферийные вычисления и FC. Анализ этих моделей показывает, что все они ориентированы на более эффективную обработку данных для новых приложений сетей IMT-2030. При этом между ними имеются существенные различия по месту развёртывания, величине задержки и уровню защищённости. Эти различия непосредственно влияют на выбор подходящей модели для каждого прикладного уровня сети следующего поколения. В таблице 5 приведено сравнение основных критериев для четырёх рассматриваемых моделей; оно служит основой для определения роли туманных вычислений в предлагаемой архитектуре.

Таблица 5 — Сравнение моделей распределённых вычислений

Характеристика	Cloud Computing	Fog Computing	Edge Computing
Место обработки	Удалённый централизованный центр обработки данных	Промежуточные узлы (шлюзы, маршрутизаторы, базовые станции) между конечными устройствами и облаком	Непосредственно на конечном устройстве
Задержка	Высокая: 100–500 мс	Низкая: менее 10 мс	Очень низкая: менее 1 мс

Характеристика	Cloud Computing	Fog Computing	Edge Computing
Пропускная способность	Большое потребление, вызывает перегрузку сети	Оптимизация: снижение передачи до 80%	Минимальная, полностью локальная обработка
Масштабируемость	Высокая, но затруднена при крупномасштабном IoT	Средняя, хорошо поддерживается иерархической архитектурой	Низкая, ограничена ресурсами устройства
Вычислительная мощность	Очень высокая	Средняя	Низкая
Хранение данных	Большой объём, централизованное	Средний объём, распределённое по узлам FC	Малый объём, локальное
Безопасность и конфиденциальность	Риски при передаче чувствительных данных через открытые сети	Повышенная: локальная обработка данных, шифрование на периферии	Максимальная: данные не покидают устройство
Поддержка мобильности	Ограниченная	Высокая: поддержка мобильности и адаптация по географическому положению	Привязана к устройству
Энергоэффективность	Низкая	Средняя	Ограничена аккумулятором устройства

Характеристика	Cloud Computing	Fog Computing	Edge Computing
Интеграция ИИ/МО	Централизованное обучение, требует передачи всех данных	Поддержка RL, FL, DRL	Прямой вывод модели на устройстве

Из приведённого сравнения следует, что FC выделяется возможностью многоуровневой распределённой обработки (N-tier), осведомлённостью о географическом положении и поддержкой гетерогенной инфраструктуры. Именно эти три свойства MEC и периферийные вычисления не обеспечивают одновременно. Практическая эффективность FC подтверждена рядом независимых экспериментальных исследований. В таблице 6 систематизированы типичные проблемы облачной модели, соответствующие решения на основе FC и количественные результаты, верифицированные в ходе реального развёртывания и моделирования.

Таблица 6 — Сравнение проблем модели Cloud и решений на основе FC

Проблема облака	Проявление	Решение в FC	Достигнутый результат
Высокая задержка	Передача данных в удалённый центр обработки не отвечает требованиям реального времени	Обработка данных на туманном узле вблизи источника генерации	Снижение задержки до 10мс для экстренных медицинских задач посредством гибридной модели планирования и балансировки нагрузки на уровне FC [42, 43]; SpeziLLM подтверждает снижение на 40–60% [44]
Перегрузка полосы пропускания	Передача всех необработанных данных в облако	Фильтрация и предварительная обработка	Снижение потребления полосы пропускания до 80% и улучшение времени

Проблема облака	Проявление	Решение в FC	Достигнутый результат
	вызывает перегрузку сети и высокие затраты	данных на уровне FC перед передачей в облако	обработки сервисов на 60% по сравнению с традиционным облаком за счёт механизма разгрузки SDN на уровне FC [45]
Ненадёжность при потере связи	Все сервисы прекращают работу при разрыве соединения с облаком	Узлы FC работают автономно, выполняя локальную обработку и принятие решений	Обеспечение непрерывности критически важных сервисов (мониторинг пациентов, управление трафиком) за счёт распределённого планирования задач и минимизации задержки [46, 47]
Ограниченная поддержка мобильности	Централизованная архитектура не адаптируется к непрерывно перемещающимся устройствам	Узлы FC динамически адаптируются к положению и перемещению устройств	В работе [48] показано снижение энергопотребления на 31,36% и задержки на 30% в среде FC 5G с высокой плотностью мобильных устройств, а в работе [49] эффективность подтверждена применением DRL в транспортных туманных сетях.
Масштабируемость для IoT	Централизованная архитектура	Распределённая иерархическая	Увеличение пропускной способности сети на 10–

Проблема облака	Проявление	Решение в FC	Достигнутый результат
	не справляется со взрывным ростом числа устройств IoT	архитектура с механизмами федерации и балансировки нагрузки между узлами FC	20% и продление срока службы устройств IoT за счёт методов энергоэффективного планирования и распределённой разгрузки [50]
Высокое энергопотребление	Централизованная обработка и непрерывная передача данных ведут к значительному расходу энергии	Архитектура Green IoT в сочетании с энергоэффективным планированием задач на уровне FC	Снижение энергопотребления на 30–50% за счёт обработки на периферии с применением энергоэффективного планирования и управление рабочим циклом [50, 51]; оптимизация планирования и балансировки нагрузки на уровне FC дополнительно снижает нагрузку на централизованную обработку [52]
Безопасность распределённой среды	Децентрализация расширяет поверхность атаки и создаёт дефицит доверия между узлами	Системы обнаружения вторжений на основе ИИ, интеграция блокчейна и	A-CIDS достигает точности 99,2%, частоты ложных срабатываний 2%, времени отклика 150 мс [53]; MLTPED-BFC

Проблема облака	Проявление	Решение в FC	Достигнутый результат
		механизмы доверия на основе МО	обеспечивает точность 98,91% [54]

На основе таблицы 6 можно сформулировать три объективных вывода о FC. Во-первых, в отношении экономии полосы пропускания: FC снижает потребление пропускной способности сети до 80% и улучшает время обработки сервисов до 60% по сравнению с традиционным облаком; этот результат подтверждён механизмом разгрузки в сочетании с SDN в многосервисной телекоммуникационной среде. Во-вторых, преимущества по задержке зависят от типа приложения: FC действительно превосходит альтернативы лишь для задач, критичных по времени, таких как экстренная медицина и плотные городские транспортные сети, достигая задержки менее 10 мс в медицинских приложениях и снижения на 50–70% по сравнению с облаком в приложениях умного города, а также одновременного снижения задержки и энергопотребления на 30% и 31,36% соответственно в среде fog 5G с мобильными устройствами. В-третьих, преимущества в области энергоэффективности и безопасности носят условный характер и проявляются только при совместном применении архитектуры Green IoT и соответствующих механизмов ИИ. В частности, методы энергоэффективного планирования и разгрузки на уровне FC снижают энергопотребление на 30–50%, а системы обнаружения вторжений с интеграцией ИИ и блокчейна достигают точности до 99,2%.

Практическая реализация концепции сети связи с ИИ порождает фундаментальное архитектурное противоречие. С одной стороны, модели ИИ нуждаются в данных с периферии сети для непрерывного обучения и адаптации к текущим условиям эксплуатации. С другой стороны, периферийные данные слишком велики по объёму и слишком чувствительны с точки зрения требований конфиденциальности для централизованной передачи в облако. Туманные

вычисления разрешают указанное противоречие, формируя промежуточный уровень обработки вблизи источника данных: достаточно мощный для локального выполнения конвейеров машинного обучения и одновременно достаточно подключённый для координации с облачным уровнем при необходимости. Тем самым туманные вычисления выступают в роли естественного уровня исполнения для задач распределённого машинного обучения, управления замкнутым контуром, локальной оптимизации сетевых ресурсов и двухуровневого федеративного обучения в тех сценариях, где требования к низкой задержке и защите персональных данных исключают полную зависимость от облачной инфраструктуры.

1.5 Федеративное обучение: принципы, архитектура и роль в сетях шестого поколения

В этих условиях в 2016 году McMahan и соавторами была предложена концепция FL, представляющая собой принципиально новый подход к совместному обучению моделей ML без передачи исходных данных между участниками процесса обучения. С того момента FL привлекло широкое внимание исследовательского сообщества и нашло применение во многих областях: мобильные сети, IoT, киберфизические системы, интеллектуальное здравоохранение и умные города [55]. По своей сути FL, предложенное Google, представляет собой подход, позволяющий участникам совместно строить общую модель обучения, сохраняя при этом базовые данные локально. Впоследствии исходная концепция FL была расширена и стала охватывать все методы децентрализованного совместного ML с защитой конфиденциальности [56]. Благодаря этому FL применимо не только к данным, горизонтально разделённым по обучающим образцам, но и к данным, вертикально разделённым по признакам. Кроме того, предложенный подход допускает масштабирование до уровня многоорганизационных федераций, объединяющих участников с разнородными локальными наборами данных. Таким образом, появляется возможность интеллектуально строить общую модель для множества субъектов, различных

источников данных и разных пространств признаков. Это позволяет всем участникам реализовать ценность совместного создания знаний на разных платформах и в разных регионах при соблюдении конфиденциальности данных. Традиционная архитектура федеративного обучения с централизованной агрегацией моделей в облаке показана на рисунке 8.

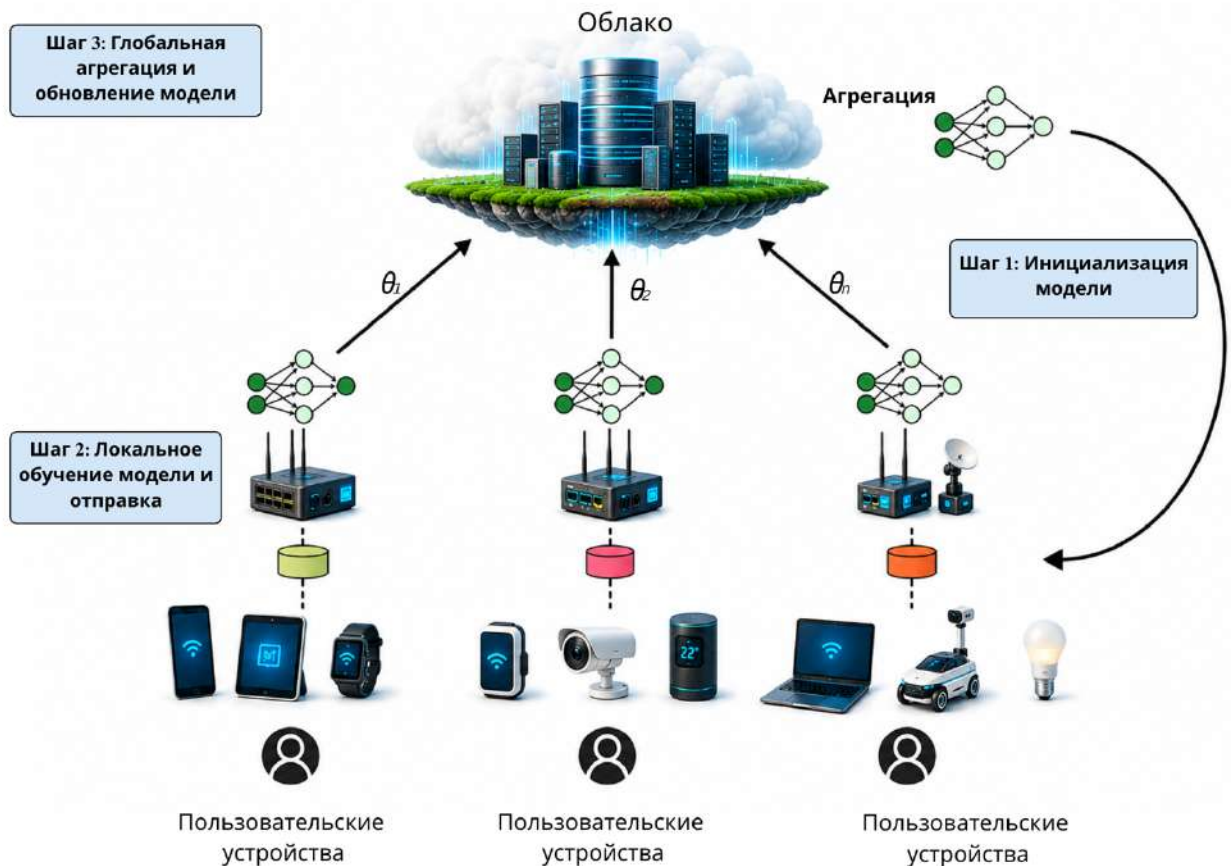


Рисунок 8 — Архитектура традиционного федеративного обучения

FL позволяет множеству пользовательских устройств совместно обучать общую модель машинного обучения под управлением центрального сервера, при этом локальные данные каждого клиента не покидают пределов устройства. Цель FL состоит в минимизации глобальной функции потерь путём нахождения оптимальных параметров общей модели.

Базовым и наиболее известным алгоритмом агрегации в FL является федеративное усреднение (FedAvg). Данный алгоритм работает в итеративном режиме и включает следующие шаги:

(1) Сервер инициализирует глобальную модель и задает гиперпараметры задачи FL, включая скорость обучения, число локальных итераций, размер минипакета и временной лимит каждого раунда коммуникации.

(2) Сервер выбирает группу клиентов и отправляет им параметры текущей глобальной модели.

(3) Каждый выбранный клиент выполняет локальное обучение модели на собственном наборе данных и отправляет обновленные параметры обратно на сервер.

(4) Сервер агрегирует полученные локальные модели для формирования новой глобальной модели, а затем распространяет ее обратно клиентам.

(5) Этот процесс повторяется до достижения целевой точности модели или максимального числа раундов.

Благодаря этому механизму исходные данные пользователей хранятся локально на устройствах, в то время как по сети на сервер передаются только градиенты или параметры модели.

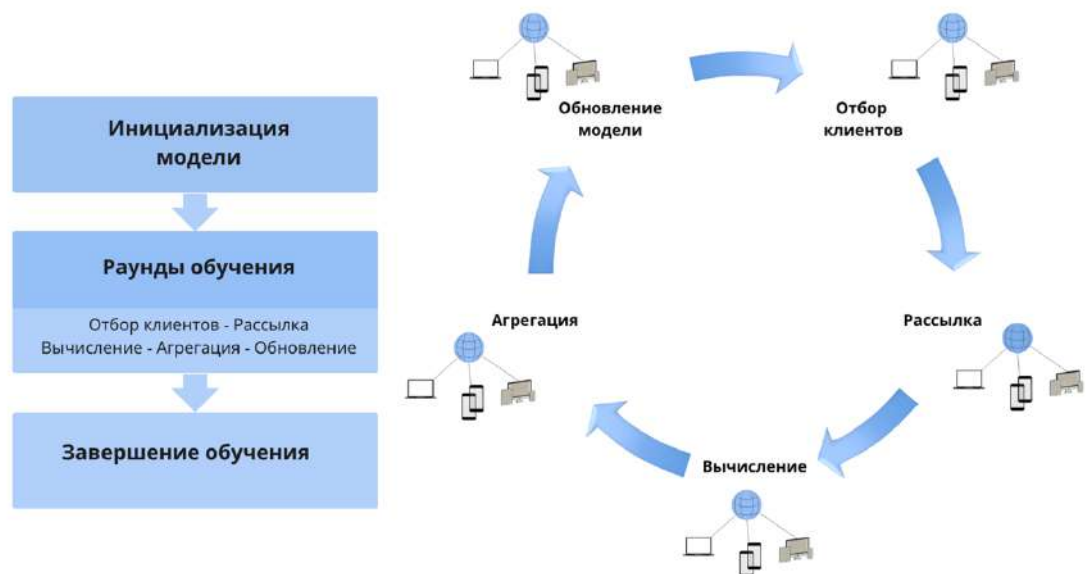


Рисунок 9 — Схема этапов федеративного обучения на основе алгоритма FedAvg

По сравнению с традиционной централизованной моделью обучения FL обладает рядом существенных преимуществ. Во-первых, данные для обучения

остаются на устройствах пользователей, что минимизирует риск утечки чувствительной информации и облегчает соблюдение нормативных требований по защите данных. Во-вторых, вместо передачи всего массива исходных данных FL требует пересылки лишь параметров модели, объём которых значительно меньше; это снижает нагрузку на пропускную способность сети. В-третьих, выполнение вывода непосредственно на устройстве после получения глобальной модели сокращает время принятия решений. Наконец, обучение на разнообразных данных от множества устройств потенциально позволяет создавать модели с лучшей способностью к обобщению, особенно в сочетании с эффективными методами отбора клиентов.

В зависимости от способа распределения данных FL подразделяется на три основных типа. Горизонтальное FL применяется в случаях, когда клиенты имеют общее пространство признаков, но владеют разными выборками данных. Вертикальное FL используется тогда, когда клиенты работают с одной и той же выборкой, но обладают разными наборами признаков. Федеративное трансферное обучение применяется, если у клиентов различаются и пространство признаков, и выборки данных. Кроме того, по архитектуре взаимодействия FL может быть организовано в виде централизованной модели с единым координирующим сервером, иерархической модели с несколькими промежуточными уровнями агрегации или одноранговой модели без центрального сервера.

FL уже активно применяется в ряде практических областей и демонстрирует ощутимые результаты. Ожидается, что FL продолжит интенсивно развиваться вместе с ростом больших языковых моделей, граничных вычислений, FC и сетей 6G [57].

1.6 Интеграция туманных вычислений и федеративного обучения в архитектуре сетей шестого поколения

При самостоятельном применении FL в традиционной двухуровневой схеме «пользовательские устройства — облачный сервер» возникает ряд существенных проблем. Во-первых, все клиентские устройства напрямую передают обновления

параметров модели на единственный центральный сервер. При масштабировании системы до миллионов узлов, характерном для сетей 6G, это создаёт значительный трафик на магистральных каналах и провоцирует перегрузку агрегирующего сервера. Во-вторых, высокая задержка при передаче обновлений в удалённое облако делает традиционный FL неприменимым для сервисов реального времени, требующих адаптации модели в пределах одного раунда обслуживания. В-третьих, единый сервер агрегации представляет собой единую точку уязвимости: его отказ или компрометация нарушают весь процесс обучения.

Применение туманных вычислений без интеграции алгоритмов федеративного обучения обеспечивает распределённую обработку данных, однако не предоставляет механизма конфиденциального совместного обучения моделей искусственного интеллекта на распределённых узлах. В подобной конфигурации локальные данные по-прежнему концентрируются на туманных узлах, что порождает потенциальные уязвимости с точки зрения соответствия требованиям регуляторных актов в области защиты персональных данных.

Интеграция туманных вычислений и федеративного обучения устраняет ограничения каждой технологии в отдельности. Туманные узлы принимают на себя роль промежуточных агрегаторов: вместо того чтобы каждое пользовательское устройство напрямую отправляло обновления модели в облако, оно взаимодействует с ближайшим туманным узлом, который выполняет локальную агрегацию и передаёт в облако уже обобщённый результат. Это позволяет одновременно сократить затраты на передачу данных в направлении облачного уровня, снизить задержку агрегации и устранить единую точку уязвимости за счёт распределения функций агрегации по множеству туманных узлов [58]. Данный подход реализуется через трёхуровневую иерархическую архитектуру, соответствующую физической структуре сетей шестого поколения и принципам Рекомендации МСЭ-Р М.2160. Принципиальная схема описанной иерархии и взаимодействие её уровней в процессе федеративного обучения представлены на рисунке 10.

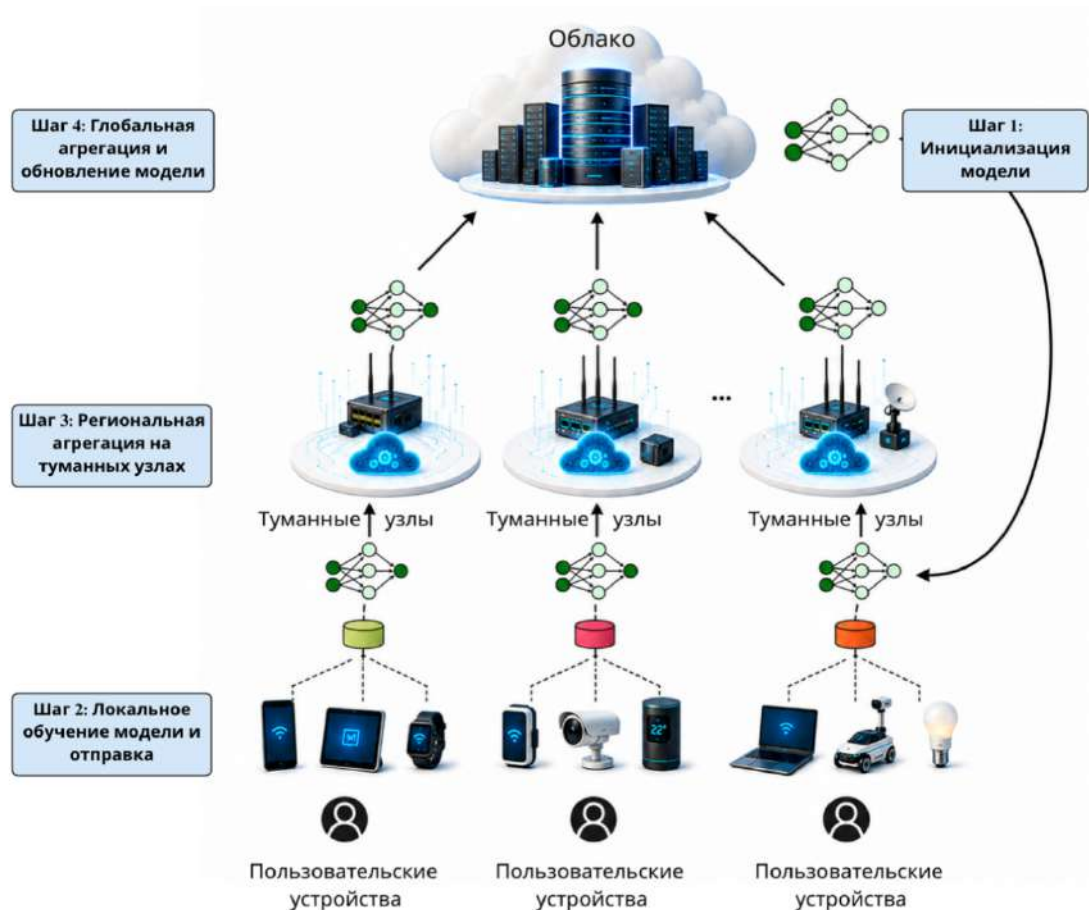


Рисунок 10 — Трёхуровневая иерархическая архитектура интеграции FL и FC в сетях 6G

– *Уровень пользовательских устройств.* На нижнем уровне располагаются конечные узлы: IoT-датчики, пользовательское оборудование, транспортные средства с поддержкой V2X, носимые устройства. Каждый узел генерирует локальные данные и выполняет локальное обучение модели ИИ на собственном наборе данных, не передавая исходные данные за пределы устройства. По завершении заданного числа локальных итераций узел отправляет только обновления параметров модели на обслуживающий его туманный узел.

– *Уровень туманных узлов.* Средний уровень образуют туманные узлы: базовые станции gNB, интеллектуальные маршрутизаторы, шлюзы IoT. Туманный узел принимает обновления от группы закреплённых за ним конечных устройств и выполняет промежуточную агрегацию. Это значительно сокращает объём данных, передаваемых на следующий уровень, поскольку на облако направляется единый агрегированный вектор, а не тысячи индивидуальных обновлений.

– *Облачный уровень.* На верхнем уровне облачный сервер принимает агрегированные обновления от всех туманных узлов, выполняет глобальную агрегацию и распространяет обновлённую глобальную модель обратно по иерархии. Облако также отвечает за инициализацию модели, хранение версий и координацию гиперпараметров обучения.

Интеграция FL и FC обеспечивает комплекс преимуществ, непосредственно релевантных для задач построения сетей 6G. Сравнительный анализ трёх архитектурных подходов представлен в таблице 7.

Таблица 7 — Сравнительная характеристика архитектур централизованного обучения, FL-облако и FL-FC

Характеристика	Централизованное обучение	FL (облачный сервер)	FL-FC (иерархическая)
Задержка агрегации	Высокая	Высокая	Низкая (локальная агрегация)
Трафик в сторону облака	Очень высокий	Высокий	Значительно снижен
Конфиденциальность данных	Отсутствует	Частичная	Высокая
Устойчивость к отказам	Низкая	Низкая	Высокая (распределённая)
Масштабируемость	Ограниченная	Ограниченная	Высокая

– *Снижение коммуникационных затрат.* Иерархическая агрегация на уровне туманных узлов принципиально изменяет структуру сетевого трафика. Вместо передачи обновлений модели от каждого пользовательского устройства непосредственно в облачный сервер, обновления сначала собираются и агрегируются на туманных узлах, после чего на облачный уровень передаётся только агрегированный вектор от каждого туманного узла. Такой подход обеспечивает существенное сокращение числа сообщений, достигающих

облачного уровня, поскольку один туманный узел обслуживает группу из множества пользовательских устройств.

– *Снижение задержки.* Выполнение промежуточной агрегации на уровне туманных узлов позволяет получить обновлённую версию модели в рамках локального взаимодействия между пользовательскими устройствами и обслуживающим их туманным узлом, без обращения к облачному серверу. Это значительно сокращает задержку обновления модели и обеспечивает её актуальность в режиме реального времени, что критически важно для сервисов с жёсткими требованиями к задержке, включающих приложения сверхнадёжной связи с малой задержкой, интеллектуальные транспортные системы и промышленный Интернет вещей.

– *Конфиденциальность и защита данных.* В традиционном FL риск утечки информации возникает при передаче градиентов напрямую на центральный сервер: исследования показали возможность частичного восстановления исходных данных методом инверсии градиентов. В архитектуре FL-FC промежуточная туманная агрегация дополнительно защищает индивидуальные обновления: облачный сервер получает уже усреднённый вектор от десятков устройств, что существенно затрудняет атаки на конфиденциальность.

– *Устойчивость и масштабируемость.* Распределение функции агрегации между множеством туманных узлов исключает единую точку отказа. При выходе из строя одного туманного узла остальные продолжают работу независимо; глобальная агрегация в облаке выполняется на основе доступных обновлений. Это свойство имеет особое значение для сетей 6G, ориентированных на поддержку критически важных приложений.

Таким образом, интеграция туманных вычислений и федеративного обучения представляет собой не просто техническое объединение двух подходов, а формирование качественно новой архитектурной парадигмы для сетей 6G. Трёхуровневая иерархия «облако — туманные узлы — пользовательские устройства» обеспечивает реализацию принципа AI-native на инфраструктурном уровне: интеллектуальные функции распределяются по всей сети, вычисления

приближаются к источнику данных, а конфиденциальность обеспечивается на алгоритмическом уровне. Именно эта архитектура принята в качестве основы для разработки моделей и методов, исследуемых в последующих главах настоящей диссертации.

1.7 Задачи диссертации

Анализ тенденций развития сетей шестого поколения, рекомендации МСЭ-Р М.2160 о принципах проектирования AI-native и резолюции 101 ВАСЭ-24 о сетях связи с искусственным интеллектом, а также системы стандартов МСЭ-Т серии Y.317x по применению машинного обучения в телекоммуникационных сетях выявил системный пробел в исследованиях. Этот пробел находится между теоретическими логическими моделями интеграции ИИ в сетевую инфраструктуру и их практической реализацией на гетерогенной физической инфраструктуре сетей ИМТ-2030. В первой главе диссертации построена трёхуровневая эталонная архитектура, объединяющая туманные вычисления и федеративное обучение в качестве практического механизма реализации концепции сети связи с искусственным интеллектом. Однако для практического применения этой архитектуры необходимо преодолеть три ограничения, возникающие при её развёртывании в реальных условиях. Первое связано с пределами пропускной способности канала связи, второе обусловлено недостаточными возможностями алгоритмов для распределённого принятия решений без зависимости от центрального сервера, третье определяется аппаратной неоднородностью граничных узлов и туманных узлов в инфраструктуре сетей шестого поколения.

На основе проведённого анализа общая цель диссертации состоит в разработке моделей и методов оптимизации сетей связи шестого поколения на основе конвергенции туманных вычислений и федеративного обучения, обеспечивающих практическую реализацию принципа “AI-native” и концепции сети связи с искусственным интеллектом в распределённой гетерогенной граничной среде. Решаемые задачи направлены на повышение производительности системы, снижение коммуникационных издержек и обеспечение

кроссплатформенного развёртывания моделей ИИ. Практическая верификация проводится на примере пространственно-временного прогнозирования городских транспортных потоков в сервисах городских сетевых вселенных — как репрезентативного сценария реализации сети связи с ИИ в условиях городской инфраструктуры сетей связи шестого поколения.

Для реализации этой стратегической цели в едином целом диссертация ставит и последовательно решает четыре основные исследовательские задачи:

– *Задача 1:* Анализ, исследование и разработка трёхуровневой интегрированной архитектуры туманных вычислений и федеративного обучения для сетей связи с искусственным интеллектом. Формируется механизм оптимального отображения функциональных компонентов на физическую инфраструктуру «облако – туманные узлы – пользовательские устройства» в соответствии с принципом AI-native, закреплённым в рекомендации МСЭ-Р М.2160. Функции обработки распределяются согласно пространственно-временным характеристикам задач, что позволяет снизить задержку на структурном уровне системы.

– *Задача 2:* Снижение коммуникационных издержек на уровне туманных вычислений путём разработки метода динамического выбора пользовательских устройств для участия в раундах федеративного обучения. Механизм основан на адаптивном пороге скорости изменения нормы градиента функции потерь, пересчитываемом непосредственно на уровне туманных узлов в каждом раунде обучения. Это позволяет исключать зашумлённые обновления и оптимизировать использование полосы пропускания канала связи в условиях массового подключения устройств, характерного для сетей 6G.

– *Задача 3:* Разработка децентрализованной структуры федеративного обучения с применением адаптивной графовой свёрточной рекуррентной нейронной сети (AGCRN). Модель глубоко моделирует сложные пространственно-временные закономерности городских транспортных потоков для поддержки сервисов CitiVerse. Доказывается способность туманного кластера к независимой работе даже при разрыве соединения с центральным облаком.

– *Задача 4*: Устранение барьера аппаратной неоднородности граничных узлов путём разработки метода кроссплатформенного развёртывания модулей федеративного обучения. Метод использует технологию WebAssembly и среду выполнения Wasmtime, что обеспечивает безопасность системы и производительность, близкую к нативному коду, на гетерогенных микропроцессорных архитектурах ARM64, ARM, x86 и RISC-V, типичных для туманных узлов в инфраструктуре сетей шестого поколения.

Решение перечисленных исследовательских задач позволяет реализовать и подтвердить практическую применимость архитектурной структуры, предложенной в первой главе. Полученные результаты вносят конкретный технический вклад в развитие сетей мобильной связи нового поколения в соответствии с направлениями международной стандартизации, закреплёнными в рекомендации МСЭ-Р М.2160 и резолюции 101 ВАСЭ-24. Методология, математические модели и результаты экспериментальной оценки по каждой задаче последовательно излагаются в главах 2, 3 и 4.

1.8 Выводы по Главе 1

1. В главе представлен обзорный анализ сетей 6G и тенденций развития базовых технологий. Рассмотрен переход от сетей пятого поколения к сетям шестого поколения на основе рекомендации МСЭ-Р М.2160 и резолюции 101 ВАСЭ-24. Подчёркивается, что 6G представляет собой смену парадигмы, при которой сеть связи становится интегрированной распределённой интеллектуальной платформой в соответствии с концепцией AI-native и сетью связи с искусственным интеллектом.

2. Проведён анализ туманных вычислений и федеративного обучения как ключевых технологий для практической реализации концепции сети связи с искусственным интеллектом. Раскрыты преимущества FC, включая снижение задержек, разгрузку полосы пропускания и локальную обработку данных в реальном времени. Установлено, что федеративное обучение является наиболее подходящим механизмом для обучения моделей ИИ в распределённой среде со

строгими требованиями к конфиденциальности данных в соответствии с резолюцией 101 ВАСЭ-24.

3. Систематически проанализированы рекомендации МСЭ-Т серии Y.317x по применению машинного обучения в телекоммуникационных сетях. Выявлены три фундаментальных пробела в исследованиях, связанные с ограничениями пропускной способности канала связи, недостаточными возможностями алгоритмов для распределённого принятия решений и аппаратной неоднородностью граничных узлов.

4. Для решения выявленных проблем предложена трёхуровневая интегрированная архитектура туманных вычислений и федеративного обучения, реализующая принцип “AI-native” и концепцию сети связи с искусственным интеллектом путём отображения функциональных компонентов на физическую инфраструктуру «облако – туманные узлы – пользовательские устройства».

ГЛАВА 2. МОДЕЛЬ ФЕДЕРАТИВНОГО МАШИННОГО ОБУЧЕНИЯ С ТУМАННЫМИ ВЫЧИСЛЕНИЯМИ И МЕТОД ВЫБОРКИ ПОЛЬЗОВАТЕЛЬСКИХ УСТРОЙСТВ НА ОСНОВЕ ДИНАМИЧЕСКОГО ПОРОГА

2.1 Введение

В предложенной в первой главе архитектуре уровень туманных вычислений выполняет функцию регионального агрегатора федеративного обучения: FC собирает обновления градиентов от граничных устройств, вычисляет взвешенное среднее и передаёт агрегированный результат на облачный уровень для формирования глобальной модели. Такая схема устраняет узкое место на участке «туманные узлы – облако»: исходные данные остаются на граничных устройствах, а в вышестоящий узел передаются только параметры модели. Тем не менее в условиях FL существует одна важная практическая проблема: обновления от устройств, локальные данные которых статистически близки к уже сложившемуся состоянию глобальной модели, несут ограниченную информационную ценность для процесса сходимости. Передача таких обновлений расходует ресурсы канала, не внося заметного вклада в улучшение глобальной модели. В связи с этим в настоящей главе предлагается механизм адаптивного выбора пользовательских устройств для участия в раундах обучения, направленный на повышение эффективности и практической применимости предложенной архитектуры.

Задача выбора пользовательских устройств для участия в раундах FL привлекла значительное внимание исследовательского сообщества. Были предложены методы, основанные на нормированных оценках обновлений [59], адаптивной максимизации вероятности участия устройств [60], диверсификации по разнообразию распределения данных [61], а также на использовании обновлений предшествующих раундов в качестве заменителей для неактивных устройств [62]. Вместе с тем перечисленные подходы обладают существенными ограничениями: они либо не учитывают динамику сходимости при неоднородном распределении данных, либо требуют детальной информации о статистических свойствах

локальных наборов данных, либо не предусматривают интеграцию с промежуточным уровнем FC для разгрузки центрального сервера. При этом вопрос об адаптивном пороговом критерии выбора, учитывающем фазу процесса сходимости и реализуемом непосредственно на узлах FC без обращения к центральному серверу, остаётся недостаточно изученным.

В настоящей главе предлагается метод выборки пользовательских устройств на основе динамического порога скорости изменения нормы градиента функции потерь в интегрированной модели федеративного обучения с туманными вычислениями. Пороговое значение адаптивно пересчитывается на уровне туманных вычислений в каждом раунде, что позволяет учитывать неоднородность локальных данных и текущую фазу сходимости модели, а также снизить объём сетевого взаимодействия между устройствами и центральным сервером.

Структура настоящей главы построена следующим образом. В разделе 2.2 представлен аналитический обзор существующих методов выборки устройств в системах FL и обоснована их недостаточность применительно к интегрированной среде FL-FC. В разделе 2.3 изложены формализованная постановка задачи и предлагаемая модель интеграции FL и FC. В разделе 2.4 представлены теоретические основы метода динамического порога и алгоритмы его реализации на каждом уровне архитектуры. В разделе 2.5 приведены результаты экспериментальной оценки предложенного метода на наборах данных MNIST и CIFAR-10. Выводы по главе суммируют полученные результаты и определяют исследовательскую задачу, решаемую в главе 3.

2.2 Обзор методов выборки устройств в федеративном машинном обучении

Значительные достижения в области FL не только изменили понимание методологий машинного обучения, но и привели к появлению новых направлений исследований, направленных на повышение эффективности распределенных вычислительных систем и сетей. Методы выбора устройств привлекли значительное внимание учёных, поскольку их применение позволяет повысить качество модели искусственного интеллекта за счёт существенного снижения

сетевой нагрузки в распределенных системах ИИ. Обеспечение эффективности таких систем требует выбора устройств, которые в достаточной степени представляют весь набор данных для модели и обладают необходимыми вычислительными ресурсами для эффективного участия в процессе обучения модели ИИ.

В последнее время было предложено множество подходов к оптимизации процесса выбора пользовательских устройств в системах федеративного обучения. Эти стратегии охватывают широкий спектр методов, включая аппроксимацию матриц низкого ранга и снижение сложности, а также квантование, случайную выборку и стохастические методы, такие как случайное перемешивание и стохастическая аппроксимация. Ярким примером является метод, предложенный в [59], в котором используется система выбора устройств на основе частоты их обновлений. В этом методе каждое устройство вычисляет функцию обновления в течение каждого цикла связи, но при этом на сервер передаются только те обновления, которые считаются значимыми на основе введенных нормированных оценок.

В [60] был предложен адаптивный метод, который использует неоднородность системы и статистику для ускорения сходимости путем максимизации вероятности выбора устройств. Кроме того, в [61] предлагалась диверсифицированная система выбора устройств с акцентом на уменьшении вариативности, вызванной частичным участием устройств. В [62] был исследован метод динамического порога, в котором обновления от неконтактных устройств заменяются обновлениями из предыдущих раундов, что позволяет уменьшить количество необходимых сетевых взаимодействий с сервером основной модели ИИ при сохранении ее производительности. В [63] для моделирования траектории весов модели использовался метод Орнштейна-Уленбека, который предполагает выбор устройств с важными обновлениями и прогноз обновления для невыбранных устройств.

Новый вектор развития FL заключается в интеграции с FC, что может стать потенциальным решением для распределения нагрузки в сети и улучшения

процессов обучения. Объединенная структура FL-FC позволяет распределить задачи обучения с центрального сервера модели ИИ на узлы FC, тем самым снижая нагрузку на центральный сервер и нагрузку на сеть (соединение между устройством и сервером модели ИИ). Благодаря близости к пользователям узлы FC не только повышают эффективность обработки запросов, но и существенно снижают нагрузку на сеть и сокращают задержки распространения, что критично для услуг URLLC. Сравнение предложенного метода с другими методами представлено в табл. 8.

Таблица 8 — Сравнение методов выборки устройств в федеративном машинном обучении

Метод	Преимущества	Ограничения	Выводы
Метод оптимальной выборки на основе нормированных оценок обновлений [59]	Снижает сетевую нагрузку за счёт передачи только значимых обновлений на основе нормированных оценок	не адаптируется к разнородным данным, требует значительных вычислительных ресурсов на устройствах	По сравнению с данным методом предлагаемый подход обеспечивает адаптацию к разнородным данным благодаря динамическому порогу скорости изменения градиента функции потерь
Адаптивный метод выборки с максимизацией вероятности участия устройств [60]	Адаптивное распределение полосы пропускания; сокращает время достижения сходимости	Высокая вычислительная сложность реализации; метод требует детальной информации о состоянии сети, предоставляя меньше	Интеграция с туманными вычислениями в предлагаемом методе снижает вычислительную сложность на стороне устройства, что повышает практическую применимость решения

		сведений о ней по сравнению с централизованными подходами	
Метод диверсифицированной выборки на основе разнообразия распределения данных [61]	Выбор устройств на основе разнообразия распределения данных устройств; уменьшает дисперсию градиента глобальной модели вследствие частичного участия устройств	Требуется детальная информация о разнообразии распределения данных и разбросе градиентов между устройствами; высокая вычислительная сложность	Предлагаемый метод не требует детальной информации о распределении данных и упрощает вычисления благодаря динамическому порогу скорости изменения градиента функции потерь
Предлагаемый метод	динамические пороги уменьшают нагрузку на сеть, ускоряется сходимость модели и интегрируются FC для разгрузки серверов центральной	требуются дополнительные тестирования на реальных системах услуг с моделями ИИ, в зависимости от параметров динамического порога. На практике FC	Обеспечивает оптимальный баланс между сетевой нагрузкой и точностью модели; интеграция с туманными вычислениями соответствует принципам эффективного использования ресурсов в системах, распределённого ИИ

	модели ИИ	еще не получили широкого распространени я ввиду наукоемкости технологии	
--	-----------	---	--

Стоит также отметить, что многие исследования сосредоточены на использовании техник выбора устройств в архитектуре FL-FC. Например, в [63] для решения проблемы дисбаланса в обучении была предложена техника выбора пользовательских устройств, предполагающая возможность участия только устройств с достаточными ресурсами. В [65] было показано, что использование облачных узлов в качестве посредников для агрегации моделей может ускорить сходимость моделей искусственного интеллекта и значительно снизить количество сетевых взаимодействий между устройствами и центральными серверами. Хотя эти решения достигли значительного развития, но они по-прежнему сталкиваются с трудностями при применении в сложных сценариях, таких как интеграция с динамическими туманными вычислениями. Эти проблемы включают в себя: гетерогенность ресурсов, устойчивость кластеров туманных вычислений, непредсказуемую задержку связи и неравномерное распределение данных между устройствами. Учитывая постоянный спрос со стороны пользователей на сетевые и вычислительные ресурсы, включая приложения с моделями искусственного интеллекта, необходимы эффективные решения для каждой из этих проблем.

2.3 Модель федеративного машинного обучения с туманными вычислениями и метод динамического порога выбора пользовательских устройств

Федеративная модель обучения использует децентрализованные системы данных, состоящие из нескольких серверов, в отличие от традиционных методов локального обучения. Это позволяет устройствам, участвующим в модели,

использовать несколько источников данных, что приводит к более точным и обобщённым прогнозам по сравнению с обучением исключительно на локальных наборах данных. Модель федеративного обучения показана на рисунке 11.

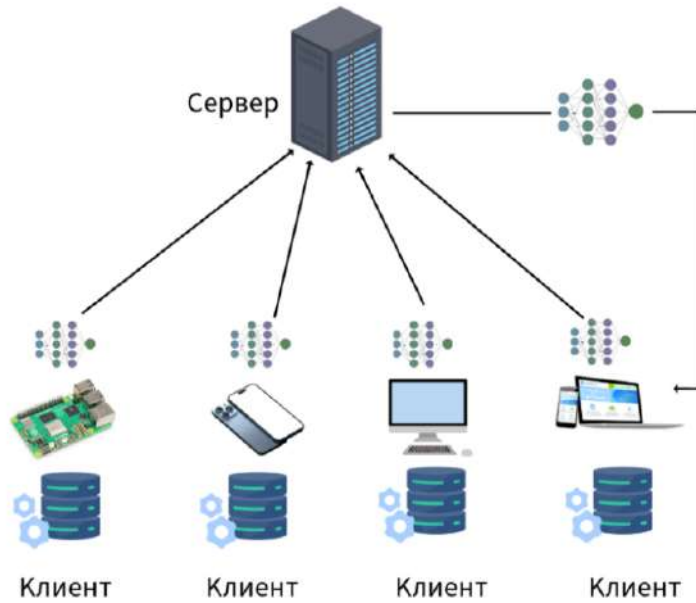


Рисунок 11 — Модель федеративного обучения

Рассмотрим настройку федеративного обучения между K устройствами, где устройство с номером k располагает локальным набором данных D_k , состоящий из N_k образцов данных. Устройства соединены через центральный агрегирующий сервер и в ходе совместного обучения определяют параметры модели w , минимизирующие эмпирический риск - усреднённое значение функции потерь по всей совокупности обучающих данных:

$$F(\theta) = \frac{1}{\sum_{k=1}^K D_k} \sum_{k=1}^K \sum_{\xi \in B_k} f(\theta, \xi) = \sum_{k=1}^K p_k F_k(\theta) \quad (1)$$

где $f(\theta, \xi)$ – составная функция потерь для образца ξ и вектора параметров θ , $p_k = D_k / \sum_{k=1}^K D_k$ доля данных, приходящаяся на k -ое устройство; $F_k(\theta) = \frac{1}{|B_k|} \sum_{\xi \in B_k} f(\theta, \xi)$ локальная целевая функция устройства k .

В условиях FL оптимальные параметры θ_k^* , минимизирующие локальную функцию потерь функцию $F_k(\theta)$, как правило, не совпадают с параметрами θ^* , минимизирующими глобальную функцию $F(\theta)$, ввиду статистической неоднородности локальных наборов данных. Оптимум глобальной задачи определяется как $F^* = \min_{\theta} F(\theta)$, а локальный оптимум каждого устройства – как $F_k^* = \min_{\theta} F_k(\theta) = F_k(\theta_k^*)$.

Прежде чем представить предлагаемый метод, рассмотрим один из наиболее распространённых алгоритмов FL - алгоритм федеративного усреднения (FedAvg), чтобы дать общее представление о механизме распределённого обучения. Данный алгоритм делит процесс обучения на раунды взаимодействия. В каждом раунде, с целью минимизации сетевой нагрузки, глобальный сервер случайным образом выбирает подмножество устройств для участия в обучении. Выбранные устройства выполняют локальные обновления методом стохастического градиентного спуска (SGD - stochastic gradient descent), после чего передают локальные параметры модели на сервер для агрегирования в единую глобальную модель.

В формальном контексте рассматриваются локальные итерации SGD, индексируемые по $t \geq 0$, где t - номер текущей итерации. Множество устройств, активных на итерации t , обозначается $S(t)$. Поскольку активные устройства выполняют шаги локального обновления синхронно в пределах каждого раунда, состав активного множества остаётся постоянным на протяжении всех T итераций одного раунда (где T — количество локальных итераций в раунде). Правило обновления FedAvg формулируется следующим образом:

$$\theta_k^{(t+1)} = \begin{cases} \theta_k^{(t)} - \eta_t g_k(\theta_k^{(t)}, \xi_k^{(t)}) & \text{for } (t+1) \bmod T \neq 0 \\ \frac{1}{m} \sum_{j \in S(t)} (\theta_j^{(t)} - \eta_t g_j(\theta_j^{(t)}, \xi_j^{(t)})) & \text{for } (t+1) \bmod T = 0 \end{cases} \quad (2)$$

где $\theta_k^{(t)}$ – локальные параметры модели для устройства k на итерации t ; η_t – скорость обучения; $g_k(\theta_k^{(t)}, \xi_k^{(t)}) = \frac{1}{b} \sum_{\xi \in \xi_k} f(\theta_k^{(t)}, \xi)$ – стохастический градиент, вычисленный на мини-пакете $\xi_k^{(t)}$ размером b , случайная выборка из локального набора данных B_k устройства k .

После каждого раунда обучения, включающего T локальных итераций SGD, сервер обновляет глобальную модель на основе агрегированных параметров, поступивших от участвующих устройств, и рассылает обновлённую модель на узлы туманных вычислений для следующего раунда.

В настоящей главе предлагается новый метод выборки пользовательских устройств для участия в федеративном обучении, разработанный на основе алгоритма FedAvg. Отличительной особенностью предлагаемого метода является применение динамического порогового критерия выбора устройств в интегрированной среде федеративного машинного обучения и туманных вычислений. Критерием выбора служит скорость изменения нормы градиента функции потерь каждого устройства: устройство включается в раунд обучения лишь в том случае, если данная скорость достигает адаптивно вычисляемого порогового значения. Такой подход позволяет минимизировать объём сетевого взаимодействия между устройствами и центральным сервером без ущерба для качества обучаемой модели.

2.4 Теоретические основы метода выборки устройств на основе динамического порога скорости изменения градиента функции потерь

В данном разделе изложены теоретические основы предлагаемого метода. Сначала рассматривается механизм градиентного спуска как базовый инструмент оптимизации в задачах машинного обучения и федеративного обучения. Затем вводится понятие скорости изменения нормы градиента как индикатора динамики сходимости модели. На этой основе формализуется механизм динамического порога и анализируется его влияние на процесс сходимости.

1. Градиентный спуск в машинном обучении и федеративном машинном обучении. Градиентный спуск — это итерационный процесс оптимизации, используемый в машинном обучении, особенно в приложениях ИИ с глубоким обучением. Он часто используется в задачах выпуклой оптимизации для определения набора собственных переменных для оптимизации модели машинного обучения. Цель градиентного спуска - вычислить значение градиента

функции потерь в каждой точке и на каждом шаге двигаться в направлении, противоположном градиенту функции потерь, пока значение градиента не приблизится к нулю, соответствующему точке минимума функции потерь. Процесс градиентного спуска включает в себя множество итераций, в том числе SGD и мини-пакетный градиентный спуск. Следующие шаги являются основополагающими для данных методов оптимизации:

- а) инициализация параметров модели;
- б) оценка модели на основе собственных переменных и функции потерь;
- с) обновление переменных в направлении, оптимизирующем функцию потерь;
- д) повтор шагов оценки и обновления (б и с) до тех пор, пока не будет выполнено условие остановки.

Формула обновления для градиентного спуска может быть записана как:

$$\theta_t \leftarrow \theta_{t-1} - \eta \nabla L(\theta_{t-1}) \quad (3)$$

где η – скорость обучения; $\nabla L(\theta_{t-1})$ – градиент функции потерь L относительно параметров модели θ на итерации t .

В FL каждое устройство с номером k выполняет шаги локальной оптимизации на основе функции потерь, которая зависит от локальных данных устройства. Глобальная модель обновляется путем усреднения обновлений, поступающих от устройств. Для каждого устройства k на итерации t обновление градиента задается как:

$$\Delta \theta = -\eta \nabla L^k(\theta_{t-1}) \quad (4)$$

2. Скорость изменения градиента как индикатор динамики сходимости.
Предположим, что после каждого локального раунда обучения модели каждое устройство k вычисляет норму своего градиента $\|\Delta \theta_t^{(k)}\|$. Чтобы оценить степень изменения градиента между раундами, скорость изменения градиента

рассчитывается как:

$$r_t^{(k)} = \frac{\|\Delta\theta_t^{(k)}\|}{\|\Delta\theta_{t-1}^{(k)}\|} \quad (5)$$

где $\|\Delta\theta_t^{(k)}\|$ – норма градиента функции потерь устройства k в раунде t ; $\|\Delta\theta_{t-1}^{(k)}\|$ норма градиента функции потерь устройства k в раунде $(t-1)$.

Большая скорость $r_t^{(k)}$ указывает на значительное изменение градиента функции потерь устройств k , предполагая, что это обновление может дать ценную информацию для улучшения глобальной модели. И наоборот, при небольшой скорости $r_t^{(k)}$ устройству k может не потребоваться отправлять обновление на сервер, так как изменение не является существенным.

3. Динамический порог на основе средней скорости изменения градиента. Политика принятия решения основана на сравнении скорости изменения градиента $r_t^{(k)}$ с порогом скорости изменения градиента τ_t для необходимых сетевых взаимодействий с центральной моделью ИИ на сервере при сохранении точности. Поскольку норма градиента функции потерь уменьшается в процессе обучения, статичный порог может негативно сказаться на процессе оптимизации по мере приближения к минимуму функции потерь.

Предлагаемый метод адаптивного порога обеспечивает фильтрацию обновлений модели ИИ, подстраиваясь в соответствии с изменениями градиента функции потерь каждого устройства в каждом раунде обучения. Данный механизм основан на среднем и стандартном отклонении скорости изменения градиента. В первом раунде устройствам присваивается начальный порог скорости изменения градиента $\tau_t=0$, что означает, что все устройства передают обновления. В последующих итерациях устройства передают свои обновления, а порог сравнения вычисляется и оценивается по скорости изменения градиента функции потерь каждого устройства, чтобы определить, нужно ли передавать обновления.

4. Среднее и стандартное отклонение скорости изменения уклона нормы градиента функции потерь. Сервер или узел туманных вычислений вычисляет

среднее значение μ_t и стандартное отклонение σ_t скорости изменения функции потерь от устройств, отправивших обновления. Если K – количество устройств в раунде t , то среднее значение и стандартное отклонение скорости изменения градиента вычисляются следующим образом:

$$\mu_t = \frac{1}{K} \sum_{k=1}^K r_t^{(k)} \quad (6)$$

$$\sigma_t = \sqrt{\frac{1}{K} \sum_{k=1}^K (r_t^{(k)} - \mu_t)^2} \quad (7)$$

5. Настройка динамического порога. Динамический порог τ_t в раунде t определяется на основе среднего и стандартного отклонения скорости изменения градиента:

$$\tau_t = \mu_t + \alpha \sigma_t \quad (8)$$

где α – коэффициент корректировки, что позволяет настраивать чувствительность порога к колебаниям скорости изменения градиента.

Когда коэффициент α велик, порог τ_t становится большим, уменьшая количество передаваемых обновлений. И наоборот, если коэффициент α мал, то порог τ_t становится ниже, что позволяет большему числу устройств передавать обновления и тем самым повышает точность глобальной модели.

6. Динамический порог и эффект сходимости. Динамический порог позволяет системе федеративного машинного обучения адаптивно изменять себя в соответствии с различными фазами процесса сходимости:

– В начале обучения градиенты демонстрируют значительную изменчивость, часто характеризующуюся большим стандартным отклонением σ_t . Порог τ_t подстраивается под эту значительную изменчивость, тем самым снижается передача большего количества обновлений модели ИИ.

– По мере приближения модели к сходимости флуктуации градиента уменьшаются, что приводит к снижению σ_t . Порог τ_t также уменьшается, что снижает частоту излишних обновлений связи.

2.5 Предлагаемая модель интеграции FL и FC

Модель интеграции FL и FC состоит из трех основных уровней, как показано на рисунке 12.

Первый уровень – уровень граничных устройств. Пользовательские устройства выполняют локальное обновление модели ИИ и рассчитывают скорость изменения градиента.

Второй уровень – уровень туманных вычислений. Узлы FC, расположенные вблизи пользовательских устройств, обрабатывают, агрегируют и фильтруют обновления перед отправкой их на центральный сервер.

Третий уровень – облачный. Глобальный агрегатор FL получает отфильтрованные обновления от уровня туманных вычислений, выполняет глобальное обновление модели и отправляет её обратно на узлы туманных вычислений.

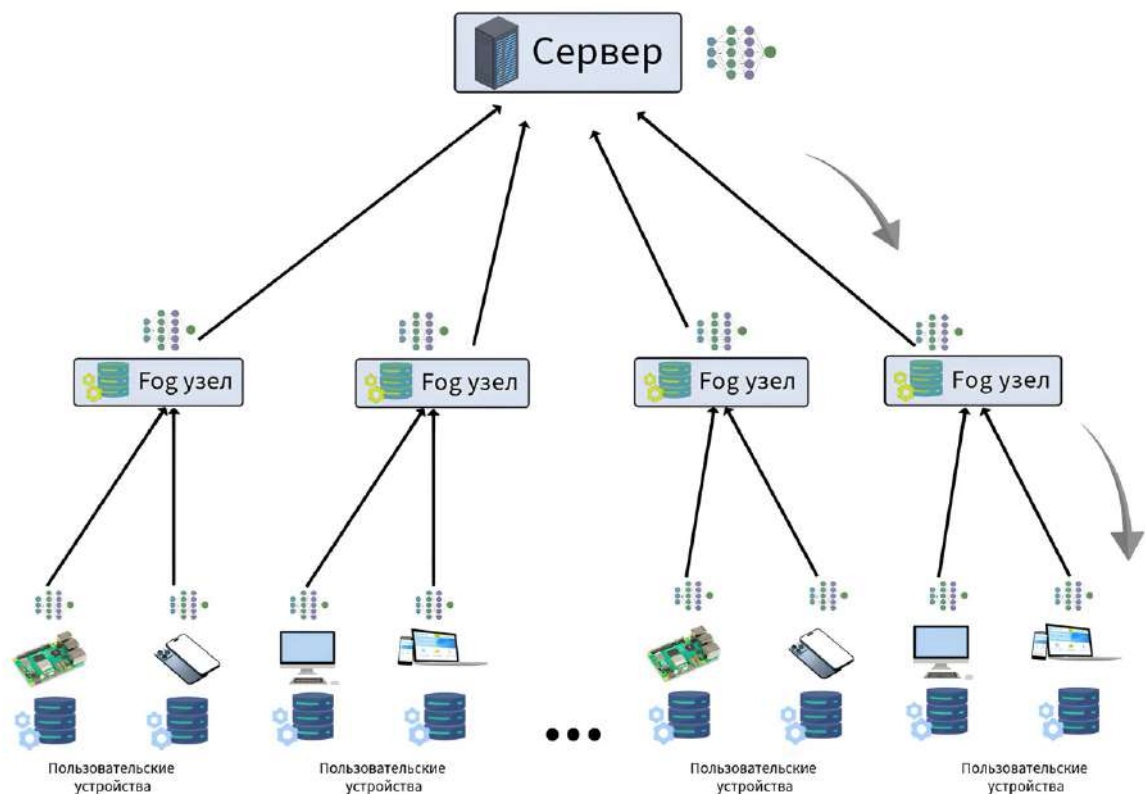


Рисунок 12 — Структура интеграции FL и FC

Сбор и обработка обновлений градиента на уровне граничных устройств.

Устройства выполняют локальную оптимизацию с помощью нескольких раундов обновления (используя алгоритмы типа SGD), при этом градиенты вычисляются на минибатчах локальных данных. Затем они вычисляют норму обновленного градиента $F^* = \min_{\theta} F(\theta)$ и скорость изменения градиента по сравнению с предыдущим раундом с помощью уравнения (5). Решения о передаче обновлений принимаются следующим образом:

– если $r_t^{(k)} < \tau_i$, то отправить «НЕТ» (не отправляйте обновление) и передавать только $r_t^{(k)}$ до ближайшего узла Fog;

– если $r_t^{(k)} \geq \tau_i$, то отправить обновление $\Delta\theta_t^{(k)}$ и $r_t^{(k)}$ до ближайшего узла тумана.

Обработка и агрегирование обновлений на уровне туманных вычислений. Узлы туманных вычислений выполняют следующие действия, чтобы минимизировать количество взаимодействия с центральным сервером и соответствующую вычислительную нагрузку:

1) сбор и агрегирование данных от пользовательских устройств: узел Fog получает $r_t^{(k)}$ значения от всех связанных устройств и сохраняет обновления градиента $\Delta\theta_t^{(k)}$ от устройства, где $r_t^{(k)} \geq \tau_i$;

2) динамический расчет порога на уровне туманных вычислений: среднее μ_i и стандартное отклонение σ_i скорости изменения градиента от устройств, посылающих обновления, вычисляются с помощью уравнений (6) и (7), динамический порог для следующего раунда устанавливается в соответствии с уравнением (8).

3) передача отфильтрованных обновлений модели ИИ: узел Fog агрегирует полученные обновления градиента и отправляет их на центральный сервер, обновления ниже порогового значения могут быть отброшены или временно сохранены.

Обновление глобальной модели на облачном уровне.

Глобальный агрегатор FL получает обновления от узлов туманных вычислений и вычисляет глобальное обновление модели на основе действительных градиентов:

$$\theta_{t+1} = \theta_t + \frac{\mu}{|S_t|} \sum_{k \in S_t} \Delta\theta_t^{(k)} \quad (9)$$

После обновления глобальной модели сервер отправляет новую модель обратно на узлы туманных вычислений для следующей итерации обучения.

Подготовка к следующей итерации.

Настройка коэффициента α основывается на результатах текущего раунда (количество полученных обновлений), сервер может настроить α для управления порогом τ_t , чтобы сбалансировать нагрузку на сеть и обеспечить необходимый уровень точности модели ИИ. Узлы FC получают обновлённую модель от глобального агрегатора FL и распространяют её среди граничных устройств, подготавливаясь к следующей итерации FL. В то же время узел Fog обновляет новый порог для проведения операции сравнения в следующем раунде.

Появление в будущем сетей, способных автономно управлять сетевыми ресурсами, оптимизировать процессы обработки данных и адаптироваться к изменениям в операционной среде без непосредственного вмешательства человека, означает значительный прогресс в сетевых технологиях. К таким сетям относятся автономные сети (Autonomous Networks, AN), определяемые МСЭ [66] как сети, способные выполнять определенные функции без вмешательства человека. Интеграция AN в модели FL-FC является потенциальным решением, концептуальная схема которого представлена на рисунке 13. В этой модели AN могут синтезировать локальные модели на основе периферийных устройств, оптимизировать обработку данных, координировать обучение моделей и обеспечивать необходимое качество функционирования системы распределённого ИИ. В данной работе предлагается интеграция AN с интегрированной моделью FL-FC, которая предназначена для минимизации задержек и экономии полосы пропускания, а также для улучшения способности системы адаптироваться к внезапным изменениям данных или условиям сети. Это решение отвечает требованиям приложений реального времени, включая URLLC, иммерсивные услуги, голографический тип коммуникаций и сетевые вселенные.

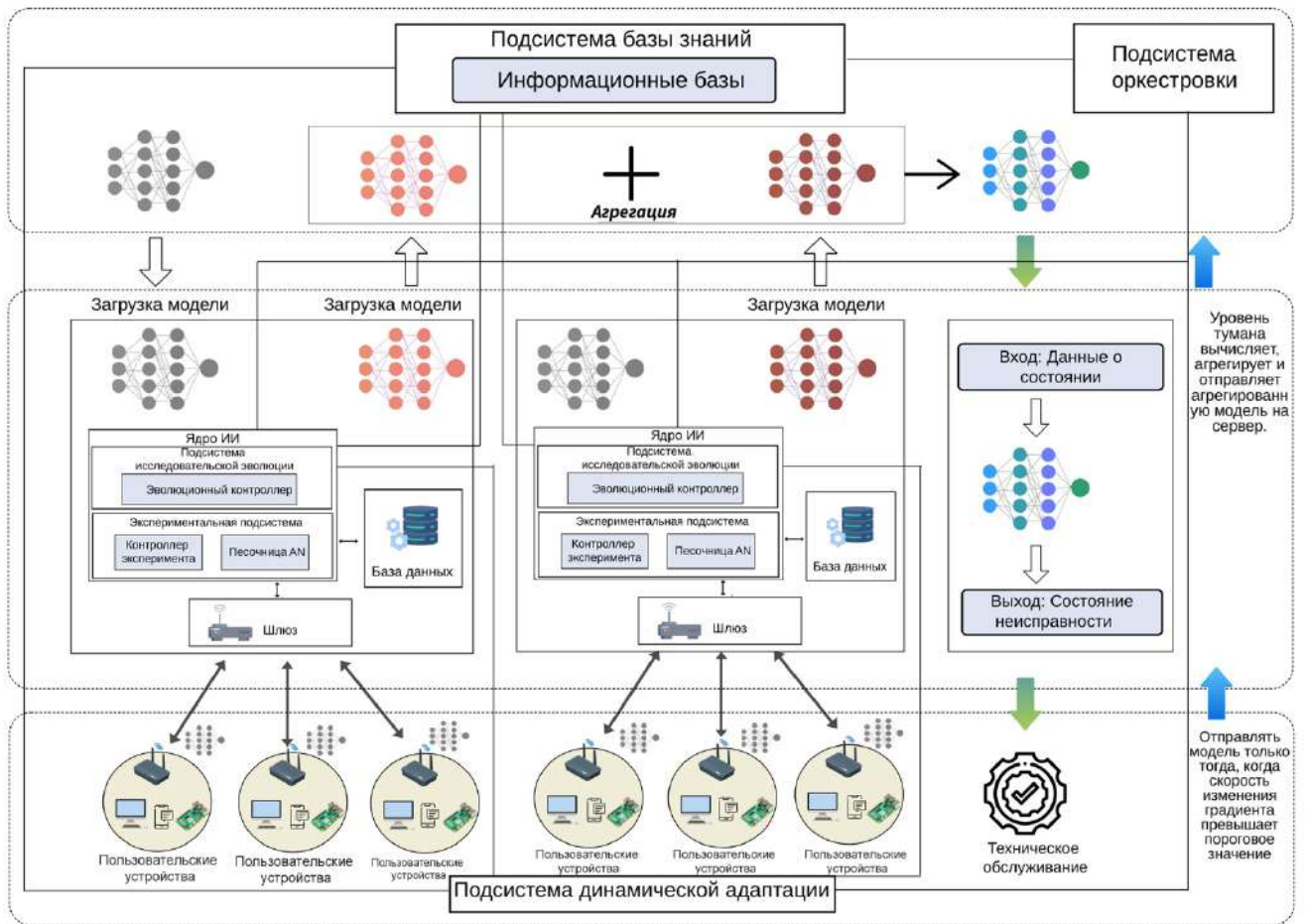


Рисунок 13 — Модель FL-FC с учетом архитектуры сетей IMT-2030 с искусственным интеллектом

Предлагаемая система реализуется посредством четырёх взаимосвязанных алгоритмов, охватывающих все уровни структуры FL – FC: обновление локальных моделей на уровне граничных устройств (Алгоритм 1), агрегацию и анализ обновлений на уровне туманных вычислений (Алгоритм 2), обновление глобальной модели на облачном уровне (Алгоритм 3) и цикл обновления порога скорости изменения градиента (Алгоритм 4).

Таблица 9 — Алгоритм обновления пользовательских устройств

Алгоритм 1: Обновление пользовательских устройств (локальные модели)

Input: Глобальная модель θ_t , порог τ_t (изначально $\tau_t=0$), размер мини-пакета B , количество эпох E , общее количество образцов данных n_k у устройства k .

Output: Сообщение $M_t^{(k)}$ отправлено в Fog уровень.

1. Инициализировать $\theta_{t+1}^{(k)} \leftarrow \theta_t$.
2. Разбить локальный набор данных B_k устройства k на мини-пакеты размером B .
3. Для каждой эпохи $i = 1, \dots, E$:
 - а. Для каждого мини-пакета b из B :
Обновление $\theta_{t+1}^{(k)} \leftarrow \theta_{t+1}^{(k)} - \eta \nabla L^k(\theta_t, b)$ (оптимизировать с учётом скорости обучения η).
 - б. Конец
4. Конец
5. Рассчитать обновление градиента $\Delta\theta_t^{(k)} = \theta_{t+1}^{(k)} - \theta_t$.
6. Вычислить скорость изменения градиента $r_t^{(k)}$ в соответствии с уравнением (1).
7. Если $r_t^{(k)} \geq \tau_t$ то:
Набор $M_t^{(k)} \leftarrow (\Delta\theta_t^{(k)}, r_t^{(k)})$ (отправить градиент).
8. Иначе:
Набор $M_t^{(k)} \leftarrow (НЕТ, r_t^{(k)})$ (отправлять только значение $r_t^{(k)}$).
9. Вернуть $M_t^{(k)}$ на слой Fog.

Таблица 10 — Алгоритм агрегации данных и их анализ на уровне FC

Алгоритм 2: Агрегация данных и их анализ на уровне FC

Input: Обновления от устройств $M_t^{(k)}$ в зоне контроля.

Output: Сообщение M_t^{Fog} отправленное на уровень сервера.

1. Собрать все $r_t^{(k)}$ от подключённых устройств.
2. Сохранить градиентные обновления $\Delta\theta_t^{(k)}$ от устройств с $r_t^{(k)} \geq \tau_t$.
3. Рассчитать среднее μ_t .
4. Рассчитать стандартное отклонение σ_t .
5. Обновить динамический порог для следующего раунда $\tau_{t+1} = \mu_t + \alpha\sigma_t$.
6. Агрегировать действительные градиентные обновления $\Delta\theta_t^{(k)}$ для отправки на сервер.
7. Установить $M_t^{(Fog)} \leftarrow (\Delta\theta_t^{Fog}, \tau_{t+1})$, где $\Delta\theta_t^{Fog}$ — агрегированное обновление градиентов от всех устройств уровня Fog, τ_{t+1} — обновлённый порог.
8. Вернуть M_t^{Fog} на уровень сервера.

Таблица 11 — Алгоритм обновления глобальной модели на облачном уровне

Алгоритм 3: Обновление глобальной модели на облачном уровне
Input: Обновления M_t^{Fog} с уровня Fog. Output: Обновлённая глобальная модель θ_{t+1}, отправленная обратно на уровень Fog. 1. Получение обновлений от узлов Fog. 1. Рассчитать глобальное обновление: θ_{t+1} 2. Вернуть глобальную модель θ_{t+1} узлам Fog для перераспределения.

Таблица 12 — Алгоритм цикла обновления порога на уровне FC

Алгоритм 4: Цикл обновления порога на уровне FC
Input: Обновление $M_t^{(k)}$ от устройств в соответствующей зоне, порог был рассчитан ранее.

Output: Обновить новую модель и порог для следующего раунда.

1. Получить новую обновлённую модель с уровня сервера.
2. Отрегулировать порог с коэффициентом α , чтобы соответствовать обучаемой модели.
3. Обновить новую модель и новый порог для устройств.

2.6 Тестирование предложенной модели и метода

Для оценки предлагаемого метода был использован набор данных MNIST, который содержит 70 тыс. размеченных изображений рукописных цифр (0–9), из которых 60 тыс. изображений входят в обучающую выборку, а 10 тыс. – в тестовую [67]. Каждое изображение имеет размер 28x28 пикселей в градациях серого. Используемая модель представляла собой простую сверточную нейронную сеть, основанную на архитектуре CNN с двумя сверточными слоями, с функциями активации выпрямленного линейного блока (rectified linear unit, ReLU). Для извлечения признаков и уменьшения размерности входных изображений применялись слои максимального объединения (Max Pooling). Затем извлеченные признаки преобразовывались в одномерный массив (flattening) и передавались в полносвязные слои, предсказывающие вероятности классов, при этом использовался механизм Dropout для уменьшения переобучения. Данная архитектура хорошо подходит для задач классификации изображений, таких как распознавание рукописных цифр в наборе данных MNIST.

Кроме того, в оценку был включен набор данных CIFAR-10 [68], состоящий из 60 тыс. цветных изображений, разделенных на 10 различных классов, из которых 50 тыс. предназначены для обучения и 10 тыс. – для тестирования. Каждое изображение имеет размер 32x32 пикселя и включает три цветовых канала, что ставит более сложную задачу классификации по сравнению с набором данных MNIST. Для этого набора данных та же простая архитектура CNN была

адаптирована с изменениями для работы с изображениями большего размера и цветовыми каналами. В частности, слои свертки были изменены для обработки входных данных с тремя каналами, а также были добавлены дополнительные слои для учета возросшей сложности данных. Слои Max Pooling и активационные признаки ReLU были сохранены для извлечения точности, а слои с полной связью и Dropout были использованы для прогнозирования вероятностей классов и уменьшения переобучения. Набор данных CIFAR-10 является стандартным тестовым набором для задач классификации изображений и обеспечивает более полную оценку предложенного метода.

Был смоделирован предложенный метод и проведено сравнение с двумя моделями: FL-FC простой модели и моделью с методом выборки устройств, основанной на алгоритме случайного выбора устройств, участвующих в обучении на каждом этапе. Кроме того, в предлагаемом методе, помимо динамического порогового значения для выбора устройств для обучения, было введено минимальное количество устройств, необходимых для участия. Это требование направлено на предотвращение ситуации, когда в обучении участвует слишком мало устройств, что может негативно сказаться на производительности модели.

Чтобы продемонстрировать эффективность метода, оценка результатов обучения происходила по следующим трём основным критериям:

- (i) время обучения и эффективность обучения (сходимость);
- (ii) объем передаваемых данных за итерацию (сетевое взаимодействие с центральным сервером модели ИИ);
- (iii) динамика средних потерь в процессе обучения.

Время обучения и сходимость модели: результаты моделирования на наборе данных MNIST, представленные на рисунке 14, показывают, что предлагаемый метод значительно улучшает время обучения по сравнению с двумя другими моделями, при этом обеспечивая необходимую сходимость, аналогичную стандартной FL-FC. Динамика средних потерь в процессе обучения на наборе MNIST представлена на рисунке 15.

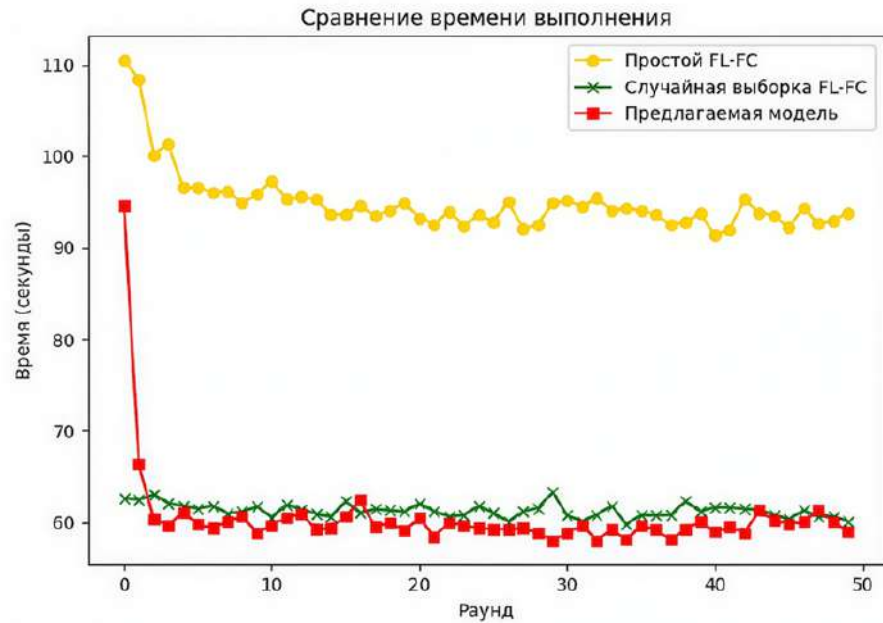


Рисунок 14 — Время выполнения (MNIST)

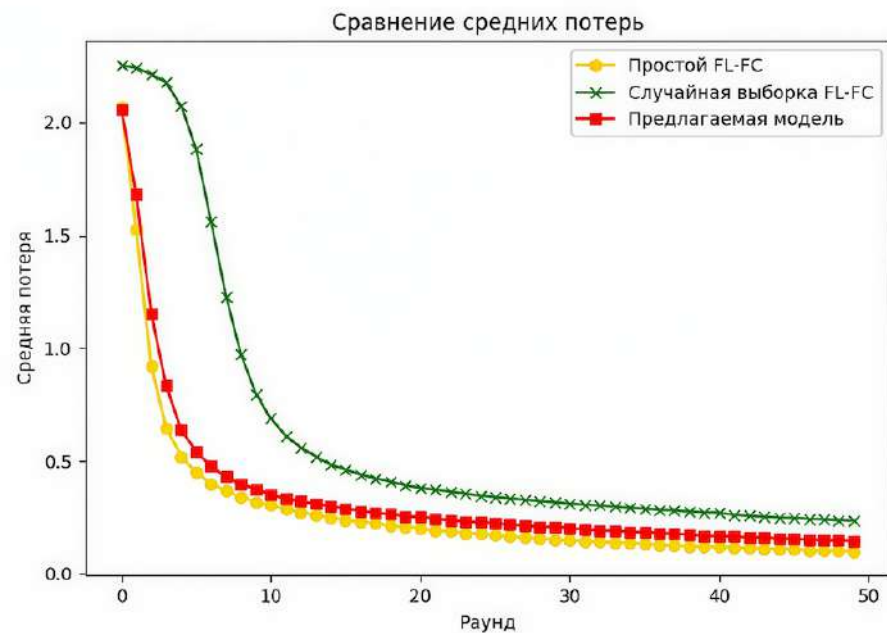


Рисунок 15 — Средние потери (MNIST)

Это демонстрирует, что применение метода динамического порогового значения для выбора устройств на основе скорости изменения градиента функции потерь не только ускоряет обучение, но и обеспечивает сходимость за меньшее количество итераций.

Точность и сетевые ресурсы: как показано на рисунке 16, предложенный метод требует меньшего сетевого взаимодействия с сервером центральной модели

ИИ для достижения уровня точности, эквивалентного FL-FC простой модели. Кумулятивный объем сетевого взаимодействия на наборе MNIST представлен на рисунке 17.

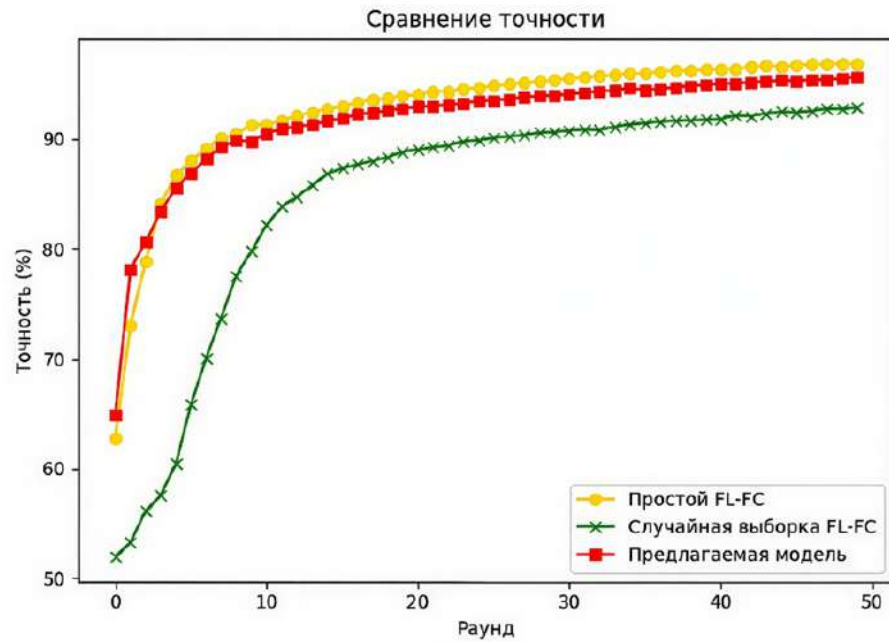


Рисунок 16 — Сравнение точности (MNIST)

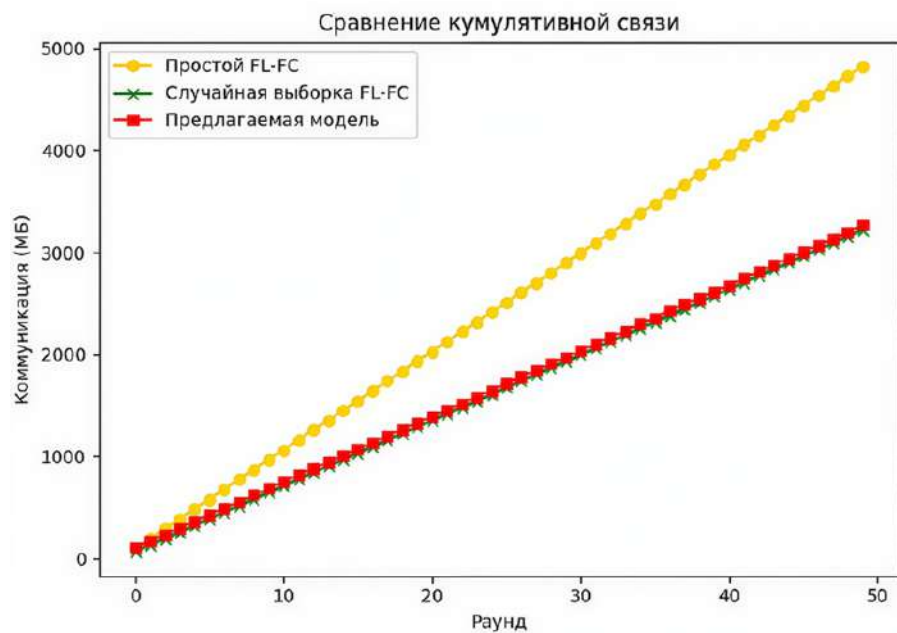


Рисунок 17 — Кумулятивная связь (MNIST)

По сравнению с базовой моделью предложенный метод позволяет снизить необходимое сетевое взаимодействие с центральным сервером модели ИИ в ходе всего процесса обучения на 39% на MNIST при сохранении точности модели ИИ

на том же уровне.

Аналогичные результаты получены на наборе данных CIFAR-10. Время выполнения и динамика средних потерь представлены на рис. 17 и 18 соответственно.

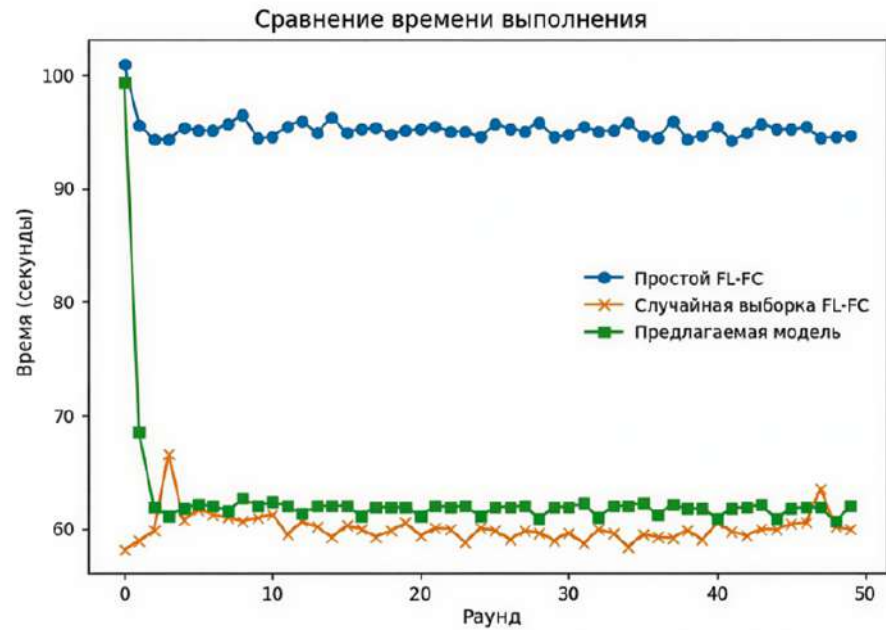


Рисунок 18 — Время выполнения (CIFAR-10)

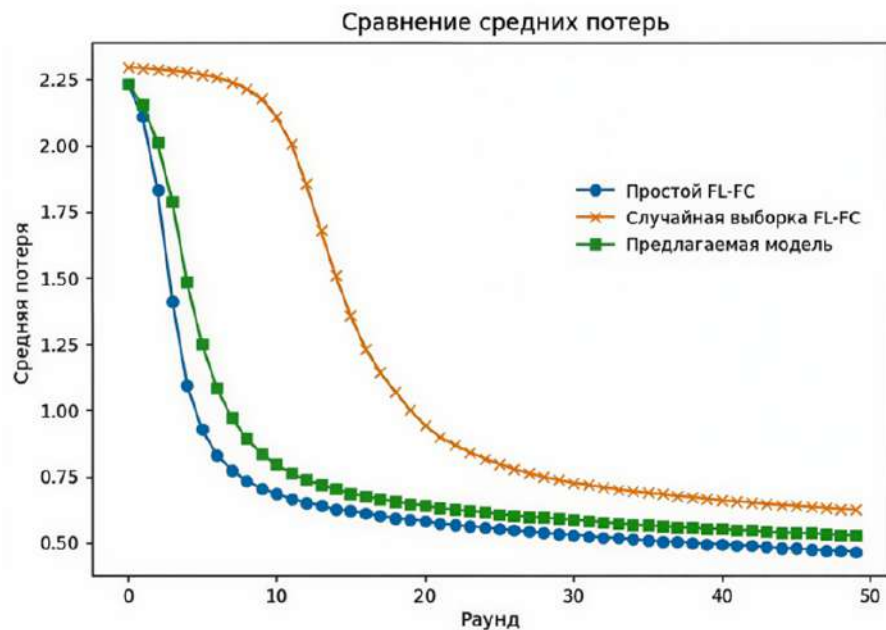


Рисунок 19 — Средние потери (CIFAR-10)

Сравнение точности и кумулятивный объём сетевого взаимодействия на наборе CIFAR-10 представлены на рисунках 20 и 21.

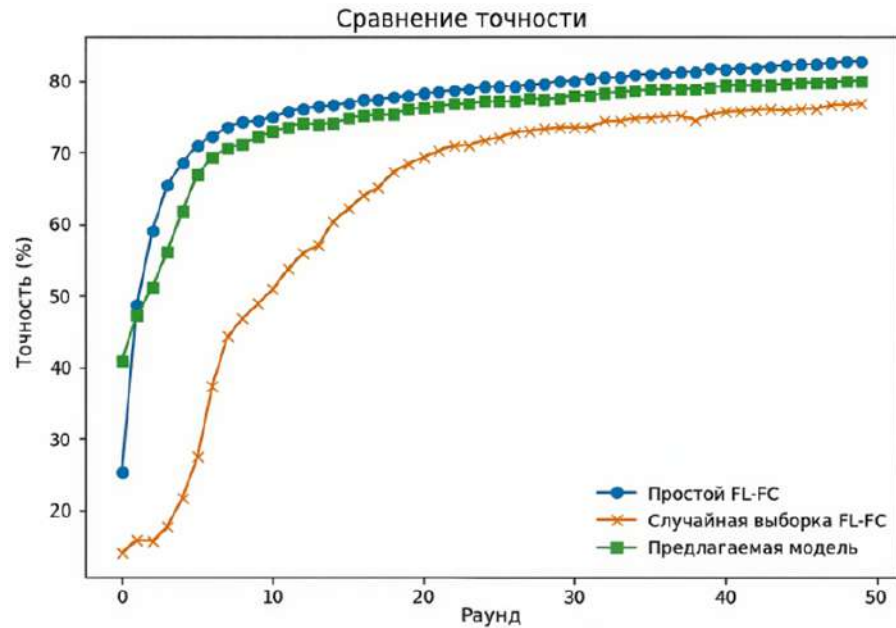


Рисунок 20 — Сравнение точности (CIFAR-10)

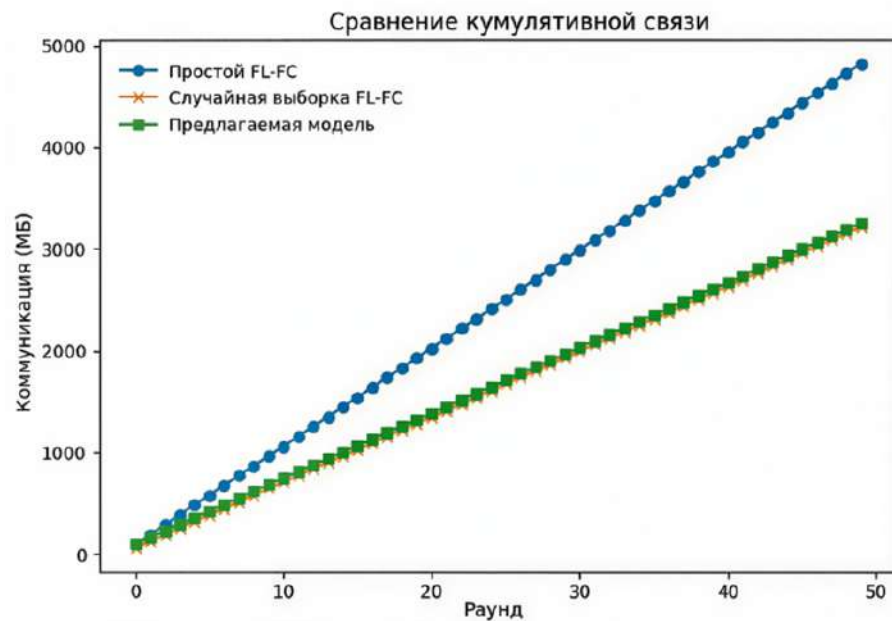


Рисунок 21 — Кумулятивная связь (CIFAR-10)

За счёт предложенного метода сетевое взаимодействие на CIFAR-10 удаётся сократить на 31% при сохранении точности модели. Среди трёх рассмотренных подходов наибольшую экономию сетевых ресурсов продемонстрировала модель случайной выборки устройств. Вместе с тем этот подход не обеспечивает устойчивую точность модели в средах с неоднородным распределением данных: при случайном исключении устройств часть из них может оказаться критически

важной для обучения. Именно этот недостаток устраняет метод динамического порога, основанный на скорости изменения градиента функции потерь. Порог адаптивно корректируется в зависимости от поступающих обновлений устройств, что позволяет поддерживать баланс между точностью модели и объёмом потребляемых сетевых ресурсов. Особенно заметен этот эффект в условиях неоднородного распределения данных на устройствах.

Полученные результаты подтверждают, что предложенный метод позволяет поддерживать высокую точность модели ИИ в условиях FL с иерархической структурой FC. При этом время обработки распределённых задач заметно сокращается по сравнению с базовыми моделями. Совместное применение отобранной группы устройств и метода динамического порога скорости изменения градиента ведёт к уменьшению числа сетевых взаимодействий между устройствами и центральным сервером.

2.7 Выводы по Главе 2

1. В главе разработан метод выборки пользовательских устройств для участия в раундах федеративного обучения в интегрированной модели с туманными вычислениями, реализующей принцип AI-native в распределённой граничной среде сетей IMT-2030. В предложенной структуре узлы туманных вычислений выполняют двойную функцию, включающую промежуточную агрегацию параметров модели и адаптивное вычисление динамического порогового критерия выбора устройств. Это позволяет принимать решения об участии устройств в каждом раунде без обращения к центральному серверу.

2. В основе предложенного метода лежит следующий принцип: скорость изменения нормы градиента функции потерь используется как критерий информационной ценности обновления каждого устройства. В каждом раунде пороговое значение адаптивно вычисляется на уровне туманных вычислений как сумма среднего значения и стандартного отклонения скоростей изменения нормы градиента участвующих устройств. Такой подход одновременно учитывает неоднородность локальных данных и текущую фазу сходимости модели.

3. Экспериментальная оценка на наборах данных MNIST и CIFAR-10 подтвердила эффективность предложенного метода. Предложенный метод снижает объём сетевого взаимодействия между устройствами и центральным сервером на 39% на наборе данных MNIST и на 31% на CIFAR-10 при сохранении точности модели на уровне базовой конфигурации FL-FC. В отличие от метода случайной выборки, предложенный подход обеспечивает устойчивую точность в условиях неоднородного распределения данных за счёт приоритизации устройств по информационной значимости их обновлений. Настраиваемость коэффициента α делает метод применимым к различным сценариям развёртывания в сетях связи шестого поколения.

ГЛАВА 3. ДЕЦЕНТРАЛИЗОВАННАЯ ОБЛАЧНАЯ СТРУКТУРА И МОДЕЛЬ ФЕДЕРАТИВНОГО ОБУЧЕНИЯ С ТУМАННЫМИ АЛЬЯНСАМИ ДЛЯ ГОРОДСКИХ СЕТЕВЫХ ВСЕЛЕННЫХ

3.1 Введение

В первой главе показано, что в предлагаемой архитектуре слой облачных вычислений обеспечивает координацию ресурсов, выступая в качестве основного управляющего центра для нижележащих уровней туманных вычислений. Однако, когда система туманных вычислений работает под управлением облака, существует большая вероятность сбоев и нарушений непрерывной работы. Это ограничение становится особенно важным в ситуациях, характеризующихся географической распределённостью узлов FC. Кроме того, в экстремальных ситуациях, когда отсутствует стабильное сетевое подключение к облаку в режиме реального времени, это ограничение становится еще более актуальным. Интеллектуальные транспортные системы (ITS, аббр. от англ. Intelligent Transportation Systems) работают в динамичной городской среде. Прогнозирование городских транспортных потоков в таких условиях требует высокой скорости обработки данных, минимальных задержек и устойчивости к нестабильности сетевого соединения. Настоящая глава посвящена второй задаче, сформулированной в первой главе - разработке иерархической децентрализованной структуры FL с туманными альянсами, способной функционировать автономно без непрерывного обращения к облачному агрегатору/оркестратору.

Прогнозирование городских транспортных потоков в городской среде относится к числу задач, требующих совместного моделирования пространственных и временных зависимостей. В современных исследованиях для описания пространственных закономерностей применяются графовые нейронные сети (GNN, аббр. от англ. Graph Neural Networks). Для моделирования временной динамики используются рекуррентные нейронные сети (RNN, аббр. от англ. Recurrent Neural Networks) и свёрточные нейронные сети (CNN, аббр. от англ. Convolutional Neural Networks) [69]. Однако большинство существующих моделей

построено на основе централизованной облачной архитектуры, в которой сервер остаётся центральным узлом агрегации и обучения. Такой подход создаёт высокую нагрузку на каналы связи, повышает риски нарушения конфиденциальности данных и снижает устойчивость системы к сбоям [70], [71]. Помимо этого, централизованное обучение не учитывает региональную неоднородность городских транспортных потоков, которая существенно варьируется в различных географических зонах.

В ряде работ предпринята попытка объединить методы федеративного обучения с моделями пространственно-временного прогнозирования, включая FedAGCN [72], FedSTN [73], T-GCN [74], FASTGNN [75] и FCGCN [76]. Результаты этих работ подтверждают эффективность подобного сочетания на реальных данных. Вместе с тем все перечисленные методы сохраняют зависимость от центрального сервера агрегации моделей искусственного интеллекта, что обеспечивает их эффективное функционирование в стабильных условиях, однако порождает существенные ограничения при практическом развёртывании. При потере соединения с облаком вследствие перегрузки сети, стихийных бедствий или технических сбоев процесс агрегации параметров модели ИИ прерывается, что приводит к нарушению функционирования системы. Кроме того, централизованная агрегация не учитывает региональную специфику данных. Транспортные шаблоны (т.н. паттерны) в различных районах города существенно различаются между собой, и единая глобальная модель неизбежно теряет локальные особенности. Совокупность указанных факторов ставит под сомнение устойчивость подобных решений в реальных условиях развёртывания.

В настоящей главе диссертации предлагается децентрализованная архитектура FL с FC на основе адаптивной графовой свёрточной рекуррентной нейронной сети DFL-FC-AGCRN. Архитектура объединяет три взаимодополняющих компонента. Первый обеспечивает децентрализованное FL с одноранговой агрегацией между узлами FC, полностью исключая центральный сервер из операционного цикла. Второй реализует уровень FC, который переносит вычисления к источникам данных и осуществляет региональную координацию без

постоянного обращения к центральному облаку. Третий представляет собой модель AGCRN, предназначенную для описания пространственно-временных зависимостей городских транспортных потоков [77].

Предлагаемая архитектура является иерархической и децентрализованной. Децентрализация понимается здесь в строгом смысле: единый облачный сервер полностью исключён из операционного цикла. Координация внутри альянсов осуществляется через узлы AFN, которые образуют промежуточный иерархический уровень между граничными устройствами и облаком. Структура настоящей главы построена следующим образом. В разделе 3.2 представлен аналитический обзор существующих методов прогнозирования городских транспортных потоков и обоснована недостаточность централизованных подходов применительно к задачам ITS в сетях IMT-2030. В разделе 3.3 изложены формализованная постановка задачи и предлагаемая структура DFL-FC-AGCRN. В разделе 3.4 описаны механизм раздельной агрегации параметров и метод кластеризации узлов. В разделе 3.5 приведены результаты экспериментальной оценки на наборах данных METR-LA и PeMSD7(M). Выводы по главе суммируют полученные результаты и определяют исследовательскую задачу, рассматриваемую в главе 4.

3.2 Анализ существующих решений и постановка задачи

Методы глубокого обучения, включая графовые нейронные сети (GNN), рекуррентные (RNN) и свёрточные (CNN) нейронные сети, нашли широкое применение в задачах прогнозирования городских транспортных потоков. В работе [77] авторы представили и доказали высокую эффективность модели AGCRN в задаче прогнозирования на основе пространственных и временных образцов данных. В работе [78] продемонстрирована высокая точность подобных подходов на реальных наборах данных о трафике. Тем не менее перечисленные модели построены на централизованной архитектуре: все исходные данные о трафике передаются на единый сервер для обучения, что порождает высокую нагрузку на каналы связи, риски нарушения конфиденциальности и уязвимость к сбоям в сети.

Именно это ограничение стимулировало развитие методов федеративного обучения применительно к задачам прогнозирования городских транспортных потоков.

В работах [72, 73, 74] представлены методы, основанные на обучении общей модели без передачи исходных данных: клиенты обмениваются только параметрами (весами) нейронной сети. Этот подход, включающий алгоритмы FedAvg и FedSTN, повышает уровень защиты данных. Однако общим ограничением данных методов остаётся использование традиционной централизованной облачной модели, в которой облако выполняет функции агрегации и распределения модели. Данные методы не исключают риски единой точки отказа и потребности в сетевых ресурсах, что может стать «узким местом» при повышении количества данных при обмене с центральной моделью ML. Также данный метод недостаточно учитывает специфику локальных данных в разных географических зонах. В некоторых исследованиях рассматривалась децентрализация с помощью блокчейна [79]. Однако это решение существенно увеличивает потребности в вычислительных ресурсах и в итоге затрудняет работу системы FL в реальном времени.

Параллельно развиваются полностью децентрализованные методы FL, не требующие центрального сервера агрегации. Lian et al. [80] предложили алгоритм D-PSGD, в котором каждый узел обменивается параметрами только с топологически смежными узлами сети, показав, что децентрализованные алгоритмы могут превосходить централизованные по скорости сходимости в условиях ограниченной пропускной способности. Turazza и др. в работе [81] предложили подход на основе сегментированного gossip-протокола для FL с использованием блок-чейна, позволяющий эффективно использовать пропускную способность между узлами без привлечения центрального сервера агрегации. Yu и др. в работе [82] разработали алгоритм CD-SGD, сочетающий компрессию градиентов и компенсацию задержек для повышения эффективности децентрализованного обучения. Однако перечисленные методы оперируют абстрактными (логическими) топологиями сети и не учитывают географическую структуру узлов. Кроме того, они не учитывают специальные архитектуры для пространственно-временного

моделирования городских транспортных потоков и не решают проблему неоднородности данных (Non-IID), характерную для территориально распределённых датчиков. Предлагаемый подход отличается интеграцией географической кластеризации (HDBSCAN, K-means) с моделью AGCRN на уровне туманных узлов, что обеспечивает адаптацию к реальной топологии транспортной сети и эффективное решение проблемы Non-IID через механизм отдельной агрегации параметров.

3.3 Предлагаемая общая структура DFL-FC-AGCRN

Сеть дорожных датчиков состоит из измерительных станций, размещённых по всей исследуемой территории. Каждая станция рассматривается как вершина, которая собирает данные о трафике в реальном времени.

Набор станций: $G = \{G_k\}_{k=1}^K$ где K - число измерительных станций городского трафика. Временные ряды данных: каждая станция G_k формирует данные временного ряда $B_k = \{(X_k, y_k)\}$, где: $X_k = \{x_k^t\}_{t=1}^T \in \mathbb{R}^{T \times d}$ входная последовательность признаков на станции k , T число временных шагов, d - число признаков, $y_k = \{y_k^t\}_{t=1}^T \in \mathbb{R}^T$ соответствующий объём трафика на станции k .

Важная особенность данных о городском трафике является высокая неоднородность (Non-IID) между измерительными станциями. Это связано с различиями в состоянии дорог, плотности транспорта и шаблонах движения в разных районах. Концептуальная схема предлагаемой архитектуры DFL-FC-AGCRN в условиях умного города представлена на рисунке 22. По этой причине наборы данных будут иметь существенные различия, что и является серьёзным вызовом при применении методов распределённого обучения.



Рисунок 22 — Концептуальная схема структуры DFL-FC-AGCRN в умном городе

Для решения задачи прогнозирования в условиях подобной неоднородности данных в качестве базовой модели глубокого обучения была выбрана архитектура AGCRN, предложенная Bai и др. [76]. Модель AGCRN устраняет два основных недостатка предшествующих методов прогнозирования городских транспортных потоков. Во-первых, традиционные модели используют только глобально разделяемые параметры, которые не отражают индивидуальные паттерны отдельных узлов сети. Во-вторых, большинство существующих подходов зависят от заранее заданного графа пространственных связей, что снижает их гибкость при изменении топологии транспортной сети.

Для каждого туманного узла F_n , строится локальный граф $G_n = (V_n, E_n)$:

- $V_n = \{s_i \mid s_i \in C_n\}$ множество вершин, где каждая вершина соответствует измерительной станции (датчику), входящей в кластер C_n ; общее число станций в кластере равно $|V_n|$;

- E_n множество рёбер, отражающих пространственные связи между станциями.

Такой локальный граф описывает топологию транспортной сети внутри кластера C_n и позволяет модели учитывать пространственные зависимости между соседними измерительными станциями. Следует подчеркнуть, что вершины графа V_n – это именно измерительные станции (датчики), а не туманные узлы: каждый туманный узел F_n управляет обучением модели на всём графе G_n своего кластера. Модель AGCRN объединяет два ключевых модуля с рекуррентной нейронной сетью для моделирования детализированных пространственно-временных зависимостей в данных о городском транспортном потоке:

1. *Адаптивное обучение параметров узла графа (измерительная станция) (NAPL, аббр. от англ. Node-Adaptive Parameter Learning)*

Модуль NAPL формирует индивидуальные параметры для каждого узла, благодаря чему модель способна обучаться специфическим паттернам временных рядов, а не только общим закономерностям:

$$\Theta = E_G \cdot W_G \quad (11)$$

где: $E_G \in \mathbb{R}^{N \times d}$ - матрица встраиваний узлов графа (embedding matrix); d - размерность встраивания; $W_G \in \mathbb{R}^{d \times C \times F}$ - общий пул весов, которые являются общими для всех вершин графа; каждая станция i извлекает из этого пула свой индивидуальный набор параметров на основе своего вектора встраивания;

Для станции i модуль NAPL извлекает параметры Θ_i из пула весов на основе вектора встраивания E_i^G .

Графовая сверточная сеть с расширением NAPL формально определяется следующим образом:

$$Z = \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) X E_G W_G + E_G b_G \quad (12)$$

где A и D — матрица смежности и диагональная матрица степеней соответственно.

2. *Генерация адаптивных графов по данным (DAGG, аббр. от англ. Data-Adaptive Graph Generation)*

Модуль DAGG автоматически обучает матрицу смежности по данным, а не использует заранее заданный граф:

$$\tilde{A} = \text{Softmax}(\text{ReLU}(E_A \cdot E_A^T)) \quad (13)$$

где: $E_A \in \mathbb{R}^{N \times d_e}$ — обучаемая матрица встраиваний узлов графа, в которой каждая строка представляет вектор признаков соответствующей станции для генерации графа; Softmax — нормализует матрицу связей; ReLU — обнуляет отрицательные значения.

3. Интеграция NAPL, DAGG and GRU

AGCRN объединяет оба модуля с GRU и одновременно моделирует пространственные и временные зависимости:

$$A_e = \text{softmax}(\text{ReLU}(EE^T)) \quad (14)$$

$$z_t = \sigma(A_e[X_{:,t}, h_{t-1}]EW_z + Eb_z) \quad (15)$$

$$r_t = \sigma(A_e[X_{:,t}, h_{t-1}]EW_r + Eb_r) \quad (16)$$

$$\hat{h}_t = \tanh(A_e[X_{:,t}, r \odot h_{t-1}]EW_{\hat{h}} + Eb_{\hat{h}}) \quad (17)$$

$$h_t = z \odot h_{t-1} + (1 - z) \odot \hat{h}_t \quad (18)$$

где: E - единая матрица встраиваний узлов графа, используемая и в NAPL, и в DAGG; $W_z, W_r, W_{\hat{h}}, b_z, b_r, b_{\hat{h}}$ — обучаемые параметры; $\odot$ — поэлементное (Адамара) умножение.

Благодаря своей структуре AGCRN: через модуль NAPL захватывает индивидуальные паттерны узлов; через модуль DAGG автоматически учит пространственные связи между временными рядами, не требуя заранее заданного графа. В работе Бай и коллег показана высокая эффективность AGCRN в задачах прогноза трафика. Это демонстрирует, что AGCRN особенно хорошо подходит для задач прогноза городских транспортных потоков с неоднородными данными, когда временные ряды из разных источников имеют разные паттерны. Поэтому в данном исследовании модель AGCRN используется как локальная модель обучения на туманных узлах

В предлагаемой модели туманные узлы классифицируются по выполняемым функциям. Выделяются два типа узлов: обучающие туманные узлы (TFN, от англ. Trainable Fog Node) и агрегационные туманные узлы (AFN, от англ. Aggregation Fog Node). Обучающие узлы получают данные от измерительных станций и используют их для обучения моделей. Агрегационные узлы выполняют функцию агрегирования результатов обучения. Каждый узел TFN F_n самостоятельно обучает модель AGCRN на локальном наборе данных.

Функция потерь для обучения модели имеет вид:

$$L(\theta_n) = \frac{1}{|B_n|} \sum_{(X,y) \in B_n} \|f_{\theta_n}(X) - y\|_2^2 \quad (19)$$

где: θ_n — параметры модели AGCRN на туманном узле F_n ; f_{θ_n} — прогнозная функция модели; $\|\cdot\|_2^2$ — квадрат L2-нормы.

Обновление параметров выполняется по градиентному спуску: $\theta_{n+1} \leftarrow \theta_n - \eta \nabla_{\theta_n} L(\theta_n)$ где η — шаг обучения. Процесс обучения модели выполняется независимо на каждом туманном узле. TFN узлы собирают, обрабатывают и обучаются на данных станций внутри своего кластера. Это одновременно защищает приватность данных и уменьшает вычислительную нагрузку на каждый узел.

Описанный выше процесс локального обучения предполагает, что датчики трафика уже распределены по кластерам, каждый из которых закреплён за соответствующим туманным узлом. Формирование таких кластеров является самостоятельной задачей: от её решения напрямую зависит качество локальных моделей и эффективность последующей агрегации. В данном исследовании для кластеризации датчиков применяется алгоритм иерархической пространственно-плотностной кластеризации с учётом шума (HDBSCAN, аббр. от англ. Hierarchical Density-Based Spatial Clustering of Applications with Noise).

Алгоритм использует несколько масштабов расстояния, что позволяет отделять плотные кластеры от разреженных областей, интерпретируемых как шум. Стоит отметить, что данный метод сильно зависит от самих данных и требует

меньше параметров задаваемых пользователем. По вычислительной сложности алгоритм сопоставим с DBSCAN, но часто превосходит К-средних и другие методы при кластеризации крупномасштабных данных [83]. Таким образом, в данном исследовании был выбран HDBSCAN как метод для кластеризации сложных сетей датчиков трафика.

Процедура HDBSCAN включает следующие шаги:

1. Вычисление попарной матрицы расстояний между станциями по формуле гаверсина. Формула гаверсина позволяет вычислять расстояние по дуге большого круга между двумя точками на поверхности Земли по их географическим координатам с учётом кривизны Земли. Поэтому она подходит для пространственной кластеризации измерительных станций и является более корректной, чем расчёт евклидова расстояния непосредственно по широте и долготе:

$$d = 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (20)$$

где R — радиус Земли (≈ 6371 km), ϕ_1, ϕ_2 и λ_1, λ_2 — широта и долгота двух точек (в радианах).

2. Построение минимального остовного дерева (MST, аббр. от англ. Minimum Spanning Tree) на основе матрицы расстояний. Каждая вершина графа соответствует измерительной станции, а вес каждого ребра определяется гаверсиновым расстоянием между соответствующими станциями. MST соединяет все точки с минимальной суммарной длиной рёбер и служит основой для построения иерархии.

3. Построение иерархии кластеров с использованием расстояния взаимной достижимости. Начиная с минимального остовного дерева MST, алгоритм модифицирует рёбра с учётом плотности и формирует «сжатое» дерево кластеров. Каждое горизонтальное сечение дерева соответствует определённому значению ε и задаёт кластеры на соответствующем уровне плотности.

4. Выделение итоговых кластеров из построенной иерархии. На заключительном этапе алгоритм HDBSCAN анализирует иерархию кластеров и выделяет группы, соответствующие плотным областям распределения данных. Выделенные кластеры могут иметь различную форму и размер. Точки, не отнесённые ни к одному из кластеров, помечаются как шум.

Для алгоритма HDBSCAN не требуется заранее задавать число кластеров. Он автоматически находит оптимальное разбиение на основе структуры данных. Это особенно важно для сценария кластеризации измерительных станций трафика, где оптимальное число кластеров заранее неизвестно и зависит от пространственного размещения станций.

Центр каждого кластера C_n вычисляется как среднее значение координат:

$$\bar{\phi} = \frac{1}{|C|} \sum_{i \in C} \phi_i, \quad \bar{\lambda} = \frac{1}{|C|} \sum_{i \in C} \lambda_i \quad (21)$$

Туманный узел F_n размещается на станции, которая находится ближе всего к центру кластера. Это обеспечивает выгодное положение для сбора и обработки данных от всех станций в кластере. Размещение туманного узла в центре минимизирует среднее расстояние до станций, снижает задержку распространения и экономит энергию узлов.

В данном исследовании для верификации разработанной структуры и метода, были использованы два реальных набора данных: METR-LA с 207 датчиками трафика и PeMSD7(M) с 228 датчиками [84]. Данные содержат измерения трафика в реальном времени и географические координаты датчиков.

После кластеризации датчиков с помощью HDBSCAN, каждый кластер датчиков представляется соответствующим туманным узлом, который является медоидом этого кластера. Целью на этом этапе метода является организация туманных узлов в альянсы для поддержки DFL в среде FC. Формирование альянсов снижает сетевую нагрузку внутри альянса, повышает стабильность системы,

сохраняет приватность данных и уменьшает задержку, что соответствует требованиям распределённого интернета вещей и 6G сетей [85].

Теоретической основой служит иерархическая модель FL, в которой туманные узлы объединяются в альянсы и образуют промежуточный слой между локальными датчиками и глобальным облаком.

Туманные альянсы — это группа туманных узлов, объединённых по географической близости, которые совместно выполняют децентрализованную агрегацию параметров модели в рамках федеративного обучения. Каждый альянс функционирует автономно и координируется через узел агрегации.

Недавние работы показывают, что такие иерархии позволяют более эффективно использовать сетевые ресурсы: модели сначала агрегируются на промежуточных узлах, а не передаются напрямую в облако. В отличие от слоя датчиков, где неравномерная плотность их распределения в пространстве и неизвестное число кластеров требуют применения методов кластеризации на основе плотности (таких как HDBSCAN), способных работать с шумом и кластерами сложной формы, слой туманных узлов уже «сглажен» и представлен более стабильными географическими репрезентативными точками. Для упрощения предлагается использовать K-means для кластеризации туманных узлов. Этот метод позволяет задать фиксированное число кластеров, минимизирует внутрикластерную дисперсию и легко поддерживает дополнительные ограничения, важные для практической реализации [86].

Число альянсов k определяется задачей развёртывания. В данном исследовании значения $k \in \{1, 2, 3, 4, 5, 6\}$ выбраны для объективной оценки плотности от грубой (мало коалиций, много узлов в каждой) до более детальной (много коалиций, мало узлов в каждой). Для корректной работы алгоритма K-means с географическими данными координаты туманных узлов (широта φ , долгота λ) преобразуются в локальную метрическую систему с помощью равнопромежуточной проекции относительно средней точки $(\bar{\varphi}, \bar{\lambda})$:

$$x_i = R \cdot (\lambda_i - \bar{\lambda}) \cdot \cos \bar{\varphi}, \quad y_i = R \cdot (\varphi_i - \bar{\varphi}), \quad (22)$$

Данная проекция ограничивает искажения на исследуемой локальной территории и используется для представления географических координат в метрической системе при работе алгоритма K-means. Независимо от этого матрица попарных расстояний D_{fog} , вычисленная по формуле гаверсина для исходных географических координат, используется для оценки расстояний между туманными узлами внутри каждого альянса и последующего выбора основного и резервного AFN. Это позволяет учитывать фактическое географическое расстояние между узлами, а не только евклидово расстояние между их проекционными координатами.

Стандартный алгоритм *K-means* может создавать несбалансированные кластеры или кластеры, состоящие из одного узла, особенно при разреженном и неравномерном распределении туманных узлов. Это связано с тем, что классический *K-means* минимизирует только суммарное внутрикластерное расстояние и не учитывает ограничение на минимальное количество узлов в каждом кластере. Поэтому отдельный центроид может оказаться ближе всего к небольшому числу узлов, что приводит к формированию слишком малых альянсов. Это проблематично для децентрализованного федеративного обучения, поскольку альянсы с малым количеством участников не позволяют эффективно агрегировать локальные модели.

Чтобы снизить этот риск, в работе вводится ограничение на минимальный размер альянса: $\text{min_size} = 2$ узла. Это ограничение гарантирует наличие в каждом альянсе как минимум двух участников и позволяет выполнять агрегацию нескольких локальных моделей внутри альянса. Кроме того, оно уменьшает влияние данных, полученных только от одного узла [87].

Дополнительно для каждого альянса проверяется его радиус R_A „, определяемый как расстояние от AFN до наиболее удалённого узла-участника. Если R_A превышает заданный порог R_{max} , система инициирует корректировку числа альянсов k или повторное разбиение туманных узлов. Это позволяет ограничить максимальное расстояние передачи данных внутри альянса, снизить

задержку взаимодействия и уменьшить объём сетевого обмена между AFN и узлами-участниками. Таким образом, учитываются ограничения туманной сети по радиусу связи, задержке передачи и доступным сетевым ресурсам.

Внутри каждого альянса мы выбираем два ключевых узла: (i) основной AFN, определяемый как минимаксный центр альянса в географическом пространстве, (ii) резервный AFN, определяемый как следующий по критерию минимаксного центра. Формально,

$$Primary\ AFN(A) = \underset{j \in A}{argmin} \max_{i \in A} D_{fog}(i, j) \quad (23)$$

$$Backup\ AFN(A) = \underset{j \in A \setminus \{Primary\ AFN\}}{argmin} \max_{i \in A} D_{fog}(i, j) \quad (24)$$

Предложенный способ представляет собой вариант выбора медоида по минимаксному критерию. В отличие от классического алгоритма k-medoids, который минимизирует сумму расстояний между центром и участниками альянса, данный критерий минимизирует максимальное расстояние от AFN до любого узла альянса. Это позволяет ограничить максимальный радиус альянса и снизить наибольшую задержку передачи данных в процессе агрегации FL. В отличие от центроида, используемого в алгоритме K-means, минимаксный центр всегда является реально существующим туманным узлом. Это упрощает его развёртывание и обеспечивает практическую возможность использования выбранного узла в качестве AFN. Кроме того, механизм резервирования повышает отказоустойчивость системы: при отказе основного AFN резервный узел берёт на себя его функции.

После того как альянсы сформированы и для каждого из них определены основной и резервный узлы AFN, можно перейти непосредственно к процессу децентрализованной агрегации модели. В отличие от традиционных подходов, где все параметры передаются на центральный сервер, в предлагаемой структуре агрегация выполняется внутри каждого альянса через узел AFN. При этом параметры модели разделяются на два типа, что позволяет сохранить баланс между глобальными знаниями и локальной адаптацией к особенностям каждого кластера.

Параметры модели делятся на два типа:

– *Глобальные параметры:* к ним относятся матрицы весов GCN-GRU W_z, W_r, W и параметры входных/выходных линейных слоёв.

– *Локальные параметры:* это векторы встраивания узлов E_n для измерительных станций в кластерах.

Как отмечалось в предыдущем разделе, в предлагаемой архитектуре каждый TFN F_n передаёт свои параметры модели θ_n узлу AFN внутри альянса. Этот процесс обмена должен использовать защищённый протокол. Протокол сохраняет приватность исходных данных на каждом устройстве.

Протокол обмена параметрами может реализовываться несколькими способами. Во-первых, используется сквозное шифрование (end-to-end encryption) данных при передаче. Во-вторых, применяются цифровые подписи для проверки подлинности отправителя. В-третьих, вводятся механизмы аутентификации для контроля доступа. Все эти меры обеспечивают конфиденциальность и целостность передаваемых параметров. (В рамках данной работы протоколы безопасности не рассматриваются более подробно и остаются предметом будущих исследований.)

Узел AFN $F_{agg,i}$ в альянсе B_i получает параметры от всех туманных узлов. Затем он вычисляет агрегированные параметры по формуле:

$$\theta_{agg,i} = \frac{1}{|B_i|} \sum_{F_n \in B_i} \theta_n \quad (25)$$

Однако агрегация выполняется с учётом разделения параметров:

- Для глобальных параметров:

$$\theta_{agg,i}^{global} = \frac{1}{|B_i|} \sum_{F_n \in B_i} \theta_n^{global} \quad (26)$$

Векторы встраивания узлов обрабатываются иначе. Локальные векторы встраивания остаются на туманных узлах. Обмен ими между туманными узлами в альянсе не выполняется. Каждый узел сохраняет свой уникальный вектор встраивания, отражающий особенности его локального окружения. Такая

стратегия разделения параметров имеет важное преимущество. Она обеспечивает обмен общими знаниями (глобальные параметры) между всеми туманными узлами альянса. Одновременно она сохраняет специфические знания (векторы встраивания узлов) для каждого кластера измерительных станций. Это позволяет модели сбалансировать две противоположные цели:

– *Достичь универсальности модели:* глобальные параметры обобщают закономерности, общие для всех измерительных станций.

– *Сохранить локальную адаптацию модели:* локальные векторы встраивания учитывают географические и климатические особенности каждого кластера. Такое разделение является важным нововведением предлагаемого метода. Оно отличает предложенный подход от стандартного федеративного обучения, где все параметры усредняются без разделения.

AFN $F_{agg,i}$ распределяет агрегированные параметры $\theta_{agg,i}$ между всеми туманными узлами альянса B_i . Для этого применяется одноранговый протокол передачи данных. Узлы AFN передают обновлённые параметры непосредственно каждому TFN внутри коалиции (например, посредством прямой адресной передачи по протоколу TCP/UDP или с использованием механизма gossip, при котором каждый узел после получения обновления пересылает его соседним узлам). Это позволяет исключить роль центрального сервера в архитектуре. Одновременно AFN сохраняет агрегированные параметры $\theta_{agg,i}$ для использования в локальной модели. Каждый туманный узел $F_n \in B_i$ обновляет свою модель агрегированными параметрами: $\theta_n \leftarrow \theta_{agg,i}$. Этот механизм гарантирует, что знания, полученные от других туманных узлов (через глобальные параметры), включаются в локальную модель, при этом она остаётся способной адаптироваться к уникальным особенностям каждой области. В данной работе предлагается облегчённый алгоритм переключения на резервный узел. Этот алгоритм построен на принципах отказоустойчивых распределённых систем. Он аналогичен подходу, описанному в работе [88], но упрощён для среды туманных вычислений малого масштаба. Предлагаемый алгоритм имеет следующие характеристики. Он автоматически

запускается на узлах-участниках туманных вычислений внутри альянса. Алгоритм не требует участия глобального облака при обнаружении сбоев. Это обеспечивает непрерывность работы системы даже при потере соединения с облачным сервером.

Таблица 13— Алгоритм переключения узла AFN

Алгоритм: RUN_FAILOVER_PROTOCOL — Протокол переключения на резервный узел AFN

Input: ALLIANCES — множество альянсов; AFN_MAP — таблица соответствия (PRIMARY, BACKUP) для каждого альянса A; TIMEOUT — предельное время ожидания ответа от PRIMARY.

Output: Обновлённая таблица AFN_MAP с актуальными значениями PRIMARY и BACKUP для каждого альянса A.

1. На каждом узле альянса A (Periodic LOOP):
2. PRIMARY, BACKUP $\leftarrow$ AFN_MAP[A]
3. IF self $\neq$ PRIMARY:
4. Отправить HEARTBEAT $\rightarrow$ PRIMARY; ожидать ACK в течение TIMEOUT
5. IF ACK не получен:
6. Отправить CHECK_PRIMARY $\rightarrow$ BACKUP
7. IF self == PRIMARY:
8. Ответить ACK на все полученные сообщения HEARTBEAT
9. IF self == BACKUP:
10. UPON receiving CHECK_PRIMARY:
11. Отправить HEARTBEAT $\rightarrow$ PRIMARY; ожидать ACK в течение TIMEOUT
12. IF ACK по-прежнему не получен:
13. // Повышение BACKUP до PRIMARY
14. PRIMARY $\leftarrow$ self
15. // Выбор нового BACKUP методом minimax-медоида в A \ {PRIMARY}
16. BACKUP $\leftarrow$ NEXT_BEST_MINIMAX(A, D_{fog}, exclude=PRIMARY)
17. AFN_MAP[A] $\leftarrow$ (PRIMARY, BACKUP)
18. Разослать BROADCAST всем узлам в A для обновления AFN_MAP

Предложенный механизм рассчитан на отказ отдельного узла и предполагает связность сети внутри альянса. Сценарий сетевого разделения, требующий механизма кворума, выходит за рамки данной работы и рассматривается как направление дальнейших исследований.

В предлагаемой распределённой структуре узлы AFN выполняют специальную роль. Они не обучают модели на данных. Вместо этого узлы AFN выступают координаторами альянса. Процесс обучения организован следующим образом: В каждом раунде участники (узлы TFN) выполняют локальные обновления на своих данных. После завершения локального обучения они отправляют обученные параметры модели в узел AFN. Узел AFN агрегирует полученные параметры, используя взвешенное усреднение, аналогично алгоритму FedAvg. Затем AFN рассылает агрегированную модель всем туманным узлам в альянсе.

Такой подход имеет важное преимущество. Поскольку узел AFN выполняет только операцию агрегации, его ресурсы не перегружаются. Вычислительные и сетевые ресурсы координатора остаются в разумных пределах. Это контрастирует с централизованной архитектурой, где облачный сервер обучает глобальную модель на всех данных. Агрегированная модель отражает общие знания всего альянса. Одновременно каждый туманный узел сохраняет свои локальные параметры. К ним относятся, например, регион-специфические векторы встраивания. Такое разделение позволяет модели быть одновременно универсальной и локально адаптивной. В предложенной архитектуре достигается дополнительное улучшение: внутриальянсовые соединения минимизированы. Это снижает задержки по сравнению с централизованной архитектурой. Сигналы передаются на короткие расстояния между узлами одного альянса. Не требуется отправка данных в облако и обратно. Распределённая региональная структура дополнительно повышает отказоустойчивость системы. При отказе одного альянса другие продолжают работать независимо. Система не зависит от одного центрального сервера, отказ которого парализовал бы всю архитектуру.

В целом, формирование коалиций и выбор узлов AFN создают эффективный уровень распределённой координации. Это применяется для реализации децентрализованного федеративного обучения. Архитектура обеспечивает три ключевых преимущества:

– *Преимущество 1: Точность обучения.* Модель сохраняет высокую точность благодаря адекватной агрегации знаний от всех узлов.

– *Преимущество 2: Эффективное использование ресурсов.* Минимальное сетевое взаимодействие и локальная обработка снижают потребление пропускной способности и вычислительных ресурсов.

– *Преимущество 3: Повышенная устойчивость.* Система функционирует без центрального сервера и остаётся жизнеспособной при отказах отдельных компонентов.

3.4 Моделирование и анализ результатов

Для оценки предложенной архитектуры использовались два открытых набора данных: METR-LA и PeMSD7(M). Первый содержит измерения скорости трафика от 207 датчиков. Второй содержит данные от 228 датчиков. Оба набора данных представляют реальные измерения на автомагистралях. Исходные данные предварительно обрабатывались перед началом обучения моделей. Каждый обучающий пример представляет собой последовательность из 12 временных шагов. Этот период соответствует одному часу наблюдений (так как шаг равен 5 минутам). Модель прогнозирует следующие 12 шагов, то есть предсказывает развитие трафика на 60 минут вперёд.

Для оценки точности прогноза использовались три горизонта прогнозирования: горизонт 5 минут (соответствует 1 шагу); горизонт 15 минут (соответствует 3 шагам); горизонт 30 минут (соответствует 6 шагам). Такой выбор горизонтов позволяет оценить способность модели к прогнозированию как на коротких, так и на более длительных интервалах времени.

В качестве базовых решений для сравнения были выбраны две модели. Модель 1 представляет собой централизованный вариант архитектуры AGCRN. Эта модель обучается на одном сервере с использованием полного набора данных без какого-либо распределения. Эта модель служит верхней границей (потенциального) качества, на которое можно рассчитывать при идеальных условиях.

Вторая модель является стандартным федеративным вариантом AGCRN, в котором несколько клиентов координируются через центральный сервер по классической схеме FedAvg. Для оценки точности прогноза используются три стандартные метрики: средняя абсолютная ошибка (MAE, аббр. от англ. Mean Absolute Error), корень из среднеквадратичной ошибки (RMSE, аббр. от англ. Root Mean Square Error), симметричная средняя абсолютная процентная ошибка (sMAPE, аббр. от англ. Symmetric Mean Absolute Percentage Error).

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (27)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (28)$$

$$\text{sMAPE} = \frac{100\%}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \quad (29)$$

где, y_i - фактическое значение, $\hat{y}_i$ - прогнозное значение, n - число прогнозируемых примеров.

Результаты анализируются и оцениваются для трёх ключевых горизонтов: 5, 15 и 30 минут, которые выбраны как контрольные точки в рамках общего 60-минутного окна прогноза (12 шагов по 5 минут) и соответствуют краткосрочному, среднесрочному и более длительному прогнозу соответственно. Анализ по нескольким горизонтам даёт целостное представление о возможностях модели на различных временных масштабах. Для оценки скорости сходимости отслеживается функция потерь после каждого раунда в течение 100 раундов.

Высокая скорость сходимости показывает эффективность обучения модели и существенно влияет на совокупную эффективность при практическом развёртывании системы. Используются следующие показатели для анализа сходимости:

- Скорость уменьшения потерь: $CR = \frac{L_1 - L_T}{L_1}$. где L_1 - значение функции потерь в первом раунде, L_T - потери в последнем раунде.
- Число раундов до достижения порога: $R_{threshold} = \min\{r \mid L_r \leq L_{threshold}\}$, где $L_{threshold}$ - заданный порог потерь.
- Средний наклон сходимости: $S_{avg} = \frac{1}{T-1} \sum_{t=1}^{T-1} \frac{L_t - L_{t+1}}{L_t}$, который измеряет среднюю относительную скорость уменьшения потерь по раундам.

В рамках предложенной структуры общие сетевые ресурсы складываются из трёх составляющих: ресурсов для сбора данных, ресурсов децентрализованного федеративного обучения и внутриальянсовых служебных ресурсов. Ресурсы для сбора данных определяются объёмом информации, передаваемой от измерительных станций (датчиков) к обучаемым туманным узлам внутри каждого кластера. Ресурсы DFL связаны с обменом параметрами модели между обучаемыми туманными узлами и агрегационным туманным узлом в процессе федеративного обучения. Внутриальянсовые служебные ресурсы обеспечивают координацию работы альянса, включая механизм контроля доступности узлов, переключение на резервный AFN и рассылку обновлённых параметров участникам альянса. Следует отметить, что ресурсы DFL и внутриальянсовые служебные ресурсы не пересекаются: первые отвечают исключительно за обмен параметрами модели в рамках федеративного обучения, а вторые за поддержание работоспособности и координацию узлов.

Формула расчета ресурсов для FL-AGCRN:

$$C_{FL-AGCRN} = C_{sensor} + C_{params} \quad (30)$$

где:

$$C_{sensor} = \sum_{i=1}^n (s_{data} \cdot r \cdot d_{i,client}) \quad (31)$$

$$C_{params} = (s_{model} \cdot n_{clients} \cdot d_{client,server} + s_{updates} \cdot n_{clients} \cdot d_{client,server}) \cdot R \quad (32)$$

DFL-FC-AGCRN:

$$C_{DFL-FC-AGCRN} = C_{sensor} + C_{alliance} \quad (33)$$

где:

$$C_{sensor} = \sum_{i=1}^n (s_{data} \cdot r \cdot d_{i,fog}) \quad (34)$$

$$C_{alliance} = \sum_{j=1}^A (s_{updates} \cdot n_j \cdot \bar{d}_j + s_{model} \cdot n_j \cdot \bar{d}_j) \cdot R \quad (35)$$

Таблица 14 — Обозначения параметров моделей FL и Fog-FL

Обозначение	Описание	Обозначение	Описание
s_{data}	размер одной записи датчика	A	число альянсов
r	число записей	n_j	число узлов в альянсе j
$d_{i,client}$	расстояние от датчика i до клиента	$d_{i,fog}$	расстояние от датчика i до туманного узла
s_{model}	размер модели	$\bar{d}_j$	среднее расстояние внутри альянса j
$s_{updates}$	размер обновлений параметров	$d_{client,server}$	расстояние от клиента до сервера
$n_{clients}$	число клиентов	R	число раундов обучения

Построение топологии сети

В среде туманных вычислений качество прогноза зависит не только от алгоритма обучения, но и от топологии сети. Поэтому задача построения эффективной топологии сети состоит из двух шагов:

1. Формирование возможных топологий «датчики → туманные узлы» с помощью алгоритма HDBSCAN, основываясь на географических координатах датчиков.

2. Анализ влияния числа альянсов туманных узлов на точность прогноза и ресурсозатратность модели федеративного обучения для задачи прогнозирования трафика.

Это позволяет сделать выбор эффективной конфигурации DFL-сети для задачи прогнозирования трафика.

Кластеризация датчиков

Цель первого этапа — подобрать гиперпараметры HDBSCAN так, чтобы преобразовать географически распределённые датчики в слой туманных узлов с компактной топологией. Для этого выполняется перебор по сетке значений параметров HDBSCAN, где входом служит заранее вычисленная матрица геодезических расстояний между датчиками D_{sensor} .

Таблица 15 — Параметры HDBSCAN

Имя	Обоз н.	Определение	Диапазон	Примечание
Минимальный размер кластера	mcs	Минимум точек в кластере	[2, 20]	Влияет на число кластеров и долю шума
Коэффициент ядра	α	Фактор для min_samples	[0.5, 0.7]	Настраивает «жесткость» плотности

Имя	Обоз н.	Определение	Диапазон	Примечание
Минимум core-точек	mins	$\text{mins} = \lfloor \alpha \cdot \text{mcs} \rfloor$	—	Чем больше mins, тем плотнее кластеры

Для практического развертывания важен допустимый радиус связи датчиков. В качестве базовой технологии беспроводного доступа рассматривается Wi-Fi HaLow (IEEE 802.11ah), который поддерживает подключение до 8192 устройств к одной точке доступа и обеспечивает теоретическое покрытие на расстоянии до ≈ 1 км. Согласно модели, предложенной Domazetović и соавт [89], достижимый радиус связи может составлять около 1,5 км при использовании схемы модуляции и кодирования MCS10. На основе данного результата в настоящей работе принимается радиус связи датчика, равный 1,5 км. На основе этого порога отбрасываем конфигурации, где макро-радиус превышает предел, и выбираем три представительных варианта. Результаты в таблице 16.

Таблица 16 — Результаты кластеризации датчиков

mcs	mins	alpha	Коэффициент силуэта	Макс. радиус кластера, км	Доля нераспределённых датчиков, %	К
2	1	0.5	0.70	0.31	3.86	79
3	2	0.7	0.52	0.93	10.94	33
5	2	0.5	0.53	1.47	15.77	18

Чтобы ни один датчик не был отброшен, дополнительно рассматриваем вариант переназначения шума в ближайший кластер по медоиду. В таблице показаны показатели до переназначения, чтобы честно отразить качество HDBSCAN. Из таблицы видно, что большое число туманных узлов приводит к высоким затратам ресурсов на развертывание систем и межальянсовые коммуникации. Для минимизации стоимости при сохранении простоты выбираем конфигурацию с 18 туманными узлами.

Оптимизация формирования альянсов

На втором этапе, опираясь на топологию из 18 туманных узлов, подбираем число альянсов. Требуется: учитывать естественную географическую структуру туманных узлов; строго соблюдать ограничение на радиус связи внутри альянса (от AFN до участников).

Радиус взаимодействия внутри альянса задаётся равным 10 км, что соответствует концепции сетей 6G, в рамках которой вычислительные функции и ресурсы распределяются по иерархии от облачного уровня к уровню туманных и периферийных вычислений. При таком подходе альянс туманных узлов, охватывающий территорию протяжённостью в десятки километров, рассматривается как единый регион периферийных вычислений с согласованным управлением и координацией ресурсов.

Рассматриваются значения $k \in \{1, 2, 3, 4, 5, 6\}$, охватывающие диапазон от полностью централизованной агрегации до конфигурации с высокой степенью децентрализации. Следует подчеркнуть, что данный этап относится к инфраструктурному проектированию сети, а не является основным предметом исследования: его цель — зафиксировать топологию развёртывания до начала основных экспериментов.

Для обеспечения статистической надёжности каждая конфигурация запускается 3 раза с различными инициализациями весов модели (seeds: 42, 123, 777) при 10 раундах и 3 локальных эпохах на раунд - итого 18 независимых прогонов. Результаты представлены как среднее $\pm$ стандартное отклонение (выборочное, $n=3$). Топология альянсов фиксируется через *K-means* по географическим координатам туманных узлов ($\text{seed} = 42$) и не изменяется между прогонами.

Таблица 17– Оценка при разном числе альянсов

k	MAE	RMSE	sMAPE	Loss	Сетевые затраты (МБ)
1	10,428 $\pm$ 0,549	18,192 $\pm$ 0,599	0,158 $\pm$ 0,003	0,330 $\pm$ 0,001	343,28

k	MAE	RMSE	sMAPE	Loss	Сетевые затраты (МБ)
2	$9,814 \pm 0,157$	$17,342 \pm 0,089$	$0,153 \pm 0,002$	$0,313 \pm 0,000$	221,40
3	$9,869 \pm 0,245$	$17,477 \pm 0,157$	$0,151 \pm 0,002$	$0,292 \pm 0,001$	178,61
4	$9,958 \pm 0,309$	$17,693 \pm 0,263$	$0,154 \pm 0,003$	$0,262 \pm 0,000$	135,20
5	$9,969 \pm 0,187$	$17,713 \pm 0,161$	$0,152 \pm 0,003$	$0,262 \pm 0,000$	132,67
6	$10,170 \pm 0,391$	$17,997 \pm 0,351$	$0,154 \pm 0,009$	$0,238 \pm 0,000$	118,59

Результаты 18 независимых прогонов выявляют устойчивую U-образную зависимость метрик точности от числа альянсов k : при малых значениях ($k=1$) высокая неоднородность данных внутри единственного альянса ухудшает агрегацию; при больших ($k \geq 4$) чрезмерная фрагментация вызывает локальное переобучение - функция потерь монотонно снижается ($0,330 \rightarrow 0,238$), тогда как MAE и RMSE на тестовой выборке деградируют. Оптимум достигается при $k \in \{2, 3\}$, где масштаб альянса соответствует естественному транспортному коридору. Стандартное отклонение MAE дополнительно подтверждает нестабильность крайних конфигураций ($k=1$: $\sigma=0,549$; $k=6$: $\sigma=0,391$) по сравнению со средними ($k=3$: $\sigma=0,245$). С учётом точности, устойчивости и коммуникационных затрат конфигурация $k=3$ обеспечивает наилучший баланс и принимается как оптимальная.

Моделирование и сравнительный анализ моделей

При радиусе связи $R = 10$ км система содержит 18 туманных узлов. Эти узлы обслуживают 207 датчиков. Датчики разбиваются на 3 независимых тренировочных альянса. Развёртывание системы на датасете METR-LA показано на рисунке 23.



Рисунок 23— Развёртывание системы на датасете METR-LA

Результаты прогноза интенсивности трафика для трёх моделей суммированы на рисунке 24. Для горизонтов 5, 15 и 30 минут после 100 раундов обучения приводятся MAE, RMSE и sMAPE.

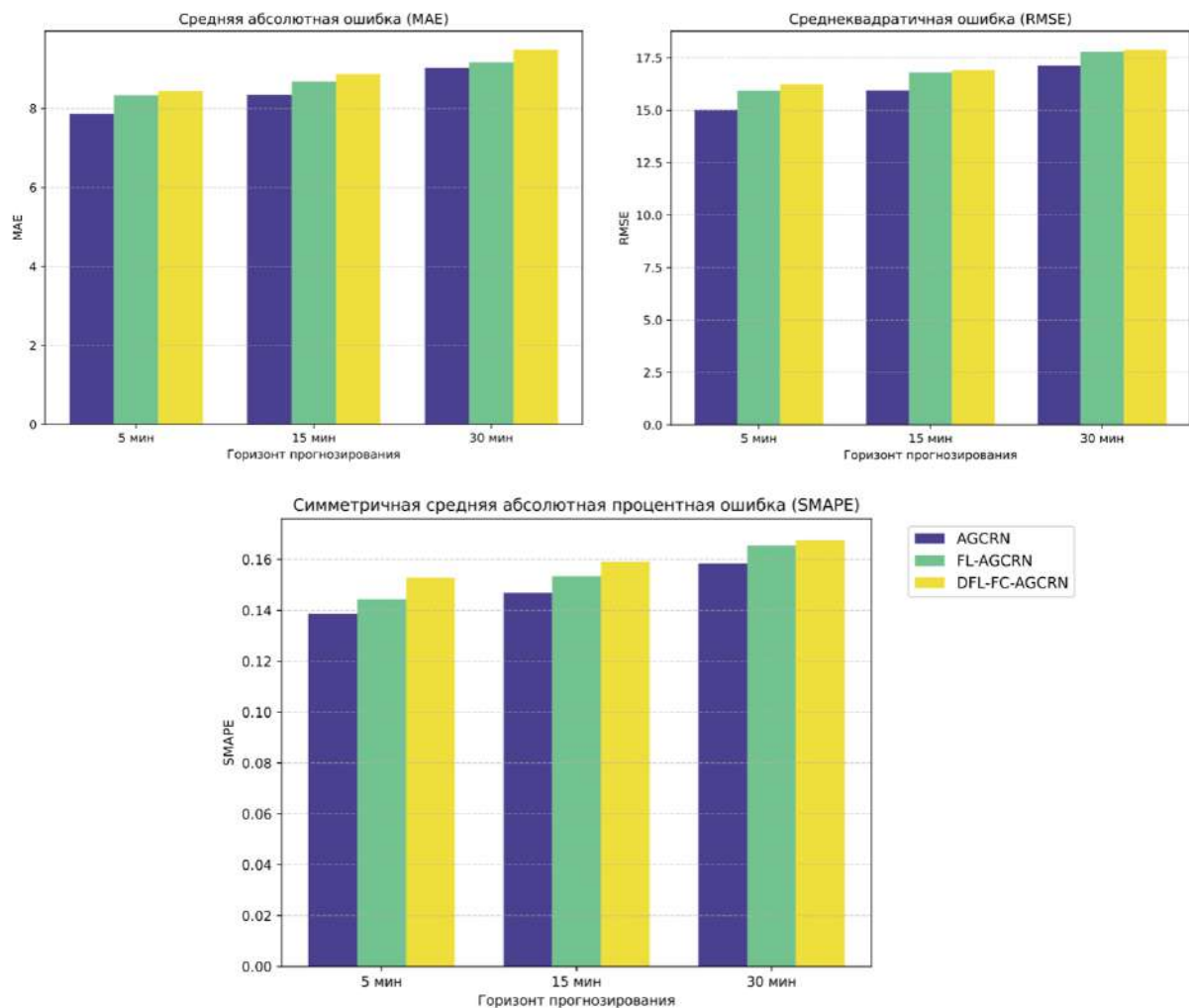


Рисунок 24 — Точность моделей на датасете METR-LA

Централизованная модель демонстрирует лучшую точность. Она достигает наименьших значений MAE среди всех сравниваемых подходов. Это ожидаемо по следующей причине: обучение проводится на полном наборе данных без каких-либо ограничений на распределение данных модели ИИ. Распределённые модели (FL-AGCRN и DFL-FC-AGCRN) показывают несколько более слабые результаты. Они не достигают показателей централизованного базового варианта. Однако точность остаётся конкурентоспособной. Это важный результат, так как позволяет судить об эффективности распределённого подхода при отсутствии доступа к полному набору данных.

У всех трёх моделей качество прогноза ухудшается при увеличении горизонта прогнозирования, что соответствует общему поведению задач прогнозирования временных рядов. Скорость деградации для DFL-FC-AGCRN сопоставима со скоростью для FL-AGCRN. Это показывает, что характеристики прогноза на больших горизонтах хорошо сохраняются в иерархической децентрализованной структуре.

В рамках данного исследования были выбраны 4 случайных датчика внутри сформированных кластеров, где были полученные примеры прогнозов. Результаты прогноза модели DFL-FC-AGCRN на реальном датасете для горизонтов 5, 15 и 30 минут показаны на рисунке 25.

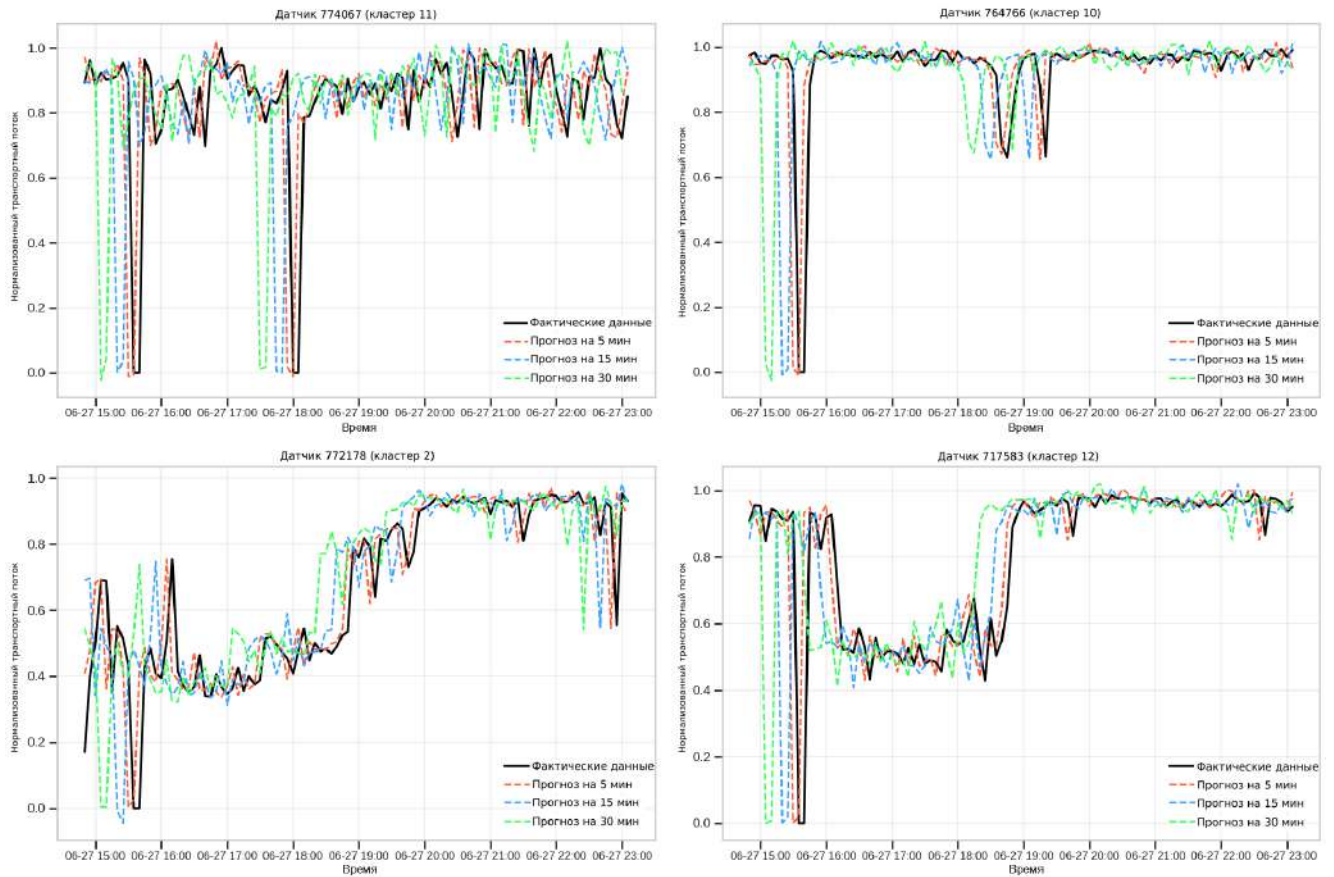


Рисунок 25 — Результаты прогноза модели DFL-FC-AGCRN для четырёх случайно выбранных датчиков в кластерах

Чтобы дополнительно продемонстрировать эффективность предлагаемой структуры и методы, была оценена скорость сходимости модели, а также необходимые затрачиваемые сетевые ресурсы. Результаты для 10, 20, 50 и 100 раундов обучения суммированы в таблице 18.

Таблица 18 — Показатели скорости сходимости

Метрика	FL-AGCRN	DFL-FC-AGCRN	Преимущество DFL-FC-AGCRN
Коэффициент снижения потерь после 10 раундов	5.79%	4.76%	Улучшение 17.8%
Коэффициент снижения потерь после 20 раундов	7.16%	6.08%	Улучшение 15.1%
Коэффициент снижения потерь после 50 раундов	8.51%	7.45%	Улучшение 12.5%

Метрика	FL-AGCRN	DFL-FC-AGCRN	Преимущество DFL-FC-AGCRN
Коэффициент снижения потерь после 100 раундов	9.26%	8.28%	Улучшение 10.6%
Раунды для достижения потерь 0.325 (FL) / 0.290	18 раундов	13 раундов	Быстрее 27.8%
Раунды для достижения потерь 0.320 (FL) / 0.285	43 раунда	33 раунда	Быстрее 23.3%

На основании результатов предложенная модель не только быстрее сходится, но и поддерживает эффективное обучение на протяжении всего процесса. Эффективность развёртывания модели подтверждается показателем эффективности использования ресурсов, который вырос с 48% до 60%. Это подтверждает практическую эффективность модели.

Таблица 19 — Влияние скорости сходимости на затраты на связь

Метрика	FL-AGCRN	DFL-FC-AGCRN	Экономия
Стоимость связи за раунд	34.33 МБ	17.71 МБ	48.4%
Затраты до достижения Loss 0,325 (FL) / 0,290 (DFL)	617,94 МБ (18 раундов)	232,18 МБ (13 раундов)	62.4%
Затраты до достижения Loss 0,320 (FL) / 0,285 (DFL)	1476,19 МБ (43 раунда)	589,38 МБ (33 раунда)	60.1%
Суммарные затраты за 100 раундов	3433.00 МБ	1771.43 МБ	48.4%

Таблица 20 — Затраты на связь на этапах реализации проекта

Раунд	FL-AGCRN	DFL-FC-AGCRN	Экономия (МБ)	Экономия
10 раундов	343.30	183.67	159.63	46.5%
20 раундов	686.60	358.41	328.19	47.8%
50 раундов	1716.50	870.27	846.23	49.3%

Раунд	FL-AGCRN	DFL-FC-AGCRN	Экономия (МБ)	Экономия
100 раундов	3433.00	1771.43	1661.57	48.4%

Оценка на расширенном датасете

Для проверки применимости предложенной модели были проведены дополнительные эксперименты на наборе данных PeMSD7(M). Это реальный набор данных из системы измерения Caltrans (Performance Measurement System, Округ 7). Набор данных содержит измерения с 228 датчиков (станций). Датчики установлены на автомагистралях. Они регистрируют скорость транспортных средств, описывая соответствующие городские транспортные потоки. Данные собираются с временным шагом 5 минут. Период сбора данных определяется задачей исследования. Обычно это период в один месяц или один квартал. Такой период позволяет захватить сезонные закономерности в движении транспорта. Преимущество этого датасета существенно. По сравнению с другими наборами (например, METR-LA или PeMSD8) он содержит значительно больше датчиков. Это позволяет оценить масштабируемость и эффективность модели. Модель должна работать с более распределёнными и сложными данными.

Для выбора конфигурации системы проведено отдельное исследование. Рассматривались параметры кластеризации датчиков и число альянсов туманных узлов. В настоящей работе под оптимальной конфигурацией понимается конфигурация, которая удовлетворяет практическим ограничениям инфраструктуры и обеспечивает наилучший баланс между точностью прогнозирования и сетевыми затратами. При выборе конфигурации учитывались следующие ограничения: радиус кластера датчиков не должен превышать 1,5 км, а радиус взаимодействия между туманными узлами внутри альянса не должен превышать 10 км. Эти ограничения связаны с задержкой распространения сигнала и стоимостью передачи данных. Сравнение конфигураций проводилось по трём критериям:

- Критерий 1, возможность практического развёртывания с учётом указанных ограничений на радиус связи;

- Критерий 2, минимизация необходимых сетевых взаимодействий между узлами;
- Критерий 3, поддержание высокой точности прогноза модели.

По результатам многокритериального сравнения была выбрана следующая конфигурация системы. 228 датчиков организованы в 38 узлов туманных вычислений. Эти 38 узлов далее разбиваются на 3 альянса (группы). Визуализация этой структуры представлена на рисунке 26.

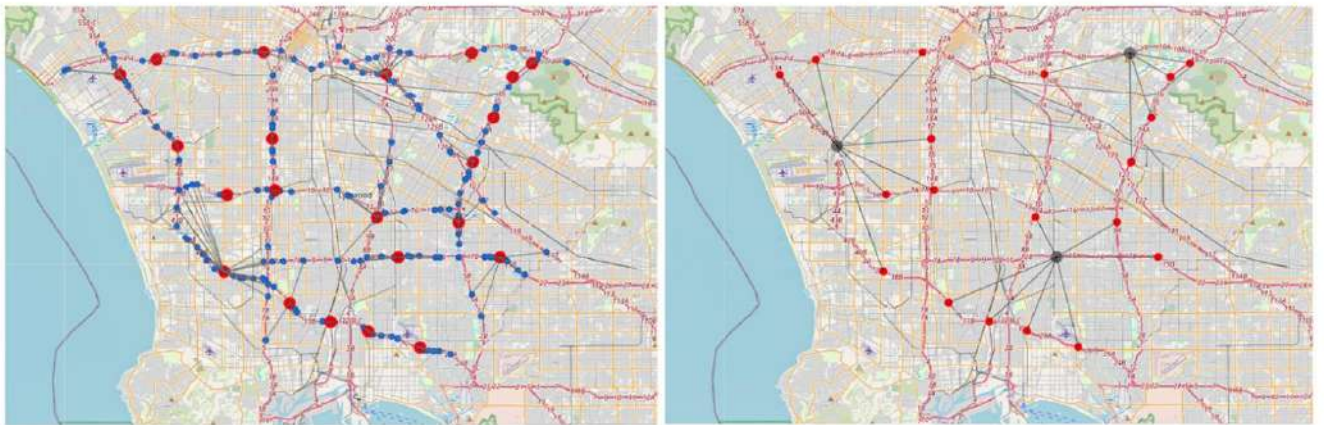


Рисунок 26 — Применение системы к набору данных PeMSD7(M)

Для 50 раундов обучения с 3 локальными эпохами предлагаемая структура сохраняет эффективность, демонстрируя достаточную точность и при этом уменьшая необходимые сетевые взаимодействия благодаря иерархической структуре и предложенному механизму альянсов. Результаты показаны на рисунках 27 и 28. Средняя экономия сетевого взаимодействия составляет около 48% после 50 раундов.

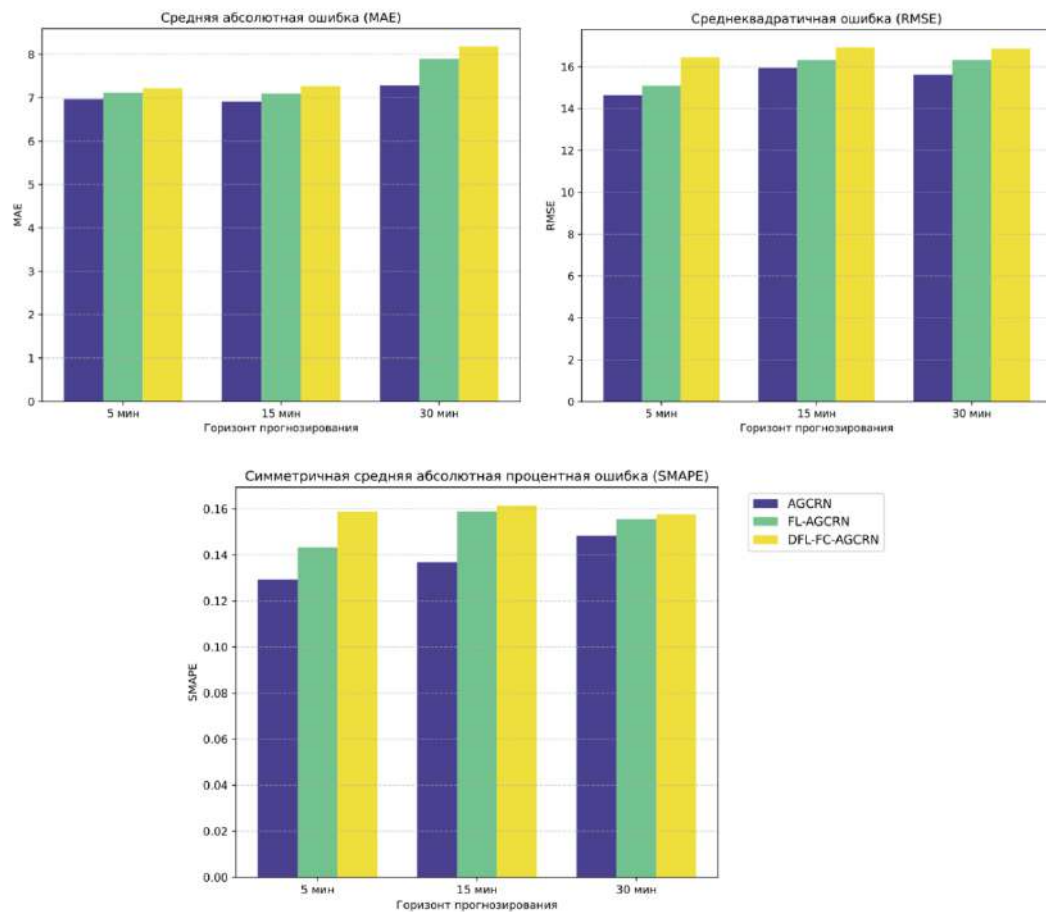


Рисунок 27 — Точность моделей на датасете PeMSD7(M)

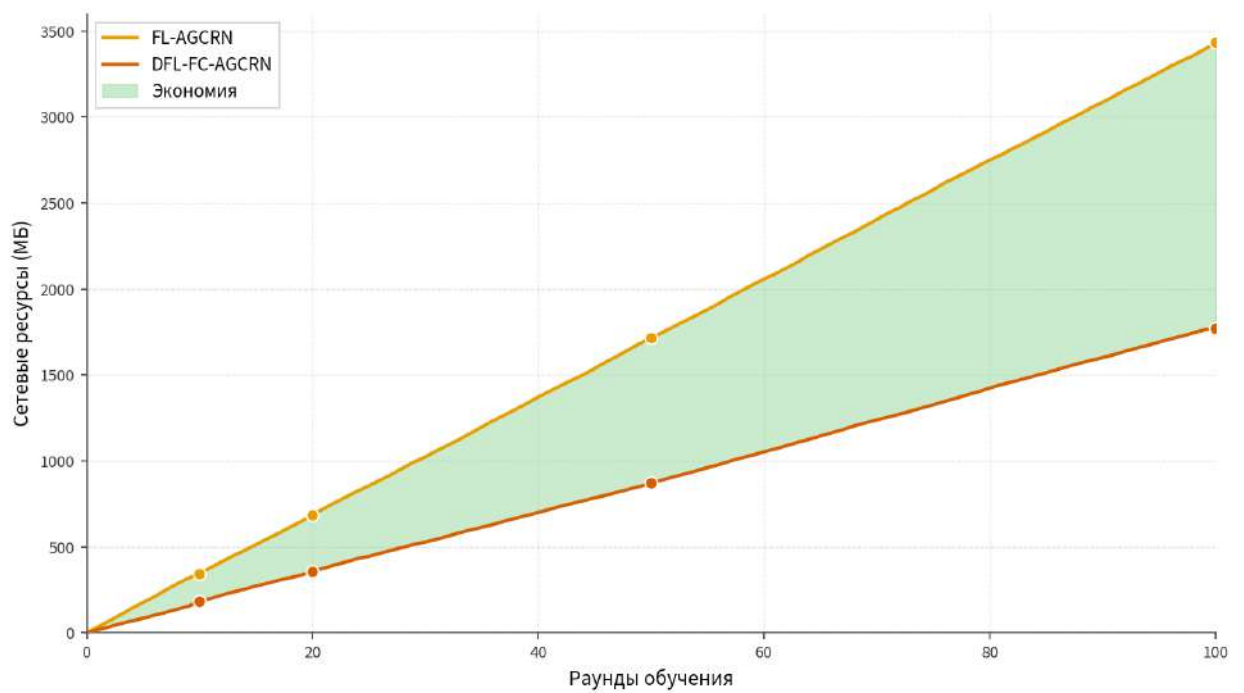


Рисунок 28 — Затраченные сетевые ресурсы по раундам обучения

3.5 Выводы по Главе 3

1. В данной главе разработан метод выборки пользовательских устройств для участия в раундах федеративного обучения в интегрированной среде FC с использованием модели AGCRN. Совместное применение трёх указанных технологий формирует подход, который обеспечивает рациональный баланс между точностью прогноза распределённой модели искусственного интеллекта и эффективностью использования сетевых ресурсов.

2. В данной главе также установлено, что децентрализованный характер предложенной структуры обеспечивает устойчивость системы. Показано, что полное исключение центрального сервера агрегации из операционного цикла обучения модели ИИ решает проблему единой точки отказа в сценариях потери связи, стихийных бедствий и чрезвычайных ситуаций, при которых подключение к центральному серверу является нецелесообразным или невозможным. Наряду с этим предложенная структура использует региональные свойства данных, что в свою очередь ускоряет сходимость модели ИИ и способствует снижению потребления сетевых ресурсов.

3. В главе проведена экспериментальная проверка на открытых реальных данных системы измерения производительности Caltrans: наборах METR-LA (207 датчиков) и PeMSD7(M) (228 датчиков). Подтверждены работоспособность и эффективность предложенного метода. Незначительное снижение точности модели при существенной экономии сетевых ресурсов и повышении отказоустойчивости признано приемлемым компромиссом для практических приложений, обрабатывающих большие объёмы данных.

4. Таким образом, в главе разработаны модель DFL-FC-AGCRN и метод формирования иерархической структуры, объединяющий локальное обучение на TFN, агрегацию на AFN, кластеризацию датчиков с помощью HDBSCAN и объединение туманных узлов в альянсы. Экспериментальная оценка на реальных наборах данных METR-LA и PeMSD7(M) подтвердила работоспособность предложенного подхода. По сравнению с базовой конфигурацией FL-AGCRN

предложенная структура обеспечивает сокращение объёма сетевого взаимодействия, ускорение сходимости и сохранение сопоставимой точности прогнозирования. В целом предложенный метод обеспечивает рациональный баланс между качеством модели, сетевыми затратами и отказоустойчивостью системы.

ГЛАВА 4. СТРУКТУРА И МЕТОД РАЗВЁРТЫВАНИЯ ФЕДЕРАТИВНОГО МАШИННОГО ОБУЧЕНИЯ НА ОСНОВЕ WEBASSEMBLY В ГЕТЕРОГЕННОЙ СРЕДЕ ТУМАННЫХ ВЫЧИСЛЕНИЙ

4.1 Введение

Стремительное развитие сетей IMT-2030 и массовое распространение устройств IoT формируют принципиально новые требования к вычислительной инфраструктуре. Обработка данных всё в большей степени смещается от централизованных облачных платформ к распределённым узлам на границе сети, где данные генерируются и должны обрабатываться локально. Именно в этом контексте Туманные вычисления (FC) приобретают роль ключевого промежуточного уровня, приближающего вычисления к источникам данных и обеспечивающего требования по задержке, которые недостижимы при исключительно облачной обработке.

Федеративное обучение (FL) органично вписывается в данную парадигму. FC позволяет множеству распределённых узлов совместно обучать глобальную модель ИИ без передачи исходных данных по сети, что одновременно решает задачи конфиденциальности, снижает нагрузку на каналы связи и распределяет вычислительную нагрузку по уровням инфраструктуры. Сочетание FC и FL образует, таким образом, естественную основу для построения распределённых интеллектуальных систем в сетях следующего поколения.

Вместе с тем между концептуальной привлекательностью данного сочетания и его практической реализацией сохраняется значительный разрыв. Инфраструктура FC в реальных сетях 6G представляет собой физически неоднородную совокупность устройств, где узлы тумана функционируют на различных аппаратных архитектурах, включая ARM64, ARM, x86 и RISC-V, под управлением операционных систем Linux, Android и RTOS, при существенно различающихся вычислительных ресурсах [90, 91]. Код, скомпилированный под одну аппаратную архитектуру, не может быть исполнен на другой без перекомпиляции, а это означает, что масштабное развёртывание единого

программного стека FL на гетерогенном парке туманных узлов превращается в самостоятельную и нетривиальную инженерную задачу.

Для устранения данного пробела в настоящей главе предлагается структура и метод кроссплатформенного развёртывания FL на основе WebAssembly, представляющего собой низкоуровневый байт-код, и среды выполнения Wasmtime. Модуль локального обучения реализуется на языке Rust, однократно компилируется в формат `wasm32-wasi` и исполняется на любом туманном узле через среду Wasmtime без необходимости перекомпиляции под конкретную аппаратную архитектуру. Модель безопасности WASI обеспечивает изолированное выполнение модуля с минимально необходимыми привилегиями доступа к ресурсам хоста (устройства), формируя тем самым дополнительный уровень защиты туманных узлов. Предложенная структура реализует на практике принцип «скомпилировано однажды, выполняется везде» применительно к задаче локального обучения FL на гетерогенной инфраструктуре FC.

Структура настоящей главы построена следующим образом. В разделе 4.2 проводится анализ существующих подходов к управлению гетерогенностью при развёртывании FL и обосновывается исследовательский пробел, на устранение которого направлена данная работа. В разделе 4.3 излагается системная модель и теоретические основы предлагаемой структуры. В разделе 4.4 описывается структура на основе WebAssembly, включая операционный цикл и алгоритмы. В разделе 4.5 представлены результаты экспериментальной оценки на модельной сети Лаборатории, в которую входят разнородные устройства Интернета вещей.

4.2 Анализ существующих подходов к развёртыванию FL на гетерогенных узлах

Развёртывание FL в распределённых средах, таких как FC, сталкивается с рядом присущих проблем системного уровня, которые хорошо отражены в обзоре научной литературы.

– *Гетерогенность систем и ресурсов.* Это один из наиболее существенных барьеров. Туманные и граничные узлы существенно различаются по аппаратному

обеспечению (ЦП, память), программному обеспечению и сетевым возможностям. Такая гетерогенность усложняет развертывание моделей искусственного интеллекта и приводит к проблемам производительности, например, к появлению «отстающих узлов», которые снижают скорость сходимости всей системы FL. Более того, эти устройства часто работают в условиях ограничений ресурсов, требуя высокой эффективности алгоритмов и моделей с точки зрения вычислений и памяти [92], [93].

– *Безопасность и конфиденциальность.* Хотя FL разработано для защиты конфиденциальности путем сохранения данных локально, другие угрозы сохраняются. Обновления модели могут быть атакованы с целью логического вывода конфиденциальной обучающей информации (атаки логического вывода) или злонамеренно изменены для повреждения глобальной модели (атаки отравления). Следовательно, безопасная среда выполнения на каждом узле имеет решающее значение.

Для решения проблемы гетерогенности было принято несколько подходов, каждый из которых имеет свои компромиссы в контексте туманных вычислений.

– *Кросс-компиляция и традиционное (естественное) развертывание.* Этот традиционный подход (с англ. Native deploy) предполагает компиляцию исходного кода отдельно для каждой целевой платформы. Хотя он может обеспечить оптимальную производительность, ему не хватает переносимости и масштабируемости. Поддержание отдельных инструментальных цепочек и конвейеров сборки для каждой пары ISA (архитектура набора команд, англ. Instruction Set Architecture) и ОС (аббр. операционная система) является дорогостоящим и подверженным ошибкам процессом при управлении разнообразным парком устройств [94].

– *Виртуализация и контейнеризация.* Контейнеризация обеспечивает важное преимущество в разработке программного обеспечения. Технологии, подобные Docker, создают унифицированную среду выполнения. Приложения упаковываются с их зависимостями. Это эффективно разрешает классическую проблему несовместимости. Однако такой подход требует значительных ресурсных

затрат. Контейнеры демонстрируют существенно более медленное время инициализации. Они требуют больших объёмов оперативной памяти. Дополнительно возрастает потребление дискового пространства. Эти ограничения делают контейнеры неприемлемыми для устройств с ограниченными ресурсами [95].

Особенно это актуально для двух категорий систем. Во-первых, для устройств интернета вещей (IoT). Во-вторых, для граничных вычислительных узлов. Существует дополнительная проблема архитектурной зависимости. Контейнеры остаются тесно связаны с аппаратной архитектурой. Для каждой архитектуры требуется отдельный образ контейнера. Это усложняет процесс развёртывания в гетерогенной инфраструктуре.

– *Существующие структуры FL.* В области федеративного обучения (FL) сложился ряд популярных фреймворков. TensorFlow Federated, PySyft и Flower представляют собой мощные инструменты. Они предоставляют комплексную поддержку разработки алгоритмов федеративного обучения. Однако эти структуры имеют существенное ограничение в своей ориентации. Они в первую очередь сосредоточены на алгоритмическом уровне, в то время как проблемы системного уровня получают ограниченную поддержку. Это касается, в частности, вопросов развёртывания и управления инфраструктурой. Фреймворки основаны на определённых предположениях относительно подготовка аппаратного, программного и сетевого окружения. Они предполагают, что среды выполнения предварительно подготовлены на клиентских устройствах. На практике это часто не соответствует реальности. Поддержка менее распространённых аппаратных архитектур остаётся проблемной. Архитектура ARM, широко используемая в мобильных и встроенных системах, получает ограниченную поддержку. Для разработчиков фреймворков она часто не является приоритетной. Это создаёт дополнительный барьер при развёртывании в гетерогенной среде.

Таблица 21 — Сравнительный анализ подходов к развертыванию моделей в
федеративном обучении

Критерий	Традиционный /естественный (с англ., Native)	Контейнеризация (Docker)	Веб-ориентированный Wasm (FLAT)	Инфраструктурно-ориентированный Wasm-FL (предлагаемый)
Переносимость	Очень низкая. Требует перекомпиляции и для каждой пары {ISA, ОС}.	Средняя. Требуются отдельные образы для каждой ISA.	Очень высокая. Выполняется в любом современном браузере.	Очень высокая. Один файл .wasm работает на любой платформе с установленной средой выполнения Wasm.
Безопасность и изоляция	Низкая. Прямое выполнение на хостовой ОС.	Высокая. Изоляция в пользовательско м пространстве, но общее программное ядро.	Очень высокая. Песочница браузера.	Очень высокая. Песочница Wasm + безопасность на основе возможностей WASI.
Время запуска	Очень быстрое	Медленное	Быстрое	Очень быстрое
Потребление памяти	Очень малое	Большое	Среднее (зависит от браузера)	Очень малое
Производитель ность	Оптимальная (эталон)	Ниже традиционной из-за накладных расходов контейнера	Ниже нативной из-за браузерной среды	Близкий к традиционному методу
Целевая среда	Серверы, встраиваемые устройства	Серверы, облако	Веб-браузеры на устройствах конечных пользователей	Безинтерфейсные узлы туманных/краевых вычислений

Wasm становится ключевой технологией, способной решить вышеуказанные проблемы. Изначально разработанный для браузера, Wasm быстро получил признание как универсальная, переносимый и безопасный метод компиляции для сред вне браузера [96]. Практическая применимость WebAssembly в среде FC в значительной степени обеспечивается двумя взаимодополняющими элементами экосистемы:

– *Автономные среды выполнения.* Созревание автономных сред выполнения, таких как Wasmtime (от Bytecode Alliance) и WasmEdge (проект CNCF), имеет ключевое значение. Эти среды выполнения оптимизированы для серверных и граничных приложений, обеспечивают высокую производительность посредством JIT/AOT (just in time/ahead of time) компиляции и поддерживают множество платформ, включая архитектуры x86 и ARM. [97].

– *Системный интерфейс WebAssembly (WASI).* WASI является ключевым компонентом, который делает Wasm полезным за пределами браузера. Он определяет стандартизированный, платформо-независимый системный интерфейс, который позволяет модулям Wasm безопасно и контролируемо взаимодействовать с ресурсами ОС хоста (например, файловыми системами, сетью, часами). В данном контексте под хостом понимается физическое устройство с установленной средой выполнения Wasmtime. Примечательно, что WASI применяет безопасность на основе возможностей [98]. По умолчанию модуль Wasm не имеет разрешений; хост должен явно предоставить «возможности» (например, доступ к определенному каталогу). Это обеспечивает соблюдение принципа наименьших привилегий естественным и надежным образом.

Основываясь на приведенном выше анализе, данная глава призвана заполнить данный критический исследовательский пробел. Ее отличительные черты и новизна лучше всего понимаются в сравнении с недавними исследованиями и эволюцией экосистемы Wasm. В применении WebAssembly к машинному обучению наблюдается чёткое разделение на два направления: веб-ориентированный подход и инфраструктурно-ориентированный подход. Веб-

ориентированное направление представлено недавним фреймворком FLAT [99], который позволяет любому устройству с браузером присоединяться к кластеру FL без установки и конфигурации. FLAT решает задачу доступности для конечных пользователей и простоты участия, где браузер одновременно выступает средой выполнения и песочницей, полностью скрывая сложность ОС и архитектуры набора инструкций.

Целью данной главы является решение проблемы развертывания единого кода на безинтерфейсных узлах FC (например, Raspberry Pi, телефоны или иных устройствах, включая разнородные устройства Интернета вещей), не имеющих пользовательского интерфейса. Основное внимание уделяется операционной и инфраструктурной сложности для системных администраторов. В предлагаемой структуре, среда выполнения Wasm является явным слоем абстракции, который требуется явно устанавливать и обслуживать на хостовой ОС. В связи с этим исследование сосредоточено на системной производительности по сравнению с нативным кодом.

Кроме того, современная экосистема Wasm в значительной степени ориентирована на ускорение этапа вывода за счёт спецификации WASI-NN [100], тогда как поддержка обучения остаётся ограниченной. Это формирует важный и актуальный исследовательский пробел. Настоящая работа вносит вклад в его устранение, предоставляя базовые показатели производительности и кроссплатформенную операционную модель, а также отвечая на вопрос о практической осуществимости обучения FL с использованием Wasm в средах FC с точки зрения производительности.

4.3 Структура и теоретические основы

Предлагается трёхуровневая физическая структура федеративного машинного обучения в архитектуре «облако — туман — пользовательское устройство», включающая сенсорный слой, слой туманных вычислений и облачный слой, как показано на рисунке 29, где каждый уровень выполняет строго

определённые функции в соответствии с его вычислительными возможностями и пространственно-временными характеристиками обрабатываемых задач.

– *Уровень сенсоров:* Данный уровень состоит из устройств сбора данных (например, камер и датчиков). Предполагается, что эти сенсоры локально подключены к туманным узлам в пределах радиуса действия связи и непрерывно передают сырые данные на соответствующий туманный узел.

– *Туманный уровень:* Этот уровень соответствует клиентам в алгоритме FedAvg и включает набор туманных узлов $\mathcal{F} = \{1, \dots, F\}$. Каждый туманный узел $f \in \mathcal{F}$ (Raspberry Pi 4, NVIDIA Jetson и др.) отвечает за сбор и локальное хранение полного набора данных B_f от сенсоров в своей доменной зоне. Набор данных считается приватным и никогда не покидает узел f . На каждом туманном узле запущена среда выполнения Wasm для исполнения модуля обучения, отправленного из Облака.

– *Облачный уровень:* Центральный сервер (оркестратор) координирует процесс FedAvg: выбирает клиентов, распространяет модель, агрегирует обновления и перераспределяет глобальную модель.

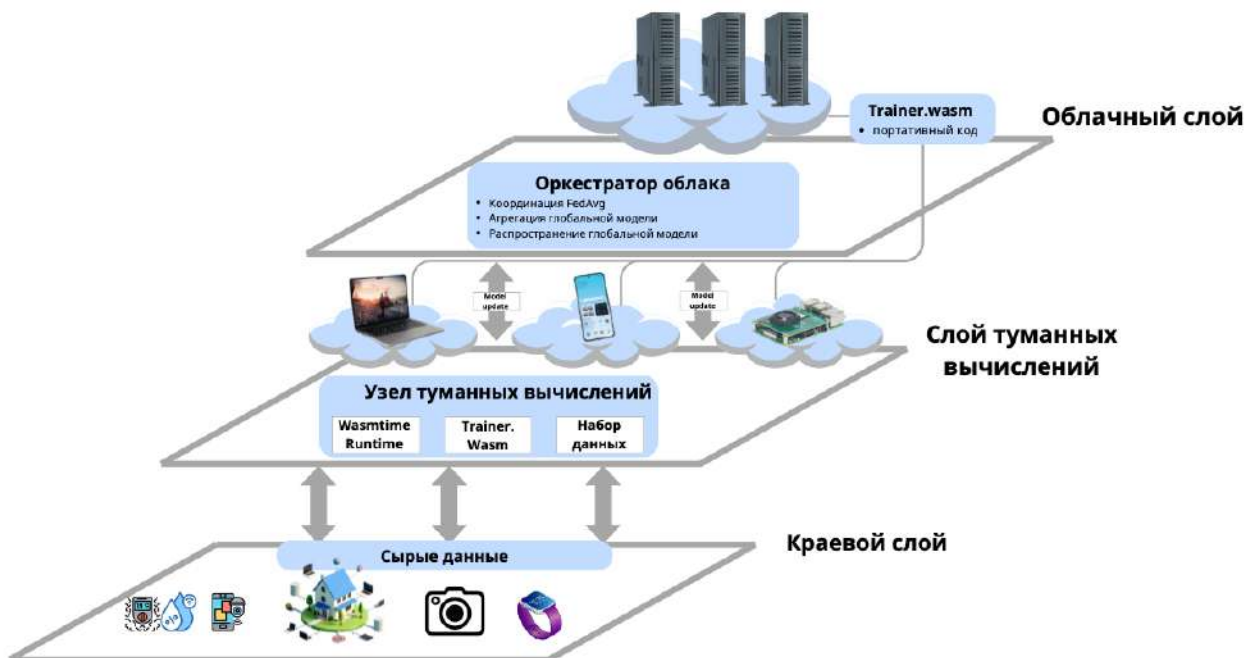


Рисунок 29 — Интегрированная структура с WebAssembly

Центральная задача FL, формально изложенная в Главе 2, состоит в минимизации глобальной функции потерь $F(\theta)$ на данных, распределенных по туманным узлам, без обмена необработанными данными между узлами или с Облаком:

$$\min_{\theta} F(\theta) = \sum_{f=1}^F \frac{D_f}{N} F_f(\theta), \quad N = \sum_{f=1}^F D_f \quad (36)$$

где: $F_f(\theta) = \frac{1}{D_f} \sum_{(x,y) \in B_f} l(\theta; x, y)$ — локальная функция потерь на приватном наборе данных B_f (размером $D_f = |B_f|$) на туманном узле f ; $l(\theta; x, y)$ — функция потерь для одного образца данных; F — количество туманных узлов.

В контексте настоящей главы рассматривается задача развёртывания федеративного обучения в распределённой инфраструктуре туманных вычислений, включающей территориально распределённые узлы с различными типами оборудования и вычислительными возможностями. Конфиденциальность исходных данных обеспечивается архитектурой федеративного обучения: данные сохраняются на локальных туманных узлах, а по сети передаются только параметры или обновления модели. Поэтому конфиденциальность рассматривается как базовая возможность федеративного обучения, тогда как основным предметом исследования является обеспечение совместимого и эффективного выполнения модели на неоднородных туманных узлах.

4.4 Предлагаемая структура на основе WebAssembly

WebAssembly по своей конструкции представляет собой низкоуровневый формат машинного кода со стандартизированной семантикой, разработанный так, чтобы не зависеть от конкретной встроенной платформы. Это свойство позволяет одному и тому же модулю выполняться на различных архитектурах с детерминированным и согласованным поведением. В данном исследовании система построена для работы на Wasmtime — высокопроизводительной среде выполнения, разработанной Bytecode Alliance, некоммерческой организацией, в число участников которой входят Mozilla, Fastly, Intel и Red Hat. Wasmtime

спроектирована на основе четырех ключевых принципов, делающих ее подходящей для промышленных внедрений [101]:

– *Быстродействие*: Wasmtime интегрирует оптимизирующий компилятор для создания высококачественного машинного кода и поддерживает как JIT, так и AOT компиляцию. Среда выполнения специально оптимизирована для быстрой инициализации экземпляров и минимизации накладных расходов при вызовах функций между хостом и гостевой системой.

– *Безопасность*: Реализация на языке Rust обеспечивает прочную основу для безопасности памяти. Проект подвергается непрерывному тестированию, финансируемому Google OSS-Fuzz, и следует строгому процессу реагирования на инциденты безопасности для оперативного устранения уязвимостей [102, 103].

– *Конфигурируемость*: Wasmtime предоставляет возможности тонкой настройки для контроля ресурсов, позволяя устанавливать лимиты на потребление ЦП и памяти для каждого экземпляра Wasm. Это делает среду выполнения пригодной для широкого спектра сред развертывания - от встраиваемых устройств с ограниченными ресурсами до крупных серверов, обрабатывающих множество одновременных экземпляров [104].

– *Соответствие стандартам*: Wasmtime проходит официальное тестирование. Система тестов разработана сообществом WebAssembly. Разработчики Wasmtime активно участвуют в работе по стандартизации. Важное обязательство - отказ от проприетарных расширений. Это гарантирует две вещи. Во-первых, модули Wasm остаются переносимыми. Во-вторых, они совместимы с другими реализациями. Для промышленной эксплуатации Wasmtime использует систему уровней поддержки. Платформы классифицируются по трём уровням (тирам). Это позволяет разработчикам оценить риски развёртывания (см. Таблица 21).

Таблица 22 – Уровни поддержки платформ в Wasmtime

Уровень	Описание	Примеры платформ
Уровень 1	Готова к промышленной эксплуатации, полностью протестирована и всесторонне поддерживается	x86_64-unknown-linux-gnu, x86_64-apple-darwin, x86_64-pc-windows-msvc
Уровень 2	Близка к готовности к промышленной эксплуатации, хорошо поддерживается, но может не соответствовать некоторым критериям Tier 1	aarch64-unknown-linux-gnu, aarch64-apple-darwin
Уровень 3	Поддерживается, но широко не тестировалась; не рекомендуется для промышленной эксплуатации	riscv64gc-unknown-linux-gnu, x86_64-unknown-freebsd, aarch64-linux-android

WebAssembly поддерживает несколько исходных языков. Можно использовать C, C++ и AssemblyScript. Однако Rust остаётся наиболее популярным выбором. Rust сочетает три качества. Во-первых, высокую производительность. Во-вторых, безопасность памяти. В-третьих, удобство разработки. Комбинация Rust и Wasm считается идеальной. Это связано с техническими достоинствами обоих инструментов. Но главное - их проектные философии совпадают. Оба подходят со схожим видением надёжности и безопасности.

– *Безопасность памяти без сборщика мусора*: Rust гарантирует безопасность памяти на этапе компиляции с помощью механизма владения (ownership) и проверки заимствований (borrow checker), устраняя необходимость в сборщике мусора (Garbage Collector, GC) во время выполнения. Это согласуется с Wasm, который также спроектирован без встроенного GC. Компиляция Rust в Wasm позволяет создавать компактные модули, не содержащие громоздкой среды выполнения GC, что уменьшает размер бинарного файла, ускоряет время загрузки

и позволяет избежать непредсказуемых пауз, обеспечивая стабильную производительность [105].

– *Высокая производительность и абстракции с нулевой стоимостью*: Rust разработан для обеспечения производительности на уровне C/C++. Его характерные «абстракции с нулевой стоимостью» (например, итераторы, замыкания, Option, Result) полностью оптимизируются компилятором в эффективный машинный код без накладных расходов во время выполнения, что соответствует цели Wasm по достижению производительности, близкой к нативной.

– *Зрелая экосистема инструментов*: Сообщество Rust предоставляет обширный инструментарий для Wasm. Такие утилиты, как wasm-pack, автоматизируют процессы сборки и упаковки (включая npm-совместимые пакеты), а wasm-bindgen генерирует связующий код для бесшовного взаимодействия Rust и JavaScript. Пакетный менеджер <cargo> упрощает управление зависимостями.

– *Стратегическое соответствие по всему стеку*: Rust хорошо подходит как для написания модулей Wasm (гостевой код), так и для создания безопасных, высокопроизводительных сред выполнения Wasm (код хоста), каноническим примером чего является Wasmtime [106]. Это позволяет создать единую экосистему, в которой разработчики могут использовать один и тот же язык и инструменты во всем стеке приложений.

Конвейер от исходного кода Rust к исполняемому бинарному файлу WASI проиллюстрирован на рисунке 30. Этот конвейер подчёркивает, что инструментальная цепочка Rust-to-WASM находится на этапе перехода и активного развития. В то время как такие инструменты, как wasm-pack, оптимизированы для веб-контекста и сред граничных вычислений IoT [107], компиляция для автономных сред выполнения, таких как Wasmtime, проще и, как правило, требует только стандартных команд Cargo. Это отражает растущую зрелость WASM как первоклассной цели компиляции в экосистеме Rust.

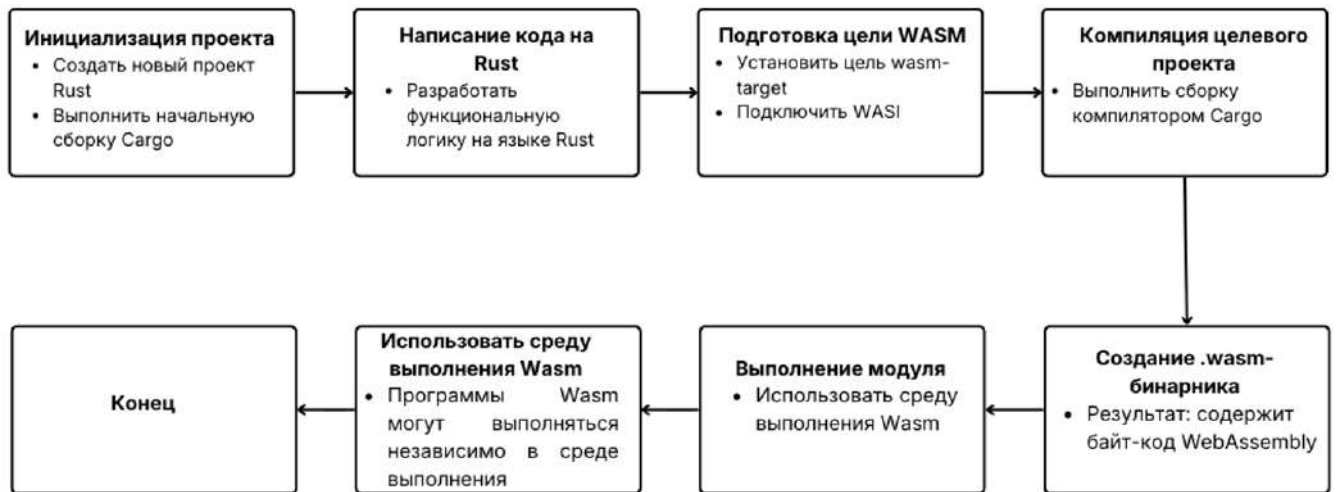


Рисунок 30 — Конвейер компиляции от исходного кода Rust к бинарному файлу WASI

По своей архитектуре Wasm является полностью изолированной вычислительной средой. По умолчанию модуль Wasm не может выполнять операции ввода-вывода, такие как чтение файлов, открытие сетевых соединений или доступ к системным часам [108]. Эта изоляция лежит в основе модели безопасности Wasm, но также создает препятствия для практических приложений. Интерфейс WASI был разработан для решения этой проблемы путем определения стандартизированного, платформо-независимого системного интерфейса, который предоставляет API для безопасного и контролируемого взаимодействия между модулем Wasm и внешним миром. Целью WASI является распространение принцип Wasm «написано однажды, работает везде» с уровня ЦП на системный уровень, позволяя одному модулю взаимодействовать с различными операционными системами через единый набор API.

Основным принципом проектирования WASI является безопасность на основе возможностей. В отличие от традиционных моделей доступа (например, POSIX), где программа запускается с привилегиями пользователя и может попытаться получить доступ к любому ресурсу, WASI использует инвертированную модель: модуль Wasm запускается с нулевыми полномочиями и отсутствием внешних привилегий. Для доступа к ресурсу хост-среда выполнения должна явно

предоставить модулю «возможность» — неподделиваемый дескриптор этого ресурса. Модуль может действовать только в отношении тех ресурсов, для которых он имеет такой дескриптор. Этот механизм заставляет разработчиков заранее указывать необходимые разрешения и естественным образом обеспечивает соблюдение принципа наименьших привилегий, одновременно устраняя проблему «запутанного заместителя» (с англ., *confused deputy problem*), распространенную в традиционных системах безопасности.

Основной вклад предлагаемой структуры и метода состоит в использовании WebAssembly в качестве инфраструктурного слоя выполнения для локального обучения, что непосредственно направлено на решение ключевых проблем: гетерогенности устройств, системной безопасности, экономии вычислительных ресурсов и алгоритмической целостности при развёртывании.

– *Архитектурная переносимость*: Модуль обучения разрабатывается на Rust и компилируется однократно в портативный бинарный файл формата wasm32 (создавая 32-битные модули Wasm). Полученный бинарный файл, *trainer.wasm*, может быть распространен из облака и выполнен на любом туманном узле, независимо от аппаратной архитектуры (ARM, x86) или операционной системы (Linux, Android). Этот подход полностью устраняет бремя кросс-компиляции и сложность управления зависимостями («ад зависимостей») на гетерогенных туманных устройствах.

– *Безопасность и изоляция*: Система использует среду выполнения Wasmtime в соответствии с WASI и его моделью безопасности на основе возможностей. По умолчанию *trainer.wasm* не имеет доступа к файловой системе, сети или системным часам. При запуске операционная система туманного узла должна явно предоставить модулю разрешение на чтение/запись только в специально отведенных каталогах. Этот механизм защищает туманные узлы от вредоносного кода обучения или сбоев, исходящих из облака.

– *Алгоритмическая целостность*: Слой Wasm служит исключительно субстратом для выполнения и не изменяет математическую логику алгоритма. Следовательно, траектория сходимости (потери/точность) модели в среде Wasm-

FL совпадает с таковой при нативной реализации, гарантируя корректность результатов обучения.

Таблица 23 — Алгоритм координации на стороне сервера в Облаке

Алгоритм 1: Координация на стороне сервера в Облаке (FedAvg с частичным участием и интеграцией Wasm)	
Input: Начальные параметры глобальной модели $\theta^{(0)}$, общее количество туманных узлов F, количество раундов связи R, политика выбора клиентов (например, случайная). Output: Финальные параметры глобальной модели $\theta^{(R)}$.	
1.	Инициализировать глобальную модель $\theta^{(0)}$.
2.	FOR каждого раунда связи $r = 0$ до $R - 1$ DO:
a.	Выбрать подмножество участвующих клиентов $S_r \subseteq \{1, \dots, F\}$ на основе политики выбора.
b.	Разослать (broadcast) текущую глобальную модель $\theta^{(r)}$ и (если обновлён) модуль обучения Wasm <code>trainer.wasm</code> вместе с конфигурационным файлом <code>config.json</code> каждому клиенту $f \in S_r$.
c.	Собрать обновления локальных моделей $(\theta_f^{(r+1)}, D_f)$ от всех клиентов в S_r , где $D_f = B_f $ — размер локального набора данных у клиента f .
d.	Агрегировать обновления, используя взвешенное усреднение:
	$\theta^{(r+1)} \leftarrow \sum_{f \in S_r} \frac{D_f}{\sum_{j \in S_r} D_j} \cdot \theta_f^{(r+1)}$
3.	END FOR
4.	Вернуть финальную глобальную модель $\theta^{(R)}$.

Таблица 24 — Алгоритм локального обучения на стороне клиента

Алгоритм 2: Локальное обучение на стороне клиента (туманный узел f)

Input: Текущая глобальная модель $\theta^{(r)}$, приватный локальный набор данных B_f , количество локальных эпох E , размер мини-пакета (batch size) B , скорость обучения η , модуль Wasm `trainer.wasm` и конфигурационный файл `config.json` (содержащий E , B , η).

Output: Обновлённая локальная модель $\theta_f^{(r+1)}$, размер набора данных D_f и опциональные метрики.

1. Получить и загрузить начальную модель $\theta \leftarrow \theta^{(r)}$.
2. Подготовить песочницу Wasm с безопасностью на основе возможностей (WASI):
 - 2.1. Предоставить ограниченный доступ к каталогам: `--dir <data_dir>` для чтения B_f , `--dir <model_dir>` для ввода-вывода модели.
 - 2.2. Применить ограничения ресурсов (например, лимиты CPU/RAM).
3. Вызвать среду выполнения Wasm:


```
wasmtime [--aot] --dir <data_dir> --dir <model_dir>
  trainer.wasm config.json
```
4. Внутри модуля Wasm (выполнение недоверенного кода):
 - а. Загрузить веса модели θ из `<model_dir>` и набор данных B_f из `<data_dir>`.
 - б. FOR каждой локальной эпохи $e = 1$ до E DO:

FOR каждого мини-пакета (x_b, y_b) размера B , выбранного из B_f DO:

$$g \leftarrow \nabla l(\theta; x_b, y_b)$$

$$\theta \leftarrow \theta - \eta \cdot g$$

где $l(\cdot)$ — функция потерь (например, перекрёстная энтропия), ∇ обозначает градиент (опционально с использованием оптимизаторов, таких как Adam).

END FOR

END FOR

с. Вычислить и сохранить обновлённую модель $\theta_f^{(r+1)} \leftarrow \theta$ и метрики (например, локальные потери/точность) в `<model_dir>`.

5. Вне Wasm (хост-среда): Считать результаты из `<model_dir>` и передать $\{\theta_f^{(r+1)}, D_f, "metrics"\}$ обратно на облачный сервер.

4.5. Экспериментальная оценка

Эксперименты проведены в научно-исследовательской лаборатории Meganetlab 6G Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. Установка эмулирует FL в среде туманных вычислений в двух вариантах: обучение на платформе Wasm и обучение с использованием стандартного нативного кода. Система реализована по трёхуровневой архитектуре «облако – туман – пользовательское устройство», где:

– *Облако*: выполняет роль центрального оркестратора, реализующего алгоритм агрегации FedAvg, рассылает глобальную модель и собирает обновления от узлов тумана. Сервер развернут на MacBook Pro M1 (8-ядерный CPU, 14-ядерный GPU, 32 ГБ ОЗУ, SSD 1 ТБ), что обеспечивает быструю обработку задач агрегации.

– *Туман*: три гетерогенных узла по аппаратной части и ОС: два Raspberry Pi 4 (ARM64, Linux) и смартфон Xiaomi 13 (ARM, Android). Такая разнородность намеренно выбрана для отражения типичной гетерогенности реальных сред туманных вычислений. Узлы тумана проводят локальное обучение на своих приватных данных и отправляют обновления градиентов серверу.

– *Датчики*: эмулируют источники сырых данных для узлов тумана. В данном исследовании данные предварительно разделены по узлам для имитации локального сбора с датчиков.

На рисунке 31 показана реальная конфигурация развертывания: три устройства тумана подключены к серверу по локальной сети, образуя полную

систему FL. Эксперимент спроектирован с разумными допущениями, чтобы сосредоточиться на оценке именно производительности Wasm. Связь между облаком и туманом осуществляется по gRPC/TCP; сырые данные в облако не передаются – только обновления модели. Предполагается стабильное сетевое соединение узлов тумана с сервером, что исключает шум от неопределённой сетевой задержки и позволяет точно измерить чистые вычислительные накладные расходы.

Набор данных - MNIST (10 классов), стандартный в исследованиях FL. Данные разделены (Non-IID) образом с использованием распределения Дирихле при $\alpha=0,5$ для эмуляции реального дисбаланса: каждый узел получает подмножество с перекосом по меткам. Конкретно, узлы имеют разное количество образцов и сконцентрированы на определённых метках, что отражает характеристики реальных распределённых данных. Моделью обучения является многоклассовая логистическая регрессия (Softmax) с 7850 параметрами (784×10 весов + 10 смещений). Выбрана простая модель, чтобы изолировать накладные расходы слоя WebAssembly без влияния сложности оптимизатора. Компактный размер модели (31,4 КБ) дополнительно минимизирует влияние пропускной способности сети на измерения.

Ключевое различие между двумя конфигурациями эксперимента состоит в используемом слое выполнения. Код локального обучения написан на языке Rust и компилируется двумя способами:

- *Базовая линия (Native)*: компиляция непосредственно в машинный код для каждой архитектуры (aarch64 для Raspberry Pi/Xiaomi; x86_64 для остальных платформ).

- *WebAssembly*: компиляция с помощью wasm32-wasi с получением портативного модуля trainer.wasm, исполняемого через среду Wasmtime на всех узлах тумана.

Среда Wasm обеспечивает изоляцию выполняемого модуля и предоставляет ему только явно заданные разрешения на доступ к ресурсам узла. В рассматриваемом эксперименте модулю разрешено читать данные из выделенного

каталога и записывать обученную модель в выходной каталог. Доступ к сети и другим областям файловой системы запрещён. Такая конфигурация ограничивает взаимодействие модуля с ресурсами туманного узла и снижает риск нарушения конфиденциальности данных при возникновении ошибок или наличии вредоносного кода в модуле обучения.



Рисунок 31 — Экспериментальное развертывание на физических устройствах

Собираемые метрики: точность и потери (для сходимости); время выполнения, пропускная способность (образцов/с), максимальный объём резидентной памяти (max RSS), задержка инициализации (с) (для системной производительности); оценочное активное время CPU и загрузка CPU (%CPU) (для оценки ресурсов).

Сходимость модели

Точность: результаты показывают, что оба метода обеспечивают практически идентичные траектории сходимости по точности. После 10 раундов традиционный подход даёт среднюю точность 90,15 %, Wasm 89,86 %, разница всего 0,29 %. При 50 раундах разрыв сокращается до 0,04 % (Native: 91,25 %, Wasm: 91,29 %). На 100 раундах Wasm достигает 91,49 % против 91,49 % у Native, что

свидетельствует о высокой стабильности при длительной сходимости (рисунок 32). Небольшое расхождение может быть связано с различиями численной точности в среде Wasm, но не оказывает существенного влияния на общую производительность.

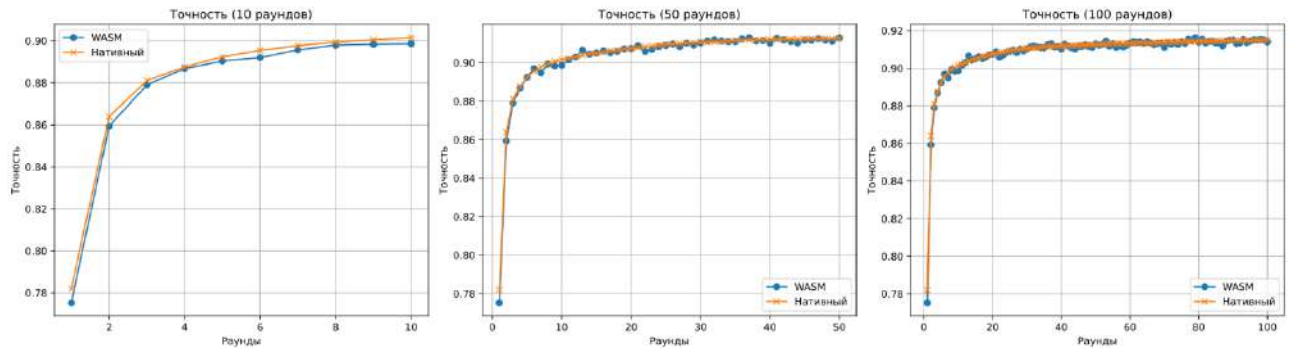


Рисунок 32 — Точность по раундам обучения

Функция потерь: тенденции снижения потерь у обоих методов практически идентичны (рисунок 33). На 10-м раунде Native: 0,339, Wasm: 0,346. К 50-му раунду оба сходятся к $\approx 0,296$ (Native: 0,296, Wasm: 0,296). После 100 раундов Native: 0,286, Wasm: 0,289 – разница пренебрежимо мала. Результаты подтверждают, что использование Wasm не влияет на корректность алгоритма FL и способность модели к сходимости.

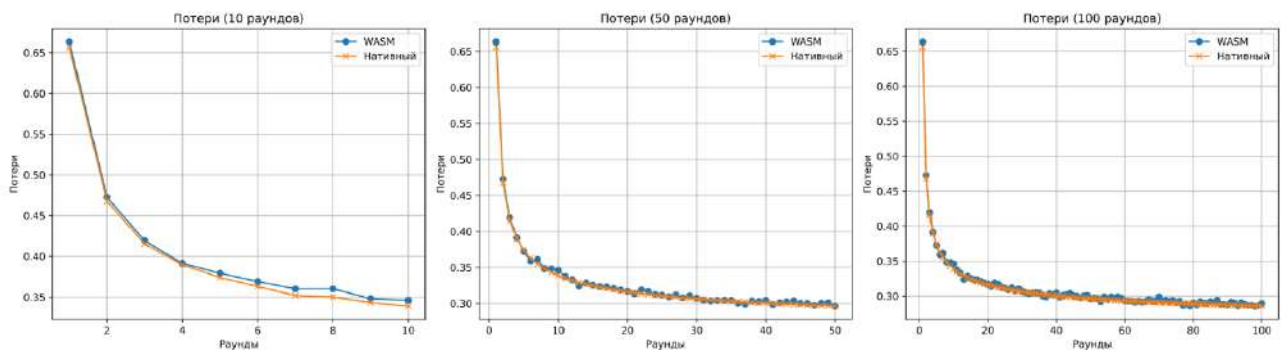


Рисунок 33 — Потери по раундам обучения

Оценка вычислительной производительности

Пропускная способность (образцов/с) демонстрирует явное преимущество традиционного развертывания кода за счёт прямого выполнения на оборудовании по сравнению с абстрактным слоем среды Wasm (рисунок 34). В эксперименте на

10 раундов Native: 120 990 образцов/с (± 5120), что в 2,35 раза выше Wasm (51 460 образцов/с ± 1230). Разрыв сохраняется при увеличении раундов: 50 раундов - Native 123 228 образцов/с (± 4890), в 2,43 раза выше Wasm (50 791 образцов/с ± 1450); 100 раундов - Native 125 098 образцов/с (± 5340), в 2,46 раза выше Wasm (50 754 образцов/с ± 1560). Отношение пропускной способности Native к Wasm составляет 2,35–2,46 раза. При этом фактическое время выполнения полного цикла обучения для Wasm увеличивается на 10,2–64,6 % по сравнению с Native в зависимости от числа раундов обучения.

Конкретно по времени выполнения:

- 10 раундов: накладные расходы 64,6 % (132,60 с против 80,51 с)
- 50 раундов: накладные расходы 23,1 % (310,84 с против 252,45 с)
- 100 раундов: накладные расходы 10,2 % (514,00 с против 466,63 с)

Примечательно, что накладные расходы уменьшаются с ростом числа раундов, что указывает на успешную амортизацию затрат на инициализацию среды Wasm в длительных нагрузках.

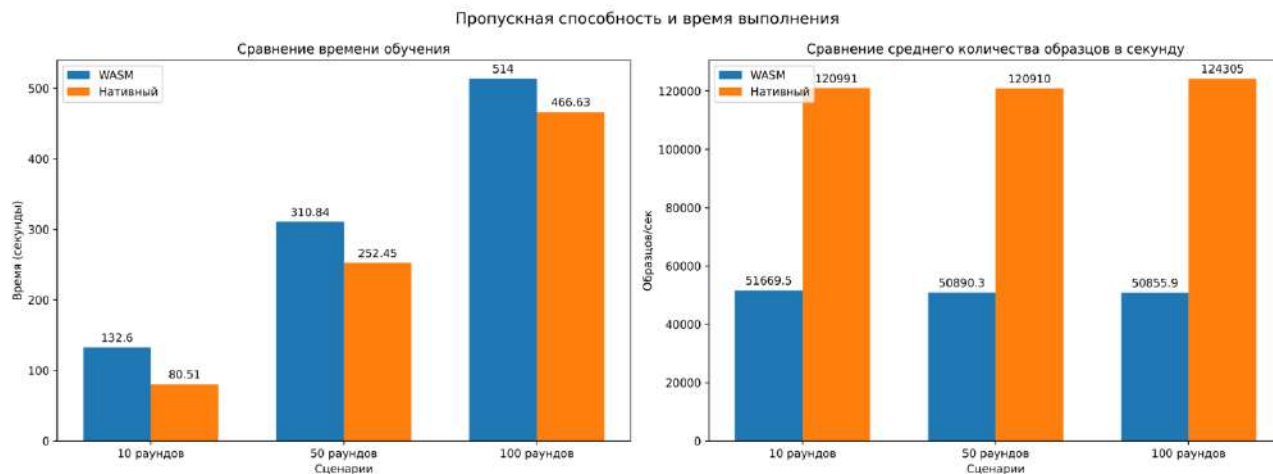


Рисунок 34. Сравнение времени обучения и средней пропускной способности

Среднее время инициализации за раунд у Wasm (2,02 с при 10 раундах) выше, чем у Native (1,57 с), однако разрыв сокращается при увеличении масштаба (рисунок 35): 50 раундов - Wasm 2,03 с против Native 1,58 с; 100 раундов - Wasm 2,02 с против Native 1,59 с. Это может быть связано с использованием АОТ-

компиляции в Wasm, снижающей время запуска по сравнению с традиционным кодом.

Анализ резидентной памяти выявляет значительное преимущество Wasm по энергоэффективности памяти. “Native” потребляет в среднем 193,0 МБ ($\pm 2,5$ МБ) и остаётся стабильным. Wasm — только 58,8 МБ ($\pm 1,2$ МБ), экономия 69,5 %. Конкретно:

- 10 раундов: Wasm 58,8 МБ против “Native” 193,3 МБ (экономия 69,6 %)
- 50 раундов: Wasm 58,8 МБ против “Native” 193,0 МБ (экономия 69,5 %)
- 100 раундов: Wasm 58,9 МБ против “Native” 193,0 МБ (экономия 69,5 %)

Это преимущество особенно важно в средах туманных вычислений, где краевые устройства часто ограничены по памяти (Raspberry Pi, шлюзы и т.п).

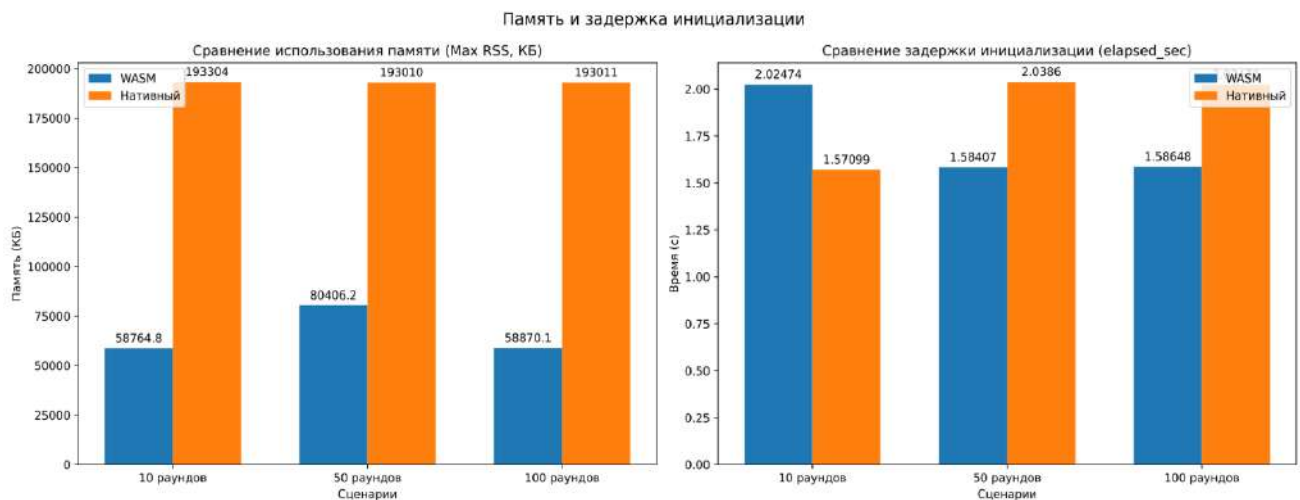


Рисунок 35 — Потребление памяти и среднее время инициализации по раундам

Оценочное активное время CPU показывает, что Wasm требует примерно на 25 % больше активного времени из-за накладных расходов runtime, однако максимальная общая загрузка %CPU ниже: Wasm ~22–25 % против “Native” 82–90 % (рисунок 36). Это свидетельствует о лучшей пригодности Wasm для сред с ограниченными ресурсами (Fog и Edge).

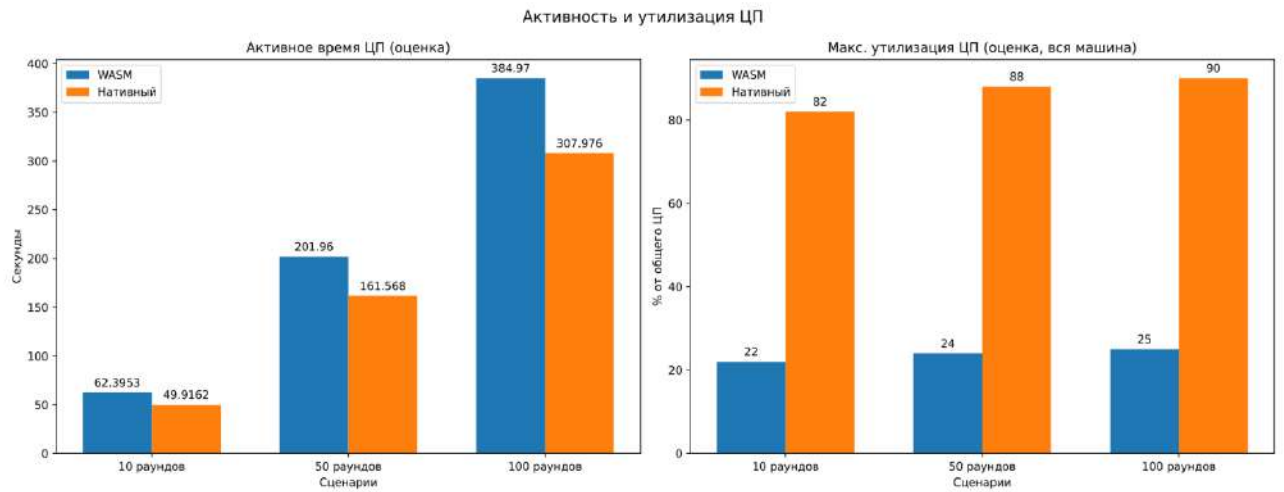


Рисунок 36 — Активное время CPU и максимальная загрузка %CPU в процессе обучения

4.6 Выводы по Главе 4

1. В данной главе представлена структура и метод кроссплатформенного развёртывания FL на гетерогенных узлах FC на основе технологии WebAssembly и среды выполнения Wasmtime. Предложенная структура и метод закрывают третий операционный пробел, сформулированный в главе 1, обеспечивая принцип «скомпилировано однажды - выполняется везде» для модуля локального обучения FL на физически неоднородной инфраструктуре FC, включающей устройства с архитектурами ARM64, ARM и x86 под управлением различных операционных систем.

2. Установлено, что применение технологии Wasm в качестве переносимого слоя выполнения не оказывает существенного влияния на алгоритмическую корректность FL. Теоретический анализ и эмпирическая оценка на физических устройствах Raspberry Pi 4 (ARM64, Linux) и Xiaomi 13 (ARM, Android) подтвердили, что предложенная структура сохраняет траекторию сходимости, эквивалентную нативному выполнению: расхождение по точности и функции потерь не превышает 0,3% после 100 раундов обучения.

3. Выявлены значительные преимущества предложенной структуры и метода по ресурсным показателям. Потребление оперативной памяти снижается на 69,5% (58,8 МБ против 193 МБ при нативном выполнении), загрузка процессора сокращается до 22–25% против 82–90% при нативном коде. Накладные расходы по времени выполнения снижаются с 64,6% (10 раундов) до 10,2% (100 раундов), что свидетельствует об успешной амортизации затрат на инициализацию среды Wasm при длительной эксплуатации.

4. Раскрыта практическая значимость предложенной структуры в контексте развёртывания распределённого ИИ в сетях шестого поколения. Модель безопасности WASI на основе возможностей обеспечивает изолированное выполнение модуля обучения с минимально необходимыми привилегиями доступа к ресурсам хоста, что формирует дополнительный уровень защиты туманных узлов от потенциально вредоносного кода. Полученные результаты дают разработчикам

и архитекторам чёткие ориентиры по компромиссам производительность/ресурсы при развёртывании систем федеративного обучения на ресурсно-ограниченных узлах и сетях туманных вычислений.

ЗАКЛЮЧЕНИЕ

В диссертационной работе получены следующие основные результаты:

1. Решена научная задача разработки и исследования интегрированной архитектуры туманных вычислений и федеративного обучения в сетях шестого поколения, а также моделей и методов её практической реализации на гетерогенной инфраструктуре, обеспечивающей реализацию принципа “AI-native” и концепции сети связи с искусственным интеллектом согласно Рекомендации МСЭ-Р М.2160 и Резолюции 101 ВАСЭ-24.

2. Проведён анализ направлений развития сетей шестого поколения на основе Рекомендации МСЭ-Р М.2160, Резолюции 101 ВАСЭ-24 и Рекомендаций МСЭ-Т серии Y.317х. Установлено, что сети связи шестого поколения представляют собой качественный переход к интегрированной распределённой платформе, объединяющей связь, вычисления и искусственный интеллект в соответствии с принципом “естественного ИИ” (с англ. AI-native). Выявлены три фундаментальных пробела, включающие ограничения пропускной способности канала связи, недостаточные возможности алгоритмов для распределённого принятия решений и аппаратную неоднородность граничных узлов. Предложена трёхуровневая интегрированная архитектура туманных вычислений и федеративного обучения, реализующая концепцию сети связи с искусственным интеллектом путём отображения функциональных компонентов на физическую инфраструктуру «облако – туманные узлы – пользовательские устройства».

3. Разработан метод выбора пользовательских устройств и модель федеративного обучения с туманными вычислениями для сетей шестого поколения, отличающийся от известных тем, что адаптивный порог выбора динамически пересчитывается на уровне туманных вычислений в каждом раунде на основе статистики скоростей изменения нормы градиента, что позволяет учитывать неоднородность локальных данных и динамику процесса сходимости, а также снизить объём сетевого взаимодействия между устройствами и центральным сервером на 39% при сохранении точности модели.

4. Разработана децентрализованная структура федеративного обучения с туманными вычислениями на основе адаптивной графовой свёрточной рекуррентной сети для прогнозирования городских транспортных потоков в сервисах городских сетевых вселенных (CitiVerse), отличающаяся от известных тем, что полностью исключает центральный сервер из цикла обучения и агрегации модели искусственного интеллекта, а механизм раздельной агрегации глобальных весов и локальных векторов встраивания обеспечивает баланс между общими знаниями и региональной адаптацией модели при решении проблемы Non-IID (неоднородных) данных, а также учитывает новую структуру туманных альянсов. Предложенный подход демонстрирует высокую эффективность: снижает потребность в сетевых ресурсах на ~48 %, по сравнению с традиционными методами федеративного обучения, ускоряет сходимость модели при сохранении точности модели прогноза.

5. Разработаны структура и метод кроссплатформенного развёртывания федеративного обучения на гетерогенных узлах туманных вычислений в архитектуре «облако – туман – пользовательское устройство», отличающиеся от известных решений тем, что система байт-кода WebAssembly используется как инфраструктурный слой выполнения локального обучения моделей федеративного искусственного интеллекта на гетерогенных туманных узлах, что устраняет необходимость в перекомпиляции и переконфигурации под каждую целевую аппаратную архитектуру, в отличие от нативного подхода и контейнеризации, тем самым ускоряя процесс развёртывания модулей федеративного искусственного интеллекта в гетерогенной среде туманных вычислений, что существенно снижает сложность развёртывания в сетях шестого поколения.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ИИ	искусственный интеллект
МСЭ	Международный союз электросвязи
ОС	операционная система
ЦОД	центр обработки данных
AGCRN	Adaptive Graph Convolutional Recurrent Network — адаптивная графовая свёрточная рекуррентная сеть
AFN	Aggregation Fog Node — агрегирующий туманный узел
AI-native	архитектура телекоммуникационной сети с естественной интеграцией искусственного интеллекта на всех уровнях
AOT	Ahead-of-Time compilation — заблаговременная компиляция
API	Application Programming Interface — интерфейс прикладного программирования
ARM	семейство архитектур процессоров на основе системы команд ARM
CitiVerse	City network Metaverse — концепция городских сетевых вселенных;
CNN	Convolutional Neural Network — свёрточная нейронная сеть
CNC	Computing and Networking Coordination — система координации сети и вычислений
CPU	Central Processing Unit — центральный процессор
DAGG	Data-Adaptive Graph Generation — генерация графов с адаптацией к данным
DBSCAN	Density-Based Spatial Clustering of Applications with Noise — пространственная кластеризация на основе плотности с обработкой шума
eMBB	enhanced Mobile Broadband — улучшенная мобильная широкополосная связь

FC	Fog Computing — туманные вычисления
FL	Federated Learning — федеративное обучение
FMSC	Fixed, Mobile and Satellite Converged networks — конвергентные фиксированные, мобильные и спутниковые сети
GCN	Graph Convolutional Network — графовая свёрточная сеть
GNN	Graph Neural Network — графовая нейронная сеть
GPU	Graphics Processing Unit — графический процессор
GRU	Gated Recurrent Unit — управляемый рекуррентный блок
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise — иерархическая пространственная кластеризация на основе плотности с обработкой шума
HRLLC	Hyper Reliable and Low-Latency Communication — гипернадёжная связь с ультрамалой задержкой
ICT	Information and Communication Technology — информационно-коммуникационные технологии
IMT-2020	International Mobile Telecommunications for 2020 — стандарт МСЭ для сетей пятого поколения
IMT-2030	International Mobile Telecommunications for 2030 and Beyond — стандарт МСЭ для сетей шестого поколения
InNetFL	In-Network aggregated Federated Learning — федеративное обучение с агрегацией внутри сети
IoT	Internet of Things — Интернет вещей
ISA	Instruction Set Architecture — архитектура набора команд процессора
ISAC	Integrated Sensing and Communication — интегрированное зондирование и связь
ITS	Intelligent Transportation Systems — интеллектуальные транспортные системы
JIT	Just-In-Time compilation — компиляция «точно в срок»

K-means	алгоритм кластеризации К-средних
MAE	Mean Absolute Error — средняя абсолютная ошибка
MCC	Mobile Cloud Computing — мобильные облачные вычисления
MEC	Mobile Edge Computing — мобильные граничные вычисления
MNIST	Modified National Institute of Standards and Technology database — открытый эталонный набор данных рукописных цифр
mMTC	massive Machine-Type Communications — массовая межмашинная связь
MSE	Mean Square Error — среднеквадратическая ошибка
NAPL	Node-Adaptive Parameter Learning — адаптивное к узлу обучение параметров
Non-IID	non-Independent and Identically Distributed — неоднородные данные
ONNX	Open Neural Network Exchange — открытый формат обмена нейросетевыми моделями
PeMSD	Performance Measurement System Dataset — обозначение наборов транспортных данных калифорнийской системы Caltrans
RAM	Random Access Memory — оперативная память
ReLU	Rectified Linear Unit — функция активации «выпрямленный линейный блок»
RIS	Reconfigurable Intelligent Surfaces — реконфигурируемые интеллектуальные поверхности
RMSE	Root Mean Square Error — среднеквадратическое отклонение
RNN	Recurrent Neural Network — рекуррентная нейронная сеть

SGD	Stochastic Gradient Descent — стохастический градиентный спуск
sMAPE	Symmetric Mean Absolute Percentage Error — симметричное среднее абсолютное процентное отклонение
STIN	Space-Terrestrial Integrated Network — интегрированная наземно-космическая сеть
TFN	Trainable Fog Node — обучаемый туманный узел
URLLC	Ultra-Reliable Low-Latency Communication — сверхнадёжная связь с малой задержкой
WasmEdge	среда выполнения WebAssembly для граничных вычислений (проект CNCF)
WebAssembly	открытый стандарт переносимого низкоуровневого байт-кода
WASI	WebAssembly System Interface — системный интерфейс WebAssembly для безопасного доступа к ресурсам операционной системы
WASI-NN	WebAssembly System Interface for Neural Networks — расширение WASI для запуска нейросетевых моделей вывода в среде WebAssembly
Wasmtime	среда выполнения модулей WebAssembly от консорциума Bytecode Alliance

СПИСОК ЛИТЕРАТУРЫ

1. Recommendation ITU-R M.2160-0 (11/2023) — Framework and overall objectives of the future development of IMT for 2030 and beyond. — Geneva: ITU-R, 2023.
2. ITU. World Telecommunication Standardization Assembly (WTSA-24). — New Delhi, India, 15–24 October 2024. — URL: <https://www.itu.int/wtsa/2024/> (дата обращения: 18.03.2026).
3. Vocabulary for Artificial Intelligence Native Telecommunication Networks: Technical Specification FG-AINN-O-011 / International Telecommunication Union, Telecommunication Standardization Sector. – 2025. – URL: https://www.itu.int/en/ITU-T/focusgroups/ainn/Documents/Vocabulary_AINTN.pdf (дата обращения: 08.08.2026).
4. Thang D. V. A Framework Integrating Federated Learning and Fog Computing Based on Client Sampling and Dynamic Thresholding Techniques / D. V. Thang, A. Volkov, A. Muthanna, I. A. Elgendy, R. Alkanhel, D. N. K. Jayakody, A. Koucheryavy // IEEE Access. — 2025. — Vol. 13. — DOI: 10.1109/ACCESS.2025.3571979.
5. Тханг Д. В., Волков А. Н. Туманные альянсы: децентрализованная облачная структура с федеративным машинным обучением для городских сетевых вселенных / Д. В. Тханг, А. Н. Волков // Труды учебных заведений связи. — 2026. — Т. 12, № 3. — С. 27–35. — DOI: 10.31854/1813-324X-2026-12-3-27-35. — EDN: PVIDRA.
6. Dao N. N. A review on new technologies in 3GPP standards for 5G access and beyond / N. N. Dao, N. H. Tu, T. D. Hoang, T. H. Nguyen, L. V. Nguyen, K. Lee, L. Park, W. Na, S. Cho // Computer Networks. — 2024. — Vol. 245. — Art. 110370. — DOI: 10.1016/j.comnet.2024.110370.
7. Measuring Digital Development: Facts and Figures 2023 / International Telecommunication Union (ITU). — Geneva: ITU, 2023. — URL:

<https://www.itu.int/itu-d/reports/statistics/facts-figures-2023/> (дата обращения: 18.03.2026).

8. Chowdhury M. Z. 6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions / M. Z. Chowdhury [et al.] // IEEE Open Journal of the Communications Society. — 2020. — Vol. 1. — P. 957–975.

9. Cisco Annual Internet Report (2018–2023) White Paper / Cisco. — 2020. — URL: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html> (дата обращения: 18.03.2026).

10. Данг В. Т. Услуги телеприсутствия и их будущее в развивающейся среде туманных вычислений: обзор исследований и примеров использования / В. Т. Данг, А. Н. Волков // Информационные технологии и телекоммуникации. — 2024. — Т. 12, № 3. — С. 29–51.

11. Cabrera J. A. 6G and the Post-Shannon Theory / J. A. Cabrera, H. Boche, C. Deppe, R. F. Schaefer, C. Scheunert, F. H. P. Fitzek // Shaping Future 6G Networks: Needs, Impacts, and Technologies. — 2021. — P. 271–294.

12. Press release: World Radiocommunication Conference approves IMT-2030 framework for 6G mobile technologies / ITU. — 2023. — URL: <https://www.itu.int/en/mediacentre/Pages/PR-2023-12-01-IMT-2030-for-6G-mobile-technologies.aspx> (дата обращения: 08.04.2026).

13. Волков А. Н. Сети связи пятого поколения: на пути к сетям 2030 / А. Н. Волков, А. С. А. Мутханна, А. Е. Кучерявый // Информационные технологии и телекоммуникации. — 2020. — Т. 8, № 2. — С. 32–43.

14. Pennanen H. 6G: The intelligent network of everything / H. Pennanen, T. Hänninen, O. Tervo, A. Tölli, M. Latva-Aho // IEEE Access. — 2024. — Vol. 13. — P. 1319–1421.

15. Аль-Кереа З. А. Х. Архитектурная эволюция сетей: распределённый интеллект и конвергенция космических, воздушных, наземных и подводных сетей связи / З. А. Х. Аль-Кереа, А. С. А. Мутханна, А. Е. Кучерявый // Электросвязь. — 2025. — № 10. — С. 63–71.

16. IMT-2020 (5G) // ITU-R Study Group 5. — URL: <https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2020/Pages/default.aspx> (дата обращения: 14.04.2026).
17. ITU-R Report M.2516 — Future technology trends of terrestrial International Mobile Telecommunications systems towards 2030 and beyond. — Geneva: ITU-R, 2019. — URL: <https://www.itu.int/pub/R-REP-M.2516> (дата обращения: 18.03.2026).
18. ITU-R Report M.2541 — Technical feasibility of IMT in bands above 100 GHz. — Geneva: ITU-R, 2022. — URL: <https://www.itu.int/pub/R-REP-M.2541> (дата обращения: 18.03.2026).
19. Othman W. M. Key enabling technologies for 6G: The role of UAVs, terahertz communication, and intelligent reconfigurable surfaces in shaping the future of wireless networks / W. M. Othman, A. A. Ateya, M. E. Nasr, A. Muthanna, M. ElAffendi, A. Koucheryavy, A. A. Hamdi // Journal of Sensor and Actuator Networks. — 2025. — Vol. 14, No. 2. — Art. 30.
20. Кучерявый А. Е. Искусственный интеллект в сетях связи / А. Е. Кучерявый, А. С. Бородин, А. С. А. Мутханна, А. Р. Абделлах, А. Н. Волков // Актуальные проблемы инфотелекоммуникаций в науке и образовании: сб. науч. ст. — Санкт-Петербург: СПбГУТ, 2021. — С. 8–18.
21. Zhu X. Enabling intelligent connectivity: A survey of secure ISAC in 6G networks / X. Zhu, J. Liu, L. Lu, T. Zhang, T. Qiu, C. Wang, Y. Liu // IEEE Communications Surveys & Tutorials. — 2024. — Vol. 27, No. 2. — P. 748–781.
22. Qaisar M. U. F. The role of ISAC in 6G networks: Enabling next-generation wireless systems / M. U. F. Qaisar, W. Yuan, O. Günlü [et al.] // IEEE Transactions on Network Science and Engineering. — 2026.
23. Lu Y. 6G: A survey on technologies, scenarios, challenges, and the related issues / Y. Lu, X. Zheng // Journal of Industrial Information Integration. — 2020. — Vol. 19. — Art. 100158.

24. Кучерявый А. Е. Новые перспективы научных исследований в области сетей связи на 2021–2024 годы / А. Е. Кучерявый, Р. В. Киричек, М. А. Маколкина [и др.]. — Санкт-Петербург: СПбГУТ, 2020.
25. Кучерявый А. Е. Сети связи ожидает блестящее наукоёмкое будущее // Электросвязь. — 2022. — № 1. — С. 3–5.
26. Ateya A. A. Enabling Metaverse and Telepresence Services in 6G Networks / A. A. Ateya, A. A. Abd El-Latif, A. Muthanna, A. Volkov, A. Koucheryavy. — 1st ed. — River Publishers, 2025. — DOI: 10.1201/9788770046749.
27. Кузнецов К. А. Метавселенные и методы их реализации в сетях связи 2030 / К. А. Кузнецов, А. Е. Кучерявый, А. С. А. Мутханна // Науч.-техн. конф. СПбНТО РЭС им. А. С. Попова. — 2024. — № 1 (79). — С. 200–201.
28. Волков А. Н. Туманные вычисления в сетях ИМТ-2030 для услуг телеприсутствия / А. Н. Волков, А. Е. Кучерявый // XIV Всерос. совещание по проблемам управления: тр. конф. — Москва, 2024. — С. 2295–2297.
29. Волков А. Н. Туманность в перспективных сетях связи для услуг телеприсутствия // Электросвязь. — 2024. — № 4. — С. 7–13.
30. Recommendation ITU-T Y.3172 — Architectural framework for machine learning in future networks including IMT-2020. — Geneva: ITU-T, 2019. — URL: <https://www.itu.int/rec/T-REC-Y.3172/en> (дата обращения: 18.03.2026).
31. Recommendation ITU-T Y.3177 — Architectural framework for artificial intelligence-based network automation for resource and fault management in future networks including IMT-2020. — Geneva: ITU-T, 2021. — URL: <https://www.itu.int/rec/T-REC-Y.3177/en> (дата обращения: 18.03.2026).
32. Recommendation ITU-T Y.3173 — Framework for evaluating intelligence levels of future networks including IMT-2020. — Geneva: ITU-T, 2020. — URL: <https://www.itu.int/rec/T-REC-Y.3173> (дата обращения: 18.03.2026).
33. Recommendation ITU-T Y.3400 — Coordination of networking and computing in IMT-2020 networks and beyond — Requirements. — Geneva: ITU-T, 2023. — URL: <https://www.itu.int/rec/T-REC-Y.3400/en> (дата обращения: 18.03.2026).

34. Recommendation ITU-T Y.3401 — Coordination of networking and computing in IMT-2020 networks and beyond — Capability framework. — Geneva: ITU-T, 2024. — URL: <https://www.itu.int/rec/T-REC-Y.3401/en> (дата обращения: 18.03.2026).
35. Recommendation ITU-T Y.3224 — Fixed, mobile and satellite convergence — Requirements and framework for in-network aggregated federated learning for enabling artificial intelligence in IMT-2020 networks and beyond. — Geneva: ITU-T, 2025. — URL: <https://www.itu.int/rec/T-REC-Y.3224/en> (дата обращения: 18.03.2026).
36. Mell P. The NIST Definition of Cloud Computing: NIST Special Publication 800-145 / P. Mell, T. Grance. — Gaithersburg: National Institute of Standards and Technology, 2011. — 7 p.
37. Mouradian C. A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges / C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, P. A. Polakos // IEEE Communications Surveys & Tutorials. — 2018. — Vol. 20, No. 1. — P. 416–464.
38. Shi W. Edge Computing: Vision and Challenges / W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu // IEEE Internet of Things Journal. — 2016. — Vol. 3, No. 5. — P. 637–646.
39. Bonomi F. Fog Computing and Its Role in the Internet of Things / F. Bonomi, R. Milito, J. Zhu, S. Addepalli // Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing (MCC'12). — New York: ACM, 2012. — P. 13–16.
40. OpenFog Consortium. OpenFog Reference Architecture for Fog Computing: IEEE Std 1934-2018. — New York: IEEE, 2018.
41. ETSI ISG MEC. Mobile Edge Computing — A Key Technology towards 5G: White Paper No. 11. — Sophia Antipolis: ETSI, 2015. — 36 p.
42. Mutlag A. A. A new fog computing resource management (FRM) model based on hybrid load balancing and scheduling for critical healthcare applications / A. A. Mutlag, M. K. Abd Ghani, O. Mohd, K. H. Abdulkareem, M. A. Mohammed, M. Alharbi, Z. J. Al-Araji // Physical Communication. — 2023. — Vol. 59. — Art. 102109.

43. Songhorabadi M. Fog computing approaches in IoT-enabled smart cities / M. Songhorabadi, M. Rahimi, A. MoghadamFarid, M. H. Kashani // *Journal of Network and Computer Applications*. — 2023. — Vol. 211. — Art. 103557.
44. Zagar P. Dynamic fog computing for enhanced LLM execution in medical applications / P. Zagar, V. Ravi, L. Aalami, S. Krusche, O. Aalami, P. Schmiedmayer // *Smart Health*. — 2025. — Vol. 36. — Art. 100577.
45. Arefian Z. Delay reduction in MTC using SDN based offloading in fog computing / Z. Arefian, M. R. Khayyambashi, N. Movahhedinia // *PLOS ONE*. — 2023. — Vol. 18, No. 5. — Art. e0286483.
46. Kaur N. A systematic review on task scheduling in fog computing: taxonomy, tools, challenges, and future directions / N. Kaur, A. Kumar, R. Kumar // *Concurrency and Computation: Practice and Experience*. — 2021. — Vol. 33, No. 21. — Art. e6432.
47. Reddy K. H. K. A service delay minimization scheme for QoS-constrained, context-aware unified IoT applications / K. H. K. Reddy, R. K. Behera, A. Chakrabarty, D. S. Roy // *IEEE Internet of Things Journal*. — 2020. — Vol. 7, No. 10. — P. 10527–10534.
48. Muhamad W. N. Energy-efficient task offloading in fog computing for 5G cellular network / W. N. Muhamad, S. S. Mohd Aris, K. Dimyati, M. A. Javed, A. Idris, D. Mohd Ali, E. Abdullah // *Engineering Science and Technology, an International Journal*. — 2024. — Vol. 50. — Art. 101628.
49. Raju M. R. DRL-based task scheduling scheme in vehicular fog computing: cooperative and mobility aware approach / M. R. Raju, S. K. Mothku, M. K. Somesula // *Ad Hoc Networks*. — 2025. — Vol. 173. — Art. 103819.
50. Alsharif M. H. Survey of energy-efficient fog computing: Techniques and recent advances / M. H. Alsharif, A. Jahid, R. Kannadasan [et al.] // *Energy Reports*. — 2025. — Vol. 13. — P. 1739–1763.
51. Reddy K. H. K. An energy efficient aware collaborative duty cycling approach for IoT enabled smart industries / K. H. K. Reddy, M. G. Rao, N. R. Pradhan,

D. S. Roy // IEEE Transactions on Industrial Informatics. — 2025. — Vol. 21, No. 3. — P. 2334–2343.

52. Yu D. A hybrid evolutionary algorithm to improve task scheduling and load balancing in fog computing / D. Yu, W. Zheng // Cluster Computing. — 2025. — Vol. 28, No. 1. — Art. 74.

53. Frimpong S. A. An adaptive collaborative intrusion detection system for vehicular fog computing networks / S. A. Frimpong, M. Han, W. Zheng, I. J. Chowdhury // Engineering Applications of Artificial Intelligence. — 2025. — Vol. 158. — Art. 111563.

54. Gowda N. C. MLTPED-BFC: machine learning-based trust prediction for edge devices in the blockchain enabled fog computing environment / N. C. Gowda, A. Bharathi Malakreddy, Y. Vishwanath, K. R. Radhika // Engineering Applications of Artificial Intelligence. — 2025. — Vol. 139. — Art. 109518.

55. Nguyen D. C. Federated learning for internet of things: A comprehensive survey / D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, H. V. Poor // IEEE Communications Surveys & Tutorials. — 2021. — Vol. 23, No. 3. — P. 1622–1658.

56. Yang Q. Federated machine learning: Concept and applications / Q. Yang, Y. Liu, T. Chen, Y. Tong // ACM Transactions on Intelligent Systems and Technology (TIST). — 2019. — Vol. 10, No. 2. — P. 1–19.

57. Yang Z. Federated learning for 6G: Applications, challenges, and opportunities / Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, S. Cui // Engineering. — 2022. — Vol. 8. — P. 33–41.

58. Данг Ван Тханг. Проблема конвергенции федеративного машинного обучения, туманных вычислений и автономных сетей в архитектуре сетей 6G / Данг Ван Тханг, А. Волков, Май Конг Хыонг, Нгуен Дык Ту // 81-я Научно-техническая конференция Санкт-Петербургского НТО РЭС им. А. С. Попова, посвящённая Дню радио : сб. докладов [Электронный ресурс] / СПбГЭТУ «ЛЭТИ». — Санкт-Петербург, 2026. — 566 с. — ISBN 978-5-7629-3618-7. — URL: <https://conf-ntores.etu.ru/2026/ru/>.

59. Chen W. Optimal client sampling for federated learning / W. Chen, S. Horvath, P. Richtarik // arXiv preprint arXiv:2010.13723. — 2020.
60. Luo B. Adaptive heterogeneous client sampling for federated learning over wireless networks / B. Luo, W. Xiao, S. Wang, J. Huang, L. Tassiulas // IEEE Transactions on Mobile Computing. — 2024. — Vol. 23, No. 10. — P. 9663–9677.
61. Wang L. Delta: Diverse client sampling for fast federated learning / L. Wang, Y. Guo, T. Lin, X. Tang // Advances in Neural Information Processing Systems. — 2023. — Vol. 36. — P. 47626–47668.
62. Chen T. LAG: Lazily aggregated gradient for communication-efficient distributed learning / T. Chen, G. Giannakis, T. Sun, W. Yin // Advances in Neural Information Processing Systems. — 2018. — Vol. 31.
63. Ribero M. Communication-efficient federated learning via optimal client sampling / M. Ribero, H. Vikalo // arXiv preprint arXiv:2007.15197. — 2020.
64. Nishio T. Client selection for federated learning with heterogeneous resources in mobile edge / T. Nishio, R. Yonetani // ICC 2019 — IEEE International Conference on Communications. — 2019. — P. 1–7.
65. Wang T. Edge-based communication optimization for distributed federated learning / T. Wang, Y. Liu, X. Zheng, H.-N. Dai, W. Jia, M. Xie // IEEE Transactions on Network Science and Engineering. — 2021. — Vol. 9, No. 4. — P. 2015–2024.
66. ITU Focus Group on Autonomous Networks (FG-AN) // ITU-T. — URL: <https://www.itu.int/en/ITU-T/focusgroups/an/Pages/default.aspx> (дата обращения: 09.04.2026).
67. LeCun Y. Gradient-based learning applied to document recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // Proceedings of the IEEE. — 2002. — Vol. 86, No. 11. — P. 2278–2324.
68. Krizhevsky A. Learning multiple layers of features from tiny images / A. Krizhevsky, G. Hinton. — 2009. — 7 p.
69. Caterini A. L. Deep neural networks in a mathematical framework / A. L. Caterini, D. E. Chang. — Berlin/Heidelberg: Springer International Publishing, 2018.

70. McMahan B. Communication-efficient learning of deep networks from decentralized data / B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas // *Artificial Intelligence and Statistics*. — PMLR, 2017. — P. 1273–1282.
71. Giannakis G. B. Decentralized learning for wireless communications and networking / G. B. Giannakis, Q. Ling, G. Mateos, I. D. Schizas, H. Zhu // *Splitting Methods in Communication, Imaging, Science, and Engineering*. — Cham: Springer International Publishing, 2017. — P. 461–497.
72. Qi T. FedAGCN: A traffic flow prediction framework based on federated learning and asynchronous graph convolutional network / T. Qi, L. Chen, G. Li, Y. Li, C. Wang // *Applied Soft Computing*. — 2023. — Vol. 138. — Art. 110175.
73. Yuan X. Fedstn: Graph representation driven federated learning for edge computing enabled urban traffic flow prediction / X. Yuan, J. Chen, J. Yang [et al.] // *IEEE Transactions on Intelligent Transportation Systems*. — 2022. — Vol. 24, No. 8. — P. 8738–8748.
74. Zhao L. T-GCN: A temporal graph convolutional network for traffic prediction / L. Zhao, Y. Song, C. Zhang [et al.] // *IEEE Transactions on Intelligent Transportation Systems*. — 2019. — Vol. 21, No. 9. — P. 3848–3858.
75. Zhang C. FASTGNN: A topological information protected federated learning approach for traffic speed forecasting / C. Zhang, S. Zhang, J. Q. James, S. Yu // *IEEE Transactions on Industrial Informatics*. — 2021. — Vol. 17, No. 12. — P. 8464–8474.
76. Feng Z. FCGCN: Feature correlation graph convolution network for few-shot individual identification / Z. Feng, H. Zha, C. Xu, Y. He, Y. Lin // *IEEE Transactions on Consumer Electronics*. — 2023. — Vol. 70, No. 1. — P. 2848–2860.
77. Bai L. Adaptive graph convolutional recurrent network for traffic forecasting / L. Bai, L. Yao, C. Li, X. Wang, C. Wang // *Advances in Neural Information Processing Systems*. — 2020. — Vol. 33. — P. 17804–17815.
78. Yu B. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting / B. Yu, H. Yin, Z. Zhu // *arXiv preprint arXiv:1709.04875*. — 2017.

79. Zhang H. Decentralized federated learning based on blockchain: concepts, framework, and challenges / H. Zhang, S. Jiang, S. Xuan // *Computer Communications*. — 2024. — Vol. 216. — P. 140–150.
80. Lian X. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent / X. Lian, C. Zhang, H. Zhang [et al.] // *Advances in Neural Information Processing Systems*. — 2017. — Vol. 30.
81. Turazza F. FedBGS: A Blockchain Approach to Segment Gossip Learning in Decentralized Systems / F. Turazza, M. Pietri, M. Picone, M. Mamei // *2025 IEEE 45th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. — 2025. — P. 760–770.
82. Yu E. CD-SGD: Distributed stochastic gradient descent with compression and delay compensation / E. Yu, D. Dong, Y. Xu, S. Ouyang, X. Liao // *Proceedings of the 50th International Conference on Parallel Processing*. — 2021. — P. 1–10.
83. Stewart G. An implementation of the HDBSCAN* clustering algorithm / G. Stewart, M. Al-Khassaweneh // *Applied Sciences*. — 2022. — Vol. 12, No. 5. — Art. 2405.
84. Chen C. Freeway performance measurement system: mining loop detector data / C. Chen, K. Petty, A. Skabardonis, P. Varaiya, Z. Jia // *Transportation Research Record*. — 2001. — Vol. 1748, No. 1. — P. 96–102.
85. Sharma M. Reliable federated learning in a cloud-fog-IoT environment / M. Sharma, P. Kaur // *The Journal of Supercomputing*. — 2023. — Vol. 79, No. 14. — P. 15435–15458.
86. Manju A. B. Hierarchical Federated Learning with Fog Nodes: Enhancing Efficiency in Smart City Networks / A. B. Manju, C. S. P. Kumar, J. Jegan, D. Jagadeeshan, S. K. Nunna // *2024 OITS International Conference on Information Technology (OCIT)*. — 2024. — P. 749–753.
87. Lähderanta T. Capacitated spatial clustering with multiple constraints and attributes / T. Lähderanta, L. Lovén, L. Ruha [et al.] // *Engineering Applications of Artificial Intelligence*. — 2024. — Vol. 127. — Art. 107182.

88. Ongaro D. In search of an understandable consensus algorithm / D. Ongaro, J. Ousterhout // 2014 USENIX Annual Technical Conference (USENIX ATC 14). — 2014. — P. 305–319.
89. Domazetović B. Performance evaluation of IEEE 802.11 ah systems / B. Domazetović, E. Kočan, A. Mihovska // 2016 24th Telecommunications Forum (TELFOR). — 2016. — P. 1–4.
90. Mittone G. Experimenting with Emerging RISC-V Systems for Decentralised Machine Learning / G. Mittone, N. Tonci, R. Birke [et al.] // Proceedings of the 20th ACM International Conference on Computing Frontiers (CF '23). — Bologna: ACM, 2023. — P. 1–10. — DOI: 10.1145/3587135.3592211.
91. Kakati S. A Cross-Architecture Evaluation of WebAssembly in the Cloud-Edge Continuum / S. Kakati, M. Brorsson // IEEE IPDPS Workshops. — 2024. — DOI: 10.1109/IPDPSW63119.2024.00154.
92. Lim W. Y. B. Federated learning in mobile edge networks: A comprehensive survey / W. Y. B. Lim, N. C. Luong, D. T. Hoang [et al.] // IEEE Communications Surveys & Tutorials. — 2020. — Vol. 22, No. 3. — P. 2031–2063.
93. Barona López L. I. Heterogeneity challenges of federated learning for future wireless communication networks / L. I. Barona López, T. B. Saltos // Journal of Sensor and Actuator Networks. — 2025. — Vol. 14, No. 2. — Art. 37.
94. Abreha H. G. Federated learning in edge computing: a systematic survey / H. G. Abreha, M. Hayajneh, M. A. Serhani // Sensors. — 2022. — Vol. 22, No. 2. — Art. 450.
95. Raza S. M. Empirical performance and energy consumption evaluation of container solutions on resource constrained IoT gateways / S. M. Raza, J. Jeong, M. Kim, B. Kang, H. Choo // Sensors. — 2021. — Vol. 21, No. 4. — Art. 1378.
96. Zhang Y. Research on webassembly runtimes: A survey / Y. Zhang, M. Liu, H. Wang [et al.] // ACM Transactions on Software Engineering and Methodology. — 2025. — Vol. 34, No. 8. — P. 1–47.

97. Eddine Khelifa S. Case study of webassembly runtimes for AI applications on the edge / S. Eddine Khelifa, M. Bagaa, A. M. Ouameur, A. Ksentini // 2024 Global Information Infrastructure and Networking Symposium (GIIS). — 2024. — P. 1–6.
98. Heinrich T. Uso de chamadas WASI para a identificação de ameaças em aplicações webassembly / T. Heinrich, N. C. Will, R. R. Obelheiro, C. A. Maziero // Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg). — SBC, 2023. — P. 237–250.
99. Garofalo M. Web-centric federated learning over the cloud-edge continuum leveraging ONNX and Wasm / M. Garofalo, M. Colosi, A. Catalfamo, M. Villari // 2024 IEEE Symposium on Computers and Communications (ISCC). — 2024. — P. 1–7.
100. Bachmeier J. WebAssembly with WASI-NN for Edge Machine Learning Inference: Experiences and Lessons Learned / J. Bachmeier, V. Yussupov, J. Henß, H. Koziolk // European Conference on Software Architecture. — Cham: Springer Nature Switzerland, 2025. — P. 343–359.
101. Introduction // Wasmtime Documentation / Bytecode Alliance. — URL: <https://docs.wasmtime.dev/> (дата обращения: 28.10.2025).
102. Xu H. Memory-safety challenge considered solved? An in-depth study with all Rust CVEs / H. Xu, Z. Chen, M. Sun, Y. Zhou, M. R. Lyu // ACM Transactions on Software Engineering and Methodology (TOSEM). — 2021. — Vol. 31, No. 1. — P. 1–25.
103. Serebryany K. OSS-Fuzz — Google's Continuous Fuzzing Service for Open-Source Software // Proceedings of the 26th USENIX Security Symposium. — Vancouver, BC: USENIX Association, 2017. — URL: <https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/serebryany> (дата обращения: 10.05.2026).
104. VanHattum A. Lightweight, modular verification for webassembly-to-native instruction selection / A. VanHattum, M. Pardeshi, C. Fallin, A. Sampson, F. Brown // Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. — 2024. — Vol. 1. — P. 231–248.

105. Hoffman K. Programming WebAssembly with Rust: unified development for web, mobile, and embedded applications. — 2019. — 220 p.
106. Wagner L. On the energy consumption and performance of webassembly binaries across programming languages and runtimes in IoT / L. Wagner, M. Mayer, A. Marino [et al.] // Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering. — 2023. — P. 72–82.
107. Prabakar A. WebAssembly Performance Analysis: A Comparative Study of C++ and Rust Implementations / A. Prabakar, R. Kiran. — 2024.
108. Hu Y. WASIT: Deep and Continuous Differential Testing of WebAssembly System Interface Implementations / Y. Hu, W. Zhang, B. Xiao [et al.] // Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles. — 2025. — P. 719–735.

Приложение

Акты реализации диссертационных исследований

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

Юридический адрес: набережная реки Мойки,
д. 61, литера А, Санкт-Петербург, 191186

Почтовый адрес: пр. Большевиков, д. 22, корп. 1,
Санкт-Петербург, 193232
Тел. (812) 3263156, Факс: (812) 3263159

<http://sut.ru>
E-mail: rector@sut.ru
ОКПО 01179934 ОГРН 1027809197635
ИНН 7808004760 КПП 784001001
ОКТМО 40909000

03.09.2026 № 119-1/54-99
на № _____ от _____

УТВЕРЖДАЮ

Проректор по научной работе,
доктор техн.наук



Рабин
Алексей Владимирович

Акт

о внедрении научных результатов, полученных Дангом Ван Тхангом
в диссертационной работе «Разработка и исследование моделей и методов
интеграции туманных вычислений и федеративного обучения в сетях связи шестого
поколения»

Комиссия в составе и.о. декана факультета Инфокоммуникационных сетей и систем В.С. Елагина, доцента кафедры сетей связи и передачи данных Р.А. Дунайцева и заведующей лабораторией кафедры сетей связи и передачи данных О.И. Ворожейкиной составила настоящий акт в том, что научные результаты, полученные результаты, полученные Дангом Ван Тхангом, использованы:

1. При чтении лекций и проведении практических занятий для бакалавров по дисциплине «Искусственный интеллект в сетях и системах связи» (Рабочая Программа регистрационный номер №_23.05/119-Д), разделы Программы:
 - Машинное обучение;
 - Приложения искусственного интеллекта.
2. При чтении лекций и проведении практических занятий для магистров по дисциплине «Современные проблемы науки в условиях перехода к сетям шестого поколения (6G)» (Рабочая Программа регистрационный номер № 24.05/537-Д), раздел Программы:
 - Граничные вычисления. Туманные вычисления. Новая микросервисная архитектура сети. Робототехника и сети связи.

3. При чтении лекций проведении практических занятий для аспирантов по дисциплине «Системы, сети и устройства телекоммуникаций» (Рабочая Программа регистрационный номер №_21.05/780-Д), раздел Программы:

- Современные тенденции развития инфокоммуникационной системы, основные направления научных исследований;

- Основные задачи построения и эксплуатации систем, сетей и устройств связи.

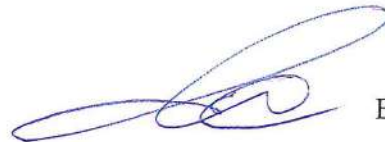
В указанных дисциплинах используются следующие новые научные результаты, полученные Дангом Ван Тхангом в диссертационной работе:

- Метод выбора пользовательских устройств и модель федеративного обучения с туманными вычислениями на основе динамического порога скорости изменения нормы градиента функции потерь, позволяющий одновременно учитывать неоднородность локальных данных и скорость изменения градиента на текущем этапе обучения модели искусственного интеллекта, а также снизить объём сетевого взаимодействия между пользовательскими устройствами и сервером на 39%.

- Децентрализованная структура и модель федеративного обучения с туманными альянсами на основе адаптивной графовой свёрточной рекуррентной нейронной сети для прогнозирования транспортных потоков в сервисах городских сетевых вселенных (CitiVerse), обеспечивающая снижение сетевого взаимодействия до 48% по сравнению с традиционными методами федеративного обучения.

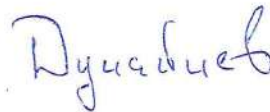
- Структура и метод развёртывания федеративного обучения на основе WebAssembly (низкоуровневого байт-кода) в гетерогенных средах туманных вычислений, обеспечивающие кроссплатформенное выполнение единого модуля обучения нейронной сети на узлах с различной аппаратной архитектурой, снижение потребления оперативной памяти на 69,5% по сравнению со стандартной реализацией при сохранении точности и скорости сходимости модели федеративного искусственного интеллекта.

И.о. декана факультета ИКСС
к.т.н., доцент



В.С. Елагин

Доцент кафедры ССиПД
к.т.н., PhD



Р.А. Дунайцев

Зав. лабораторией кафедры ССиПД



О.И. Ворожейкина