УДК 378.016+004.056.2 Вестник УрФО № 3(61) / 2026, с. 19–29
DOI: 10.14529/secur260302

Частикова В.А., Валеев Т.Н.

СОВРЕМЕННЫЕ МЕТОДЫ ФЕДЕРАТИВНОГО ОБУЧЕНИЯ: СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ И УСТОЙЧИВОСТИ

Chastikova V.A., Valeev T.N.

MODERN METHODS OF FEDERATED LEARNING: COMPARATIVE ANALYSIS OF EFFICIENCY AND ROBUSTNESS

В данной статье рассматриваются методы и алгоритмы в модели федеративного обучения, а также проводится сравнительный анализ при условии наличия во время обучения гетерогенных данных. Особое внимание уделяется таким алгоритмам и их расширениям, как FedAvg, FedProx, SCAFFOLD, FedNova, FedOpt, FLTrust и LoRA-FL. При работе были использованы следующие метрики: точность обучения, скорость сходимости, устойчивость к атакам типа Byzantine и эффективность работы при гетерогенных данных. Результатом работы является выявление преимуществ алгоритмов, а также перспективные направления развития модели федеративного обучения.

Ключевые слова: Федеративное обучение, non-IID данные, гетерогенность, агрегация моделей, Byzantine-атаки, коммуникационная эффективность, гетерогенное обучение.

This article examines methods and algorithms used in the Federated Learning model and provides a comparative analysis under conditions of heterogeneous data during training. Particular attention is paid to algorithms and their extensions, including FedAvg, FedProx, SCAFFOLD, FedNova, FedOpt, FLTrust, and LoRA-FL. The study employs the following evaluation metrics: training accuracy, convergence speed, resistance to Byzantine attacks, and performance efficiency under heterogeneous data conditions. The results of the study identify the advantages of the considered algorithms and outline promising directions for the further development of federated learning models.

Keywords: federated learning, non-IID data, heterogeneity, model aggregation, Byzantine attacks, communication efficiency, heterogeneous learning.

Введение

Примерно в 2016–2017 годах был фактически впервые применение принцип федеративного обучения, данный метод использовался для обучения

нейронных сетей на основе мобильных устройств клиентов, отличительной чертой которой стало выполнение требований конфиденциальности. Важной отличительной чертой данного способа стал централизированный метод при том, что данные не покидали пределов своих клиентов, т.е. сервер получал лишь обновления самой модели (градиенты, веса), после чего происходила агрегация данных обновлений, тем самым изменяя глобальную модель. И соответственно данный процесс проходил за счет множества циклов [1, 2].

Несмотря изначально на безопасность основополагающей идеи в ней все же присутствуют аспекты, которые требуют внимания: статическая гетерогенность (non-IID), системная гетерогенность (различные вычислительные ресурсы), уязвимость к атакам типа Byzantine.

Научная новизна данной статьи заключается в рассмотрении алгоритмов и методов федеративного обучения, а также их сравнение, с последующим выделением проблем и возможных направлений развития данной области.

Базовые алгоритмы и методы федеративного обучения

В основе большинства алгоритмов агрегации лежит одна и та же дилемма: чем проще протокол, тем хуже он справляется с неоднородностью данных, и наоборот. Показательнее всего эта закономерность видна на примере FedAvg – исторически первого и самого распространённого алгоритма, который просто усредняет локальные модели клиентов:

$$w_{t+1} = \sum_k \frac{n_k}{n} w_k,$$

где t – раунды,

w_t – модель для обучения,

k – клиенты.

На независимых и одинаково распределённых (IID) данных такой подход сходится быстро, но стоит данным стать статистически неоднородными – и модель начинает страдать от «клиентского дрейфа», из-за которого точность резко падает [1].

Именно поиск баланса между простотой и устойчивостью к дрейфу породил следующее поколение алгоритмов – FedProx, SCAFFOLD, FedNova и FedOpt. Каждый из них решает эту задачу по-своему: одни ограничивают отклонение весов «мягко», через штраф в функции потерь, другие – «жёстко», через прямую коррекцию направления обновлений или отказ от усреднения весов как такового.

Таблица 1

Оптимизирующие алгоритмы

Алгоритм	Механизм	Формула	Задача	Замечания
FedProx	Проксимальный штраф в целевой функции	$\frac{\mu}{2}\|w - w_t\|^2$, где μ ограничивает отклонение локальных весов от глобального консенсуса	Смягчает клиентский дрейф	Требует подбора μ
SCAFFOLD	Контрольные векторы корректируют направление обновлений	$g_k = \nabla F_k - c_k + c$, где g_k – обновление клиента, ∇F_k – локальный градиент, c_k – контрольный вектор клиента, c – глобальный контрольный вектор.	Заметно повышает точность на non-IID данных	Растут коммуникационные издержки [4]
FedNova	Нормализация обновлений по числу локальных шагов	$w_{t+1} = w_t - \eta \frac{\sum_{i \in S_t} \frac{n_i}{n} * \frac{\Delta_i}{\tau_i}}{\sum_{i \in S_t} \frac{n_i}{n}}$, где η – скорость обучения, S_t – случайная выборка клиентов, n_i – размер данных клиента, τ_i – число локальных шагов, Δ_i – суммарный локальный градиент.	Устраняет смещение модели в сторону устройств с большим числом итераций	Не устраняет статистическую неоднородность как таковую
FedOpt	Разница моделей трактуется как псевдоградиент	$g_t = \sum_{i \in S_t} \frac{n_i}{n}(w_t - w_{ti})$	Позволяет применять продвинутую оптимизацию поверх агрегированного результата [6]	Дополнительная сложность на сервере

Иными словами, если FedProx и SCAFFOLD решают проблему статистического дрейфа, то FedNova устраняет другую его причину – рассинхронизацию локальных вычислений, – а FedOpt меняет саму логику агрегации, отказываясь от прямого усреднения весов.

Отказоустойчивость агрегации

Но даже идеально настроенный алгоритм агрегации остаётся уязвим, если часть клиентов присылает аномальные обновления – по ошибке или намеренно. Классическое взвешенное усреднение никак от этого не защищено, поэтому поверх базовых алгоритмов применяют робастные стратегии – от простого отсечения выбросов до полноценной валидации через доверенный эталон.

Таблица 2

Методы агрегации

Метод	Принцип защиты	Уровень защиты
Coordinate-wise Median	Отсекает экстремальные отклонения по каждому параметру отдельно	Базовый
Trimmed Mean	Отбрасывает маргинальные значения по краям распределения, усредняет остальное	Базовый
Krum / Multi-Krum	Анализирует пространственное расположение обновлений, отбирает те, у которых больше «ближайших соседей»	Средний–высокий
Bulyan	Комбинирует Multi-Krum (фильтрация) и Trimmed Mean (усреднение отфильтрованного)	Максимальный
FLTrust	Валидирует градиенты клиентов через небольшую доверенную выборку на сервере	Максимальный, но принципиально иной подход

Как видно из таблицы, рост защиты почти всегда достигается комбинированием методов (Bulyan) либо принципиально иной логикой (FLTrust) – прямое усреднение к этому моменту уже не спасает [7, 8].

Гетерогенность в федеративном обучении

Проблема дрейфа и уязвимость агрегации – лишь следствия более глубокой особенности федеративного обучения: клиенты изначально неоднородны, причём неоднородны по двум независимым осям – статистической (какие у них данные) и модельной (какие у них ресурсы).

Статистическая гетерогенность изучена лучше всего и встречается чаще: она проявляется как неоднородность меток, неоднородность признаков, дисбаланс количества образцов или концептуальный сдвиг. Удобный способ измерить её степень – параметр Дирихле α: при $\alpha \to 0$ неоднородность становится экстремальной (у каждого клиента данные только одного класса), при

$\alpha \to \infty$ распределение приближается к однородному IID. Часть решений этой проблемы уже знакома – это SCAFFOLD и FedOpt из первого раздела; к ним добавляются персонализированное FL (Per-FedAvg), общий публичный датасет (FedMix) и дистилляция знаний через публичные данные [9].

Модельная гетерогенность возникает по другой причине – из-за разницы в вычислительных ресурсах устройств, из-за которой обучить единую тяжёлую архитектуру на всех клиентах попросту невозможно. Адаптация к этому ограничению происходит на трёх уровнях.

Таблица 3

Уровни архитектуры

Уровень	Что адаптируется	Принцип
Данных	Предобработка	Локальная аугментация под аппаратные лимиты устройства
Модели	Архитектура	Прунинг, масштабирование, дистилляция знаний, раздельное обучение
Сервера	Координация клиентов	Кластеризация по архитектурным профилям, мета-агрегация

Из перечисленных методов дистилляция знаний стоит особняком: на сервер передаются не «сырые» веса или градиенты, а логиты (мягкие метки), полученные на публичном датасете. Это позволяет полностью абстрагироваться от архитектурных различий клиентских моделей и свести потерю точности к минимуму [10].

Масштабирование на сверхбольшие модели

Модельная гетерогенность обостряется многократно, когда речь заходит о по-настоящему больших архитектурах – LLM на клиентских устройствах с ограниченной памятью попросту не помещаются. Это породило отдельный класс решений, снижающих издержки не за счёт адаптации архитектуры, а за счёт сокращения того, что вообще передаётся или обучается.

Таблица 4

Методы для LLM

Подход	Процесс	Эффект
LoRA-FL, FedIT, FlexLoRA	Узлы обновляют только низкоранговые адаптеры	Трафик сокращается в 100–1000 раз без потери качества
Prefix-tuning	Оптимизируются только мягкие промпты, веса модели заморожены	Минимальная нагрузка на клиента

Окончание табл. 4

Split Learning	Слои сети распределяются между клиентом и сервером	Разгружается память конечного устройства
In-context learning	Клиенты формируют только контекстные примеры	Основные вычисления полностью уходят на сервер

Эта же логика – разгрузить клиента, перераспределив нагрузку, – прослеживается и на уровне сетевой топологии, только там перераспределяется не вычислительная, а координационная нагрузка. Иерархическое FL вводит промежуточные граничные серверы (Edge), снижающие задержки и разгружающие центральный координатор. P2P-FL идёт дальше и вовсе отказывается от центрального узла в пользу прямого обмена градиентами между клиентами – это максимально повышает устойчивость к единой точке отказа (SPOF), но заметно усложняет гарантии математической сходимости. Наконец, Federated X Learning выходит за рамки топологии как таковой, объединяя саму парадигму федеративного обучения с трансферным, континуальным обучением и обучением с подкреплением [13].

Материалы и методы

Экспериментальная часть работы проводилась в симуляции федеративного обучения в среде Python, где клиентами выступали независимое процессы на отдельном кластере. В данном случае сервер, где проводилась симуляция, оснащен четырьмя NVIDIA A100 по 40 Гб памяти.

Более того, использовались Python 3.10, PyTorch 2.1, PySyft 0.8, обеспечивающие дифференциальную приватность и безопасные вычисления.

Для эмуляции системы, в которой данные гетерогенны использовалась библиотека FEMNIST, где данные распределены между большим количеством клиентов, но при этом не являются одинаковыми, создавая ту самую гетерогенность. Применялся данная библиотека на основе LEAF для моделей федеративного обучения, позволяющий набор инструментов для исследований в данной области. Для окончательных замеров точности использовался датасет Shakespeare, включающий в себя тексты Шекспира, а также включенный в библиотеку LEAF.

В ходе работы эксперименты проводились по пять раз с начальным seed, который менялся от 0 до 4, тем самым создавая вариативность. А итоговые значения приводились в формате усредненного значение ± отклонение. А также количество раундов в работе использовалось 200, а размер батча 64.

В качестве метрик в работе использовались следующие:

– максимально достигнутая точность
– раунды необходимые для достижения порога в 75% точности
– точность, достигнутая при наличии Byzantine-атак на систему
– точность при отношении гетерогенности в системе.

Сравнительный анализ эффективности методов

На рис. 1 показана динамика точности на тестовой выборке для шести ведущих алгоритмов агрегации в условиях сильного non-IID (параметр Дирихле $\alpha = 0.1$). Как можно заметить все методы начинают примерно одинаково по точности, а различия начинают появляться примерно после 50 раунда.

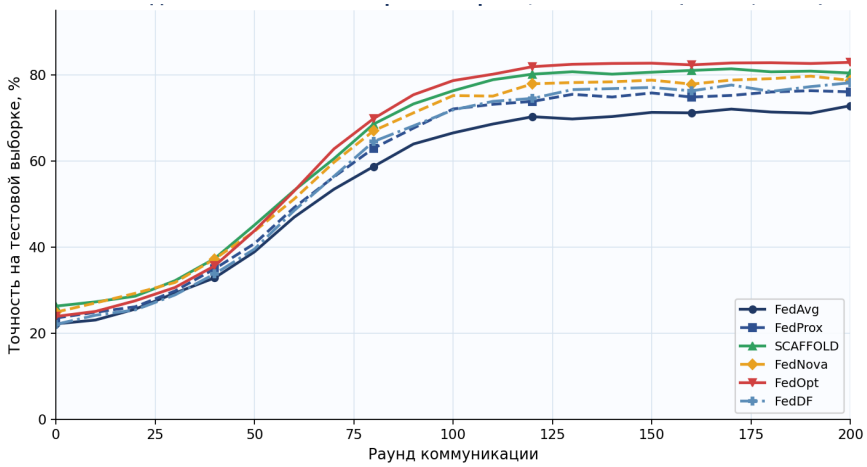


Рис. 1. Динамика точности алгоритмов агрегации FL

На рис. 2 представлен график с осями: горизонтальный обозначает количество раундов для достижения 75% точности, вертикальный обозначает предельную точность обучения на проверочной выборке.

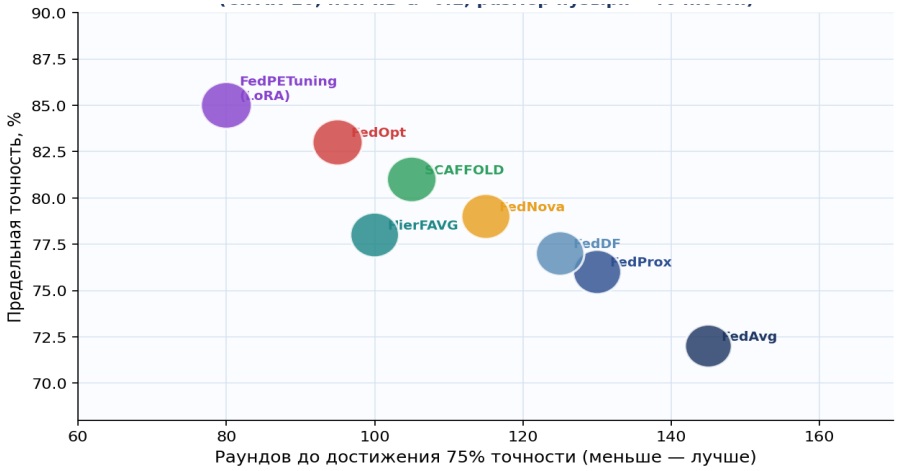


Рис. 2. Компромисс «скорость сходимости – предельная точность»

На нижеследующем рис. 3 представлен график, показывающий степень защищенности от Byzantine-атак (имитации случайных обновлений) при возрастающие доле злонамеренных клиентов от 0% до 30%.

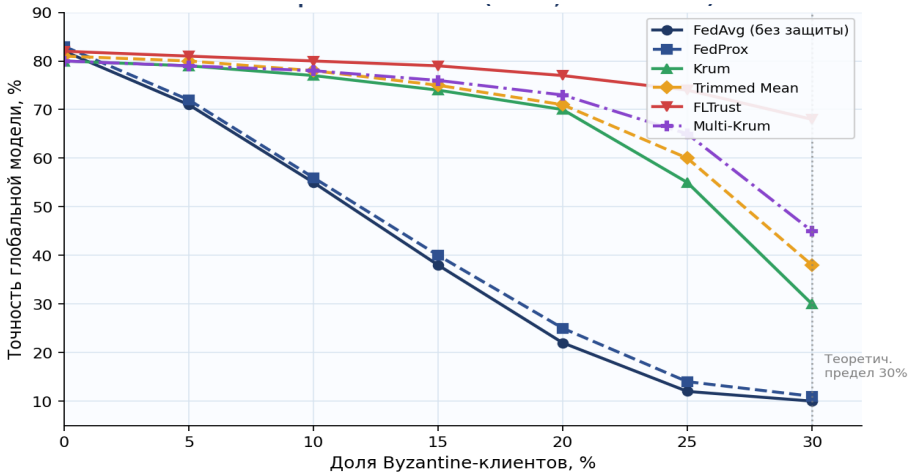


Рис. 3. Точность методов FL при различной доле Byzantine-клиентов

На рис. 4 представлен график точности методов относительно того, как сильно падает эффективность при неоднородных данных во время обучения, где 0,05 означает экстремально неоднородный, а IID – полную однородность.

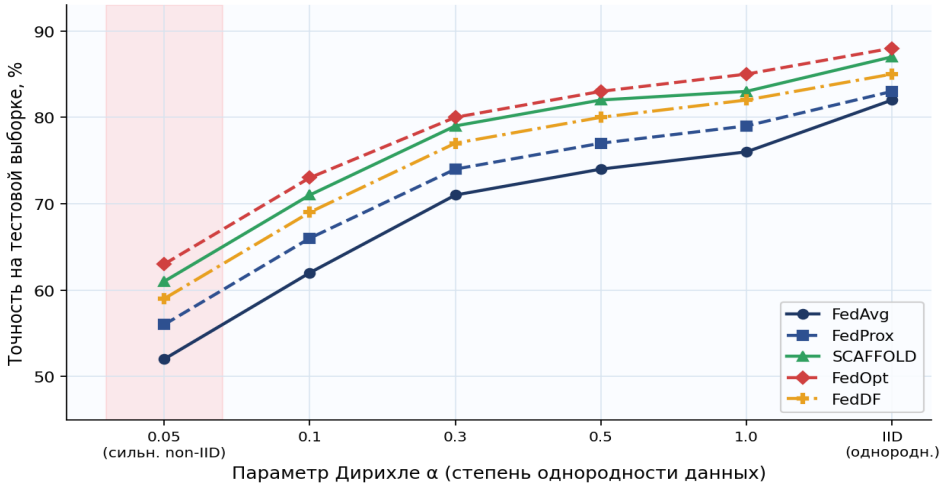


Рис. 4. Точность методов FL в зависимости от степени неоднородности данных

Представленная ниже таблица наглядно показывает эффективность рассмотренных ранее методов по основным метрикам.

Таблица 5

Сводная таблица

Метод	Точность	Раундов до 75%	Доп. вычисления	Byzantine-устойчивость
FedAvg	72.1	145	Базовый	Нет
FedProx	76.3	130	+0.5% (μ-штраф)	Нет
SCAFFOLD	81.2	105	+5% (вариаты)	Частично
FedNova	79.0	115	+1% (норм.)	Нет
FedOpt	83.1	95	+2% (оптим.)	Нет
FedDF	77.4	125	+15% (дист.)	Нет
Krum	78.5	130	+8% (отбор)	Да (при <30%)
FLTrust	82.0	110	+3% (оценка)	Да (при <40%)
LoRA-FL	85.2	80	+20% (PEFT)	Нет
HierFAVG	78.2	100	+10% (иерарх.)	Нет

Заключение

Во-первых, по качественным показателям лучше всего себя показали FedOpt и SCAFFOLD обеспечивая лучшую точность (больше на 9–11%), а также имеют при этом более быструю сходимость (быстрее на 40–50 раундов).

Во-вторых, насущной проблемой для системы все также остаются атаки типа Byzantine, в первую очередь влияя на точность результирующей точности. И среди прочих методов лучше всего себя показал FLTrust, сохранив 68% точности при 30% злонамеренных клиентов. В то время как традиционные методы не справляются уже при 10–15%.

В-третьих, наилучшую скорость обучения показал метод LoRA-FL во много раз превзойдя другие методы, что делает его практически единственным методом для обучения таких моделей как LLM, включающий большое количество параметров.

Подводя итоги, стоит отметить, что из всех рассмотренных методов не было характерного претендента, отвечающий всем требованиям сразу: безопасность от Byzantine-атак, точность, скорость. Что может в свою очередь определять основные направления развития методов в модели федеративного обучения.

Литература / References

1. A Comprehensive Survey on Client Selection Strategies in Federated Learning // Computer Networks. – 2024. – Vol. 242. – P. 109–124.

2. A Contemporary Survey of Recent Advances in Federated Learning // Computer Science Review. – 2024. – Vol. 51. – URL: https://www.sciencedirect.com/science/article/pii/S2542660524001926 (дата обращения: 23.04.2026).

3. A Survey of Security Strategies in Federated Learning // MDPI Sensors. – 2024. – Vol. 24, № 5. – URL: https://www.mdpi.com/2998474 (дата обращения: 23.04.2026).

4. A Survey on Efficient Federated Learning Methods for Foundation Model Training // International Joint Conference on Artificial Intelligence (IJCAI). – 2024. – URL: https://www.ijcai.org/proceedings/2024/0919 (дата обращения: 23.04.2026).

5. A Systematic Review of Federated Generative Models // arXiv preprint arXiv:2405.16682. – 2024.

6. A Systematic Review of Federated Learning: Aggregation Methods // Journal of Network and Computer Applications. – 2023. – Vol. 218. – URL: https://www.sciencedirect.com/science/article/pii/S1084804523001339 (дата обращения: 23.04.2026).

7. Cao, X. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping / X. Cao [et al.] // Network and Distributed System Security Symposium (NDSS). – 2022.

8. Emerging Trends in Federated Learning: From Model Fusion to Federated X Learning // International Journal of Machine Learning and Cybernetics. – 2024. – Vol. 15. – P. 451–468.

9. Federated Learning: A Cutting-Edge Survey of the Latest Advancements // Computer Communications. – 2024. – Vol. 215. – P. 112–125.

10. Federated Learning: Overview, Strategies, Applications and Tools // IEEE Access / PMC. – 2024. – Vol. 12. – URL: https://pmc.ncbi.nlm.nih.gov/ articles/PMC11466570/ (дата обращения: 23.04.2026).

11. Li, T. Federated Optimization in Heterogeneous Networks (FedProx) / T. Li [et al.] // Proceedings of machine learning and systems (MLSys). – 2020. – Vol. 2. – P. 429–450.

12. McMahan, H. B. Communication-Efficient Learning of Deep Networks from Decentralized Data / H. B. McMahan [et al.] // Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS). – 2017. – P. 1273–1282.

13. Recent Advancements in Federated Learning: State of the Art // MDPI Electronics. – 2024. – Vol. 13, № 8. – URL: https://www.mdpi.com/3035012 (дата обращения: 23.04.2026).

ЧАСТИКОВА Вера Аркадьевна, кандидат технических наук, доцент кафедры кибербезопасности и защиты информации федерального государственного бюджетного образовательного учреждения высшего образования «Кубанский государственный технологический университет». 350072, Краснодарский край, г. Краснодар, ул. Московская, д. 2. E-mail: chastikova_va@mail.ru

ВАЛЕЕВ Тамерлан Назирович, магистр кафедры кибербезопасности и защиты информации федерального государственного бюджетного образовательного учреждения высшего образования «Кубанский государственный технологический университет». 350072, Краснодарский край, г. Краснодар, ул. Московская, д. 2. E-mail: tamer.valeev2014@yandex.ru

CHASTIKOVA Vera Arkadyevna, Candidate of Technical Sciences, Associate Professor of the Department of Cybersecurity and Information Protection at the Federal State Budgetary Educational Institution of Higher Education “Kuban State Technological University”. 350072, Krasnodar Krai, Krasnodar, Moskovskaya St., 2. E-mail: chastikova_va@mail.ru

VALEEV Tamerlan Nazirovich, Master’s Degree Student of the Department of Cybersecurity and Information Protection at the Federal State Budgetary Educational Institution of Higher Education “Kuban State Technological University”. 350072, Krasnodar Krai, Krasnodar, Moskovskaya St., 2. E-mail: tamer.valeev2014@yandex.ru