Реализация алгоритмов федеративного обучения для ненезависимых и неоднородно распределённых данных

Кинда Мрейш
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
Санкт-Петербург, Россия
Алеппо Университет
Алеппо, Сирия
kinda-mresh@hotmail.com

Салар Сали
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
Санкт-Петербург, Россия
salar.sali97@gmail.com

Иван И. Холод
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
Санкт-Петербург, Россия
iiholod@etu.ru

Яссер Низамли
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
Санкт-Петербург, Россия
yanizamli@stud.etu.ru

Тарек Альнажар
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
Санкт-Петербург, Россия
tariq.najjar.7@yandex.com

***Аннотация*. Федеративное обучение (FL) стало трансформационным подходом для обучения моделей машинного обучения на децентрализованных источниках данных при сохранении конфиденциальности. В данном исследовании оценивается эффективность двух известных стратегий FL, Federated Averaging (FedAvg) и Federated Batch Normalization (FedBN), на двух различных наборах данных: NF-UNSW-NB15, предназначенном для обнаружения сетевых вторжений, и Dumpers, содержащем данные сенсоров коммерческих транспортных средств. Исследуется влияние не-независимых и неидентично распределённых (non-IID) искажений данных, включая сдвиги распределения признаков, сдвиги распределения меток и изменения в соотношениях между признаками и метками. Для набора данных NF-UNSW-NB15 FedBN достиг точности тестирования 0.9999 в сценариях с одинаковыми метками и разными признаками, тогда как FedAvg продемонстрировал приблизительно такой же результат при различных искажениях. На наборе данных Dumpers FedBN значительно превзошёл FedAvg, достигнув точности тестирования 0.8156 для смещения распределения признаков, в то время как FedAvg показал результат 0.7473. Эти результаты подчёркивают превосходную способность FedBN справляться с гетерогенными распределениями данных, что делает его надёжным решением для проблем non-IID.**

Ключевые слова: алгоритмы федеративного обучения, распределение данных non-IID, фреймворк Flower, модель нейронной сети, классификация, набор данных Dumpers, набор данных NF-UNSW-NB15, Federated Averaging, Federated Batch Normalization

I. Введение

Федеративное обучение (FL) – это децентрализованный подход к машинному обучению, который позволяет нескольким участникам обучать общую модель, сохраняя данные локальными и конфиденциальными [1]. В отличие от традиционных методов, где данные централизованы, FL гарантирует, что данные остаются на отдельных устройствах или в учреждениях. Участники обучают модели локально и отправляют только обновления, такие как веса или градиенты, на центральный сервер, который агрегирует их для улучшения глобальной модели без доступа к исходным данным [2]. Эта техника защиты конфиденциальности снижает риски утечек данных, что делает FL подходящим для таких чувствительных областей, как здравоохранение, финансы и мобильные приложения [3]. Она повышает конфиденциальность, снижает затраты на передачу данных и соответствует нормативным требованиям по защите данных, таким как Общий регламент защиты данных (GDPR) и Закон о переносимости и подотчетности медицинского страхования (HIPAA) [2]. FL позволяет проводить совместное обучение на распределенных наборах данных без агрегации частной информации, что делает его идеальным для Интернета вещей, автономных транспортных средств и персонализированного искусственного интеллекта [3].

В данной работе мы сравниваем две различные стратегии федеративного обучения, используя два

разных набора данных. Каждый набор данных разделен на четыре партии с использованием четырех различных алгоритмов разбиения non-IID данных для оценки влияния распределений данных на производительность модели.

II. Связанная работа

В федеративном обучении проблема non-IID данных среди клиентов значительно влияет на производительность модели. Две заметные стратегии, решающие эту проблему, — это Federated Averaging (FedAvg) и Federated Batch Normalization (FedBN). FedAvg, предложенная McMahan и др. (2017) [1], агрегирует локально обученные модели, усредняя их веса. Хотя эта стратегия эффективна в сценариях IID, её производительность ухудшается при non-IID данных из-за специфических распределений данных на клиентских устройствах, что приводит к расходимости модели. Эмпирические исследования показали, что FedAvg испытывает трудности с сходимостью в условиях высокой гетерогенности данных, часто приводя к снижению точности модели по сравнению с централизованным обучением [1]. Для смягчения этого Li и др. (2021) [4] предложили FedBN, которая сохраняет локальные слои нормализации батчей во время агрегации. Этот подход решает проблемы сдвига признаков в non-IID сценариях, таких как различные медицинские устройства для визуализации или разнообразные условия автономного вождения, позволяя клиентам адаптироваться к своим специфическим распределениям данных. Экспериментальные результаты показали, что FedBN превосходит как FedAvg, так и FedProx в этих контекстах. В частности, FedBN показала на 10–15 % более высокую точность в условиях non-IID, а также более быструю скорость сходимости, так как она смягчает негативное воздействие сдвигов признаков [4]. Mhaisen и др. (2020) проанализировали влияние иерархического федеративного обучения (HFL) на распределения non-IID данных, сосредоточив внимание на оптимальных стратегиях распределения нагрузки на устройства для уменьшения деградации производительности. Их результаты показали, что использование агрегации на уровне устройств перед глобальными обновлениями улучшило сходимость модели и снизило затраты на коммуникацию. Было сообщено, что в условиях сильно искажённых данных иерархическая агрегация повысила точность до 10 % по сравнению со стандартными подходами FL при меньшей задержке на каждый раунд коммуникации [5].

В нашей работе мы сосредотачиваемся на оценке точности модели нейронной сети в области федеративного обучения на двух различных наборах данных, один из которых собран с нескольких самосвалов, используемых в строительных работах, а второй предназначен для системы обнаружения сетевых вторжений (NIDS), чтобы создать надёжную систему машинного обучения для обеих задач.

III. Подход

В этом разделе рассматриваются типы сдвигов данных non-IID с объяснением двух алгоритмов FL, которые используются в нашей работе.

A. *Сдвиги данных non-IID в федеративном обучении*

- Как упоминалось в начале работы, мы использовали четыре различных метода для разделения набора данных на четыре части, основываясь на следующих сдвигах данных non-IID [3]:
- Сдвиг распределения признаков: Хотя условное распределение $\mathcal{P}(y|x)$ остаётся одинаковым для всех клиентов ($\mathcal{P}i(y|x) = \mathcal{P}j(y|x)$ для всех клиентов i и j), могут возникать различия в маргинальных распределениях $\mathcal{P}i(y|x)$ среди клиентов.
- Сдвиг распределения меток: Распределение меток $\mathcal{P}(y)$ может различаться среди клиентов, даже если условное распределение признаков $\mathcal{P}(y|x)$ остаётся неизменным.
- Одни и те же признаки, разные метки: Хотя маргинальное распределение $\mathcal{P}(x)$ общее для всех клиентов, их условные распределения $\mathcal{P}i(y|x)$ могут различаться.
- Одинаковые метки, разные признаки: Хотя маргинальное распределение $\mathcal{P}(y)$ общее для всех клиентов, их условные распределения $\mathcal{P}i(x|y)$ могут различаться.

Процесс интеграции моделей от различных клиентов является основным вызовом в федеративном обучении. Главная сложность заключается в определении оптимального набора параметров для глобальной модели с учётом вклада нескольких клиентов. Математически эта проблема обычно выражается как (1):

$$\min_{w} F(w), \text{ where } F(w) := \sum_{k=1}^{m} p_k F_k(w) \quad (1)$$

где m представляет общее количество клиентов, p_k это неотрицательный вес, такой что $\sum_k p_k = 1$, а F_k обозначает локальную целевую функцию для клиента k. Термин p_k определяет влияние модели каждого клиента на итоговую агрегированную глобальную модель [6]. Используя фреймворк Flower, мы реализовали различные стратегии федеративного обучения для обработки распределений данных non-IID на наборах данных Dumpers и NF-UNSW-NB15.

B. *Федеративное усреднение (FedAvg)*

Федеративное усреднение служит основным и широко используемым методом агрегации. Процесс обучения происходит в итерационных раундах. В начале каждого цикла центральный сервер распределяет последнюю глобальную модель среди случайно выбранного подмножества из mm клиентов из общего числа K клиентов. Эти выбранные клиенты уточняют параметры модели локально, используя стохастический градиентный спуск (SGD) для минимизации функции потерь F_k на своих соответствующих обучающих данных. После завершения локальных обновлений, откорректированные параметры отправляются обратно на сервер, где они объединяются с использованием метода взвешенного усреднения. Агрегированные параметры определяют глобальную модель для

следующей итерации [6]. Рис. 1 ниже представляет псевдокод для алгоритма федеративного усреднения.

Algorithm FedAvg

1: function ClientUpdate Run on local nodes
2: **while** $iter < max\_iter$ **do**
3: //Received initialize model M^{init} from aggregation node
4: $M_i^{iter} = M^{init}$
5: **for** batch $b \in B$ **do**
6: $M(w)_i^{iter} \triangleq M(w)_i^{iter} - \eta \nabla l(w; b)$
7: **end for**
8: $\{M_i^{iter}, n_i\}$ to the aggregation node
9: **if** ConvergenceCheck **then**
10: break
11: **end if**
12: $iter = iter + 1$
13: **end while**
14:
15: function ServerAggregation Run on aggregation node
16: **for** each participant $i \in P$ **do**
17: Received $\{M_i^{iter}, n_i\}$ from local nodes
18: **end for**
19: $M(w)^{init} \triangleq \sum_{i=1}^{P} \frac{n_i}{N} M(w)_i^{iter}$
20: Send M^{init} to local nodes

Рис. 1. Псевдокод алгоритма федеративного усреднения

C. Федеративное усреднение с локальной нормализацией по батчам (FedBN)

Нормализация по батчам (Batch Normalization, BN) – это метод, используемый для стабилизации и ускорения обучения глубоких нейронных сетей. Он нормализует активации каждого слоя, используя среднее значение и дисперсию, вычисленные по мини-пакету данных во время обучения [7]. Параметры BN:

- Среднее (μ): Среднее значение активаций в мини-пакете.
- Дисперсия (σ^2): Дисперсия активаций в мини-пакете.
- Масштаб (γ) и Смещение (β): Обучаемые параметры, которые масштабируют и смещают нормализованные активации.

В федеративном обучении параметры BN (μ и σ^2) вычисляются локально на данных каждого клиента. Агрегация этих параметров между клиентами может привести к снижению производительности, так как распределения данных могут значительно различаться (данные non-IID) [4].

FedBN изменяет процесс агрегации, исключая параметры BN из шага усреднения, при этом агрегируются другие параметры модели [4]. Локальные обновления нормализации по батчам следуют по формуле (2):

$$\mu_k = \frac{1}{B}\sum_{i=1}^{B} x_{i,k}, \quad \sigma_k^2 = \frac{1}{B}\sum_{i=1}^{B}(x_{i,k} - \mu_k)^2 \tag{2}$$

где: μ_k и σ_k^2 — среднее и дисперсия батча на клиенте K; B — размер батча; $x_{i,k}$ — представляет локальные выборки признаков.

Вместо того чтобы усреднять эти статистики глобально, FedBN сохраняет их локально, обеспечивая, чтобы каждый клиент сохранял свои собственные параметры нормализации (3):

$$w_{\text{global}}^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{N} w_k^{(t)}, \tag{3}$$

где γ_k и β_k — параметры масштаба и смещения BN для клиента k [4]. Рис. 2 представляет псевдокод для FedBN.

Algorithm Federated Learning using FedBN

Notations: The user indexed by k, neural network layer indexed by l, initialized model parameters: $w_{0,k}^{(l)}$, local update pace: E, and total optimization round T.

1: **for** each round $t = 1, 2, \ldots, T$ **do**
2: **for** each user k and each layer l **do**
3: $w_{t+1,k}^{(l)} \leftarrow SGD(w_{t,k}^{(l)})$
4: **end for**
5: **if** $\text{mod}(t, E) = 0$ **then**
6: **for** each user k and each layer l **do**
7: **if** layer l is not BatchNorm **then**
8: $w_{t+1,k}^{(l)} \leftarrow \frac{1}{K}\sum_{k=1}^{K} w_{t+1,k}^{(l)}$
9: **end if**
10: **end for**
11: **end if**
12: **end for**

Рис. 2. Псевдокод алгоритма федеративной локальной нормализации по батчам

IV. Эксперименты

В этом разделе представлены экспериментальная настройка и цели анализа наборов данных Dumpers и NF-UNSW-NB15. Наше исследование сосредоточено на классификации видов деятельности самосвалов и потоков данных между источниками и пунктами назначения в сети с использованием федеративного обучения при различных распределениях данных non-IID. Кроме того, мы исследуем оптимальные конфигурации гиперпараметров для нейронной сети в разных экспериментальных условиях. Для проведения этих экспериментов мы используем сервер Flower и клиентские реализации Flower, применяя две стратегии в рамках фреймворка Flower. Эксперименты выполняются на облачных виртуальных машинах со следующими характеристиками оборудования: процессор Intel Xeon Ice Lake с тактовой частотой 2,0 ГГц, оснащенный 2 физическими ядрами и 4 логическими потоками, кэш-памятью 24,25 МБ, 16 ГБ оперативной памяти и жестким диском объемом 300 ГБ.

A. Наборы данных

В этом разделе мы рассмотрим наборы данных, используемые в нашем исследовании.

1) Набор данных Dumpers

The Набор данных о сенсорных показателях коммерческих автомобилей, предоставленный компанией Smartilizer Scandinavia AB, содержит числовые данные, собранные с двух самосвалов, работающих на объекте рекультивации грунта недалеко от Гётеборга. Этот набор данных ценен для анализа эксплуатационной динамики коммерческих автомобилей в строительных условиях, предоставляя информацию о поведении и производительности транспортных средств в реальных условиях. Набор данных включает измерения (признаки), такие как временная метка, скорость, показания гироскопа и акселерометра [8]. В общей сложности он содержит 1 699 983 точки данных с пятью метками: Холостой ход (Idle), Движение (Driving), Загрузка (Loading), Разгрузка (Dumping) и Двигатель выключен (Engine-off).

2) Набор данных NF-UNSW-NB15

Формат набора данных UNSW-NB15 на основе NetFlow, названный NF-UNSW-NB15, был разработан и

размечен в соответствии с соответствующими категориями атак. Этот набор данных предназначен для использования в системах обнаружения сетевых вторжений (NIDS) на основе машинного обучения. Общий объем потоков данных составляет 1 623 118 записей. Он включает 14 признаков (числовых и категориальных типов данных) и 10 различных подкатегорий сетевых потоков, для классификации которых мы разрабатываем распределённую нейронную сеть. Образцы данных классифицируются в десять подкатегорий: Benign (безопасный трафик), Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms [9].

B. Параметры модели нейронной сети

Мы построили модель нейронной сети, которая использует различные функции активации (tanh, softsign, ReLU и т.д.), а также несколько оптимизаторов (Adam, Adadelta и SGD) с экспоненциальным уменьшением скорости обучения. Кроме того, мы использовали различные значения для dropout между скрытыми слоями, а для обучения выбрали несколько значений размера пакета и количество эпох (локальных и глобальных), чтобы достичь наилучшей точности для наших задач.

C. Результаты

Эксперимент проводится с использованием двух различных наборов данных: Dumpers и NF-UNSW-NB15, которые оба фокусируются на задаче классификации. Он структурирован в четыре основных сценария, основанных на распределении данных между четырьмя клиентами: Feature Distribution Skew (FDS), Label Distribution Skew (LDS), Same Label Different Features (SLDF) и Same Feature Different Labels (SFDL). Для каждого типа распределения данных non-IID мы реализовали два различных алгоритма из фреймворка Flower: FedAvg и FedBN.

1) Сценарии для набора данных NF-UNSW-NB15

- Сценарии алгоритма Federated Averaging

Для изменений в распределении признаков и меток наибольшая точность теста, равная 0.99, была достигнута при 4 эпохах на клиенте и 2 эпохах на сервере. Для сценария «одни и те же признаки, разные метки» лучшая точность 0.99 была получена при 3 эпохах на клиенте и 2 эпохах на сервере, а для сценария «одни и те же метки, разные признаки» оптимальная точность 0.99 была достигнута при 8 эпохах на клиенте и 2 эпохах на сервере. Все модели нейронных сетей использовали коэффициент Dropout 0.3 и оптимизатор Adadelta. Рис. 3, 4, 5, 6 ниже иллюстрируют изменения non-IID для набора данных NF-UNSW-NB15 с использованием алгоритма FedAvg.

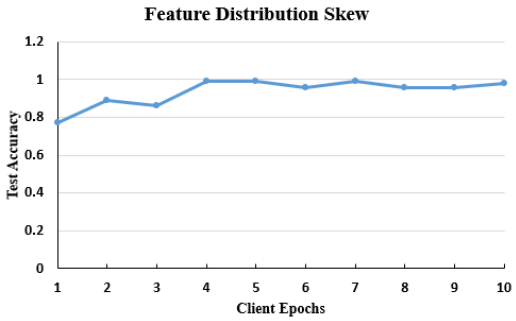

Рис. 3. Результаты точности теста для изменений в распределении признаков с использованием алгоритма FedAvg на наборе данных NF-UNSW-NB15

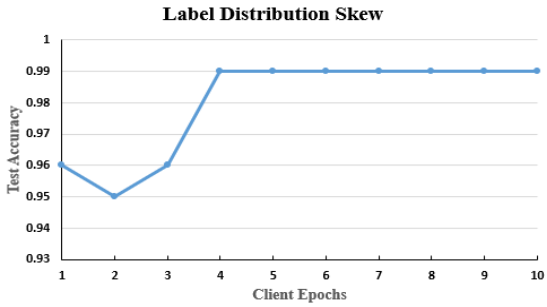

Рис. 4. Результаты точности теста для изменений в распределении меток с использованием алгоритма FedAvg на наборе данных NF-UNSW-NB15

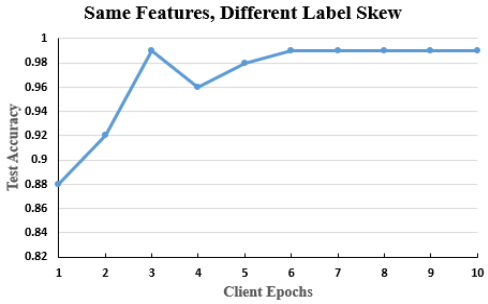

Рис. 5. Результаты точности теста для изменений в распределении SFDL с использованием алгоритма FedAvg на наборе данных NF-UNSW-NB15

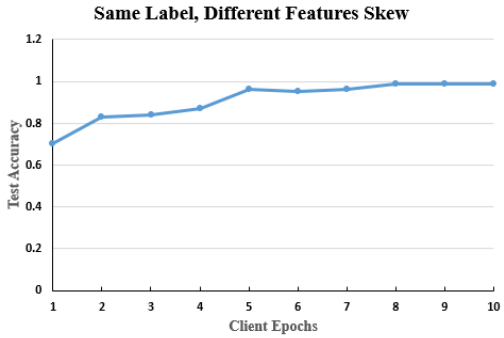

Рис. 6. Результаты точности теста для изменений в распределении SLDF с использованием алгоритма FedAvg на наборе данных NF-UNSW-NB15

- Сценарии алгоритма Federated Batch Normalization

Для изменений в распределении признаков наивысшая точность теста 0.9998 была достигнута при 7 эпохах на клиенте, 10 эпохах на сервере, оптимизаторе Adam и коэффициенте Dropout 0.1. Для изменений в распределении меток лучшая точность 0.8194 была получена при 10 эпохах на клиенте, 10 эпохах на сервере, оптимизаторе Adadelta и отсутствии dropout между скрытыми слоями. Для сценария «одни и те же признаки, разные метки» оптимальная точность 0.9964 была достигнута при 3 эпохах на клиенте, 10 эпохах на сервере, оптимизаторе Adam и коэффициенте Dropout 0.2. Для сценария «одни и те же метки, разные признаки» наилучшая точность 0.9999 была получена при 7 эпохах на клиенте, 10 эпохах на сервере, оптимизаторе SGD и коэффициенте Dropout 0.5 или 0.4. Рис. 7, 8, 9, 10 ниже показывают все изменения non-IID для набора данных NF-UNSW-NB15 и алгоритма FedBN.

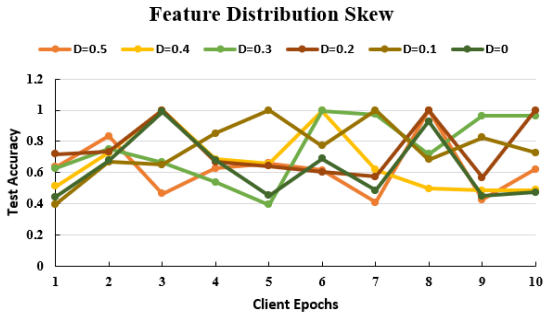

Рис. 7. Результаты точности теста для изменений в распределении признаков с использованием алгоритма FedBN на наборе данных NF-UNSW-NB15

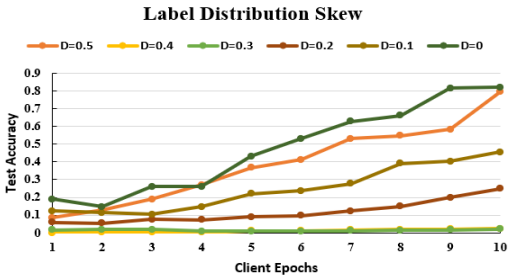

Рис. 8. Результаты точности теста для изменений в распределении меток с использованием алгоритма FedBN на наборе данных NF-UNSW-NB15

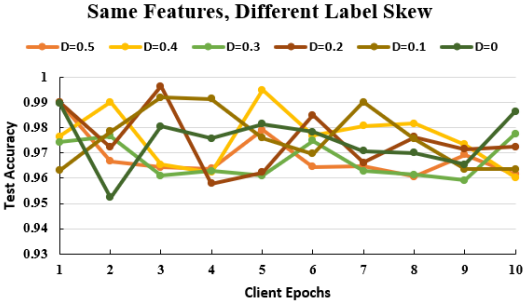

Рис. 9. Результаты точности теста для изменений в распределении SFDL с использованием алгоритма FedBN на наборе данных NF-UNSW-NB15

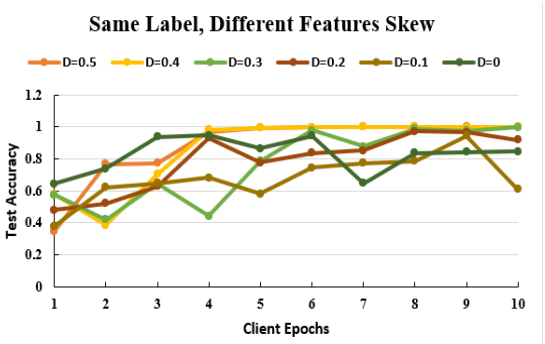

Рис. 10. Результаты точности теста для изменений в распределении SLDF с использованием алгоритма FedBN на наборе данных NF-UNSW-NB15

2) Случаи с набором данных Dumpers

- Случаи с Federated Averaging

Для набора данных Dumpers, используя алгоритм FedAvg, наибольшая тестовая точность для искажения распределения признаков составила 0.5879, достигнута при 10 эпохах для клиентов и серверов, оптимизаторе Adam и коэффициенте Dropout 0.4. В случае искажения распределения меток наилучшая точность составила 0.3974 при 1 эпохе для клиента, 10 эпохах для сервера, оптимизаторе Adadelta и коэффициенте Dropout 0.2. Для искажения с одинаковыми признаками и различными метками мы получили тестовую точность 0.7473, используя 3 эпохи для клиентов, 10 эпох для сервера, оптимизатор SGD и отсутствие Dropout между скрытыми слоями. Наконец, для искажения с одинаковыми метками и различными признаками наибольшая точность составила 0.4176 при 1 эпохе для клиента, 10 эпохах для сервера, оптимизаторе Adam и коэффициенте Dropout 0.5. Рис. 11, 12, 13, 14 ниже иллюстрируют искажения non-IID для набора данных Dumpers с FedAvg.

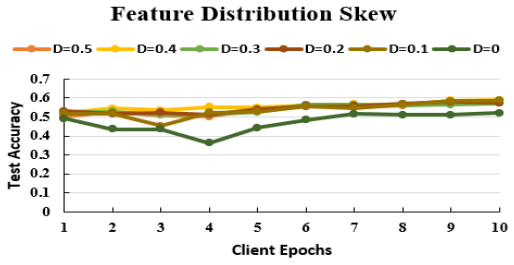

Рис. 11. Результаты тестовой точности для искажения распределения признаков с использованием алгоритма FedAvg на наборе данных Dumpers

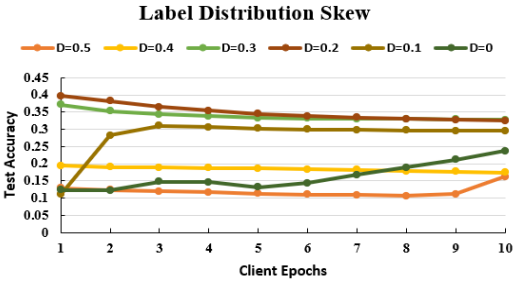

Рис. 12. Результаты тестовой точности для искажения распределения меток с использованием алгоритма FedAvg на наборе данных Dumpers

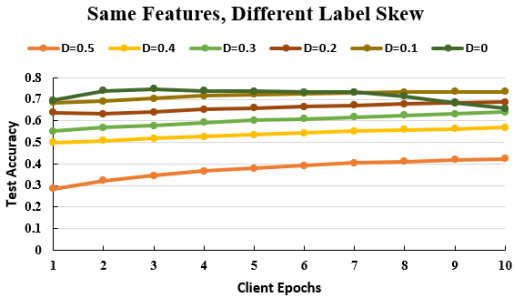

Рис. 13. Результаты тестовой точности для искажения SFDL с использованием алгоритма FedAvg на наборе данных Dumpers

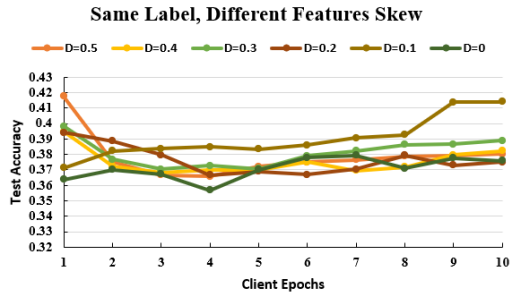

Рис. 14. Результаты тестовой точности для искажения SLDF с использованием алгоритма FedAvg на наборе данных Dumpers

- Случаи с использованием Federated Batch Normalization

Для набора данных Dumpers, использующего алгоритм FedBN, наибольшая точность теста для смещения распределения признаков составила 0,8156, что было достигнуто при 8 эпохах для клиентов, 10 эпохах для сервера, оптимизаторе Adam и коэффициенте Dropout 0,2. Для смещения распределения меток лучшая точность составила 0,3703 при 7 эпохах для клиентов, 10 эпохах для сервера, оптимизаторе Adadelta и коэффициенте Dropout 0,5. В случае смещения одинаковых признаков, разных меток, мы получили точность теста 0,7668, используя 10 эпох для клиентов и сервера, оптимизатор Adam и коэффициент Dropout 0,5. Наконец, для смещения одинаковых меток, разных признаков, наивысшая точность составила 0,4742 при 6 эпохах для клиентов, 10 эпохах для сервера, оптимизаторе Adam и коэффициенте Dropout 0,3. Рис. 15, 16, 17, 18 ниже иллюстрируют не-IID смещения для набора данных Dumpers с FedBN.

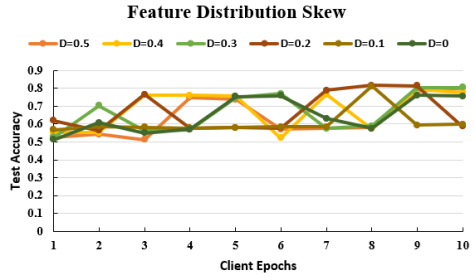

Рис. 15. Результаты точности тестирования при смещении распределения признаков с алгоритмом FedBN на наборе данных Dumpers

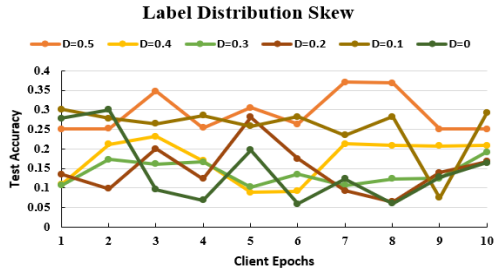

Рис. 16. Результаты точности тестирования при смещении распределения меток с алгоритмом FedBN на наборе данных Dumpers

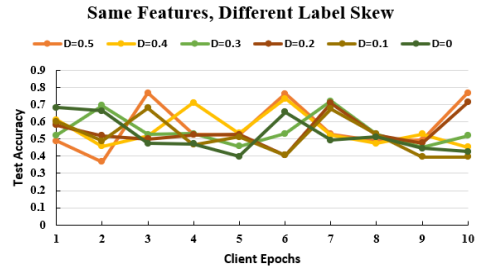

Рис. 17. Результаты точности тестирования при смещении SFDL с алгоритмом FedBN на наборе данных Dumpers

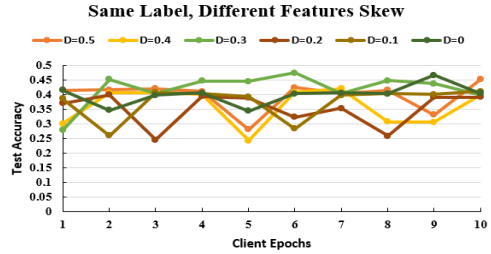

Рис. 18. Результаты точности тестирования при смещении SLDF с алгоритмом FedBN на наборе данных Dumpers

В итоге, для набора данных NF-UNSW-NB15 мы часто наблюдаем схожие результаты по точности тестирования для алгоритмов FedAvg и FedBN при различных конфигурациях эпох клиента, эпох сервера, оптимизаторов и коэффициентов dropout. В то время как для набора данных Dumpers алгоритм FedBN демонстрирует лучшие результаты по сравнению с FedAvg, особенно в сценариях с смещением распределения признаков и одинаковыми признаками с разными метками, где достигается значительно более высокая точность. Табл. 1 ниже показывает наилучшую точность тестирования для каждого набора данных и каждой стратегии Federated Learning.

ТАБЛИЦА I. ЛУЧШАЯ ТОЧНОСТЬ ТЕСТИРОВАНИЯ ДЛЯ КАЖДОГО НАБОРА ДАННЫХ

Набор данных NF-UNSW-NB15		**Набор данных Dumpers**	
Стратегия FL	*Точность тестирования*	*Стратегия FL*	*Точность тестирования*
FedAvg	0.9999	FedAvg	0.7473
FedBN	0.9999	FedBN	0.8156

V. Заключение

В этом исследовании сравнивались алгоритмы Federated Averaging (FedAvg) и Federated Batch Normalization (FedBN) на наборах данных NF-UNSW-NB15 и Dumpers в условиях различных распределений данных, не являющихся IID. Для NF-UNSW-NB15 оба алгоритма достигли высокой точности, при этом FedBN немного превзошел FedAvg в сценариях, таких как сдвиг распределения признаков. На наборе данных Dumpers FedBN продемонстрировал более высокую производительность, особенно при обработке сдвига распределения признаков и одинаковых признаков с разными метками, достигая значительно более высокой точности по сравнению с FedAvg. Эти результаты подчеркивают эффективность FedBN в адаптации к гетерогенным данным. В целом, выбор стратегии FL должен учитывать характеристики распределения данных, при этом FedBN оказался надежным решением для задач с данными, не являющимися IID. Эта работа предоставляет ценные идеи для улучшения федеративного обучения в чувствительных и распределенных приложениях.

Список литературы

[1] McMahan H.B. et al. Communication-Efficient Learning of Deep Networks from Decentralized Data: arXiv:1602.05629. arXiv, 2023.

[2] Yang Q. et al. Federated Machine Learning: Concept and Applications // ACM Trans. Intell. Syst. Technol. 2019. Vol. 10, № 2. P. 1–19.

[3] Kairouz P. et al. Advances and Open Problems in Federated Learning: arXiv:1912.04977. arXiv, 2021.

[4] Li X. et al. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization: arXiv:2102.07623. arXiv, 2021.

[5] Mhaisen N. et al. Analysis and Optimal Edge Assignment For Hierarchical Federated Learning on Non-IID Data // IEEE Trans. Netw. Sci. Eng. 2022. Vol. 9, № 1. P. 55–66.

[6] Li T. et al. Federated Optimization in Heterogeneous Networks: arXiv:1812.06127. arXiv, 2020.

[7] Batch Normalization: Theory and TensorFlow Implementation [Electronic resource]. URL: https://www.datacamp.com/tutorial/batch-normalization-tensorflow (accessed: 01.03.2025).

[8] Commercial Vehicles Sensor Data Set [Electronic resource]. URL: https://www.kaggle.com/datasets/smartilizer/commercial-vehicles-sensor-data-set (accessed: 01.03.2025).

[9] Queensland T.U. of et al. ML-Based NIDS Datasets [Electronic resource] // School of Information Technology and Electrical Engineering. URL: https://www.itee.uq.edu.au/research/cyber-security/research-areas (accessed: 19.01.2025).