electronics

Article

FedCon: Scalable and Efficient Federated Learning via Contribution-Based Aggregation

Wenyu Gao, Gaochao Xu and Xianqiu Meng *

College of Computer Science and Technology, Jilin University, Changchun 130012, China; gaowy22@mails.jlu.edu.cn (W.G.); xugc@jlu.edu.cn (G.X.)
* Correspondence: mengxq23@mails.jlu.edu.cn

Abstract: With the increasing application of federated learning to medical and image data, the challenges of class distribution imbalances and Non-IID heterogeneity across clients have become critical factors affecting the generalization ability of global models. In the medical domain, the phenomenon of data silos is particularly pronounced, leading to significant differences in data distributions across hospitals, which in turn hinder the performance of global model training. To address these challenges, this paper proposes FedCon, a federated learning method capable of dynamically adjusting aggregation weights, while accurately evaluating client contributions. Specifically, FedCon initializes aggregation weights based on client data volume and class distribution and employs Monte Carlo sampling to effectively simplify the computation of Shapley values. Subsequently, it further optimizes the aggregation weights by comprehensively considering the historical contributions of clients and the similarity between clients and the global model. This approach significantly enhances the ability to generalize and update the stability of the global model. Experimental results demonstrate that, compared to existing methods, FedCon achieved a superior generalization performance on public datasets and significantly accelerated the convergence of the global model.

Keywords: federated learning; importance weight; Shapley value; Monte Carlo sampling; client contribution evaluation

check for updates

Academic Editor: Antoni Morell

Received: 4 February 2025
Revised: 28 February 2025
Accepted: 1 March 2025
Published: 4 March 2025

Citation: Gao, W.; Xu, G.; Meng, X. FedCon: Scalable and Efficient Federated Learning via Contribution-Based Aggregation. *Electronics* **2025**, *14*, 1024. https://doi.org/10.3390/electronics14051024

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, with the growing demand for privacy protection and data security [1], federated learning (FL) has garnered significant attention as a distributed machine learning paradigm. By offloading model training to client devices and enabling localized data processing, FL effectively safeguards privacy and ensures data security. In an era characterized by highly distributed data and an increasing emphasis on privacy, the importance of FL has become increasingly significant. It has demonstrated vast potential in practical applications across various fields, such as enhancing disease diagnosis models through distributed training in healthcare [2], fraud detection and credit scoring in financial services [3], optimizing voice assistants and personalized recommendation systems on smart devices [4], and enabling data collaboration and decision-making in the Internet of Things (IoT) [5–7]. In addition, the integration of mobile edge computing and FL has been used to address the issues of computational efficiency and latency in real-time applications [8]. However, FL faces significant challenges in practice, especially due to the non-independent and identically distributed (Non-IID) [9–11] nature of data. The substantial heterogeneity in data characteristics across clients often leads to degraded performance and instability in global model aggregation. Effectively addressing these challenges is critical to enhancing the practicality and generalizability of FL.

Although FL offers significant advantages in privacy protection and distributed computation, its practical applications still face several key challenges, including data heterogeneity [12], communication overhead, and the optimization of privacy-preserving mechanisms. Among these, data heterogeneity stands out as a key challenge that significantly impacts the effectiveness of FL, particularly due to substantial differences in the data volume and class distribution across clients. For example [13], in medical image analysis tasks, different hospitals may have imbalanced patient data: some hospitals' data may predominantly focus on a few disease categories, while others may span a wider range of categories. This data heterogeneity drives client models to optimize toward their local objectives [14], making it challenging to effectively contribute to a global model. Therefore, addressing data heterogeneity to enhance the robustness and generalization performance of global models has become a key challenge and a significant focus in the field of FL [15].

To address the problem of data heterogeneity, various aggregation algorithms have been proposed. The classic FedAvg [16] algorithm determines aggregation weights based on the data volume from each client, making it suitable for scenarios with a relatively uniform data distribution. FedProx [17] introduced a regularization term to constrain model updates from clients, alleviating the model drift caused by data heterogeneity. FedMA [18] employs a feature alignment approach to match models across different clients, enhancing the effectiveness of aggregation. MOON [19] strengthened the similarity between local and global models through contrastive learning. SCAFFOLD [20] addressed the Non-IID problem by introducing control variates. However, existing fixed weighting methods often do not accurately reflect the true contribution of each client to the updates of the model [21], leading to the underestimation of the impact of certain clients, which in turn affects the fairness and generalization ability of global models.

In this paper, we propose a novel weight aggregation algorithm, FedCon, which integrates precise Shapley value computation with dynamic adjustment of historical weights, offering an efficient and robust solution for FL. Federated learning has emerged as a promising paradigm for training machine learning models across distributed devices, while preserving data privacy. However, the inherent challenges of data heterogeneity (Non-IID data) and imbalanced client contributions significantly hinder the performance and generalization ability of global models. Traditional aggregation methods, such as FedAvg, often rely on simplistic weighting schemes based on data volume, which fail to account for the varying quality and distribution of client data. This leads to suboptimal model performance, especially in scenarios where clients have highly diverse data distributions or limited computational resources.

FedCon addresses these challenges by introducing a dynamic and contribution-aware aggregation mechanism that quantifies client contributions more accurately and adapts to the evolving characteristics of client data. Unlike existing methods, FedCon leverages Shapley values to fairly evaluate client contributions, while employing an efficient Monte Carlo sampling technique to reduce the computational overhead. Additionally, FedCon incorporates historical weights and cosine similarity between client and global models to stabilize updates and enhance the robustness of the global model. By dynamically adjusting aggregation weights based on data size, distribution characteristics, and model similarity, FedCon significantly improves the generalization ability of global models, particularly in Non-IID scenarios.

Our main contributions are summarized as follows:

- We propose FedCon, a novel weighted aggregation method that initializes aggregation weights by considering both the size of the dataset and the distribution of data

categories for each client. Client contributions are calculated using Shapley values, and an efficient Monte Carlo sampling method is employed to significantly reduce computational complexity.
- By comprehensively considering the data distribution characteristics of clients and the similarity between models, FedCon dynamically adjusts a similarity matrix between the client models and the global model. This enhances the correlation between aggregation weights and model quality, while reducing the sampling frequency through the use of a cosine similarity matrix.
- FedCon incorporates long-term performance and historical contribution weights from clients, stabilizing model updates and improving the robustness and generalization ability of the global model. This approach ensures that the global model is less sensitive to noisy or malicious updates, making it more reliable in real-world deployments.

Through extensive experiments, we demonstrate that FedCon outperformed state-of-the-art federated learning algorithms in terms of model accuracy, convergence speed, and robustness to data heterogeneity. Our work not only addresses the critical challenges posed by Non-IID data but also provides a scalable and efficient solution for federated learning in practical applications.

The remainder of the paper is organized as follows: Section 2 provides a review of related work. In Section 3, we present the FedCon framework. Section 4 details the experimental results, comparing FedCon with various baselines and demonstrating its performance, and includes a sensitivity analysis of hyperparameters and an ablation study. Section 5 concludes the paper and outlines future work.

2. Related Work

In this section, we focus on reviewing the weight aggregation algorithms in FL and related research addressing the challenges in FL under data heterogeneity scenarios.

FL relies on weight aggregation algorithms to combine client model updates into a global model. The classical method, FedAvg [16], calculates the weight of each client based on the size of their dataset. While it performs well in scenarios with independent and identically distributed data [22], its performance significantly degrades when the data distributions among clients vary widely. To address this issue, FedAvgM [22] introduced a momentum term, which alleviates model oscillations caused by data heterogeneity. FedProx [17] incorporates a regularization term to constrain the local updates of each client, reducing bias in model updates and improving the stability of the global model. Furthermore, FedNova [23] proposed a normalization strategy for model updates, addressing communication efficiency and asynchronous update issues, and greatly enhancing the convergence speed of the algorithm. MOON [19] employs contrastive learning, using the contrastive loss between local and global models to enhance model personalization. FedDC [24] introduced a distribution calibration mechanism that adjusts client model updates to better align with the global model, further improving the quality of global model training.

In FL, evaluating the contribution of each client to the global model training is crucial, due to disparities in client data distributions and the heterogeneity of the model training processes. A traditional method for assessing the contribution [25,26] of clients involves calculating the difference between the model of client updates and the global model, typically by measuring the difference in model weights [27]. This difference is assumed to indicate the role of the client in optimizing the global model.

However, this approach neglects crucial factors like data quality and client-specific characteristics, both of which are essential for accurately assessing the true contribution of a client. Shapley values [28], originating from cooperative game theory, offer a fair method for assessing contributions. They quantify the marginal contribution of each client to the global model based on their individual impact on the performance of the model. While Shapley values theoretically provide the most fair contribution assessment, their computational complexity increases exponentially with the number of clients, making them impractical for large-scale systems [29].

In addition to relying on model updates and dataset size, the integration of data quality [30] offers a new perspective on contribution evaluation. In previous work, many approaches have combined local models based solely on the size of each client dataset. However, dataset size alone does not offer a meaningful insight into the data distribution of the client or the performance of the model. To further improve the stability of the global model, FedDyn [31] dynamically adjusts the customized loss function of each client. By integrating a weighted smoothing process of historical model updates and applying a decay factor to the weight strategy, FedDyn [31] strikes a balance between short-term performance and long-term contributions. FedMA [18] adjusts the global model aggregation based on parameter alignment across layers, improving the consistency between local and global models. Cosine similarity-based strategies further accelerate the global model convergence by reinforcing alignment between client models and the global model [32,33].

3. FL with Precise User Contribution Evaluation

3.1. Process of FL

FL is a distributed machine learning method, where multiple clients (such as smartphones and IoT devices) independently train models on local data, without transmitting the data to a central server. The server periodically collects the local model updates from the clients, aggregates them into a global model, and then sends the global model back to the clients to continue training.

- Initialization of the Global Model: The server initializes a global model θ_0 and distributes it to all clients.
- Local Training: Each client k trains using its local dataset B_k. The client updates its local model parameters $\theta_k(t)$ by optimizing the local objective function $f_k(\theta)$.
  The local objective function of the client k is typically given by

$$f_k(\theta) = \frac{1}{|B_k|} \sum_{i=1}^{|B_k|} \ell(\theta, x_i, y_i), \tag{1}$$

  where $\ell(\theta, x_i, y_i)$ is the loss function, which represents the loss for the model parameters θ given the input x_i and label y_i.
- Sending Local Model Updates: After several rounds of training on the local data, each client k sends its model update $\Delta\theta_k(t) = \theta_k(t) - \theta(t-1)$ to the server.
- Aggregation of Updates: The server collects the model updates from all clients and aggregates them into the global model $\theta(t)$ using a weighted average. Specifically, the server weights the updates based on the data size of each client or other factors:

$$\theta(t) = \sum_{k=1}^{K} \frac{n_k}{N} \theta_k(t), \tag{2}$$

  where $\theta_k(t)$ is the model of client k after the t-th round, n_k is the size of the dataset of the client k, and $N = \sum_{k=1}^{K} n_k$ is the total size of all datasets from the clients.
- Iterative Training: The server sends the updated global model $\theta(t)$ back to all clients, and the clients continue training on their local data, repeating the above process.

Traditional FL methods assume that the client data are independent and identically distributed (IID). However, in the case of data heterogeneity (Non-IID), significant differences in data distributions across clients lead to inconsistent contributions to the global model. Traditional weighted aggregation methods typically assign weights based on data size, failing to accurately reflect the true impact of each client, especially when there are large discrepancies in data quality. This can cause low-quality data from certain clients to negatively affect the global model, impairing its stability and generalization ability. Therefore, accurately assessing client contributions and dynamically adjusting weights is key to addressing data heterogeneity.

To tackle this issue, FedCon introduced an improved aggregation strategy. Unlike traditional methods, FedCon uses initialization weights based on both client data volume and category distribution, allowing for a more precise evaluation of contributions and effectively mitigating the impact of data heterogeneity on the global model of each client. Additionally, FedCon combines Monte Carlo sampling with a cosine similarity matrix to compute Shapley values, reducing computational complexity, while enhancing the accuracy of the client contribution assessment. Specifically, cosine similarity is used to measure the alignment between local models and the global model, reducing the reliance on sample size and further improving computational efficiency.

Furthermore, FedCon incorporates a dynamic adjustment mechanism for historical weights and Shapley values to ensure smooth transitions in the global model during training. This smooth transition strategy prevents drastic fluctuations in the training process, improving both the stability and accuracy of the model.

In summary, FedCon addresses the challenges posed by data heterogeneity with innovative strategies, such as accurate client contribution assessment, reduced computational costs, and dynamic weight adjustment, significantly improving the training effectiveness and stability of the global model.

3.2. Framework Overview

To address the challenge of improving model training accuracy with client data heterogeneity in FL, this paper proposes an approach called FL with Precise User Contribution Evaluation (**FedCon**). This method allows for precise adjustment of the weights of each client during the model aggregation process, thereby more effectively enhancing the stability of the global model in scenarios with consistent data.

The training process of FedCon is illustrated in Figure 1. First, the clients download the initialized global model θ_0 from the server. Based on the global category distribution $\mathcal{D}_g$ and the local data volume $|\mathcal{D}_i|$, the clients calculate the category discrepancies between local and global data, obtaining the initialization weights w_i^{init}. Proper initialization of w_i^{init} facilitates faster convergence of the global model. Subsequently, each client trains its local model $\theta_i^{(t)}$ on the local dataset $\mathcal{D}_i$ for E local epochs and uploads the local model parameters $\theta_i^{(t)}$ and category discrepancies DQ_i to the server. Using the Monte Carlo sampling method, the server computes a Shapley value $\phi_i^{(t)}$ for each client and calculates the cosine similarity matrix $S^{(t)}$ between the local models and the global model θ^t. The Shapley values $\phi_i^{(t)}$ are used to adjust a similarity matrix, yielding the client contributions $C_i^{(t)}$. To account for the long-term performance of local data, a forgetting algorithm is applied to weight the historical aggregation weights, combined with the current contributions $C_i^{(t)}$, resulting in the current aggregation weights $w_i^{(t)}$. The server aggregates the local models using these weights, to update the global model θ^{t+1}.

Finally, the updated global model θ^{t+1} is distributed back to the clients. This process is repeated in FedCon until the global model θ^{T} converges. The process of FedCon is summarized in Algorithm 1.

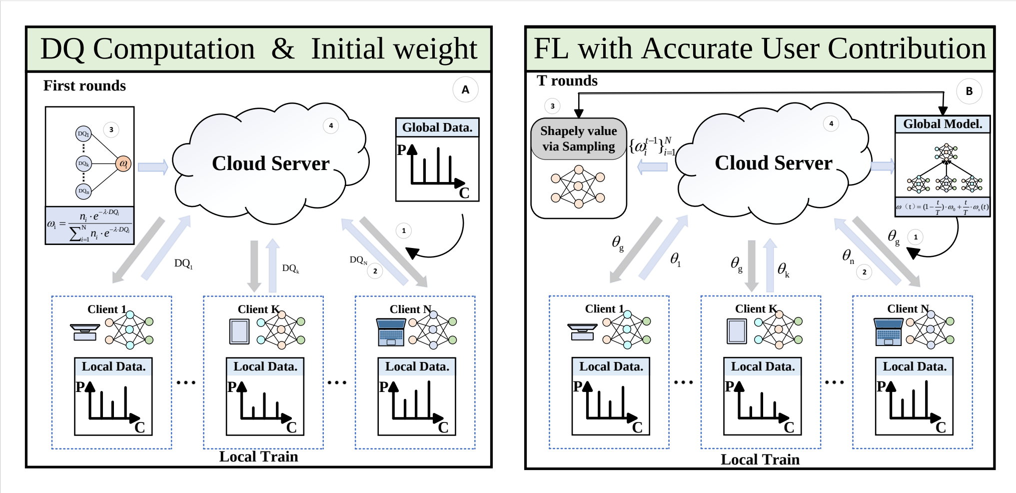

Figure 1. The FedCon framework: (**A**) the calculation of client data quality in the first round using discrepancies between global and local data distributions to determine initialization weights; (**B**) the dynamic adjustment of aggregation weights based on precise client contribution computations (using Shapley values and similarity metrics) in each round, improving the model convergence stability.

Algorithm 1: FedCon Training Process

Input: N clients, initial model θ_0, total rounds T, local epochs E, global data $\mathcal{D}_g$, local data $\mathcal{D}_i$ for each client $i \in \{1, \ldots, N\}$, discrepancy parameter λ.
Training Process:
for $t = 0$ **to** $T - 1$ **do**
  Distribute global model θ_t to all clients.
  for each client i **do**
    Train on local data $\mathcal{D}_i$ for E epochs, compute discrepancy DQ_i Equation (3).
    Send local model $\theta_{i,t}$ and DQ_i to server.
  end for
  if $t = 0$ **then**
    Compute aggregation weights w_i Equation (17) based on discrepancy DQ_i.
  else
    Calculate Shapley values ϕ_i Equation (19) for each client $i \in \{1, \ldots, N\}$ based on their contribution.
    Compute similarity matrix $\mathcal{S}$ Equation (20) .
    Adjust weights w_i Equation (21) using similarity matrix $\mathcal{S}$ and Shapley values ϕ_i.
    Update weights w_i by considering historical weights Equation (22).
  end if
  Aggregate all models $\{\theta'_{i,t}\}$, using computed weights w_i to update global model θ_{t+1}.
end for
Output: Final global model θ_T.

3.3. Client Contribution Evaluation

In FL, accurately evaluating client contributions is critical for addressing the challenges posed by data heterogeneity and model performance disparities. Variations in client data distributions and local training dynamics can lead to biased model updates, undermining the robustness and generalization ability of the global model. Effective contribution evaluation not only ensures fair aggregation but also promotes model stability and performance, making it a cornerstone of improving FL outcomes. This section introduces a comprehensive framework for client contribution assessment, addressing initialization weight design, Shapley value computation, and sampling optimization to enhance evaluation accuracy and efficiency.

3.3.1. Initial Weight

For the initialization of global model weights, data quality plays a decisive role in local model training. We can compute the data quality difference for each client, which is the difference between the data category distribution of the client and the global distribution. Let the data distribution of the i-th client be D_i, and the global category distribution be D_g. The data quality difference between the client and the global data distribution can be expressed as follows (we use the Kullback–Leibler [29] divergence to measure):

$$\mathrm{DQ}_i = D_{KL}(D_i \parallel D_g) = \sum_{k=1}^{K} D_i(k) \log \frac{D_g(k)}{D_i(k)}, \tag{3}$$

where $D_i(k)$ and $D_g(k)$ represent the probability distributions of category k for the i-th client and the global data, respectively. A larger Kullback–Leibler divergence indicates a greater quality difference between the data of the client and the global data.

Inspired by the convergence error bound proposed by FedAvg [16], which primarily focuses on the influence of client data size on the aggregation weight w_i, we further consider the data quality discrepancy DQ_i. By minimizing the global optimization error bound, we derive a new aggregation weight formula. The derivation shows that, during model aggregation, both data quantity and data quality should be considered. Our reasoning and analysis are based on the following four standard assumptions:

Assumption 1. *Stability of the Loss Function: Suppose the loss function Loss of each client is stable, which means that the gradient variations of the client are also stable. Specifically, this is expressed as follows:*

$$\|\nabla Loss_i(a) - \nabla Loss_i(b)\| \leq Loss\|a - b\|. \tag{4}$$

This ensures that no excessive gradient fluctuations will occur during the local training process of the clients.

Assumption 2. *The Gradient of Each Client is Unbiased: The gradient of each client is unbiased, meaning the expected gradient of client i equals the global gradient:*

$$\mathbb{E}[\nabla Loss_i(\theta_i)] = \nabla Loss(\theta_g). \tag{5}$$

Assumption 3. *The Global Loss Has a Lower Bound: The global loss function $Loss(\theta_g)$ has a lower bound, denoted as $Loss_{\min}$, i.e.,*

$$Loss(\theta_g) \leq Loss_{\min}, \tag{6}$$

where this ensures that the global loss does not become too low, preventing excessive deterioration of the model performance.

Assumption 4. *Data Volume and Class Imbalance: The contribution weight of each client is proportional to its local data volume n_i, and inversely proportional to the data quality discrepancy DQ_i. Specifically, the difference between the gradient of the local model and the global model is* *bounded by a constant C_{dist}, which depends on the data distribution quality of the client. Formally, we assume that*

$$\|\nabla Loss_i(\theta_i)\|^2 - \|\nabla Loss(\theta_g)\|^2 \leq C_{dist} \cdot DQ_i, \tag{7}$$

where the difference between the local gradient and the global gradient is bounded and influenced by the data distribution of the client. The constant C_{dist} quantifies the bound on the gradient discrepancy due to the data distribution of the client.

Based on the assumptions above, we observe that both the data volume n_k and the data quality discrepancy DQ_k play critical roles in determining the contribution of each client to the global model. Specifically, Assumption 4 establishes a relationship between the local gradient and the global gradient, which is bounded by the data quality discrepancy DQ_k. This implies that clients with larger data volumes and smaller data quality discrepancies are more likely to provide gradients that align closely with the global optimization direction. To incorporate these insights into the global optimization process, we define the global objective function $F(\theta)$ as a weighted sum of the local objective functions, where the weights w_k reflect both the data volume and the data quality of each client. This formulation ensures that the global model is optimized in a manner that balances the contributions of all clients, while prioritizing those with higher data quality and larger data volumes. The global objective function is defined as follows:

$$F(\theta) = \min \sum_{k=1}^{K} \omega_k \cdot f_k(\theta) \tag{8}$$

where w_k is the contribution weight of client k, satisfying $\sum_{k=1}^{K} w_k = 1$, K is the total number of clients, and $f_k(\theta)$ Equation (1) represents the loss function of the client k.

In FL, the performance of the global model is influenced not only by the volume of data for each client but also by the quality of the data, particularly the distributional alignment with the global data. To account for both factors, we introduce a regularization term that jointly considers the data volume n_k and the data quality discrepancy DQ_k. This regularization term ensures that clients with larger data volumes and better data quality contribute more significantly to the global model. To account for the impact of data quality and volume, we introduce a regularization term based on the data quality measure DQ_k and data volume n_k. The global objective function is extended as follows:

$$F(\theta) = min\left(\sum_{k=1}^{K} w_k \cdot f_k(\theta) + \lambda \cdot \sum_{k=1}^{K} g(n_k, \mathrm{DQ}_k)\right), \tag{9}$$

where λ is a hyperparameter controlling the trade-off between local loss and data quality, and $g(n_k, \mathrm{DQ}_k)$ is a function that quantifies the joint impact of data volume n_k and data quality DQ_k.

To minimize the objective function, we assume that the function $g(n_k, \mathrm{DQ}_k)$ is defined as follows:

$$g(n_k, \mathrm{DQ}_k) = n_k \cdot \mathrm{DQ}_k, \tag{10}$$

where DQ_k represents the data quality of client k, measured as the Kullback–Leibler (KL) divergence between the local data distribution D_k and the global data distribution D_g.

The optimization problem is to minimize the global objective function $F(\theta)$, which incorporates both the local loss functions and the regularization term accounting for data quality and volume:

$$F(\theta) = \min_{\omega_1,\omega_2,\cdots,\omega_K} \left(\sum_{k=1}^{K} w_k \cdot f_k(\theta) + \lambda \cdot \sum_{k=1}^{K} n_k \cdot \mathrm{DQ}_k \right). \tag{11}$$

To ensure the rationality of the weights w_i, we impose the following constraints:

$$\sum_{i=1}^{N} w_i = 1, \quad \text{and} \quad w_i \geq 0 \quad \text{for all } i. \tag{12}$$

To solve this constrained optimization problem, we introduce the Lagrangian function, which combines the global objective function with the constraint using a Lagrange multiplier β:

$$\mathcal{L}(\omega_1, \omega_2, \cdots, \omega_K, \beta) = \sum_{k=1}^{K} w_k \cdot f_k(\theta) + \lambda \cdot \sum_{k=1}^{K} n_k \cdot \mathrm{DQ}_k + \beta \left(\sum_{k=1}^{K} w_k - 1 \right), \tag{13}$$

where β is the Lagrange multiplier. To find the optimal weights w_k, we take the partial derivative of the Lagrangian function $\mathcal{L}$ with respect to w_k and set it to zero:

$$\frac{\partial \mathcal{L}}{\partial w_k} = f_k(\theta) + \beta = 0. \tag{14}$$

$$\beta = -f_k(\theta). \tag{15}$$

By taking the derivative of the objective function and applying Lagrange multipliers, we obtain the optimized expression for each weight w_i of the client:

$$w_k \propto n_k \cdot e^{-\lambda \cdot DQ_k}. \tag{16}$$

This indicates that the weight of the client is exponentially related to both its data size n_k and the data quality discrepancy DQ_k. Clients with larger data sizes and smaller data quality discrepancies receive higher weights, thus contributing more to the global model update.

Upon further simplification, we obtain the final weight formula:

$$w_k = \frac{n_k \cdot e^{-\lambda \cdot DQ_k}}{\sum_{k=1}^{K} n_k \cdot e^{-\lambda \cdot DQ_k}}, \tag{17}$$

where this formula shows that the weight of the client depends not only on their data size n_k, but also on the data quality discrepancy DQ_k. Specifically, when n_k is larger and DQ_k is smaller, the client's aggregation weight w_k increases. This indicates that clients with larger data volumes and more balanced data distributions (i.e., smaller discrepancies with the global distribution) are assigned higher weights. This formula calculates the weight w_k of client k, considering both the data size n_k and the data quality discrepancy DQ_k. Clients with larger data sizes or smaller data quality discrepancies are assigned higher weights. The exponential function $e^{-\lambda \cdot DQ_k}$ adjusts the influence of data quality discrepancy, with λ controlling the strength of this effect. Finally, the weights are normalized across all clients to ensure a fair distribution of their contributions to the global model. This weighting strategy allows for a more accurate weight distribution, effectively mitigating the negative effects of data distribution imbalances on the global model and improving its accuracy and generalization capability. λ is a hyperparameter that controls the influence of the data quality difference on the weight coefficient. This weight coefficient p_k combines the data size and the data quality difference of each client, so that clients with higher data quality and larger data sizes contribute more to the global model. Based on experimental results, it is recommended that the value of λ be set within the range $[0.5, 0.7]$.

FedCon ensures privacy by adopting a decentralized approach to handling sensitive client data. Specifically, data category distributions are computed locally on clients, and the server only collects the model update differences. This approach significantly reduces the risk of leaking sensitive information, as the server does not have direct access to the raw data or explicit data categories of individual clients. Instead, it only receives aggregated updates in the form of model differences, which prevents any reverse-engineering of client-specific data attributes.

In contrast to traditional FL methods, which may expose information through the exchange of raw model updates or data aggregation strategies, FedCon's design limits the information shared with the server. This approach ensures that the server cannot infer private data details, such as specific data categories or class distributions, from the model updates. The privacy guarantees of FedCon are particularly significant in scenarios involving sensitive or personal data, such as medical records, where privacy is of utmost importance.

When compared to other privacy-preserving FL techniques, such as differential privacy (DP) [34] and secure multi-party computation (SMPC) [35], FedCon offers a distinct trade-off. While DP [34] techniques typically add noise to the model updates to preserve privacy, and while SMPC [35] employs cryptographic techniques to secure computations, FedCon maintains privacy through model update differences, which provides an efficient alternative, without the need for heavy computational overhead. FedCon's method is particularly suitable for scenarios where a balance between privacy and computational efficiency is required.

In this subsection, we have discussed how to initialize weights based on the category discrepancies across clients, in order to accurately assess the contribution of each client to the global model and lay a stable foundation for subsequent model training. Next, after initializing the weights, we use the Monte Carlo sampling method to calculate Shapley values, enabling a more precise evaluation of the contribution of each client.

3.3.2. Shapley with Monte Carlo Sampling

In FL, the Shapley value, originating from cooperative game theory, provides a fair way to quantify each marginal contribution of clients to the global model training process. The core idea is to evaluate the contribution of each client by considering all possible orders in which the clients could be added to the training process. The Shapley value ϕ_k for client k is calculated as follows:

$$\phi_k = \sum_{S \subseteq K \setminus \{k\}} \frac{|S|!(|K| - |S| - 1)!}{|K|!} [F(S \cup \{k\}) - F(S)], \tag{18}$$

where S is a subset of clients excluding client k, $F(S)$ represents the contribution of subset S to the global model, and $F(S \cup \{k\})$ is the contribution when client k joins the subset. The term $\frac{|S|!(|K|-|S|-1)!}{|K|!}$ is a weight factor that ensures that each possible order in which clients join the training process is given equal importance. The main idea behind the Shapley value calculation is to evaluate the marginal contribution of each client to the global model. For each subset of clients S, we compute the difference in performance $F(S \cup \{k\}) - F(S)$, which represents the gain in the global model's performance when client k is added. Then, this contribution is weighted by the likelihood of S occurring in any given order.

To calculate the Shapley value for a client, we evaluate its marginal contribution to the global model by considering all possible orders of client inclusion. For two clients C_1 and C_2, assume the following performance values for the global model: $F(\emptyset)$, $F(\{C_1\})$, $F(\{C_2\})$, $F(\{C_1, C_2\})$. The Shapley value ϕ_1 for client C_1 is computed by averaging its marginal contributions across all possible orders of inclusion. Specifically, this is given by $\phi_1 = \frac{1}{2}[F(\{C_1\}) - F(\emptyset) + F(\{C_1, C_2\}) - F(\{C_2\})]$. Similar calculations can be performed for C_2 and C_3. This process ensures that the contribution of each client to the global model is fairly quantified, which is especially important in Non-IID scenarios where the data distribution varies across clients.

Calculating the Shapley value allows for the precise evaluation of the contribution of each client. This is particularly beneficial in Non-IID scenarios, where it helps improve the stability and accuracy of the global model. However, due to its computational complexity, which grows exponentially with the number of clients, approximate methods such as Monte Carlo sampling are often used to simplify the computation. These methods ensure that Shapley values can be efficiently applied for weight aggregation in FL, while maintaining theoretical fairness and effectiveness.

To efficiently evaluate the contribution of each client to the global model and compute the Shapley values, we use a simplified method based on Monte Carlo sampling. In this process, we rank and select clients based on their data volume n_i and data category discrepancies DQ_i, to optimize the sampling procedure. During Monte Carlo sampling, we select clients for sampling based on the computed weight w_i. This strategy significantly improves the sampling efficiency and accuracy, while reducing unnecessary computations. Through the selected subsets S_m, we can approximate the marginal contribution of client k and obtain an approximation of the Shapley value:

$$\phi_k = \frac{1}{M} \sum_{m=1}^{M} [F(S_m \cup \{k\}) - F(S_m)], \quad (19)$$

where S_m represents the m-th randomly selected subset of clients, and $F(S_m)$ and $F(S_m \cup \{k\})$ represent the model performance with and without client k in subset S_m, respectively. Moreover, M is the number of sampling iterations.

In our experiments, we selected 30% of the clients. Specifically, the process involved first sorting the clients based on their weights w_i, and selecting the top 30% of clients as the sampling set. Then, the local models of these clients were aggregated to form a global model, which was evaluated for accuracy on a test set. Next, by constructing different subsets of clients (sampled from different combinations), the contribution of each client to the accuracy of the global model was computed. The Shapley value for each client was estimated by calculating the average of its contributions across all possible subsets, reflecting the impact of the client on the performance of the global model.

The marginal contribution function is calculated based on the accuracy of the client on the test set. By comparing the performance of different subsets, this effectively estimates the Shapley value. This strategy ensures that we can efficiently evaluate the contribution of each client to the global model in large-scale systems, without having to traverse all possible client combinations. Through experiments, we found that Monte Carlo sampling, while ensuring computational efficiency, provides sufficiently accurate approximations of the Shapley values, thus achieving a reasonable evaluation of the contributions from clients.

To better understand the advantages of this strategy, we compared the complexity of using Monte Carlo sampling with the traditional Shapley value computation method. In the traditional Shapley value calculation, all combinations of clients need to be traversed, resulting in a time complexity of $O(2^N \cdot N)$, where N is the number of clients. In contrast, Monte Carlo sampling performs five iterations of sampling, each time selecting 30% of the clients, with a time complexity of $O(M \cdot 0.3N)$, where M is the number of sampling iterations. Thus, by selecting the clients for Monte Carlo sampling, we significantly reduce the computational complexity, especially when the number of clients is large, making this strategy an effective way to reduce the computational burden. However, it is important to note that Monte Carlo sampling may introduce potential unfairness [36,37] in client selection, as random sampling does not guarantee absolute fairness in representing all types of clients. To mitigate this limitation, stratified sampling can be employed, where clients are grouped based on specific characteristics (e.g., data distribution, device type, or contribution level), and samples are drawn from each group proportionally. This approach not only considers the weight of each client but also balances the representation of different client types, thereby enhancing the fairness and robustness of the contribution evaluation process.

In this subsection, we calculate the Shapley values using the Monte Carlo sampling method to accurately assess the contribution of each client. To further reduce computational costs, we introduce a cosine-similarity-based weighted Shapley value method, aiming to simplify the contribution evaluation process.

3.3.3. The Weighted Shapley Value Method Based on Cosine Similarity

Although Monte Carlo sampling can effectively simplify the calculation of Shapley values, its accuracy relies on the number of samples. This becomes particularly challenging when the amount of client data is large and model training time is long, leading to higher computational costs. To reduce this overhead, we introduce a cosine similarity matrix between the local models and the global model to optimize the sampling process.

In FL, the similarity between local models and the global model is usually measured using cosine similarity. Let the local model of the i-th client be θ_i and the global model be θ_g. The cosine similarity between them, denoted as $\cos(\theta_i, \theta_g)$, is defined as follows:

$$\cos(\theta_i, \theta_g) = \frac{\theta_i \cdot \theta_g}{\|\theta_i\| \|\theta_g\|}, \tag{20}$$

where $\theta_i \cdot \theta_g$ denotes the dot product of the model vectors, and $\|\theta_i\|$ and $\|\theta_g\|$ represent the Euclidean norms of the local and global models, respectively. The higher the similarity value, the closer the local model is to the global model.

Monte Carlo sampling can effectively simplify the computation of Shapley values. However, even with the Monte Carlo method, the number of samples still affects the accuracy of the approximation, especially when the client data volume is large and the model training time is long. Monte Carlo sampling requires multiple iterations to estimate a marginal contribution, which still incurs significant computational overhead. Effectively conducting sampling remains an important challenge.

Therefore, we considered combining the cosine similarity matrix between the local models and the global model, to reduce the number of samples. When the cosine similarity between the local model and the global model is high, this indicates that the update direction of the client is more aligned with the global model. Thus, by incorporating a cosine similarity matrix, we can better capture the relationship between the client and the global model, avoiding potential biases from relying purely on Shapley values. The adjustment formula is as follows:

$$w_i = \alpha \cdot \phi_i + (1 - \alpha) \cdot \cos(\theta_i, \theta_g), \tag{21}$$

where ϕ_i represents the Shapley value of client i, and $\cos(\theta_i, \theta_g)$ represents the cosine similarity between the local model θ_i of client i and the global model θ_g. α is a hyperparameter that controls the trade-off between the Shapley value and the cosine similarity contribution, with $0 \leq \alpha \leq 1$.

By adjusting α, we can control the influence of Shapley values and cosine similarity on the weights. When $\alpha = 1$, the formula relies solely on the Shapley value, and when $\alpha = 0$, the formula depends only on the cosine similarity. In practice, clients with higher similarity to the global model contribute more to the global model, so increasing the weight of such clients can help reduce redundant sampling for them.

After accurately calculating the client weights, we introduce a smooth transition mechanism to ensure the stability of the global model training, aiming to reduce the impact of weight fluctuations on the training process.

3.3.4. Smooth Transition of Weights

We propose a dynamic adjustment mechanism to achieve a smooth transition between the initial weights and the subsequent weights based on Shapley values, while ensuring training stability. In each aggregation round, the initial weights and Shapley values are gradually combined to ensure that the data contribution represented by the initial weights in the first round gradually transitions to the contribution represented by the Shapley values.

We use a round-based linear decay to adjust the combination ratio of the initial weights and the Shapley values. As the number of training rounds increases, the influence of the initial weights gradually decreases, and ultimately, the weight is fully determined by the Shapley values. The decay formula is as follows:

$$w_i^t = \left(1 - \frac{t}{T}\right) \cdot w_i^{t-1} + \frac{t}{T} \cdot w_s^t, \tag{22}$$

where $w_i^{(t)}$ is the dynamic weight of client i at round t. $w_i^{(t-1)}$ is the weight of client i in the previous round. $w_s^{(t)}$ is the Shapley weight of client i at round t, which represents the contribution of client i to the global model at the current round. T is the total number of training rounds, and t is the current training round.

This linear decay mechanism ensures a gradual transition from the initial weights to the Shapley-based weights, allowing the model to rely on the initialization in the early stages of training for stability, while progressively incorporating more accurate client contributions as the training advances. This dynamic adjustment mechanism effectively minimizes the potential instability during the early stages of training when data contributions are not fully determined. Introducing Shapley values too early could cause significant fluctuations in the training process, especially when there are substantial differences in client data distributions.

From a theoretical perspective, the introduction of initial weights helps prevent overfitting in the early stages of training. Since the model relies more on these stable initial weights when the data are limited, this avoids over-dependence on a few high-quality clients. As training progresses, the increasing weight of the Shapley values ensures that the true contributions of each client are accurately assessed in the later stages. This mechanism effectively mitigates the influence of clients with imbalanced or heterogeneous data distributions, ensuring the stability and reliability of the global model.

Through this decay mechanism, we can ensure that, in each aggregation round, the initial weights gradually transition to the contribution represented by the Shapley values, thereby maintaining the stability and accuracy of the model throughout the training process, and ensuring that the global model effectively learns the true contribution of each client. As the number of training rounds increases, the influence of the Shapley values gradually increases, ultimately ensuring that the true contribution of each client is accurately reflected, thereby improving the performance and fairness of the global model.

4. Experiments

Our experiments were conducted on a computer equipped with an Intel Core i5-12400K CPU, 32.00 GB of RAM, and an NVIDIA GeForce RTX 3060 GPU.

In a real-world deployment, several challenges arise, particularly related to hardware and network requirements. Edge devices or clients involved in federated learning typically vary widely in computational capacity, storage, and network bandwidth. These variations can significantly impact the efficiency and stability of the learning process. Moreover, the communication between clients and servers in a real-world setting can be influenced by factors such as network latency, bandwidth limitations, and the reliability of the network connection, all of which may affect model convergence and performance.

For FedCon, real-world deployment would require careful consideration of hardware resources on both the client and server sides. Clients with limited computational resources might require optimization techniques to ensure efficient model training and updates. On the server side, sufficient computational power would be necessary to handle the aggregation of updates from potentially many clients. Furthermore, maintaining privacy, as we achieved in the simulated environment by only sharing model updates instead of raw data, would need to be rigorously preserved in real-world scenarios, especially when dealing with sensitive data.

Although the current experiments were limited to simulations, we believe that the design of FedCon focused on efficient privacy-preserving updates, making it a strong candidate for real-world deployment as hardware and network infrastructure continue to improve. We are confident that, with appropriate optimization and resource allocation, FedCon could be deployed in practical federated learning environments in the near future.

4.1. Data Partitioning

To validate the effectiveness of the proposed method, we conducted experiments on four commonly used datasets: CIFAR10, CIFAR100, HAM10000, and HAR. These datasets cover image classification and time-series tasks. CIFAR10 and CIFAR100 are typically used for image classification, HAR is a time-series dataset, and HAM10000 is a medical image dataset for skin lesion classification. To evaluate the performance of the algorithm in healthcare applications, we also included three medical datasets: OrganAMNIST, OrganCMNIST, and OrganSMNIST. These datasets contain different types of medical images and are used for organ classification tasks and various medical image classifications, further demonstrating the applicability and performance of the algorithm in the medical field.

For the HAM10000 dataset, due to its inherent class imbalance, we performed preprocessing by selecting 20 samples per class for the test set and applied only the Dirichlet partitioning method for data division. For the remaining datasets, we adopted the standard 9:1 ratio to split the data into training and test sets. The same partitioning standard was also applied to the HAR dataset. Two data partitioning methods are shown in the Figures 2 and 3.

The first data partitioning method was based on the Dirichlet distribution, referred to as NIID-1. The core idea of this method is to generate a class probability distribution for each client by setting a parameter vector α. Specifically, for this method, we set $\alpha = 0.1$ for the 10 clients, ensuring that the data distribution for each client had a certain degree of randomness and diversity.

The second partitioning method is called the "Pathological method", referred to as NIID-2. In this method, 2 out of the 10 clients have evenly distributed data, while the remaining 8 clients each contain only 2 or 3 categories of samples. Although the amount of data per client is equal, the distribution of categories is significantly imbalanced. This partitioning method aims to simulate the severe class imbalance often found in real-world medical data scenarios, where some clients may only have data from certain categories, while other categories may be almost entirely absent. This method provided an effective testing scenario for evaluating the ability of the model to adapt to class imbalance and distribution discrepancies.

For CIFAR10 and CIFAR100, we conducted experiments with three data partitioning methods: Homo, NIID-1, and NIID-2, where Homo represents a uniform distribution. For HAM10000 and HAR, we only conducted experiments with the NIID-1 data partitioning method. The following are two examples of data partitioning for CIFAR10. For the three MedMNIST datasets: OrganAMNIST, OrganCMNIST, and OrganSMNIST, due to the unequal data distribution across classes, they did not conform to homogeneous (Homo) partitioning. Therefore, we only performed experiments using two Non-IID partitioning methods: NIID-1 and NIID-2.

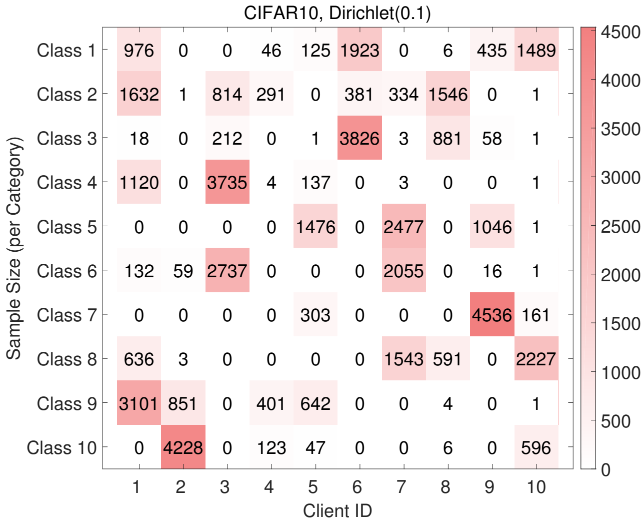

Figure 2. The above chart is the heatmap of data partitioning for CIFAR10-NIID-1.

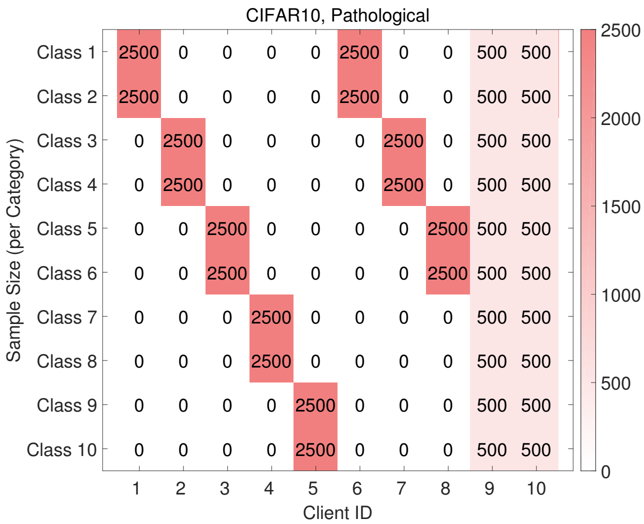

Figure 3. The above chart is the heatmap of data partitioning for CIFAR10-NIID-2.

4.2. Baselines

In this study, we selected eight methods for comparative analysis, to evaluate their performance in scenarios with data heterogeneity. These methods included FedAvg [16], FedProx [17], FedDyn [30], FedNova [23], MOON [19], FedDC [24], SCAFFOLD [20], and Local training. The Local method refers to a scenario where each client only performed local training, without participating in any form of global aggregation, providing a baseline for assessing the performance of individual clients without global model sharing. Through this approach, we were able to gain a comprehensive understanding of the strengths and weaknesses of the different methods in handling data heterogeneity, the model convergence, and accuracy.

4.3. System and Model Settings

All baseline methods were configured with the hyperparameters recommended in their respective original papers. These methods have been widely applied in the FL field and validated in multiple studies, ensuring high reliability and standardization. By comparing with these well-established baseline methods, we could ensure the validity of our experimental results and clearly demonstrate the significant improvements achieved by our proposed FedCon method in various scenarios. In the experiments, we used standard hyperparameter settings: a learning rate of 0.01, a batch size of 64, and a total of 100 communication epochs. These settings were based on extensive experiments with common datasets, ensuring that each method was fairly compared under the same experimental conditions. During model evaluation, we focused on three key metrics: accuracy, loss, and training stability, to comprehensively assess the performance of each method. These evaluation criteria allow us to deeply analyze the performance of each baseline method in the context of data heterogeneity and to validate the advantages of the FedCon method in addressing data imbalance and client heterogeneity.

On the CIFAR10 and CIFAR100 datasets, we used a convolutional neural network (CNN) architecture consisting of two convolutional layers and three fully connected layers. For the HAM10000 dataset, a customized modification based on the ResNet18 network was employed. On the HAR dataset, a CNN architecture with three convolutional layers and three fully connected layers was used. On the OrganAMNIST, OrganCMNIST, and OrganSMNIST datasets, we used a CNN architecture consisting of two convolutional layers followed by four fully connected layers.

4.4. Experiment Evaluation

4.4.1. Accuracy Analysis

Table 1 presents a comparison of the global model performance on the CIFAR10 and CIFAR100 datasets under both Independent and two Non-Independent Data Partitioning methods. Since the HAR datasets have significantly uneven sample distributions across classes, NIID-2 partitioning could not be applied. Therefore, experiments on these datasets were only conducted using the NIID-1 partition. Table 2 presents a performance comparison on four medical datasets under the NIID-1 and NIID-2 data partitions. Since the class distributions in the HAR and HAM10000 datasets are similar, they did not meet the requirements for Homo and NIID-2 partitioning, so experiments were only conducted under the NIID-1 partition. In contrast, the OrganAMNIST, OrganCMNIST, and OrganSMNIST datasets satisfy both NIID-1 and NIID-2 partitioning.

The experimental results in the table fully demonstrate the significant advantages of FedCon across the various datasets and scenarios, especially its outstanding performance in handling data heterogeneity and complex classification tasks. On the HAM10000 dataset under the NIID-1 setting, the accuracy of FedCon was significantly improved compared to the other methods, with an increase of 8.09% over FedDyn [30] and 10.49% over SCAFFOLD [20], showcasing its stronger robustness and adaptability in addressing the class imbalance issues in medical imaging data. On the HAR dataset under the NIID-1 setting, FedCon achieved an accuracy of 93.52%. It outperformed FedNova [23], SCAFFOLD [20], and FedDyn [30] by 3.51%, 3.07%, and 3.97%, respectively. This highlights its superior ability to handle time-series data. FedCon demonstrated clear advantages on the OrganAMNIST, OrganCMNIST, and OrganSMNIST datasets across both NIID-1 and NIID-2 settings. On OrganAMNIST, it outperformed FedProx by 6.90% and FedDyn [30] by 3.48%. For OrganCMNIST, it achieved an 86.03% accuracy, surpassing FedDyn [30] by 2.00%. On OrganSMNIST, FedCon achieved a 69.27%, outperforming FedDyn [30] by 2.82%. These results highlight FedCon's superior ability to handle data heterogeneity and class imbalance in medical image classification tasks.

Table 1. The accuracy comparison across various settings and datasets demonstrates that FedCon consistently outperformed these state-of-the-art methods. The experimental results were statistically significant, with the data representing the average of three runs.

Datasets	**HAR**		**CIFAR10**			**CIFAR100**	
Metrics	**NIID-1**	**Homo**	**NIID-1**	**NIID-2**	**Homo**	**NIID-1**	**NIID-2**
Local	86.49 ± 0.32	56.14 ± 0.45	46.71 ± 0.28	44.26 ± 0.31	18.55 ± 0.22	13.34 ± 0.18	12.46 ± 0.15
FedAvg	89.45 ± 0.25	68.97 ± 0.38	60.63 ± 0.30	65.00 ± 0.35	33.65 ± 0.28	24.82 ± 0.20	25.46 ± 0.22
FedProx	89.81 ± 0.27	71.37 ± 0.40	64.86 ± 0.32	64.95 ± 0.33	27.62 ± 0.25	27.39 ± 0.23	29.80 ± 0.26
FedDyn	89.55 ± 0.26	71.83 ± 0.39	66.32 ± 0.31	66.97 ± 0.34	34.34 ± 0.29	30.57 ± 0.24	31.97 ± 0.27
FedNova	90.01 ± 0.28	69.36 ± 0.37	63.36 ± 0.29	61.25 ± 0.30	29.36 ± 0.26	24.95 ± 0.21	25.24 ± 0.20
SCAFFOLD	90.45 ± 0.29	72.43 ± 0.41	65.52 ± 0.33	66.70 ± 0.35	33.56 ± 0.28	29.6 ± 0.25	30.97 ± 0.26
MOON	89.65 ± 0.27	71.28 ± 0.40	64.84 ± 0.31	65.17 ± 0.34	32.87 ± 0.27	28.39 ± 0.24	29.88 ± 0.25
FedDC	89.38 ± 0.26	71.69 ± 0.39	65.26 ± 0.32	66.43 ± 0.33	32.64 ± 0.27	30.05 ± 0.25	31.53 ± 0.26
FedCon	**93.52 ± 0.18**	**76.63 ± 0.35**	**70.26 ± 0.30**	**70.94 ± 0.32**	**42.52 ± 0.31**	**42.68 ± 0.28**	**45.73 ± 0.30**

Table 2. The accuracy comparison on the medical datasets also demonstrates that FedCon outperformed the other methods.

Datasets	**HAM10000**	**OrganAMNIST**		**OrganCMNIST**		**OrganSMNIST**	
Metrics	**NIID-1**	**NIID-1**	**NIID-2**	**NIID-1**	**NIID-2**	**NIID-1**	**NIID-2**
Local	33.33 ± 0.45	58.23 ± 0.50	58.17 ± 0.51	48.35 ± 0.48	48.33 ± 0.47	37.61 ± 0.42	37.59 ± 0.41
FedAvg	36.19 ± 0.47	51.76 ± 0.52	51.73 ± 0.53	64.00 ± 0.55	64.39 ± 0.56	59.76 ± 0.54	59.67 ± 0.53
FedProx	36.86 ± 0.48	74.59 ± 0.60	74.58 ± 0.61	64.31 ± 0.57	64.30 ± 0.58	65.73 ± 0.59	65.71 ± 0.60
FedDyn	48.53 ± 0.55	78.01 ± 0.65	79.53 ± 0.66	81.82 ± 0.68	83.22 ± 0.69	66.45 ± 0.62	68.00 ± 0.63
FedNova	47.13 ± 0.54	53.46 ± 0.53	51.76 ± 0.52	68.25 ± 0.60	64.41 ± 0.59	61.49 ± 0.58	59.77 ± 0.57
SCAFFOLD	46.13 ± 0.53	74.66 ± 0.62	74.55 ± 0.63	64.50 ± 0.61	64.48 ± 0.62	65.77 ± 0.63	65.96 ± 0.64
MOON	34.06 ± 0.46	74.58 ± 0.61	74.63 ± 0.62	64.30 ± 0.60	64.30 ± 0.61	65.82 ± 0.63	64.75 ± 0.62
FedDC	47.13 ± 0.54	78.19 ± 0.66	78.61 ± 0.67	82.83 ± 0.70	82.43 ± 0.69	67.97 ± 0.65	68.06 ± 0.66
FedCon	**56.62 ± 0.50**	**81.49 ± 0.68**	**81.99 ± 0.69**	**84.71 ± 0.72**	**86.03 ± 0.73**	**69.27 ± 0.67**	**70.57 ± 0.68**

Overall, FedCon demonstrated significant advantages across all datasets and partitioning scenarios, particularly excelling in handling non-independent and identically distributed data, with enhanced robustness and convergence stability. On the CIFAR100 dataset under the NIID-2 setting, FedCon achieved an accuracy improvement of over 20%, proving its outstanding performance in complex tasks and on heterogeneous data. Its innovative aggregation strategy dynamically adjusts client weights, accurately evaluates the data distribution and historical contributions, and effectively enhances the generalization ability of the global model.

The exceptional performance of FedCon indicates its high application potential in scenarios with data heterogeneity. Compared to traditional methods, FedCon not only significantly improved the global model performance but also excelled in stability and fairness, verifying its effectiveness and practicality in high-complexity and class-imbalanced tasks. In summary, FedCon demonstrated remarkable robustness and effectiveness in addressing Non-IID data, with particularly outstanding results in challenging scenarios.

To further evaluate the scalability and robustness of FedCon, additional experiments were conducted on the FashionMNIST (fmnist) dataset, with a significantly larger number of clients. Specifically, the number of clients was increased to 50, 100, 200 and experiments were performed under both NIID-1 and NIID-2 data partitioning scenarios. The results of these experiments are summarized in Tables 3–5 which demonstrate that FedCon continued to deliver a strong performance, even when the number of clients was significantly increased.

The experimental results indicate that FedCon maintained its effectiveness in large-scale settings, achieving competitive accuracy with both data partitioning schemes. This confirms that the method is not only effective in small-scale experiments, but also scales well with increasing numbers of clients. Furthermore, the ability of FedCon to adaptively adjust client weights based on their contribution and data quality ensures its robustness, even as the number of participating clients grows.

The scalability of FedCon is a crucial aspect of its design, as federated learning is inherently constrained by the number of clients involved in the system. The experimental results show that the computational cost of estimating Shapley values, which is a key component of FedCon's aggregation process, remained manageable, even with 50 clients. This suggests that FedCon can effectively handle larger-scale federated learning scenarios, which is important for real-world applications where the number of clients can be substantial.

Overall, the results underline FedCon's strong scalability and its ability to handle the complexities associated with larger, more diverse client populations. The ability to effectively manage data heterogeneity and client contributions across a larger number of clients further enhances the generalization ability of the global model, making FedCon a promising approach for large-scale federated learning applications.

Table 3. Scalability of FedCon on the FMNIST dataset with NIID-1 and NIID-2 data partitioning for 50 clients, shown in Table 3.

Metrics	Local	FedAvg	FedProx	FedDyn	FedNova	SCAFFOLD	MOON	FedDC	FedCon
NIID-1	81.46	82.31	83.28	83.65	84.12	86.05	83.19	85.74	**88.46**
NIID-2	82.19	84.58	85.26	85.81	84.71	75.48	85.85	85.72	**90.56**

Table 4. Scalability of FedCon on the FMNIST dataset with NIID-1 and NIID-2 data partitioning for 100 clients, shown in Table 4.

Metrics	Local	FedAvg	FedProx	FedDyn	FedNova	SCAFFOLD	MOON	FedDC	FedCon
NIID-1	78.32	79.45	80.12	80.67	81.23	82.15	80.89	82.74	**83.81**
NIID-2	79.14	81.56	82.34	82.89	81.75	72.63	82.91	82.68	**87.25**

Table 5. Scalability of FedCon on the FMNIST dataset with NIID-1 and NIID-2 data partitioning for 200 clients, shown in Table 5.

Metrics	Local	FedAvg	FedProx	FedDyn	FedNova	SCAFFOLD	MOON	FedDC	FedCon
NIID-1	76.45	77.32	78.15	78.64	79.12	80.05	78.19	80.74	**81.6**
NIID-2	77.19	79.58	80.26	80.81	79.71	70.48	80.85	80.72	**86.05**

4.4.2. Convergence Analysis

To validate the convergence of FedCon, we performed 100 rounds of global model training on the CIFAR10 and CIFAR100 datasets under three data partitioning schemes: Homo, NIID-1, and NIID-2. Figure 4 shows the convergence results for CIFAR10-Homo and CIFAR100-Homo. The results indicate that FedCon demonstrated stable and fast convergence under all partitioning schemes, outperforming other methods, particularly in reducing fluctuations and improving the convergence stability when facing different data distributions. Figure 5 shows the convergence results for CIFAR10-NIID-1 and CIFAR10-NIID-2. FedCon also exhibited faster convergence in these two Non-IID settings. Figure 6 displays the convergence results for CIFAR100-NIID-1 and CIFAR100-NIID-2, where FedCon continued to maintain a faster convergence rate, further demonstrating its strong generalization ability on more complex, larger datasets.

Figure 7 shows the convergence results for HAM10000-NIID-1 and HAR-NIID-1. FedCon outperformed the other methods on these datasets, showing stable convergence with significantly faster speeds, indicating its stronger robustness when handling heterogeneous and medical data. In contrast, the baseline methods struggled with convergence, often oscillating or converging to local optima. Figures 8–10 show the convergence results on the OrganAMNIST, OrganCMNIST, and OrganSMNIST datasets under the NIID-1 and NIID-2 data partitioning strategies, respectively.

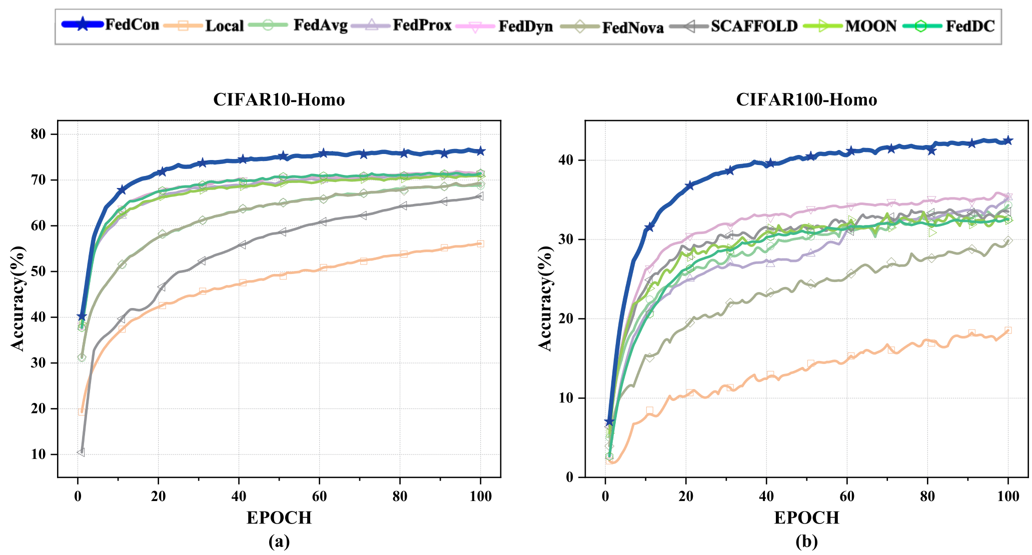

Figure 4. Convergence analysis of different methods on CIFAR10 and CIFAR100 datasets under independent and identically distributed (Homo) settings. (**a**) shows the convergence analysis of CIFAR10 under Homo settings, and (**b**) shows the convergence analysis of CIFAR100 under Homo settings.

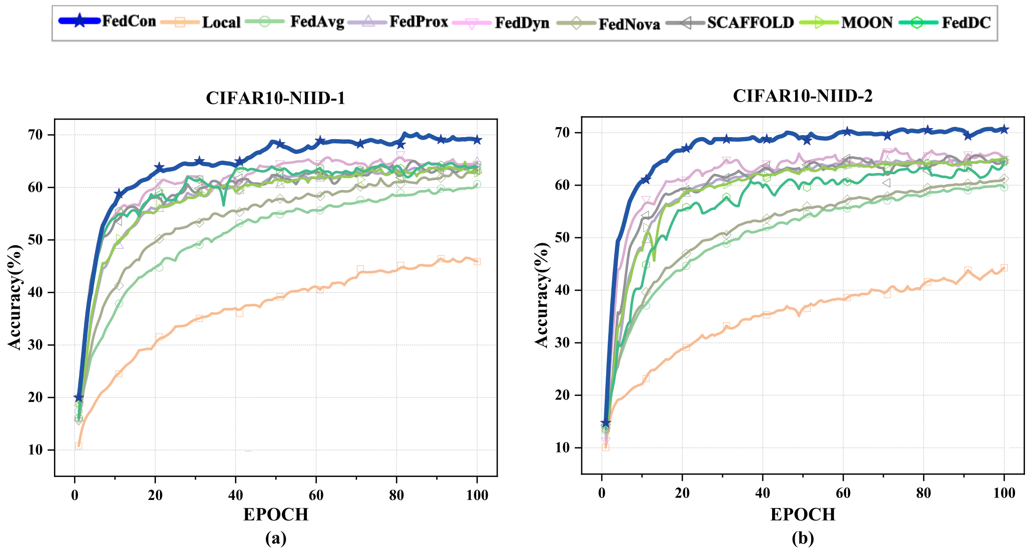

Figure 5. Convergence analysis of different methods on the CIFAR10 dataset under two Non-IID data partitioning strategies, NIID-1 and NIID-2. The figure above demonstrates that the FedCon method achieved a faster convergence and outperformed the other methods in terms of performance. (**a**) shows the convergence analysis of CIFAR10 under NIID-1 settings, and (**b**) shows the convergence analysis of CIFAR10 under NIID-2 settings.

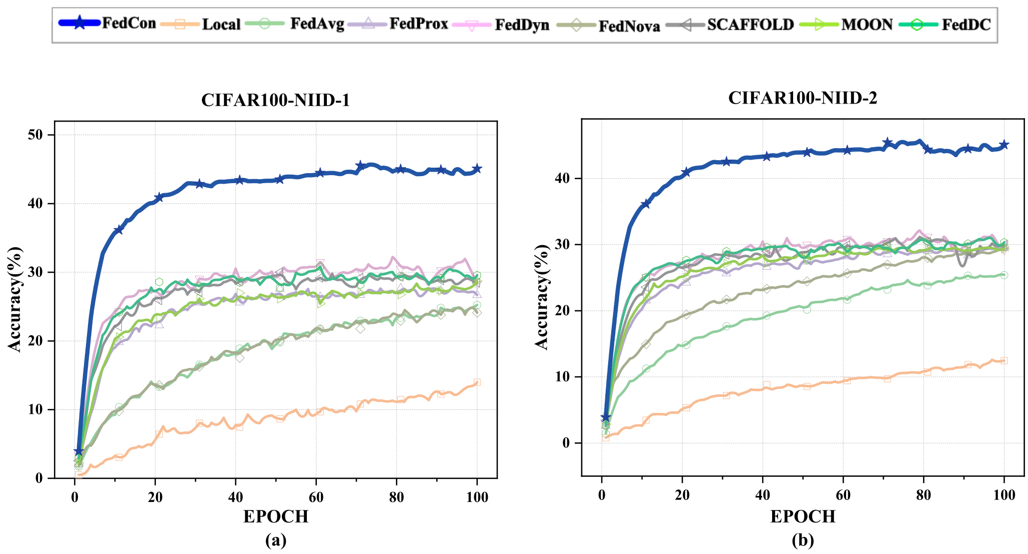

Figure 6. Convergence analysis of various methods on the CIFAR100 dataset under two Non-IID data partitioning strategies, NIID-1 and NIID-2. (**a**) shows the convergence analysis of CIFAR100 under NIID-1 settings, and (**b**) shows the convergence analysis of CIFAR100 under NIID-2 settings.

Overall, FedCon demonstrated fast and stable convergence in both IID and Non-IID environments, showing remarkable adaptability to various data distributions. Whether dealing with evenly distributed data or highly heterogeneous datasets, FedCon effectively handled data diversity and complexity. Its strong generalization ability ensures that the model maintains excellent performance even when faced with different tasks and datasets. This characteristic makes FedCon widely adaptable in practical applications, especially when dealing with complex medical image data or multiple data sources, providing consistent and efficient learning results.

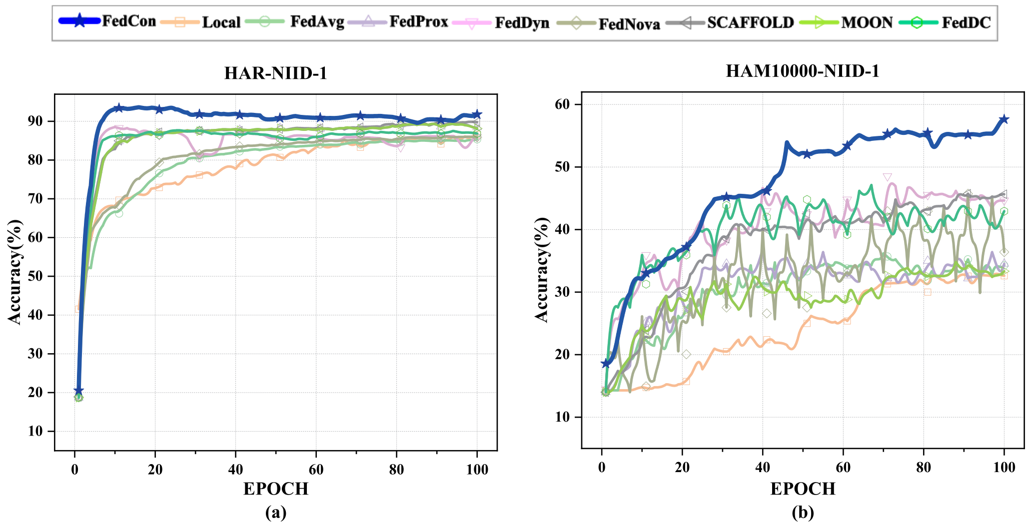

Figure 7. Convergence analysis of different methods on the HAR and HAM10000 datasets under the NIID-1 partitioning strategy (since these two datasets have different numbers of classes, only the NIID-1 partitioning strategy could be applied). (**a**) shows the convergence analysis of HAR under NIID-1 settings, and (**b**) shows the convergence analysis of HAM10000 under NIID-1 settings.

In comparison to traditional aggregation methods, FedCon shows significant advantages, particularly in convergence speed and stability. FedCon converges more quickly to lower error rates and demonstrates greater robustness when confronted with data inconsistencies and class imbalance issues. By accurately evaluating client contributions and dynamically adjusting weights, FedCon not only optimizes the training efficiency of the global model but also ensures the quality of the model aggregation results, making it a crucial method for various practical applications.

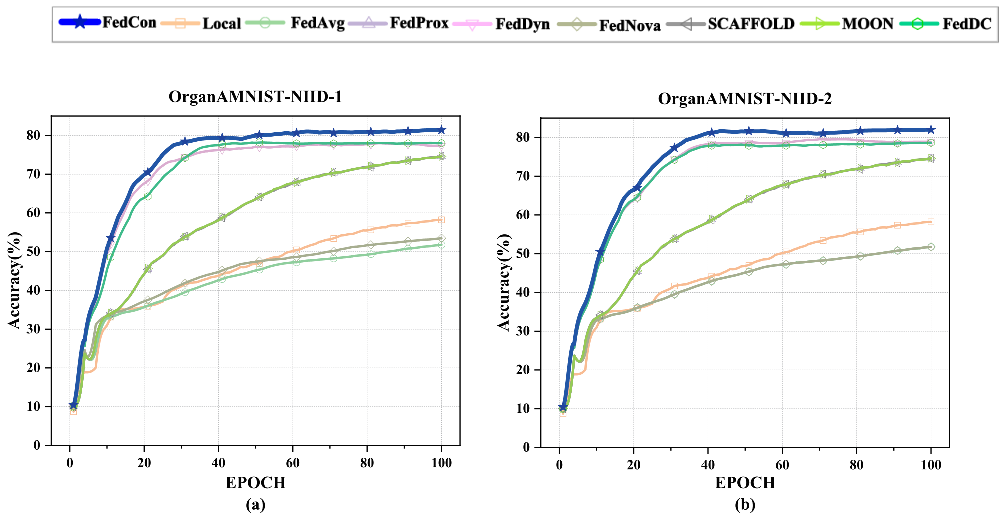

Figure 8. Convergence analysis of different methods on the OrganAMNIST dataset under the NIID-1 and NIID-2 Non-IID data partitioning strategies. (**a**) shows the convergence analysis of OrganAMNIST under NIID-1 settings, and (**b**) shows the convergence analysis of OrganAMNIST under NIID-2 settings.

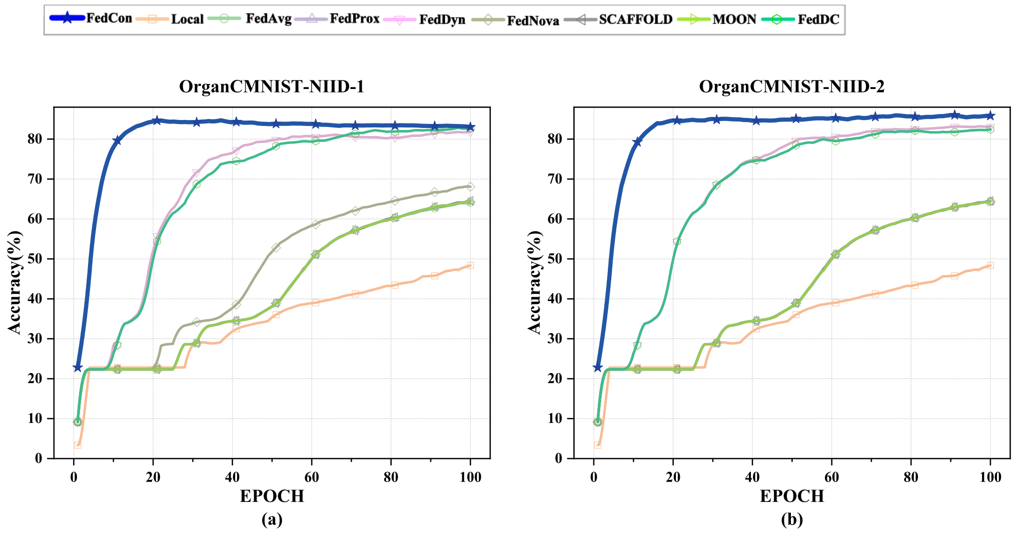

Figure 9. Convergence analysis of different methods on the OrganCMNIST dataset under the NIID-1 and NIID-2 Non-IID data partitioning strategies. (**a**) shows the convergence analysis of OrganCMNIST under NIID-1 settings, and (**b**) shows the convergence analysis of OrganCMNIST under NIID-2 settings.

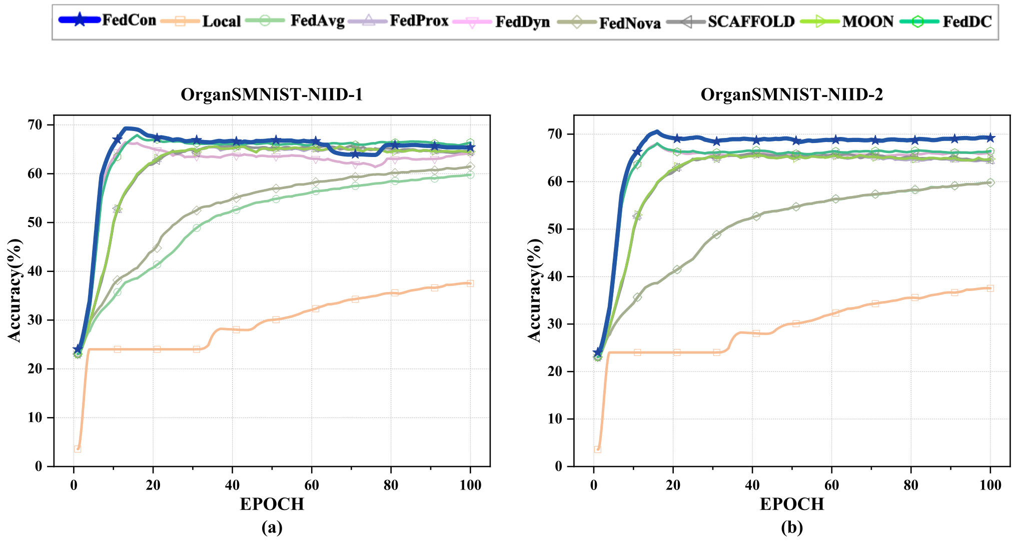

Figure 10. Convergence analysis of different methods on the OrganSMNIST dataset under the NIID-1 and NIID-2 Non-IID data partitioning strategies. (**a**) shows the convergence analysis of OrganSMNIST under NIID-1 settings, and (**b**) shows the convergence analysis of OrganSMNIST under NIID-2 settings.

4.4.3. Stability Analysis

As shown in the Figure 11, FedCon exhibited good stability during 100 rounds of global model iterations. On the CIFAR10 and CIFAR100 datasets, FedCon, compared to the baseline methods, was able to update smoothly, with no significant fluctuations in the convergence process. However, on the HAM10000 and HAR datasets, the baseline methods showed instability in updating, particularly on the HAM10000 dataset, where noticeable fluctuations and oscillations in the model updates occurred.

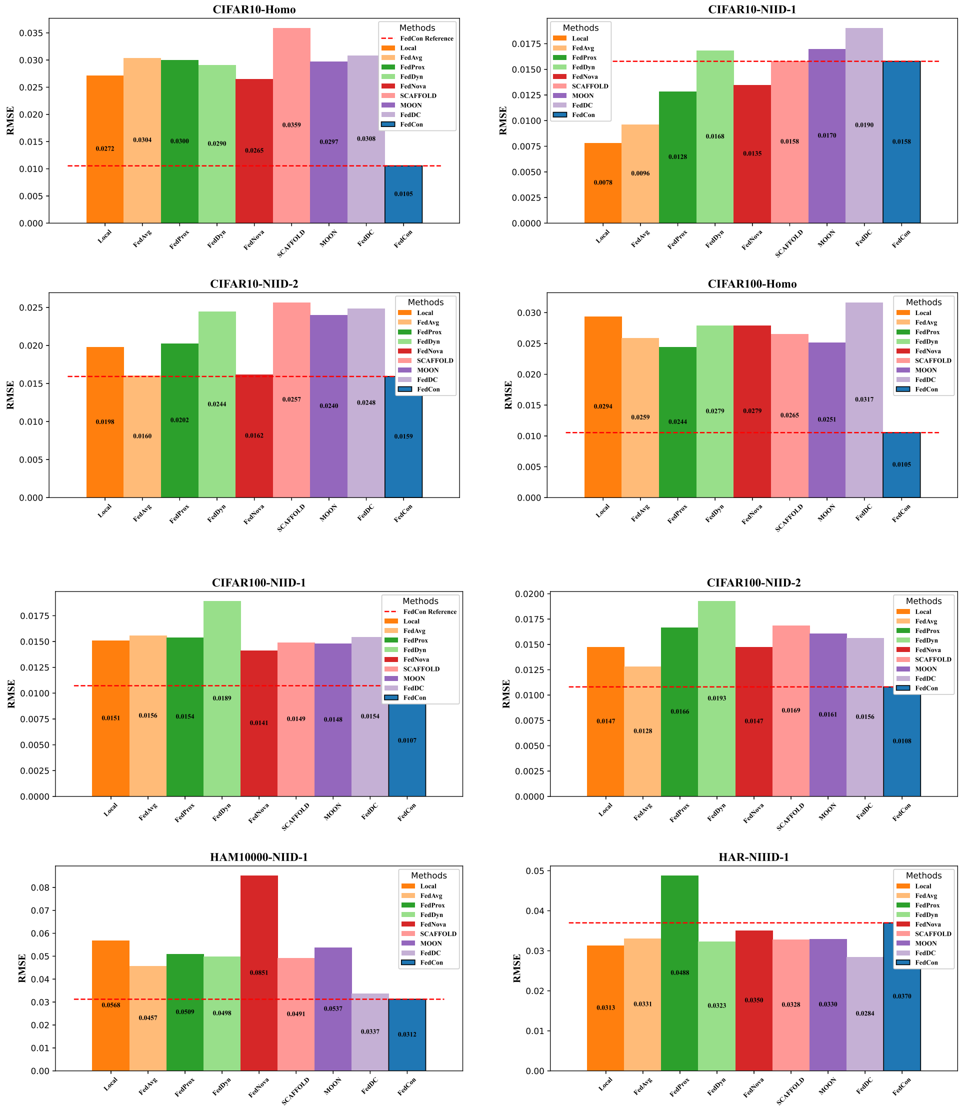

Figure 11. The figure above presents the RMSE of the eight baseline methods and the FedCon method across the different datasets and partitioning strategies. Notably, the performance of FedCon was particularly remarkable on the HAM10000 dataset. On other datasets, FedCon also demonstrated superior performance, highlighting its advanced capabilities.

In contrast, FedCon demonstrated stable updates on both the HAM10000 and HAR datasets, especially on the HAM10000 dataset, where it showed significant advantages. FedCon was able to maintain stable updates in more complex data environments, avoiding the convergence instability issues often encountered by the baseline methods when handling data imbalances and heterogeneous data. This further confirmed the robustness and effectiveness of FedCon in dealing with heterogeneous data.

FedCon effectively enhances the stability of model updates by incorporating a historical weight mechanism. In each round of updates, it combines current weights with historical information, smoothing the update process and reducing fluctuations caused by data imbalances or heterogeneity. Historical weights retain useful information, mitigating the impact of noise or outliers, while a decay mechanism (controlled by α) dynamically balances the importance of current and historical information. This mechanism excels in handling complex data, avoiding oscillations and instability, and improving the convergence performance. The experimental results demonstrate that this approach was indeed effective in achieving stable and robust model updates.

4.4.4. Time Efforts Analysis

Communication overhead is a key metric for evaluating the performance of federated learning. To assess this, we conducted experiments on the OrganSMNIST dataset using both NIID-1 and NIID-2 data partitioning strategies, comparing the performance of FedCon with baseline methods. We calculated the communication time by measuring the model upload and download times between clients and the server, and we also recorded the overall cycle time per round. Based on these two metrics, we performed a detailed analysis.

The results are shown in Figures 12 and 13, which compare the communication time and accuracy, as well as the cycle time and accuracy, for various federated learning algorithms under the NIID-1 and NIID-2 data partitioning strategies. In Figures 12a and 13a, the bar charts represent the communication time per round, while the line charts depict the accuracy trends. Similarly, in Figures 12b and 13b, the bar charts display the cycle time per round, providing a detailed view of the time per round, while the line charts show the model accuracy.

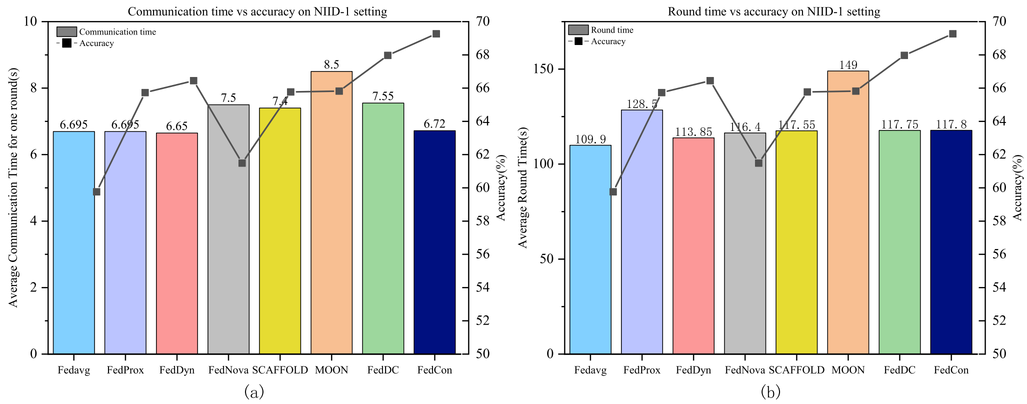

Figure 12. Comparison of communication time, round time, and accuracy for NIID-1 setting. (**a**) shows communication time and accuracy on NIID-1 setting. (**b**) shows round time and accuracy on NIID-1 setting.

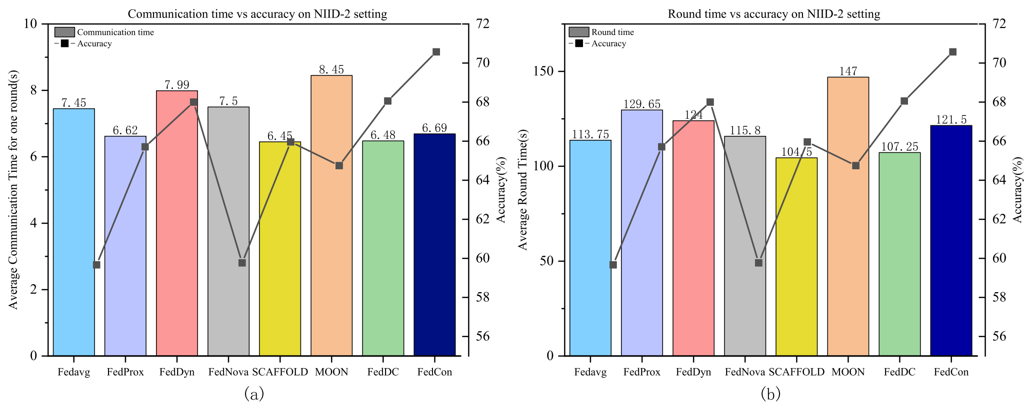

Figure 13. Comparison of communication time, round time, and accuracy for NIID-2 setting. (**a**) shows communication time and accuracy on NIID-2 setting. (**b**) shows round time and accuracy on NIID-2 setting.

As seen from the figures, although the communication time of our proposed FedCon was not the shortest, the performance improvement was acceptable. MOON [19] had the longest communication time. This is because FedCon involves extensive computation on the server side to accurately evaluate client contributions, which slightly increases the time per round. However, compared to FedProx [17] and MOON [19], FedCon still showed an advantage. FedProx [17] and MOON [19] were slower than FedCon, mainly due to the additional computational overhead they introduce. FedProx [17] adjusts client updates with a regularization term, which increases the computational burden. Meanwhile, MOON [19] uses a contrastive learning approach, requiring extra computation and storage to handle model similarities. As a result, both methods took longer per training round, leading to a slower overall convergence.

In summary, FedCon achieved an effective balance between time overhead and accuracy, significantly optimizing the communication efficiency. Although its communication time was not the shortest, the substantial improvement in accuracy fully justified this trade-off. By performing efficient computations on the server side, FedCon can precisely evaluate client contributions, allowing it to outperform baseline methods even with slightly higher communication costs. Given the significant performance gains, the communication cost is acceptable.

4.4.5. Attack Analysis

To evaluate the robustness of FedCon against potential attacks during the transmission of model parameters between clients and the server, we conducted a comprehensive attack analysis. Specifically, we simulated scenarios where a small fraction of clients (10%) are malicious and attempt to disrupt the global model by adding noise to their gradients. Although the noise intensity is moderate, it can still negatively impact the global model's performance. We compared FedCon with FedAvg under both NIID-1 and NIID-2 data partitioning scenarios, with and without attacks, to analyze the robustness of FedCon in adversarial environments.

In our experiments, the results are shown in the Tables 6 and 7, the malicious clients introduced Gaussian noise into their gradients before transmitting them to the server. This simulated real-world scenarios where attackers attempt to compromise the integrity of the federated learning process by injecting small but persistent perturbations. We then analyzed the impact of these attacks on the global model's performance, focusing on metrics such as accuracy. The results demonstrate that FedCon exhibited strong robustness against such attacks. Under both NIID-2 and NIID-2 data partitioning, FedCon maintained a stable model performance and achieved a higher accuracy compared to FedAvg in the presence of malicious clients. The convergence curves show that FedCon's performance degradation was minimal, even when 10% of clients were malicious, while FedAvg experienced significant performance drops under the same conditions. FedCon's robustness can be attributed to its dynamic weight adjustment mechanism, which leverages Shapley values and historical contributions to identify and down-weight clients whose updates deviate significantly from the global model's direction. Additionally, the incorporation of cosine similarity between client and global models further enhances FedCon's ability to detect and isolate malicious updates. These features enable FedCon to mitigate the impact of adversarial behavior and maintain reliable model performance.

In conclusion, FedCon not only addresses the challenges of data heterogeneity but also provides strong resilience against client-side attacks, even in scenarios with a moderate noise intensity. This makes FedCon a robust and secure solution for federated learning in real-world applications where adversarial conditions may exist.

Table 6. Accuracy Drop rates of FedAvg and FedCon with and without attacks under NIID-1 scenario on FMNIST Dataset.

Metrics (NIID-1)	With-Attack	Without-Attack	Gap
FedAvg	82.22	87.85	↓6.41%
FedCon	86.45	87.04	↓0.68%

Table 7. Accuracy drop rates of FedAvg and FedCon with and without attacks under NIID-2 scenario on FMNIST Dataset.

Metrics (NIID-2)	With-Attack	Without-Attack	Gap
FedAvg	82.05	87.48	↓6.21%
FedCon	87.3	89.35	↓2.29%

4.5. Hyperparameter Sensitivity Analysis

In this section, we conducted an ablation study aimed at tuning two hyperparameters, λ and α, in the FedCon algorithm to determine their optimal value ranges. The experiments were conducted on the CIFAR10 dataset with two data partitioning scenarios, NIID-1 and NIID-2. By evaluating the training results of different hyperparameter combinations, we were able to comprehensively analyze the performance of FedCon under these settings. Specifically, λ controls the sensitivity of client contribution evaluation, while α affects the decay of historical weights. Through multiple rounds of experiments, we systematically adjusted these two parameters and recorded the performance of the global model after each adjustment, to identify the optimal hyperparameter combination.

The experimental results Figures 14 and 15 show that the best hyperparameter combination was λ in the range [0.5, 0.7] and α in the range [0.3, 0.4]. This configuration successfully balances high performance with stability and convergence speed during the training process.

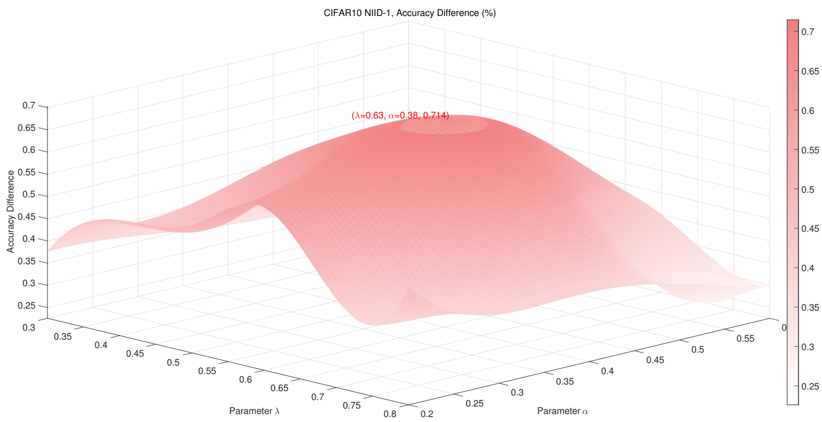

Figure 14. The above figure shows the hyperparameter analysis under CIFAR10 with the NIID-1 data partitioning.

Further analysis revealed that when the value of λ is too small ($\lambda < 0.5$), the sensitivity of client contribution evaluation is insufficient, leading to a decline in model performance. Conversely, when the value of λ is too large ($\lambda > 0.7$), the model becomes susceptible to interference from noisy data, which affects its stability. Similarly, when the value of α is too small ($\alpha < 0.3$), the decay of historical weights is too rapid, making it difficult for the model to fully utilize historical information. On the other hand, when the value of α is too large ($\alpha > 0.4$), the influence of historical weights becomes too strong, potentially slowing down the convergence speed of the model.

This ablation experiment not only provided the optimal hyperparameter selection for the FedCon algorithm but also lays a solid foundation for subsequent experiments and applications. By determining the optimal ranges for λ and α, we can ensure that FedCon performs excellently in various Non-IID scenarios, thereby offering strong support for further research in the field of federated learning.

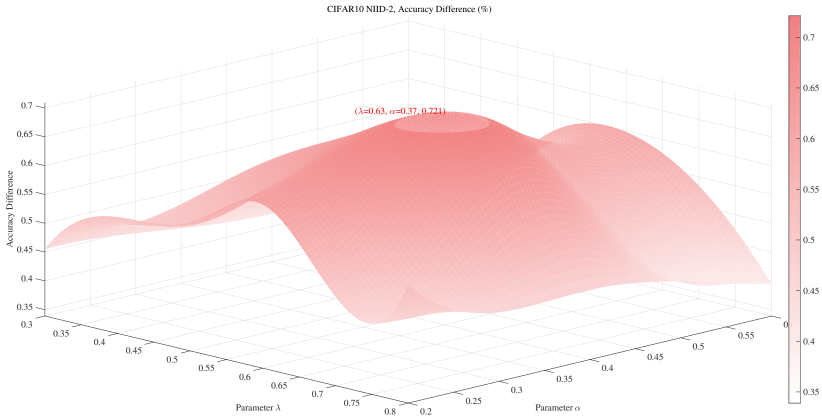

Figure 15. The above figure shows the hyperparameter analysis under CIFAR10 with the NIID-2 data partitioning.

4.6. Ablation Experiment

In this section, we conducted an ablation study to investigate the impact of each component of the proposed FedCon method, specifically the initial weight calculation, Shapley value computation, and cosine similarity for client contribution evaluation. To better understand the contribution of each component, we divided FedCon into two key parts: Part A, which focused on the importance of the initial weight calculation, and Part B, which dealt with the precise client contribution evaluation through Shapley values and simplified cosine similarity.

We designed three experiments to evaluate these parts:

- FedCon without Part A: Excluding the initial weight calculation.
- FedCon without Part B: Excluding Shapley value computation and cosine similarity.
- Full FedCon: Including all components.

Additionally, we performed baseline experiments with the same setup for comparison. The results are presented in violin plots to clearly illustrate the performance differences across different configurations. This ablation study provided a detailed analysis of how each component contributes to the overall performance of FedCon. From the Figure 16, it is evident that the contributions of the different components to the performance of FedCon were quite significant. In particular, the initialization weights had a notable impact on improving the model. The two components complement each other, and FedCon achieved a good performance across various data partitioning strategies by effectively combining both parts.

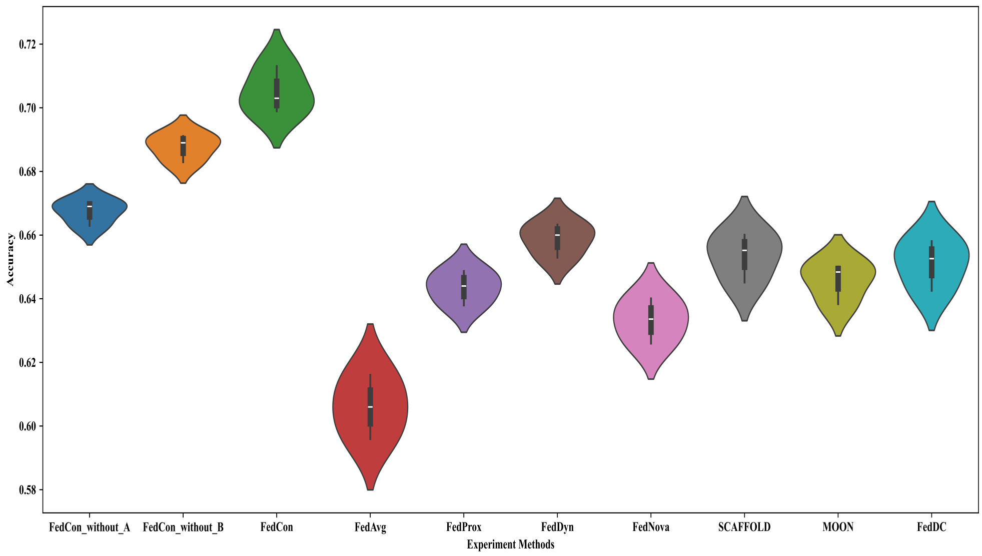

Figure 16. Violin plot comparing FedCon without A, FedCon without B, FedCon, and other baseline methods.

5. Conclusions and Future Work

In this paper, we proposed FedCon, a novel global model aggregation method designed to enhance convergence stability and performance in heterogeneous data environments. Experimental results demonstrated that FedCon significantly improved model accuracy and stability across multiple datasets, particularly in Non-IID scenarios. A key innovation of FedCon is its precise evaluation of client contributions, achieved by adjusting sampling frequency based on a similarity matrix between local and global models. Monte Carlo sampling is employed to efficiently compute Shapley values, reducing computational complexity. Additionally, FedCon achieved faster convergence and smaller fluctuations over 100 communication rounds by leveraging weight initialization and historical weights. The cosine similarity matrix ensures consistent update directions, further enhancing the aggregation stability. FedCon achieved high model accuracy and stability, but its computational cost is suboptimal, especially for Shapley value calculation and dynamic client weight adjustment. While the sampling strategy improves client contribution evaluation, it is not perfectly fair, particularly under highly imbalanced client data distributions. These limitations hinder FedCon's applicability to large-scale federated learning. Future work should prioritize optimizing fairness and computational efficiency to enhance FedCon's practicality and scalability.

Author Contributions: Conceptualization, W.G. and G.X.; methodology, W.G.; software, W.G.; validation, W.G. and G.X.; formal analysis, W.G.; investigation, W.G.; resources, G.X.; data curation, W.G.; writing—original draft preparation, W.G.; writing—review and editing, W.G.; visualization, W.G.; supervision, G.X.; project administration, X.M.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Development Project of Jilin Province of China (No. 20200401076GX).

Data Availability Statement: The data used in this study are publicly available datasets and can be found in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R. Advances and open problems in federated learning. *Found. Trends® Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
2. Xu, J.; Glicksberg, B.S.; Su, C.; Walker, P.; Bian, J.; Wang, F. Federated learning for healthcare informatics. *J. Healthc. Inform. Res.* **2021**, *5*, 1–19. [CrossRef] [PubMed]
3. Chatterjee, P.; Das, D.; Rawat, D. Securing financial transactions: Exploring the role of federated learning and blockchain in credit card fraud detection. 2023, *preprint*.
4. Chronis, C.; Varlamis, I.; Himeur, Y.; Sayed, A.N.; Al-Hasan, T.M.; Nhlabatsi, A.; Bensaali, F.; Dimitrakopoulos, G. A survey on the use of Federated Learning in Privacy-Preserving Recommender Systems. *IEEE Open J. Comput. Soc.* **2024**, *5*, 227–247. [CrossRef]
5. Ullah, F.; Srivastava, G.; Xiao, H.; Ullah, S.; Lin, J.C.-W.; Zhao, Y. A scalable federated learning approach for collaborative smart healthcare systems with intermittent clients using medical imaging. *IEEE J. Biomed. Health Inform.* **2023**, *28*, 3293–3304. [CrossRef]
6. Li, J.; Sun, G.; Duan, L.; Wu, Q. Multi-objective optimization for UAV swarm-assisted IoT with virtual antenna arrays. *IEEE Trans. Mob. Comput.* **2023**, *23*, 4890–4907. [CrossRef]
7. Li, J.; Sun, G.; Wu, Q.; Niyato, D.; Kang, J.; Jamalipour, A.; Leung, V.C. Collaborative ground-space communications via evolutionary multi-objective deep reinforcement learning. *IEEE J. Sel. Areas Commun.* **2024**, *42*, 3395–3411. [CrossRef]
8. Li, L.; Xu, G.; Liu, Z.; Ge, J.; Jiang, W.; Li, J. Task execution latency minimization for energy-sensitive IoTs in wireless powered mobile edge computing: A DRL-based method. *Comput. Netw.* **2024**, *251*, 110633. [CrossRef]
9. Li, Q.; Diao, Y.; Chen, Q.; He, B. Federated learning on non-iid data silos: An experimental study. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 965–978.
10. Luo, M.; Chen, F.; Hu, D.; Zhang, Y.; Liang, J.; Feng, J. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 5972–5984.
11. Zhu, H.; Xu, J.; Liu, S.; Jin, Y. Federated learning on non-IID data: A survey. *Neurocomputing* **2021**, *465*, 371–390. [CrossRef]
12. Dai, Y.; Chen, Z.; Li, J.; Heinecke, S.; Sun, L.; Xu, R. Tackling data heterogeneity in federated learning with class prototypes. In Proceedings of the AAAI Conference on Artificial Intelligence, Montréal, QC, Canada, 8–10 August 2023; pp. 7314–7322.
13. Guan, H.; Yap, P.-T.; Bozoki, A.; Liu, M. Federated learning for medical image analysis: A survey. *Pattern Recognition* **2024**, *151*, 110424. [CrossRef] [PubMed]
14. Chen, H.; Zhang, Y.; Krompass, D.; Gu, J.; Tresp, V. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; pp. 11285–11293.
15. Ye, M.; Fang, X.; Du, B.; Yuen, P.C.; Tao, D. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Comput. Surv.* **2023**, *56*, 1–44. [CrossRef]
16. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
17. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
18. Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; Khazaeni, Y. Federated learning with matched averaging. *arXiv* **2020**, arXiv:2002.06440.
19. Li, Q.; He, B.; Song, D. Model-contrastive federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10713–10722.
20. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. Scaffold: Stochastic controlled averaging for federated learning. In Proceedings of the International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 5132–5143.
21. Kim, D.; Woo, H.; Lee, Y. Addressing Bias and Fairness Using Fair Federated Learning: A Synthetic Review. *Electronics* **2024**, *13*, 4664. [CrossRef]
22. Liu, W.; Chen, L.; Chen, Y.; Zhang, W. Accelerating federated learning via momentum gradient descent. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 1754–1766. [CrossRef]
23. Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; Poor, H.V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7611–7623.
24. Gao, L.; Fu, H.; Li, L.; Chen, Y.; Xu, M.; Xu, C.-Z. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10112–10121.
25. Liu, Z.; Chen, Y.; Zhao, Y.; Yu, H.; Liu, Y.; Bao, R.; Jiang, J.; Nie, Z.; Xu, Q.; Yang, Q. Contribution-aware federated learning for smart healthcare. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Conference, 22 February–1 March 2022; pp. 12396–12404.

26. Zhang, W.; Wang, X.; Zhou, P.; Wu, W.; Zhang, X. Client selection for federated learning with non-iid data in mobile edge computing. *IEEE Access* **2021**, *9*, 24462–24474. [CrossRef]
27. Liu, Z.; Chen, Y.; Yu, H.; Liu, Y.; Cui, L. Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning. *ACM Trans. Intell. Syst. Technol. (TIST)* **2022**, *13*, 1–21. [CrossRef]
28. Wang, G.; Dang, C.X.; Zhou, Z. Measure contribution of participants in federated learning. InProceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2597–2604.
29. Ye, R.; Xu, M.; Wang, J.; Xu, C.; Chen, S.; Wang, Y. Feddisco: Federated learning with discrepancy-aware collaboration. In Proceedings of the International Conference on Machine Learning, Zhuhai China, 17–20 February 2023; pp. 39879–39902.
30. Sattler, F.; Müller, K.-R.; Wiegand, T.; Samek, W. On the byzantine robustness of clustered federated learning. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 8861–8865.
31. Acar, D.A.E.; Zhao, Y.; Navarro, R.M.; Mattina, M.; Whatmough, P.N.; Saligrama, V. Federated learning based on dynamic regularization. *arXiv* **2021**, arXiv:2111.04263.
32. Li, Z.; Chen, Z.; Wei, X.; Gao, S.; Yue, H.; Xu, Z.; Quek, T.Q. Exploiting Complex Network-based Clustering for Personalization-Enhanced Hierarchical Federated Edge Learning. *IEEE Trans. Mob. Comput.* **2024**, *23*, 14852–14870. [CrossRef]
33. Li, Z.; Chen, Z.; Wei, X.; Gao, S.; Ren, C.; Quek, T.Q. HPFL-CN: Communication-efficient hierarchical personalized federated edge learning via complex network feature clustering. In Proceedings of the 2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Stockholm, Sweden, 20–23 September 2022; pp. 325–333.
34. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.; Poor, H.V. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
35. Kalapaaking, A.P.; Khalil, I.; Yi, X. Blockchain-based federated learning with SMPC model verification against poisoning attack for healthcare systems. *IEEE Trans. Emerg. Top. Comput.*, **2023**, *12*, 269–280. [CrossRef]
36. Ezzeldin, Y.H.; Yan, S.; He, C.; Ferrara, E.; Avestimehr, A.S. Fairfed: Enabling group fairness in federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Montréal, QC, Canada, 8–10 August 2023; pp. 7494–7502.
37. Lyu, L.; Xu, X.; Wang, Q.; Yu, H. Collaborative fairness in federated learning. In *Federated Learning: Privacy and Incentive*; Springer: Cham, Switzerland, 2020; pp. 189–204.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.