Analysis of Non-IID Data Processing and Model Generalization in Federated Learning

Cao shihao

January 1, 2025

Abstract

With increasing concerns about data privacy and security, federated learning has emerged as a promising distributed machine learning approach, serving as a key technology to address data privacy protection and efficient collaborative learning. However, one of the core challenges in federated learning is how to handle non-independent and identically distributed (Non-IID) data. Due to differences in data distributions across devices or users, the training and generalization performance of federated learning models are often significantly impacted. This paper explores the issue of Non-IID data in federated learning and analyzes the performance of two representative algorithms—MOON and FedAvg—under Non-IID conditions. Through experimental reproduction and comparative analysis of these algorithms, the study evaluates their differences in training effectiveness, convergence speed, and generalization ability. Experimental results demonstrate that while both algorithms can mitigate the negative effects of Non-IID data to some extent, they exhibit significant performance fluctuations in highly heterogeneous data environments and show weaker generalization on certain devices. This study provides a systematic experimental analysis of Non-IID data handling in federated learning, revealing the strengths and limitations of existing algorithms and proposing future optimization directions. With the widespread adoption of edge computing and intelligent devices, federated learning is expected to promote more personalized and efficient model training while ensuring data privacy. Future research will continue to explore ways to enhance federated learning performance in more complex environments, particularly in scenarios with high device heterogeneity and significant data distribution differences.

Keywords: Federated Learning, Non-Independent and Identically Distributed, MOON Algorithm, FedAvg Algorithm, Data Privacy.

1 Introduction

1.1 Background and Significance of Federated Learning

With the rapid development of information technology, the way data is generated and stored has undergone significant changes. Driven by smart devices, the Internet of Things, and mobile applications, massive amounts of user data are now distributed across different devices and regions. How to effectively utilize this data for model training while ensuring data privacy and security has become a critical challenge in the field of artificial

intelligence. Federated Learning (FL) [1], as an emerging distributed machine learning approach, offers an effective solution. Unlike traditional centralized learning methods, Federated Learning enables local model training on user devices and uploads only model parameters rather than the data itself to a central server. This approach significantly reduces the risk of data privacy breaches and complies with data privacy protection regulations. Figure 1 illustrates the framework of the Federated Learning process.

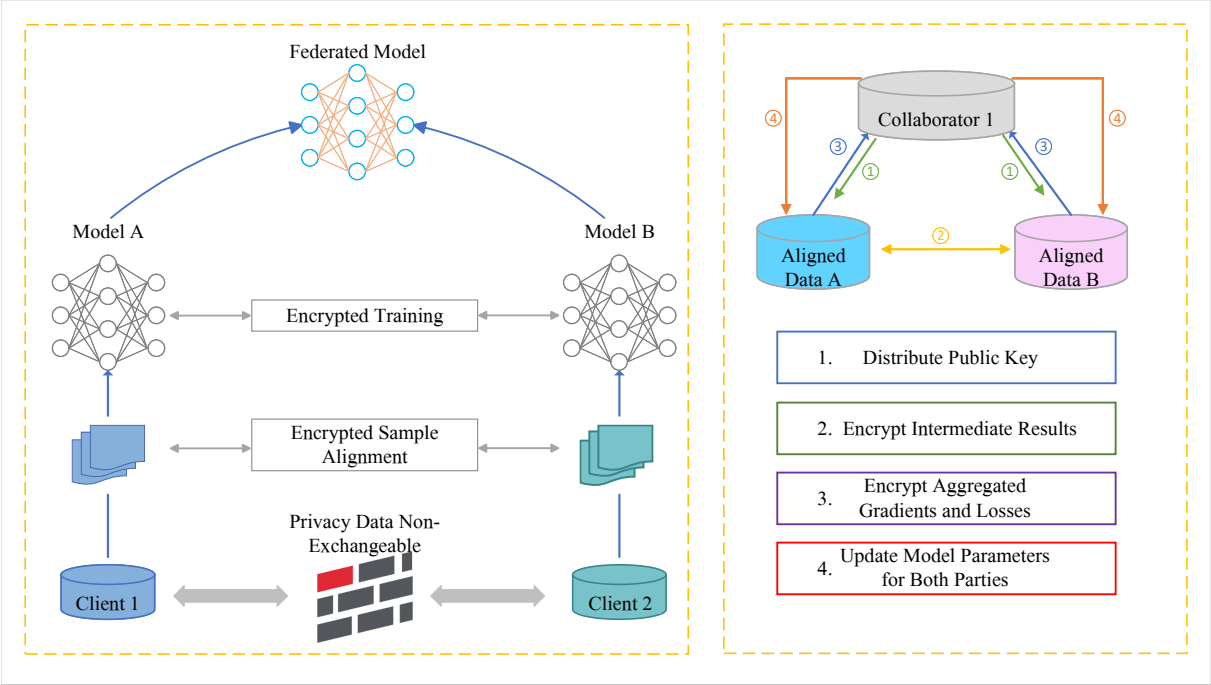

Figure 1. Federated Learning Framework Diagram

Federated Learning was first proposed by Google in 2016 [2] to address challenges such as data decentralization, privacy protection, and cross-device learning. Its core idea is to shift machine learning tasks from centralized data storage to distributed device endpoints. Each device performs model training locally, and only the trained model parameters are aggregated on a central server without transmitting the original data. This approach not only avoids the privacy risks associated with centralized storage and processing of user data in traditional methods but also enhances training efficiency on edge devices.

As data privacy concerns have become increasingly prominent, especially with the introduction of privacy protection regulations such as GDPR [3], Federated Learning's advantages in safeguarding user privacy and ensuring compliance with data regulations have become particularly evident. In industries like healthcare, finance, and the Internet of Things (IoT), data is often distributed across different devices or organizations, and this data is typically highly sensitive. Federated Learning offers these industries an innovative solution that enables effective model training while preserving data privacy. Furthermore, by reducing the need for data transmission, Federated Learning helps alleviate network bandwidth pressure and enhances the real-time performance and efficiency of training to some extent.

1.2 Challenges of Non-IID Data

Although Federated Learning provides significant assurance for data privacy protection, it also faces substantial challenges. Traditional machine learning and deep learning typically assume that data is independently

and identically distributed (IID), meaning that the training data distribution is uniform globally. However, in real-world Federated Learning scenarios, data is often non-independently and identically distributed (Non-IID), making model training and generalization more complex [4].

In Federated Learning applications, inconsistencies in data distribution arise from multiple factors, including device heterogeneity, user behavior differences, and the regional or temporal nature of the data [5]. For example, text data, sensor data, and medical records collected from different user devices often exhibit significant differences in feature distribution. These disparities may lead to poor performance of the global model on certain devices or even prevent convergence on some devices. Moreover, the presence of Non-IID data often makes global model training more unstable, affecting the model's convergence speed and generalization ability.

To address the challenges posed by Non-IID data, researchers have proposed various improvement methods, such as data resampling [6], differentiated optimization algorithms [7], and adversarial training [8]. However, the effectiveness of these methods remains controversial across different application scenarios and often requires high computational and storage costs. Therefore, designing effective algorithms to handle Non-IID data and enhance the generalization ability of Federated Learning models remains a critical topic in current Federated Learning research.

1.3 Model Generalization in Federated Learning

Model generalization refers to a trained model's ability to perform well on unseen new data. High generalization capability means that a model not only achieves good performance on training data but also maintains stable performance on test data or in real-world applications [9]. In Federated Learning, model generalization is particularly crucial because the data on different devices may have distinct characteristics and distributions. Models trained using traditional centralized learning methods may struggle to adapt to these distributional differences [10]. Therefore, improving the generalization ability of Federated Learning models on Non-IID data has become a key research focus.

To enhance model generalization, researchers have proposed various approaches. On one hand, differentiated optimization and updates on the device side, such as personalized model training [11], can help each device better adapt to its local data distribution. On the other hand, model aggregation strategies [12] have garnered significant attention, including methods like weighted averaging and gradient aggregation. These strategies aim to mitigate the impact of data distribution differences across devices on the global model.

While these methods can alleviate the Non-IID problem to some extent, numerous challenges remain. Particularly in cross-device environments, balancing the trade-offs between local and global learning and addressing biases in data distribution are critical issues that warrant further investigation.

1.4 Objectives and Significance of This Study

This study aims to deeply analyze the challenges of handling Non-IID data in Federated Learning and evaluate the generalization capabilities of different algorithms when confronted with such data. By reproducing representative algorithms in the field of Federated Learning, such as MOON and FedAvg, this paper conducts experimental evaluations of their performance in handling Non-IID data and analyzes their strengths and weaknesses. By comparing the experimental results of different algorithms, this study seeks to provide

a theoretical foundation for optimizing Federated Learning algorithms and offer guidance for future research directions.

Through this study, we aim to gain a better understanding of the advantages and challenges of Federated Learning models under Non-IID data conditions and provide new ideas for addressing practical problems. With the continuous growth of smart devices and large-scale distributed data, Federated Learning is expected to play an increasingly important role in various fields, particularly in data privacy protection and cross-device collaborative learning. Therefore, enhancing the generalization capabilities of Federated Learning algorithms, especially in handling Non-IID data, will have a profound impact on future artificial intelligence applications.

This chapter has briefly introduced the background and significance of Federated Learning, highlighted its importance in data privacy protection, and discussed the challenges posed by Non-IID data to the generalization capabilities of Federated Learning models. As Federated Learning technology continues to evolve, effectively handling Non-IID data and improving model generalization capabilities have become critical research directions in this field. The following chapters will provide a detailed overview of Federated Learning technologies, experimental design and methodology, and an analysis and discussion of experimental results, offering a comprehensive analytical framework for understanding this issue.

2 Overview of Federated Learning Technology

2.1 Definition and Architecture of Federated Learning

Federated Learning is a distributed machine learning [13] approach that enables multiple devices or clients located in different geographic regions to collaboratively train a global model without uploading local data to a central server. This approach addresses the challenges of data privacy and security. Unlike traditional centralized machine learning methods, Federated Learning distributes computational tasks to individual devices, where each device trains the model using its local data. Only the trained model parameters or updated gradients are sent to a central server, which aggregates this information to update the global model.

The basic architecture of Federated Learning typically consists of the following core components:

1. **Clients**: These devices are responsible for local data storage and model training. The data stored locally on each client often has different distributions (e.g., medical data, mobile application data) and does not leave the client. Each client trains a model using its local data and uploads the updated model parameters to the central server.

2. **Central Server**: The central server coordinates communication between clients, aggregates model parameters uploaded from clients, and generates an updated global model. Typically, the server uses aggregation algorithms (e.g., FedAvg) to compute a weighted average of the client models, resulting in a globally consistent model.

3. **Communication Protocols**: Communication protocols play a crucial role in Federated Learning by ensuring secure and efficient data transmission between clients and the server. Since data is not directly transmitted, these protocols mainly handle the transfer of model parameters or gradient updates, thereby preventing data leakage.

4. **Aggregation Algorithms**: Aggregation algorithms are central to Federated Learning [14]. They typically use averaging methods (e.g., FedAvg) to combine updates from clients into a global model. Effectively balancing the model updates from different clients to meet the needs of the global model is an important research direction.

As illustrated in Figure 2, the process flow diagram of the core components of Federated Learning demonstrates the iterative interactions between clients and the server. This diagram showcases how clients use local data to train models, upload updates to the server, and how the server aggregates these updates and distributes a new global model.

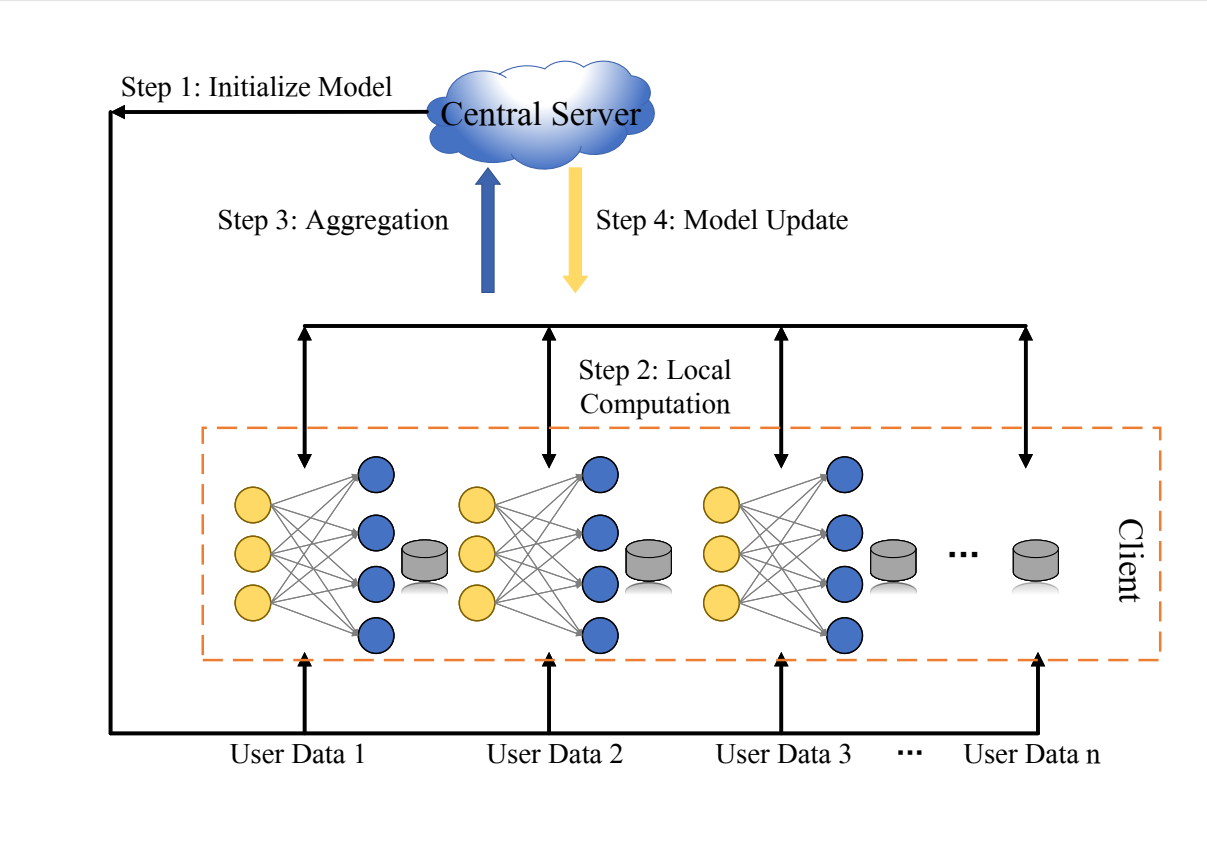

Figure 2. User-Server Learning Process

By referring to the training steps of Algorithm 1, we can clearly see how clients and the server collaborate in Federated Learning and the critical role aggregation algorithms play in the entire process. This distributed learning approach not only protects user data privacy but also enhances the efficiency and personalization of model training.

Algorithm 1 User-Server Model Training Steps

Input: Initial global model parameters $\mathbf{w}_0$

Output: Converged global model parameters $\mathbf{w}_T$

Procedure:

for each communication round $t = 1, 2, \ldots, T$ **do**

Step 1: Local users download the initialized global model $\mathbf{w}_{t-1}$ from the central server.

Step 2: Each user trains the global model $\mathbf{w}_{t-1}$ on their local data to obtain local model parameters $\mathbf{w}_i^{(t)}$.

Step 3: Local users upload the updated model parameters $\mathbf{w}_i^{(t)}$ to the central server.

Step 4: The central server aggregates the local model parameters from all users:

$$\mathbf{w}_t = \sum_{i=1}^{N} \frac{n_i}{n} \mathbf{w}_i^{(t)}$$

where n_i denotes the number of data samples for user i, and n is the total number of data samples across all users.

end for

Return: The converged global model parameters $\mathbf{w}_T$.

In summary, Federated Learning achieves a balance between data privacy preservation and model performance improvement through its unique distributed architecture. By processing data locally while enabling global model learning, it reduces reliance on centralized data storage, thus lowering the risk of data breaches. Moreover, training on local devices enhances data processing efficiency and improves the model's adaptability to local data characteristics. Federated Learning offers a secure and efficient solution for industries with stringent data privacy requirements, such as healthcare, financial services, and personal smart devices. It enables cross-device data utilization and knowledge sharing without compromising user privacy. With ongoing technological advancements and deeper applications, Federated Learning is poised to drive the development of personalized services and intelligent applications while safeguarding privacy.

2.2 The Importance of Federated Learning in Data Privacy and Regulatory Compliance

With the increasing global focus on data privacy and security, particularly driven by privacy protection regulations such as the GDPR, ensuring the security and privacy of user data has become a key concern across industries [15]. In this context, Federated Learning, as a novel distributed machine learning technology, provides an effective solution. Unlike traditional centralized learning methods, Federated Learning allows data to be processed and trained on local devices without uploading it to a central server. This design significantly reduces the risk of data leakage and effectively protects user privacy.

The core advantage of Federated Learning is that data remains on the device at all times, and the training process is entirely local. Traditional centralized learning methods rely on gathering all data on a server for centralized processing, which not only increases the risk of data leakage but also incurs high costs for data storage and transmission. In Federated Learning, data never leaves the local device; all data storage and processing stay on the user's device, effectively safeguarding privacy. Due to local storage and computation, Federated

Learning avoids potential leakage issues associated with centralized storage and complies with the strict data privacy requirements of many regions, particularly privacy protection regulations like the EU GDPR, which mandate that companies ensure the security and privacy of user data when processing it.

In addition, Federated Learning implements the principle of minimal sharing during the data sharing process [16]. Unlike traditional methods, Federated Learning only requires the transmission of model parameters or gradients, rather than raw data. This means that even if the data is intercepted during transmission, only the model parameters are exposed, not the sensitive user data. Therefore, even if the model updates are maliciously obtained, the risk of original data leakage is still avoided, which is of significant importance for complying with privacy protection regulations like the GDPR. User data always remains local and is not misused, giving Federated Learning a natural advantage in addressing data privacy concerns.

With the rise of multinational corporations and organizations, data privacy regulations have become increasingly complex on a global scale, with varying requirements for data storage and processing across different countries and regions [17]. Federated learning offers these enterprises a compliant method of data processing. In industries with stringent data privacy requirements, such as healthcare [18] and finance [19], federated learning ensures that data storage and processing align with local legal and regulatory standards. For instance, some countries impose strict restrictions on cross-border data transfers. Federated learning addresses this issue effectively by eliminating the need for data transfer. Through distributed computing, federated learning not only keeps data on users' local devices but also ensures compliance with regional regulations, mitigating the legal risks associated with cross-border data transfers.

Finally, Federated Learning strikes a balance between personalized services and privacy protection. In traditional machine learning, personalized models often require the use of private user data, which brings the risk of data privacy leakage [20]. However, Federated Learning ensures personalized training on local devices, allowing each device to train on its own specific data, generating personalized models without involving data leakage. This approach offers more personalized services tailored to user needs, improving the user experience while avoiding the risk of data misuse.

In summary, Federated Learning provides an innovative solution for data privacy protection and regulatory compliance. It not only makes data storage and processing more secure but also helps enterprises and organizations comply with privacy protection regulations across different regions globally. With the continuous development of technology, Federated Learning is expected to be applied in more industries, driving further advancements in privacy protection and personalized services in intelligent devices and distributed computing environments.

3 Experimental Design and Methodology

3.1 Literature Review and Article Selection

In this course, conducting a literature review of top conferences and relevant papers is crucial for better understanding Federated Learning. To deeply explore Federated Learning and its applications in Non-IID (Non-Independent and Identically Distributed) data processing, this experiment reproduction refers to three influential papers from top conferences in the field of Federated Learning. The selection of these articles is based on their broad impact on Federated Learning technology, algorithm optimization, and applications, and they

provide important theoretical foundations and technical support for the experimental design and methodology of this paper.

The first paper is "Scaffold: Stochastic Controlled Averaging for Federated Learning," published at the 2020 ICML conference [21]. This paper introduces the SCAFFOLD algorithm, which aims to address the gradient discrepancy caused by client data heterogeneity in Federated Learning. The SCAFFOLD algorithm effectively corrects client drift by incorporating a stochastic controlled averaging method and demonstrates its convergence on arbitrarily heterogeneous data. The paper also highlights the advantages of the SCAFFOLD algorithm in reducing communication requirements, particularly when there is similarity between clients. This paper provided valuable insights into the challenges in Federated Learning and guided the direction of this study.

The second paper is "Model-Contrastive Federated Learning," published at the 2021 CVPR conference [22]. This paper introduces the application of model contrastive learning in Federated Learning, aiming to improve Federated Learning performance by contrasting representations learned by different models. The paper suggests that the global model can extract better feature representations than the local models because the global model is trained using a more comprehensive dataset. The paper experimentally verifies this intuition and proposes a model contrastive loss function to promote consistency between the representations of different models. This work provides a new perspective on feature representation learning in Federated Learning and offers important theoretical support for the experimental design in this paper.

The third paper is "Federated Learning on Non-IID Data Silos: An Experimental Study," published in 2020 [23]. This paper systematically explores the impact of different Non-IID (Non-Independent and Identically Distributed) data partitioning strategies on the performance of Federated Learning algorithms by introducing NIIDBench. The paper presents six different Non-IID data partitioning strategies and conducts extensive experimental evaluations of four state-of-the-art Federated Learning algorithms on nine datasets. The experimental results provide insights for the future development of Federated Learning algorithms, especially in addressing the challenges posed by Non-IID data distributions. This paper provides a solid foundation for understanding how Federated Learning technologies respond to real-world challenges.

These three papers make significant contributions to Federated Learning from different perspectives, providing essential background knowledge and theoretical support for this research. The work in "Scaffold" offers an effective algorithm for handling client heterogeneity, while "Model-Contrastive Federated Learning" explores new methods for feature representation learning. "Federated Learning on Non-IID Data Silos: An Experimental Study" provides experimental research and evaluation for Federated Learning in Non-IID data environments. The selection of these papers helps us gain a deep understanding of the development of Federated Learning technologies and the strategies used to address real-world challenges, thus providing a solid foundation for the subsequent experimental design and methodology.

3.2 Experimental Environment Setup

This experiment was conducted on the Windows 11 operating system. To ensure compatibility and stability, this platform was selected. Windows 11 provides good support for deep learning and scientific computing tasks, offering stable performance, especially when running various data analysis and machine learning tools.

In terms of hardware, a GPU that supports CUDA 11.0 was used, providing significant acceleration for

deep learning tasks. CUDA (Compute Unified Device Architecture) is a parallel computing platform developed by NVIDIA, which efficiently utilizes the GPU for high-performance computing, especially for training large-scale datasets and complex models, significantly reducing training time. The use of CUDA enabled the completion of large-scale dataset model training in a shorter time and ensured the efficiency of the training process.

For software configuration, Python 3.6.20 was used as the primary programming language for scientific computing and machine learning. Python 3.6.20 is a relatively mature version, widely compatible with machine learning libraries and deep learning frameworks, enabling smooth execution of various computational tasks required for this experiment. The ease of use and extensive library support of Python make it the language of choice in data science and deep learning fields. Additionally, Python 3.6.20 ensures compatibility with various tools and frameworks, enabling efficient execution when handling large datasets.

The choice of foundational libraries is crucial for the smooth progression of the experiment. Firstly, NumPy version 1.19.5 was used, one of the most commonly used scientific computing libraries in Python, which provides efficient multidimensional array objects and mathematical functions. NumPy's high-performance matrix operations and support for large datasets enabled efficient data preprocessing and computation during the experiment. Secondly, scikit-learn version 0.23.2 was selected. This library offers various machine learning algorithms and data processing tools, useful for model evaluation and hyperparameter tuning in the experiment. Finally, SciPy version 1.5.4 was employed to support more advanced mathematical computations, especially in optimization and statistical analysis. The close integration of SciPy and NumPy ensured the stable and efficient execution of various computational tasks during the experiment.

The selection of deep learning frameworks is a key factor in this experiment. PyTorch version 1.0.0 and torchvision version 0.2.1 were used. PyTorch, as an open-source deep learning framework, is widely used in both academia and industry. Its dynamic graph feature makes model debugging and development more flexible, especially suitable for experimental research. torchvision, as a computer vision library for PyTorch, provides a large number of pre-trained models, image transformation tools, and dataset handling functions, making it ideal for image classification tasks.

This combination of software configuration and dependency libraries provided an efficient and stable computing environment for the experiment, ensuring that deep learning tasks could run smoothly.

3.3 Dataset and Model Selection

In the experiment, we used three commonly used datasets: CIFAR-100, CIFAR-10, and MNIST. These datasets are widely used to evaluate the performance of algorithms in image classification tasks. Each dataset has different characteristics, enabling the testing of model performance under various difficulty levels and diverse tasks, especially under non-independent and identically distributed (Non-IID) data.

- CIFAR-10 [24]: CIFAR-10 is an image dataset containing 10 categories, with 6000 32x32 color images per category, for a total of 60,000 images, divided into training and test sets. Compared to CIFAR-100, CIFAR-10 has fewer categories, making it suitable for quickly validating and debugging models. Its relatively simple task makes it an ideal choice for evaluating algorithm performance in a short period of time. Due to its fewer categories, CIFAR-10 is especially suitable for preliminary evaluation of model accuracy and training effectiveness, particularly in the early stages of the experiment.

- CIFAR-100 [24]: CIFAR-100 is an image dataset containing 100 categories, with 600 32x32 color images per category, for a total of 60,000 images, divided into training and test sets. With its 100 categories, CIFAR-100 represents a multi-class task with a higher level of difficulty. The lower resolution of each image imposes higher demands on the model's accuracy and generalization ability. In this experiment, CIFAR-100 was selected to evaluate the model's performance in handling large-scale multi-class tasks, especially in environments with uneven data distribution, testing the capability of the MOON algorithm.

- Tiny-ImageNet [25]: Tiny-ImageNet is a relatively large image dataset containing 200 categories, with 500 training images and 50 validation images per category, totaling approximately 100,000 training images. Each image has a resolution of 64x64, with more complex content and higher resolution. Compared to CIFAR-100 and CIFAR-10, Tiny-ImageNet has more categories and higher image resolution, making it better suited for testing deep learning models on more complex datasets. For evaluating the performance of the MOON algorithm on higher-resolution images and multi-class tasks, Tiny-ImageNet is an ideal dataset.

In the data preprocessing phase, we standardized the image data, ensuring that each data point has a mean of 0 and a standard deviation of 1. This processing method helps accelerate the training process and improve the model's convergence speed. For the federated learning framework, we divided the dataset into multiple subsets, with each client's dataset containing samples from a subset of categories in CIFAR-100, CIFAR-10, or MNIST. This non-independent and identically distributed (Non-IID) setup simulates the heterogeneity between devices in real-world environments and aims to test the adaptability and effectiveness of the MOON algorithm under different data distributions.

By using these three datasets, this study aims to comprehensively evaluate the performance of the MOON algorithm when handling Non-IID data, analyze its training effectiveness on both complex and simple datasets, and provide an in-depth analysis of the adaptability of federated learning in real-world applications.

3.4 Parameter Tuning and Experimental Setup

When conducting experiments, parameter settings are one of the key factors affecting model performance. To reproduce the MOON algorithm, we carefully configured a series of experimental parameters to ensure that the experiments proceed efficiently and yield valid results. The core of the experiment is to ensure that the MOON algorithm can effectively train and optimize the model on Non-Independent and Identically Distributed (Non-IID) data.

First, the datasets used in the experiments are CIFAR-100, CIFAR-10, and Tiny-Imagenet, and the data storage path is set to './data/'. To simulate the heterogeneity of devices in real-world scenarios, the datasets for each client have an imbalanced category distribution, meaning that each client's data contains different image categories, thus constituting Non-IID data. This data partitioning strategy better simulates the actual challenges in federated learning and tests the algorithm's generalization ability under imbalanced data distribution.

For model configuration, ResNet-50 was selected as the base model. The hyperparameter settings are as follows: the learning rate (lr) is set to 0.1, batch size ($batch\text{-}size$) is 64, and training epochs ($epochs$) are set to 5 to ensure sufficient local training on each client. To avoid overfitting and improve the model's generalization

ability, the dropout probability ($dropout\_p$) is set to 0.0, i.e., dropout is not used. In addition, $L2$ regularization (reg) is applied to the CIFAR-100 dataset to control model complexity and prevent overfitting.

Core Parameter Table 1:

Parameter Name	Default Value	Description
–model	'resnet50'	Neural network type for training.
–dataset	'cifar100'	Dataset used for training.
–partition	'homo'	Data partition strategy: 'homo' or 'non-iid'.
–batch-size	64	Batch size for training.
–lr	0.1	Learning rate.
–epochs	5	Number of local training epochs.
–n_parties	2	Number of nodes in the cluster.
–alg	'moon'	Algorithm: 'fedavg', 'scaffold', or 'moon'.
–comm_round	50	Maximum communication rounds.
–device	"cuda:0"	Device: "cpu" or "cuda:0".

Table 1. Key Experimental Parameters

In the distributed setup, the number of clients ($n\_parties$) is set to 2, simulating a simple distributed training environment. To reduce communication overhead and improve training efficiency, FedAvg is chosen as the communication strategy, and the maximum number of communication rounds ($comm\_round$) is set to 50. In each training round, the clients upload their local training results to the server, which aggregates and updates the global model based on the clients' updates.

All experimental settings and parameters are passed through the command line and retrieved using the `get_args()` function, ensuring flexible parameter configuration and reproducibility of the experiments. To ensure consistency of the experimental results, the initialization random seed (`init_seed`) is set to 0 for all experiments. Furthermore, during the experiments, all training log information is recorded into files with timestamps, facilitating post-experiment result analysis and comparison.

Through these carefully designed experimental settings, this paper effectively evaluates the performance of the MOON algorithm in federated learning, especially in handling Non-IID data.

Finally, to further clarify the application of the MOON algorithm in federated learning, this section will provide a detailed description of Algorithm 2, which outlines the federated learning training process. This process involves interaction between clients and the central server to achieve distributed model training. In each communication round, the clients update their models through local training and upload the results to the server for aggregation, thus continuously improving the performance of the global model. Algorithm 2 specifically describes this process:

Algorithm 2 Federated Learning Training Process

Input: dataset, model, hyperparameters, communication rounds
Output: Trained global model
Initialize: global model ← InitializeModel(model)
for each communication round $r = 1$ to R **do**
    Sample clients S_r from all clients
    for each client $i \in S_r$ in parallel **do**
        $w_i \leftarrow$ ClientUpdate(client i, global model, dataset)
    end for
    Aggregate: $w_{\text{global}} \leftarrow$ FedAvg(w_i for all $i \in S_r$)
    Update global model with aggregated weights w_{global}
end for
Return: global model

4 Experimental Results

4.1 Evaluation Metrics

In this experiment, we adopted two primary evaluation metrics to assess the performance of the MOON algorithm on Non-IID data: accuracy and loss function. These metrics are derived from the implementation of the MOON algorithm and have been widely used as standard measures of algorithm performance in the original paper.

1. Accuracy

Accuracy is the most commonly used metric for classification tasks, measuring the proportion of correctly predicted samples to the total number of samples. The formula for accuracy is as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100\% \tag{1}$$

In this experiment, accuracy was used to evaluate the classification performance of the MOON model on the test dataset after each training round. Higher accuracy indicates that the model can effectively perform classification tasks, particularly in scenarios with heterogeneous devices and data distributions, while maintaining strong generalization capabilities.

2. Loss Function

The loss function is a standard measure of prediction error and serves as a critical optimization target during the training process. In the MOON algorithm, the loss function consists of two components: the traditional cross-entropy loss and the model-contrastive loss. Specifically, the MOON loss function is defined as:

$$\ell = \ell_{\text{sup}}(w_i^t; (x, y)) + \mu \ell_{\text{con}}(w_i^t; w_i^{t-1}; w^t; x) \tag{2}$$

Here, ℓ_{sup} represents the traditional cross-entropy loss, which measures the classification accuracy of the model on the current input x. ℓ_{con} denotes the model-contrastive loss, which encourages consistency between the local model and the global model representations, thereby enhancing the stability and robustness of the

global model. The model-contrastive loss is similar in form to the NT-Xent loss used in contrastive learning and is defined as follows:

$$\ell_{\text{con}} = -\log \frac{\exp(\sin(z, z_{\text{glob}})/\tau)}{\exp(\sin(z, z_{\text{glob}})/\tau) + \exp(\sin(z, z_{\text{prev}})/\tau)} \tag{3}$$

Here, τ is the temperature coefficient, $\text{sim}(z, z_{\text{glob}})$ represents the similarity between the representation z generated by the current local model and the representation z_{glob} generated by the global model, while $\text{sim}(z, z_{\text{prev}})$ denotes the similarity between the current local model representation and the representation generated by the previous local model. By incorporating this loss term, the MOON algorithm encourages each client to maintain consistency with the global model during local training while reducing divergence from the previous round's model, thereby improving the convergence speed and performance of the global model.

μ is a hyperparameter that balances the weights of the cross-entropy loss and the model-contrastive loss. By adjusting the value of μ, the influence of the model-contrastive loss can be flexibly controlled during the training process, thereby impacting the model's learning dynamics and overall performance.

In summary, accuracy and the loss function are critical metrics for evaluating the effectiveness and performance of the MOON algorithm in federated learning tasks. Accuracy reflects the model's classification ability on the test set, while the loss function provides clear guidance for the optimization process, helping the algorithm adjust model parameters to achieve better generalization performance. During the experiments, these two metrics were used to comprehensively assess the performance of the MOON algorithm under Non-IID data conditions.

4.2 Analysis and Comparison of Experimental Results

In this section, we provide a detailed analysis and comparison of experimental results for different algorithms across three datasets. Specifically, we reproduced three classical federated learning algorithms: MOON, FedAvg, and SCAFFOLD. These algorithms were evaluated through multiple communication rounds and local training epochs on CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets. The primary evaluation metric was accuracy, and the performance of these algorithms was compared under varying experimental settings in terms of communication rounds and local training epochs. Below, we present the results of each group of experiments and provide a comparative analysis of the performance of these algorithms.

Method	**Rounds**						
	5	**10**	**20**	**30**	**50**	**75**	**100**
Moon	10.45%	62.13%	65.14%	66.21%	66.87%	67.32%	68.03%
FedAvg	11.32%	61.56%	62.44%	63.29%	63.51%	64.11%	64.23%
SCAFFOLD	9.27%	42.86%	53.98%	57.62%	61.36%	62.41%	64.38%

Table 2. Accuracy of different algorithms at different communication rounds on three datasets

As shown in Table 2, there are significant differences in accuracy performance across different algorithms under varying communication rounds. First, the MOON algorithm outperforms FedAvg and SCAFFOLD at every communication round. Notably, MOON's accuracy consistently increases with more communication

rounds, eventually reaching 68.03% at 100 rounds, demonstrating its ability to improve the global model's performance through extended training. In contrast, FedAvg exhibits a relatively stable performance. While its accuracy gradually improves with increasing communication rounds, the improvement is relatively modest, achieving 64.23% at 100 rounds.

SCAFFOLD performs poorly in the initial rounds, with an accuracy of only 9.27% after the first 5 rounds. However, its performance improves with additional communication rounds, reaching 64.38% at 100 rounds. These results suggest that the MOON algorithm benefits from the incorporation of model-contrastive loss, which effectively facilitates collaborative learning between local models and the global model, thereby enhancing global model accuracy. On the other hand, although FedAvg shows stable performance, it lacks specific optimizations for Non-IID data, resulting in slower performance improvement compared to MOON under the same number of communication rounds. SCAFFOLD's calibration mechanism, while useful in later stages, is hindered by the heterogeneity of model parameters during the initial training rounds, leading to suboptimal performance in the early stages. The visualization of the data from the table is shown in Figure 3.

Method	**local epochs**				
	1	5	10	20	40
Moon	62.61%	68.42%	69.57%	69.03%	67.21%
FedAvg	62.55%	66.38%	66.12%	64.97%	63.85%
SCAFFOLD	0%	64.11%	66.21%	67.37%	65.05

Table 3. Accuracy of different algorithms in different Local epochs on the cifar-10 dataset

As shown in Table 3, there are significant differences in the performance of the three algorithms when varying the number of local training epochs on the CIFAR-10 dataset. First, the MOON algorithm outperforms FedAvg and SCAFFOLD at every local training epoch, especially when the number of local epochs is small, as MOON achieves good accuracy early on. For example, with only 1 local epoch, MOON achieves an accuracy of 62.61%, slightly higher than FedAvg's accuracy of 62.55%. As the number of local training epochs increases, MOON's accuracy continues to rise, reaching 69.57% after 10 local epochs. Although the accuracy slightly drops at 20 and 40 epochs, its overall performance remains stable and high.

In contrast, FedAvg exhibits relatively stable performance, with accuracy increasing from 62.55% to 63.85% as the number of local epochs grows, showing a slower rate of improvement. In fact, during 10 and 20 local epochs, FedAvg's accuracy even lags behind MOON, indicating that FedAvg was not able to effectively improve the global model's performance over multiple training rounds on the CIFAR-10 dataset. SCAFFOLD performs relatively poorly, especially in the early stages of training. With just 1 local epoch, SCAFFOLD's accuracy is 0%, but as the number of local epochs increases, its accuracy gradually improves. At 40 local epochs, SCAFFOLD achieves an accuracy of 65.05%, but still significantly lags behind both MOON and FedAvg. The visualization of the data from Table 3 is shown in Figure 4.

As shown in Table 4, there are noticeable differences in the average accuracy of the three algorithms across different datasets. The MOON algorithm performs the best on all three datasets, particularly on the CIFAR-10 and CIFAR-100 datasets, where MOON achieves accuracies of 68.8% and 67.8%, outperforming the other two algorithms. This indicates that MOON can effectively improve the model's performance on

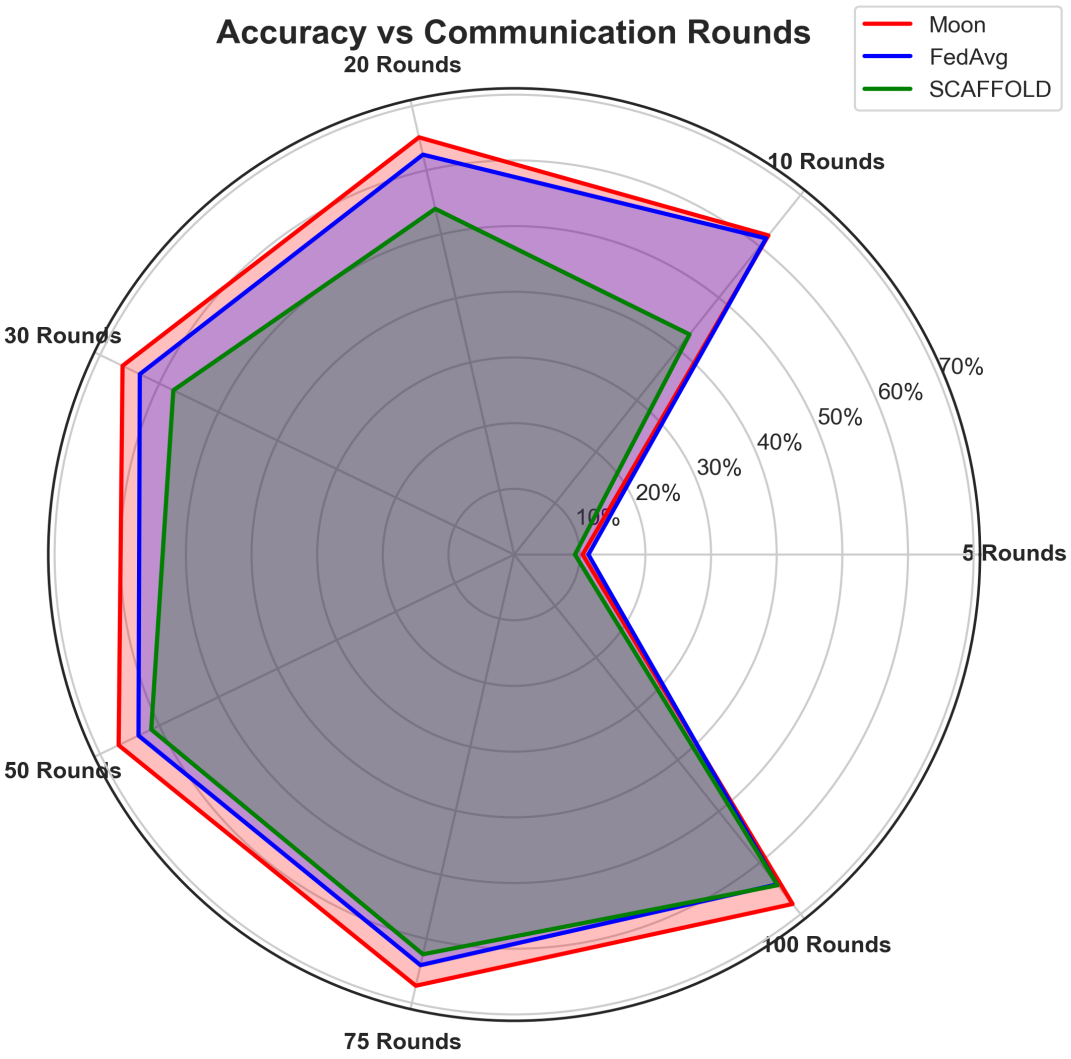

Figure 3. Visualization of the accuracy of different algorithms and different communication rounds on three datasets

Method	CIFAR-10	CIFAR-100	Tiny-Imagenet
Moon	68.8%	67.8%	25.1%
FedAvg	66.7%	64.3%	23.1%
SCAFFOLD	66.5%	52.8%	16.2%

Table 4. Average accuracy of different algorithms on three datasets

these datasets, especially when dealing with more complex data distributions, enabling it to better preserve and learn the features of the data.In contrast, FedAvg achieves accuracies of 66.7% and 64.3% on the CIFAR-10 and CIFAR-100 datasets, respectively. While its performance is slightly lower than that of MOON, it still maintains relatively stable results. On the Tiny-ImageNet dataset, however, FedAvg's accuracy drops to 23.1%, which is poorer compared to the other datasets. This may be due to the higher number of categories

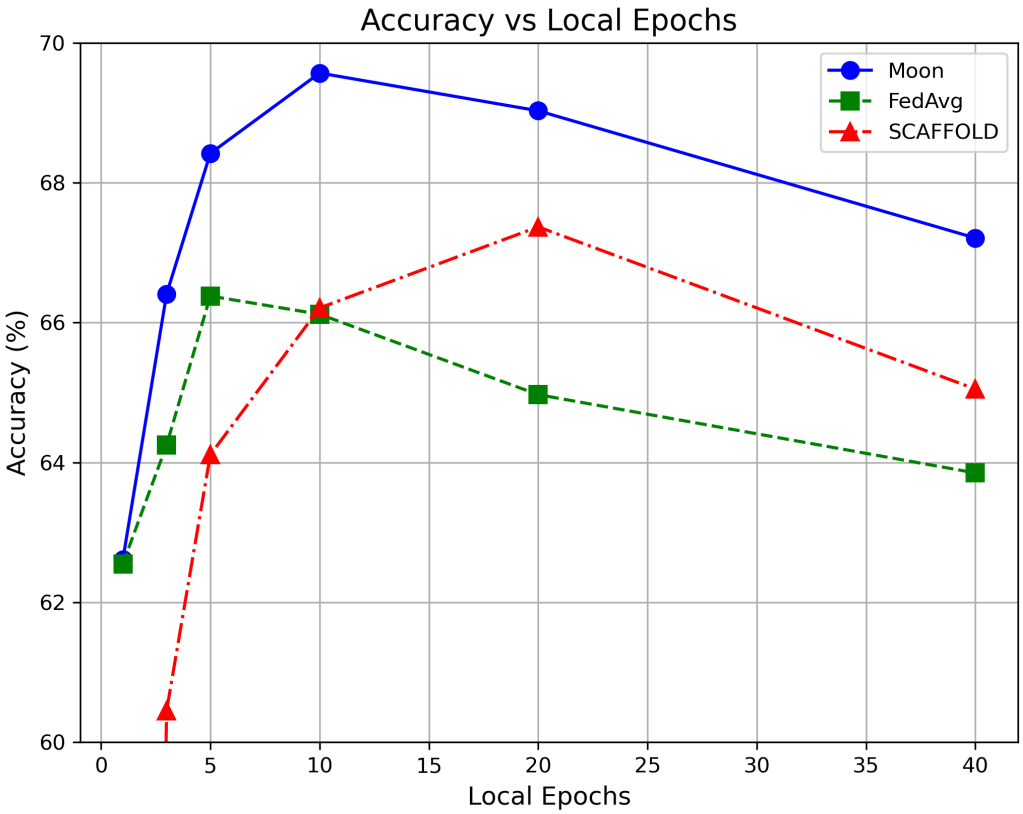

Figure 4. Accuracy visualization of different algorithms based on different Local epochs results in the cifar-10 dataset

and the increased image complexity in the Tiny-ImageNet dataset, making it harder for the model to learn and generalize effectively.

SCAFFOLD achieves accuracies of 66.5% and 52.8% on the CIFAR-10 and CIFAR-100 datasets, respectively, performing worse than both MOON and FedAvg. Particularly on the Tiny-ImageNet dataset, SCAFFOLD's accuracy is only 16.2%, indicating that the algorithm performs weakly when handling larger and more complex datasets. This may be due to its inability to effectively converge in environments with high heterogeneity.The visualization of the data from Table 4 is shown in Figure 5.

In summary, MOON demonstrates significant advantages in handling Non-IID data, improving model generalization, and addressing challenges across various datasets. Although FedAvg and SCAFFOLD can also offer some performance in certain scenarios, MOON's strength lies in its introduction of a more adaptive model contrastive learning strategy, which enables it to maintain better stability and accuracy in complex and heterogeneous environments. These results provide strong evidence for further optimization of federated learning algorithms, especially in applications involving multiple devices and large-scale datasets.

5 Conclusion

This study, through the experimental replication of three classic federated learning algorithms—MOON, FedAvg, and SCAFFOLD—provides an in-depth examination of their performance on different datasets and

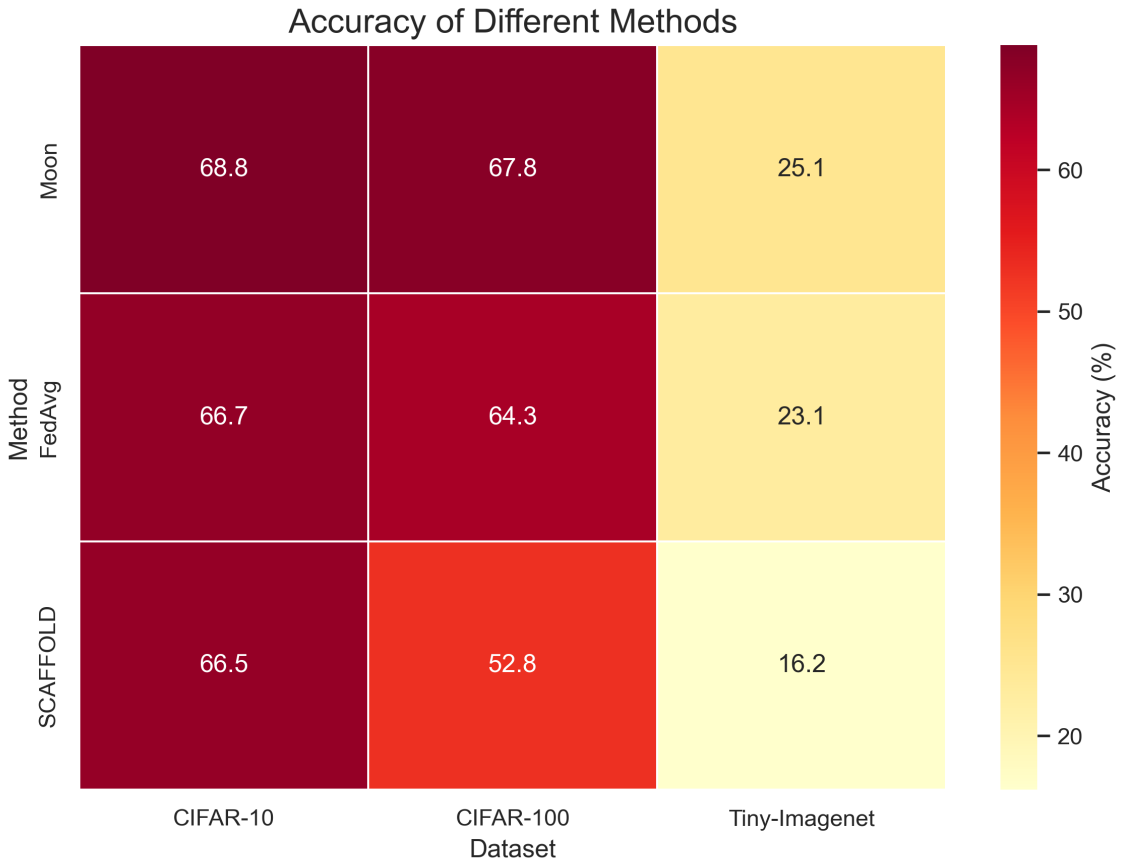

Figure 5. Visualization of the average accuracy of different algorithms on three datasets

offers a detailed analysis of their effectiveness in handling Non-IID data. The experimental results demonstrate that the MOON algorithm has a clear advantage in accuracy and model convergence speed, particularly in addressing data heterogeneity and multi-device scenarios, showcasing robust performance. This conclusion aligns with existing literature and further confirms the application potential of MOON in federated learning. Meanwhile, FedAvg and SCAFFOLD also demonstrated their effectiveness to some extent, but their performance improvement was relatively slow when confronted with more complex datasets, especially in the case of datasets like Tiny-ImageNet, where their accuracy lags behind that of MOON.

In terms of technical implementation, the reproducibility of this experiment has been well ensured. We strictly followed the algorithm details from related literature, used the same experimental environment and datasets, and achieved algorithm replication through precise parameter configuration. However, the implementation of federated learning involves several complex technical issues, particularly in areas such as data privacy protection, model synchronization, and communication costs. While most algorithms can be replicated within a standard framework, ensuring consistency and convergence of all parameters remains a challenging task due to data heterogeneity and the complexity of distributed computing environments. To better address these challenges, future research may require further innovation in algorithm design, particularly in improving communication efficiency and enhancing model stability.

Federated learning, as an emerging distributed training method, has evident advantages in data privacy protection. By keeping data on local devices, federated learning effectively avoids the data leakage risks inherent in traditional centralized training methods, complying with modern data privacy regulations. However,

federated learning still faces many challenges, particularly in terms of data heterogeneity, model performance, and communication costs. In practical applications, the training process often becomes very complex due to uneven data distribution among devices. Moreover, communication overhead is also a significant issue in federated learning. Improving communication efficiency and reducing training time while ensuring privacy is currently a hot topic in research.

From a personal perspective, this experimental replication work not only allowed me to gain a deeper understanding of the core principles and technical implementations of federated learning, but also made me appreciate the complexity of distributed learning in practical applications. During the optimization of the algorithm, I continuously adjusted and tested different parameter configurations, and by comparing the performance of different algorithms, I gradually accumulated valuable experience. Although I encountered many challenges during the experiment, each attempt and improvement deepened my understanding of federated learning. In the future, I hope to further explore the application of federated learning in more real-world scenarios, especially in large-scale datasets and high-heterogeneity device environments, and contribute to the widespread adoption of this technology in the industry.

References

[1] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

[2] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016.

[3] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[4] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[5] Qi Xia, Winson Ye, Zeyi Tao, Jindi Wu, and Qun Li. A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing*, 1(1):100008, 2021.

[6] Zhenheng Tang, Zhikai Hu, Shaohuai Shi, Yiu-ming Cheung, Yilun Jin, Zhenghang Ren, and Xiaowen Chu. Data resampling for federated learning with non-iid labels. In *Proceedings of the International Workshop on Federated and Transfer Learning for Data Sparsity and Confidentiality in Conjunction with IJCAI, Montreal, Canada*, pages 21–22. FTLIJCAI, 2021.

[7] Lin Chen, Xiaofeng Ding, Zhifeng Bao, Pan Zhou, and Hai Jin. Differentially private federated learning on non-iid data: Convergence analysis and adaptive optimization. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[8] Qinbin Li, Bingsheng He, and Dawn Song. Adversarial collaborative learning on non-iid features. In *International Conference on Machine Learning*, pages 19504–19526. PMLR, 2023.

[9] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8300–8311, 2021.

[10] Sawsan AbdulRahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, Chamseddine Talhi, and Mohsen Guizani. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal*, 8(7):5476–5497, 2020.

[11] Thanh Chi Lam, Nghia Hoang, Bryan Kian Hsiang Low, and Patrick Jaillet. Model fusion for personalized learning. In *International Conference on Machine Learning*, pages 5948–5958. PMLR, 2021.

[12] Pian Qi, Diletta Chiaro, Antonella Guzzo, Michele Ianni, Giancarlo Fortino, and Francesco Piccialli. Model aggregation techniques in federated learning: A comprehensive survey. *Future Generation Computer Systems*, 150:272–293, 2024.

[13] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *Acm computing surveys (csur)*, 53(2):1–33, 2020.

[14] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

[15] Mohammed Aledhari, Rehma Razzak, Reza M Parizi, and Fahad Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8:140699–140725, 2020.

[16] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[17] Brett Aho and Roberta Duffield. Beyond surveillance capitalism: Privacy, regulation and big data in europe and china. *Economy and Society*, 49(2):187–212, 2020.

[18] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–23, 2022.

[19] David Byrd and Antigoni Polychroniadou. Differentially private secure multi-party computation for federated learning in financial applications. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–9, 2020.

[20] Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR)*, 54(2):1–36, 2021.

[21] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[22] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

[23] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 965–978. IEEE, 2022.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

[25] Gabriele Cianfrøcco, Massimiliano Pontil, Leonardo Cavallari, Paola Gandolfo, and Luca Benini. Tiny images: A 150 million image database for machine learning. *Journal of Machine Learning Research*, 15:949–974, 2014.