Optimizing Federated Learning on Non-IID Data with Clustering and Model Sharing

Qufei Zhang*, Yunshuang Wang†, Gengsheng Li‡, Barry Cardiff§, Pasika Ranaweera¶
*Information Networking Institute, Carnegie Mellon University, United States
†Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
†Key Laboratory of Cyberspace Security Defense, Beijing, China
†School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
‡Laboratory of Brain Atlas and Brain-inspired Intelligence, Institute of Automation,
Chinese Academy of Sciences, Beijing, China
‡School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
§¶School of Electrical and Electronic Engineering, University College Dublin, Ireland
*qufeiz@andrew.cmu.edu, †wangyunshuang@iie.ac.cn, ‡ligengsheng2024@ia.ac.cn, §barry.cardiff@ucd.ie,
¶pasika.ranaweera@ucd.ie

***Abstract*—The increased amount of data generated by edge computing has necessitated the development of efficient methods to leverage this vast information. Federated Learning (FL) offers a promising solution by enabling distributed model training while preserving privacy. However, FL faces challenges with Non-Independent and Identically Distributed (Non-IID) data, which can impact model accuracy and convergence. To address this, we propose a three-stage framework that effectively trains models in Non-IID scenarios. Our approach effectively reduces the effect of Non-IID by classifying IID clients, evaluating model performance through model sharing, and dynamically adjusting the weighting of each client to perform client selection. We created extreme Non-IID environments with diverse client representation. Tested on the CIFAR-10 dataset, our method improves maximum accuracy by up to 3.03% compared to other state-of-the-art and traditional methods such as FedProx and FedAvg, demonstrating its effectiveness in Non-IID scenarios.**

***Index Terms*—Federated learning, machine learning, non-iid, clustering, model sharing, weighted aggregation**

I. Introduction

Federated Learning (FL) is a deep learning approach where training occurs on various decentralized edge devices (clients) using their data, protecting data privacy by limiting data sharing [1]. After a local model is trained on clients with their data, the training results will be sent to a central server for aggregation. The updated global weights are sent back to the clients, where training continues. This cycle is referred to as a communication round [2]. In FL, each client's datasets play a significant role in global model performance.

- IID (Independent and Identically Distributed): IID data refers to a scenario where data across different clients is statistically uniform. This means that each client's local dataset fairly represents the overall dataset across all clients. The data samples are independent of each other and drawn from the same distribution.
- Non-IID (Non-Independent and Identically Distributed): This is more typical in real-world applications of FL, where data across different clients can be highly heterogeneous in terms of distribution.

Non-IID data distributions can significantly reduce the global model's performance, leading to models that perform well on some clients but poorly on others [3]. Real-world data, which includes objects, values, attributes, and some other aspects, is fundamentally Non-IID. However, there are few effective approaches to Non-IID image-based tasks. Traditional FL algorithms assume that data on each client is identically distributed, an assumption rarely true in practical scenarios [4], leading to models that are biased toward the majority representation, difficult to converge, and lack overall generalization ability. In this work, we propose a novel approach that integrates a clustering algorithm with a performance-based weighting scheme to handle the Non-IID data more effectively. Clients are clustered based on the similarity of their data distributions. During the training stage, each local model is evaluated on multiple data distributions through model sharing. This approach allows our method to classify and prioritize IID clients while simultaneously leveraging the contributions of Non-IID clients that can positively impact the model. Our method mitigates the bias introduced by Non-IID data while still allowing it to contribute meaningfully, such as by increasing data diversity and the number of training samples, which is crucial for creating a more balanced and representative model. Contributions of this paper are,

- We proposed a Cluster and Accuracy-Based Federated Aggregation (CAFA), a three-stage FL framework with two sub-algorithms: Cluster-Based Client Classification (CBCC) and Accuracy-Based Federated Aggregation (ABFA) to mitigate the Non-IID issue in image datasets.
- We conducted a performance analysis of the proposed algorithm in terms on accuracy and convergence speed.
- We compared the proposed algorithms with FedAvg and FedProx in an extreme Non-IID situation.

II. Related Work

FedAvg [4] is the most used FL algorithm; it performs a simple aggregation step to average the weight of all client

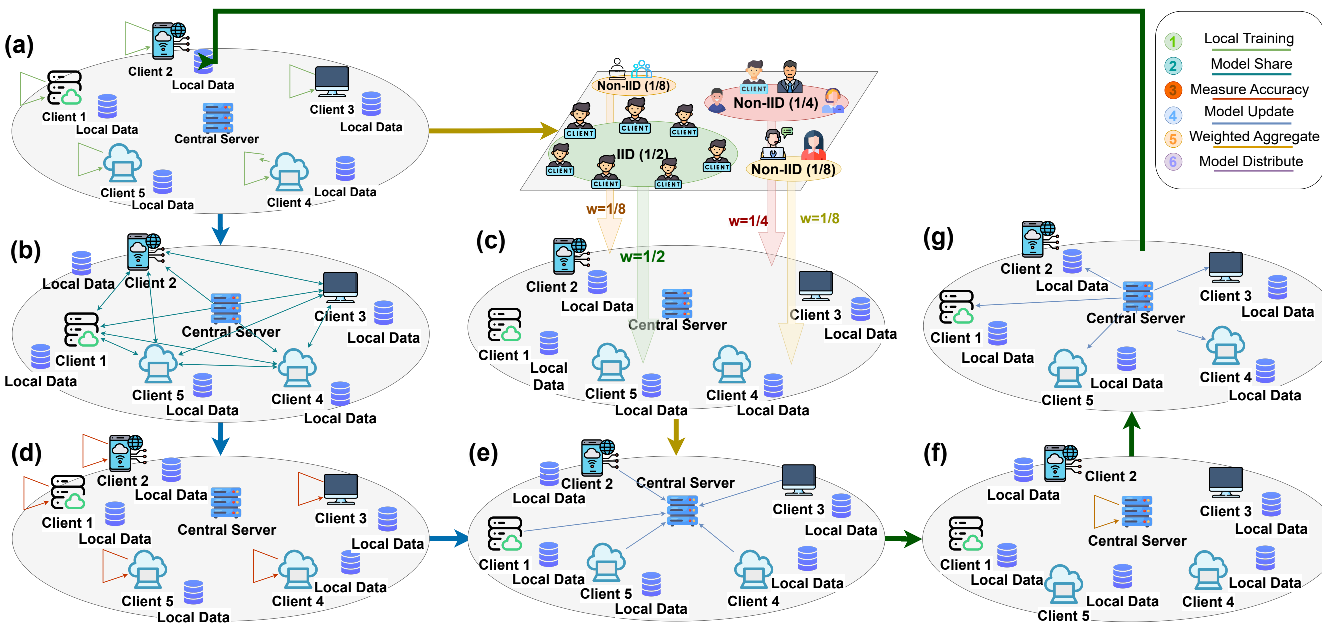

Fig. 1: Proposed clustering and accuracy-based federated aggregation method.

models. However, it performs poorly in Non-IID scenarios. One common strategy to address Non-IID is client selection, which selects a subset of clients that are beneficial for training. Taïk et al. [5] proposed a data-quality-based scheduling (DQS) algorithm for selecting clients using the ratio of data quality value to bandwidth cost based on client reputation and diversity. Wang et al. [6] utilized a double Deep Q-learning Network to automatically select the optimal subset of clients for each training round. However, in these methods, the server needs to maintain an additional dataset to evaluate client models, which might be difficult to obtain. There are also methods that do not require an additional test set. Cho et al. [7] accelerates model convergence by selecting clients with lower local losses. Li et al. [8] proposed a determinantal point process-based algorithm to select high-quality clients. However, they still suffer from information waste because of only selecting a subset of clients for training, impacting generalization ability. Another common solution is weighted aggregation, which adjusts the impact of different clients. Nguyen et al. [9] utilized deep reinforcement learning to adaptively determine the aggregation weight for each client. Ma et al. [10] introduces hierarchical aggregation to leverage the importance of different layers in client models. However, these methods struggle to directly account for the data distribution and quality of all clients. Additionally, a model-sharing approach called FedDif [11] enables models to be trained on different data distributions by diffusing trained models among clients in each epoch. However, this method has a significant computational cost on clients due to the repeated training. Another widely used method is FedProx (Federated Proximal) [12], which modifies the local training objective by introducing a proximal term, constraining local updates, and preventing them from deviating too far from the global model. In this paper, we will test the performance of FedProx and FedAvg against our proposed algorithm, CAFA, to evaluate how well each method handles Non-IID data. CAFA combines the advantages of client selection and weighted aggregation while enabling the utilization of different data distributions in a less computation-intensive way. Our detailed contributions are summarized as follows in Table I:

TABLE I: Emphasizing the Contribution of This Paper in Contrast to the State-of-the-Art.

Context	**[5]**	**[6]**	**[7]**	**[8]**	**[9]**	**[10]**	**[11]**	**CAFA**
Client selection	✓	✓	✓	✓				✓
ML-based client selelction		✓						✓
Weighted aggregation					✓		✓	✓
Combine client selection and weighted aggregation							✓	✓
No need additional test set			✓	✓		✓		✓
Local model testing on various data distributions pre-aggregation							✓	✓

III. Proposed FL Framework

A. *First Stage: Clustering-based Client Classification (CBCC)*

The steps of the proposed CBCC algorithm shown in Fig. 1a and Fig. 1c are as follows:

1) Extracting Data Distribution Features: Feature extraction aims to capture the data distribution features for each client. In CBCC, feature extraction is realized by statistically analyzing the class distribution of the data contained by each client. Specifically, CBCC begins by iterating through the data of each client, counting the occurrences of each class. For example, in each client i $(i = 1, 2, \ldots, n)$, we extract a class distribution feature $\mathbf{v}_i$ as shown in Eq. (1):

$$\mathbf{v}_i = \left(\frac{n_{i,1}}{N_i}, \frac{n_{i,2}}{N_i}, \dots, \frac{n_{i,m}}{N_i} \right) \tag{1}$$

Where $n_{i,j}$ is the number of samples of class j in client i's dataset, and N_i is the total number of samples for client i which can be calculated as shown in Eq. (2):

$$N_i = \sum_{j=1}^{m} n_{i,j} \tag{2}$$

Once all clients have undergone this process, we obtain the class distribution features, which indicate the proportion of each class present in the client's data (see Fig. 2).

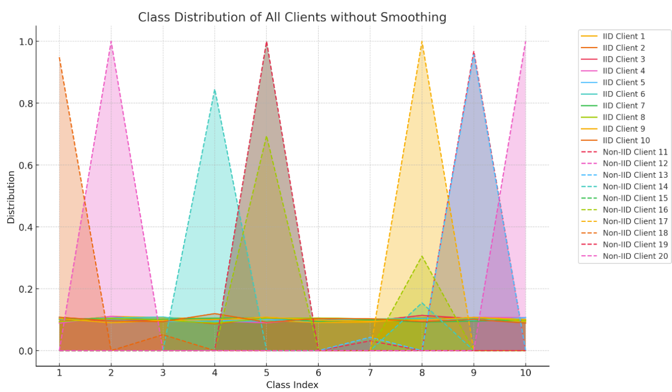

Fig. 2: Visualization of category distribution features of IID and Non-IID clients. The category distribution features for 10 IID and 10 Non-IID clients. Each curve represents the occurrence frequency of different categories in the data of each client. IID clients exhibit more uniform distributions.

2) Selecting Optimal Cluster Number: Before clustering, the Silhouette Score [13] is employed to determine the optimal number of clusters, k. The following steps outline the process:

1. Range of k Values: Test a range of k values, typically from a small number (e.g., 2) to a larger number (e.g., $n-1$, where n denotes the number of clients).

2. Silhouette Scores: For each k, perform K-means clustering and calculate the Silhouette Score $s(i)$ for each client i as shown in Eq. (3):

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{3}$$

where $a(i)$ is the mean distance to the other clients in the cluster which client i belongs to, and $b(i)$ is the mean distance to the clients in the nearest cluster of client i. The mean Silhouette Score $\bar{s}_k$ of all clients for each k can be obtained as shown in Eq. (4) and Eq. (5):

$$\bar{S}_k = \frac{1}{n} \sum_{i=1}^{n} s(i) \tag{4}$$

$$k_{\text{optimal}} = \arg\max_{k} \bar{S}_k \tag{5}$$

3) Applying K-means Clustering: Based on the extracted features and the determined optimal k, the K-means algorithm is utilized to cluster the class distribution features. K-means is a widely used unsupervised learning algorithm that clusters extracted features by minimizing the distance of samples to their respective cluster centers as shown in Eq. (6):

$$\arg\min_{C} \sum_{j=1}^{k} \sum_{\mathbf{v}_i \in C_j} \|\mathbf{v}_i - \mu_j\|^2 \tag{6}$$

where μ_j is the centroid of cluster C_j. This characteristic enables it to group similar clients together, making the data distributions more uniform (Visualization results are inV-A).

4) Classifying IID and Non-IID Clients: Once clustering is completed, the cluster containing the most clients is selected as the IID cluster, while the remaining clusters are designated as Non-IID clusters (see the top of Fig.1c). All clients within the IID cluster are considered IID clients, while clients in the Non-IID clusters are regarded as Non-IID clients. This process is as shown in Eq. (7), where $|C_j|$ is the number of clients in cluster C_j:

$$C_{\text{IID}} = \arg\max_{j} |C_j| \tag{7}$$

It is important to note that no cluster or client is strictly IID in real-world scenarios. In this context, an IID cluster represents a more uniform data distribution that approximates IID, with less variation between clients, thereby reducing the effects of Non-IID data.

5) Assigning CBCC Weights: The weight of each client is relative to the size of the cluster it belongs to, i.e., the number of clients contained in the cluster, to the number of global clients (see 1e). For example, after clustering with CBCC, if clusters $C_1, C_2, \dots, C_k$ consist of $n_1, n_2, \dots, n_k$ clients respectively, the weight for each cluster C_i will be calculated naively as shown in Eq. (8):

$$w_i = \frac{|C_i|}{\sum_{j=1}^{k} |C_i|} \tag{8}$$

For all clients belonging to C_i, their weights will be assigned as w_i.

B. Second Stage: Accuracy-based Federated Aggregation (ABFA)

ABFA shown in Fig. 1a, 1b, and 1d share models among clients and evaluates their performance on the data of different clients. Based on per-round accuracy, larger weights are assigned to better performing models. This allows the global model to focus on clients with a positive impact.

1) Sharing Models: A subset of clients is first randomly selected, then the client models in this subset are completely shared with each others to ensure that each client can receive models from all other clients, as shown in Fig. 1b.

2) Measuring Accuracy and Assigning Weights: For each client in the subset, all models received from other clients will be tested on its local validation dataset, as shown in Fig. 1d. Denote the weighted accuracy as $A_1, A_2, \dots, A_i$, the accuracy of each client on the selected clients' dataset as

$a_1, a_2, \ldots, a_j$, the validation data size as $S_1, S_2, \ldots, S_m$, the weighted accuracy of each client will be calculated as shown in Eq. (9):

$$A_i = \frac{\sum_{j=1}^{m} a_j \cdot S_j}{\sum_{k=1}^{m} S_k} \tag{9}$$

The weights will be normailized as shown in Eq. (10):

$$w_i^A = \frac{A_i}{N} = \frac{A_i}{\sum_{j=1}^{m} A_j} \tag{10}$$

C. Third Stage: Clustering and Accuracy-based Federated Aggregation (CAFA)

Algorithm 1 Clustering and Accuracy-based Federated Aggregation

```
procedure CAFA(clients, num_clusters, K)
    dist ← [calc_dist(clients[i].data) for i in clients]
    clusters ← kmeans(dist, num_clusters)
    cluster_wts ← [size(c) for c in clusters]
    sel ← random_select(clients, K)
    acc_wts ← []
    cbcc_wts ← []
    for i in sel do
        cbcc_wts.append(cluster_wts[clients[i].ClusterID])
    end for
    for i in sel do
        θ_i ← train(clients[i].data)
        acc ← [evaluate(θ_i, clients[j].val) for j in sel]
        acc_wts[i] ← weighted_mean(acc)
    end for
    normalize(cbcc_wts), normalize(acc_wts)
    comb_wts ← [cbcc_wts[i] × acc_wts[i] for i in sel]
    normalize(comb_wts)
    θ_global ← ∑(θ_i × comb_wts[i] for i in sel)
    return θ_global
end procedure
```

The proposed CAFA shown in Fig. 1 combines CCBC and ABFA:

1) Selecting Optimal Cluster Number: Utilize Silhouette Score to automatically select the optimal number of clusters before performing K-means clustering (Eqs. (1)(2)(3)(4)(5)).

2) Applying K-means Clustering: Use K-means clustering to cluster clients into different clusters based on their data distribution features and compute CBCC weights (Eqs. (6)(7)(8)).

3) Sharing Models and Measuring Accuracy: Evaluate local models on a global data distribution during a client-to-client sharing process, assign weights based on accuracy per round (Eqs. (9)(10)).

4) Assigning Weights and Aggregating Model: w_i^K stands for the CBCC weight, w_i^A stands for the ABFA weights, W_i stands for the model trained in client i and W stands for the model after aggregation as shown in Eq. (11):

$$W = \frac{\sum_{i=1}^{n} w_i^K \cdot w_i^A \cdot W_i}{\sum_{j=1}^{n} w_j^K \cdot w_j^A} \tag{11}$$

IV. Experimental Setting

A. Experimental Setup

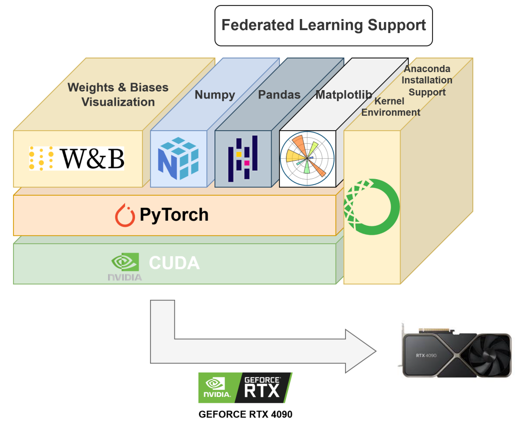

Fig. 3: Experimental setup.

- Dataset: The CIFAR-10 dataset was used, containing images in 10 categories, with 6,000 images per category, the training set includes 50,000 photos, and the testing set includes 10,000 photos, totaling 60,000 images, and these images are 32x32 pixels. 20% of the training set is used as the validation set.
- Client Setup: The experiment was set up with 20 clients.
- Model Architecture: The VGG19 [14] was used as the image classification model in all clients and the global server.
- Number of Training Rounds: All algorithms were conducted over 150 rounds.

The codes are available at: https://github.com/Gengsheng-Li/Non-IID-Robust-Federated-Learning-Algorithms.git

B. Implementation Details and Dataset Partitioning

The experiment was implemented in Python using the NumPy and PyTorch libraries. The code is executed on a NVIDIA RTX 4090D, the CPU is 18 vCPU AMD EPYC 9754 128-Core Processor and the memory is 60GB.

The dataset is divided into multiple clients based on class labels, forming a Non-IID imbalanced data distribution. The amount of data required for each client is randomly generated based on the number of clients and the number of categories held by each client. Here, each client has 20%-40% of the total categories (i.e., 2-4 in CIFAR-10).

C. Smoothing Algorithm

In order to make the curves clearer and more understandable, the Time Weighted Exponential Moving Average algorithm is applied to the accuracy curves shown in Fig. 6.

V. Experimental Results and Evaluation

A. Optimal k

Fig. 4 illustrates the Silhouette Score curve, giving an example of how the Silhouette Score varies with k. A higher Silhouette Score indicates better clustering quality [15]. Therefore, during the experiments, we used curves like this to automatically select k to ensure the quality of clustering.

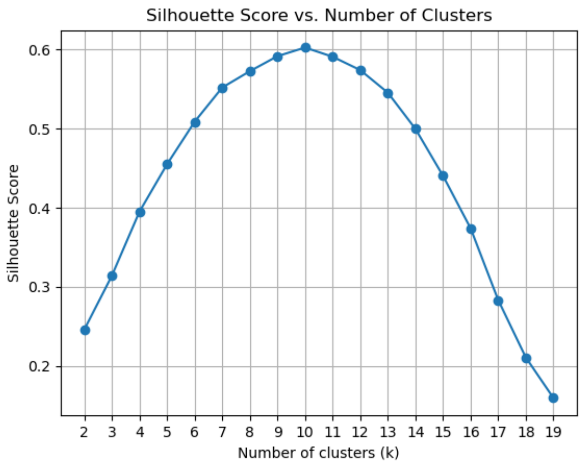

Fig. 4: Silhouette score curve.

B. Analysis of CBCC

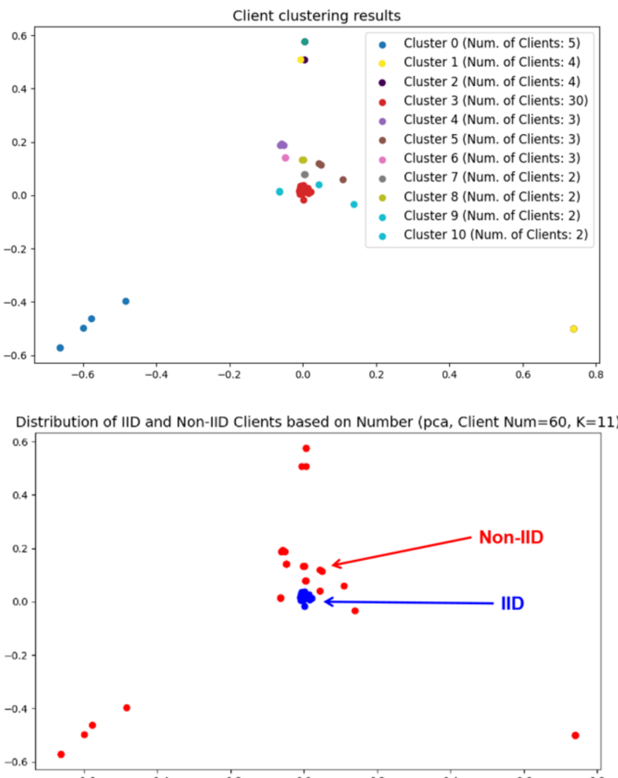

Fig. 5: Visualization of CBCC's result through PCA dimensionality reduction under Case 3 of Table III (50% IID client).

The analysis of CBCC is based on the following prior knowledge: IID clients' data distributions are more concentrated in the feature space, while Non-IID clients' data distributions tend to be more dispersed. A series of experiments across various data partition scenarios were conducted, including scenarios where 25% of the clients are IID, and 50% of the clients are IID.

TABLE II: Performance of CBCC where 25% of the clients are IID.

25% IID	**Num. of Clients**		**K**	**Acc**	**Precision**	**Recall**	**F1**
	IID	**Non-IID**					
Case 1	5	15	9	0.95	0.83	1.00	0.91
Case 2	10	30	11	1.00	1.00	1.00	1.00
Case 3	15	45	12	1.00	1.00	1.00	1.00
Average	-			0.98	0.94	1.00	0.97

TABLE III: Performance of CBCC where 50% of the clients are IID.

50% IID	**Num. of Clients**		**K**	**Acc**	**Precision**	**Recall**	**F1**
	IID	**Non-IID**					
Case 1	10	10	7	0.95	0.91	1.00	0.95
Case 2	20	20	10	1.00	1.00	1.00	1.00
Case 3	30	30	11	1.00	1.00	1.00	1.00
Average	-			0.98	0.97	1.00	0.98

Tables II and III present the classification metrics accuracy, precision, recall, and F1-score, of CBCC under the scenarios of 25% and 50% IID clients. These results demonstrate that CBCC can identify IID and Non-IID clients with extremely high accuracy and adapt well to different scenarios. Notably, CBCC performs better in scenarios with a larger number of clients. Furthermore, in all scenarios, the recall is 1.00, indicating no IID clients are misclassified as Non-IID clients.

To visualize the classification results of CBCC, PCA is applied to map each client's high-dimensional category distribution features to a two-dimensional space. This mapping allows us to present the data distribution of different clients and CBCC's classification results clearly and intuitively within a two-dimensional coordinate system. Case 3 from Table III is selected for visualization.

Fig. 5 provides a pair of clear visualization results before and after applying CBCC. The classification of IID and Non-IID clients demonstrates that CBCC is not only effective at grouping similar clients but also at identifying IID clients. Since IID clients tend to exhibit the most similar data distributions, they naturally form the largest cluster, which we refer to as the IID cluster.

C. Analysis of CAFA

TABLE IV: Average Accuracy (%) Across Communication Rounds.

Method/Rounds	**0–29**	**30–59**	**60–89**	**90–119**	**120–149**
FedAvg (FA)	55.36%	77.1%	79.05%	81.06%	79.68%
FedProx	50.96%	70.13%	79.52%	77.2%	78.68%
CAFA	**57.78%**	**79.37%**	**82.29%**	**82.71%**	**83.07%**

TABLE V: Best Accuracy Achieved by Each Algorithm and the Improvement of CAFA.

Algorithm	**Best Accuracy (%)**	**CAFA Improvement(%)**
FedAvg (FA)	82.1%	1.35%
FedProx	80.42%	3.03%
CAFA	**83.45%**	—

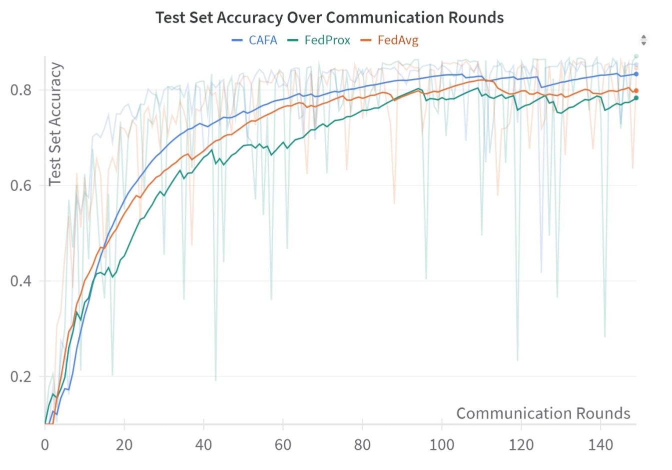

Fig. 6: Comparing the proposed CAFA with FedAvg and FedProx.

As shown in Table IV and Table V, CAFA improves accuracy, stability, and convergence speed compared to FedProx and FedAvg; it achieved the highest accuracy (3.03% over FedProx and 1.35% over FedAvg) on the CIFAR-10 dataset and consistently ranks highest in average accuracy across rounds, indicating the model on average is less affected by fluctuations due to Non-IID nature of the data. Though these improvements are limited in terms of percentage considering a general situation we considered, the improvement accumulates more when there are more non-IID clients in the distribution (up to 5.25% improvement in accuracy over the other algorithms in more extreme settings). Showing those results is beyond the scope of this research paper. Further, CAFA also converges the fastest in the experiment, reaching 66.8% accuracy (+8.07% over FedProx and +4.69% over FedAvg) after the first 30 rounds. It is interesting to note that both FedProx and FedAvg outperform CAFA in the beginning (i.e., up to around 10th round) but deteriorate over time, espeicially for FedProx. This is likely due to FedProx's proximal term, which helps in the beginning by minimizing the divergence of local updates. However, it becomes restrictive given the extreme Non-IID data distribution of the experiment, making it difficult for local models to learn from their data. The result aligns with our intuition: CBCC performs unsupervised clustering directly based on client data distribution characteristics; similar clients are grouped into clusters to form more uniform data distributions. ABFA adds an evaluation step to ensure the priorities of different clients can be adjusted dynamically based on its global distribution performance. This is crucial because CBCC may fail to pay attention to Non-IID clients that contribute positively to the global model for the diversity of their data and extra training samples. Higher priorities are assigned to clients with better-evaluated performance, allowing them to have a greater impact on the global model. CAFA takes into account both local and global data distribution characteristics, making it highly effective in enhancing the global model.

VI. Conclusion

Currently, there are few effective algorithms to solve Non-IID challenges in FL, and the traditional FedAvg treats the impact of each client on the global model equally, which leads to the inability to adapt to Non-IID scenarios. We implemented Non-IID data distribution scenarios based on CIFAR-10. We proposed and demonstrated that CAFA can effectively cluster similar clients to create more uniform distributions, accurately identify and prioritize IID clients. Through model sharing, Non-IID clients can contribute positively based on evaluations of their impact on a global scale. CAFA notably improves the accuracy, robustness, convergence speed, and generalization of the global model under Non-IID scenarios. In the future, the adaptability of these algorithms on more datasets and modalities warrants further research.

References

[1] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[2] I. Kholod, E. Yanaki, D. Fomichev, E. Shalugin, E. Novikova, E. Filippov, and M. Nordlund, "Open-source federated learning frameworks for iot: A comparative review and analysis," *Sensors*, vol. 21, no. 1, p. 167, 2020.

[3] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[5] A. Taïk, H. Moudoud, and S. Cherkaoui, "Data-quality based scheduling for federated edge learning," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. IEEE, 2021, pp. 17–23.

[6] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *IEEE INFOCOM 2020-IEEE conference on computer communications*. IEEE, 2020, pp. 1698–1707.

[7] Y. J. Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 10 351–10 375.

[8] A. Li, L. Zhang, J. Tan, Y. Qin, J. Wang, and X.-Y. Li, "Sample-level data selection for federated learning," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[9] N. H. Nguyen, P. L. Nguyen, T. D. Nguyen, T. T. Nguyen, D. L. Nguyen, T. H. Nguyen, H. H. Pham, and T. N. Truong, "Feddrl: Deep reinforcement learning-based adaptive aggregation for non-iid data in federated learning," in *Proceedings of the 51st International Conference on Parallel Processing*, 2022, pp. 1–11.

[10] X. Ma, J. Zhang, S. Guo, and W. Xu, "Layer-wised model aggregation for personalized federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 092–10 101.

[11] S. Ahn, S. Kim, Y. Kwon, J. Park, J. Youn, and S. Cho, "Communication-efficient diffusion strategy for performance improvement of federated learning with non-iid data," *arXiv preprint arXiv:2207.07493*, 2022.

[12] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[13] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," in *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*. IEEE, 2020, pp. 747–748.