# A Cluster-based Privacy-Enhanced Hierarchical Federated Learning Framework with Secure Aggregation

Chia-Yu Lin*, Chih-Hung Han†, Wei-Chih Yin†, and Ted T. Kuo‡
* *Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan*
† *Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan*
‡ *College of Artificial Intelligence, National Yang Ming Chiao Tung University, Tainan, Taiwan*
Corresponding Author: Chia-Yu Lin (sallylin0121@ncu.edu.tw)

2024 International Conference on Computing, Networking and Communications (ICNC) | 979-8-3503-7099-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICNC59896.2024.10556276

*Abstract*—**Traditional machine learning typically requires training datasets on local machines or data centers. However, this approach may raise concerns related to data privacy and security. To address these issues, federated learning was proposed. However, federated learning, which involves a server communicating with multiple client devices, can significantly burden the server. Even when using hierarchical federated learning, there is still a considerable cost associated with communication at intermediate nodes. To further alleviate the communication cost burden on intermediate nodes, the most direct approach is to have each intermediate node select a subset of clients for training and accept their model parameters. However, client training data distributions are not uniform, leading to a state known as Non-Independent and Identically Distributed (Non-IID). Unthinkingly selecting clients for training may result in more imbalanced data selection and bias the model training in specific directions. Therefore, we propose the "post-clustering selection", where clients with similar data distributions are grouped together, and a certain proportion of clients are selected as representatives for training. This approach allows intermediate nodes to reduce communication costs while avoiding the selection of clients with highly imbalanced data distributions. Finally, we integrate differential privacy and secure aggregation to enhance privacy protection and present a framework called 'Cluster-based Privacy-Enhanced Hierarchical Federated Learning Framework with Secure Aggregation (CPE-HFL)." From experiments, we reduce the communication volume by up to 29% while maintaining accuracy. Additionally, the accuracy improves more in cases with clustering than those without clustering. The proposed framework can reduce communication costs and effectively protect clients' privacy while maintaining model accuracy.**

## I. Introduction

The 2016 enactment of the General Data Protection Regulation (GDPR) by the European Union significantly impacted machine learning. Traditional machine learning, which typically involves centralizing training data on machines or data centers, began to raise concerns about data privacy and security in light of this regulation. Thus, federated learning (FL) [1]–[3], which is a distributed learning framework in which multiple clients jointly train a global model under the supervision of a server, was proposed. In each round of federated learning, the server selects several clients, utilizes their private local data to train the global model, and collects the trained model weights from them. Finally, the server aggregates the model weights submitted by all clients to form a new round of the global model. To achieve the goal of protecting client privacy, clients only submit model weights without the need to disclose their private data, thus mitigating privacy concerns to some extent. However, in real-world scenarios, the FL architecture with one server overseeing multiple clients faces significant challenges. Servers receive models trained by all clients, resulting in significant network communication traffic due to the large size of these models. This, in turn, necessitates substantial storage capacity on servers to store parameters from each client's model, ultimately reducing training efficiency. As the number of participating clients increases, the communication cost burden on the server also escalates.

To address the communication cost issues faced by Federated Learning (FL), hierarchical federated learning (HFL) [4]is emerged. In HFL, clients upload their models to their respective regional nodes, known as zone aggregators. These zone aggregators perform an initial aggregation to create regional models. Consequently, the server doesn't need models from every client; it only needs to receive regional models from multiple zone aggregators, which can then be aggregated into a global model. This can significantly reduce the storage and communication costs required by the server. Additionally, because the client's model parameters have already been aggregated into a regional model, even if the regional model is subjected to an attack, it cannot accurately reflect the data used by the client. This enhances the privacy of client data during the upload process from the zone aggregator to the server.

However, while HFL can alleviate the burden on servers, it may shift the load to zone aggregators. With many clients participating in training, zone aggregators may require enhanced computational capabilities and increased hardware capacity to manage the influx of client models and associated communication costs. Furthermore, recent research [5], [6] has pointed out potential privacy concerns related to the model weights uploaded by clients after training. Attackers may attempt to reverse-engineer the original data used for training by analyzing the model weights. Even if the client's training weights are aggregated into a regional model at the zone aggregator, providing privacy during subsequent model

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

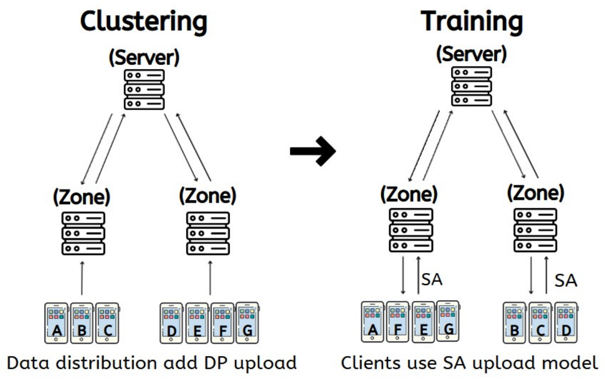

Fig. 1. The architecture of CPE-HFL.

transmission between the zone aggregator and the server, privacy leakage remains possible when clients upload their weights to the zone aggregator.

To further mitigate the communication costs at the zone aggregator and enhance privacy, we propose a "Cluster-based Privacy-Enhanced Hierarchical Federated Learning Framework with Secure Aggregation (CPE-HFL)," which integrates post-clustering selection, differential privacy, and secure aggregation into HFL. The overall structure is shown in Fig 1:

1) **Differential Privacy (DP)**: About federated learning clustering, current research focuses on clustering based on post-training model weights. However, these weights are not endowed with protective measures, meaning that even the model weights returned by clients could leak privacy. In the proposed CPE-HFL, clients return their data distribution instead of sending the model weights. We then apply differential privacy to this data distribution, preventing the server from discerning the exact data distribution of individual clients while still effectively clustering them.
2) **Clustering**: Once the participating clients are determined, they submit their data distributions, which have been processed with differential privacy, from the zone aggregator to the server. Subsequently, the server employs clustering algorithms such as K-means, Hierarchical Clustering, and Affinity Propagation based on these data distributions to cluster the clients, allowing clients with similar data distributions to be grouped under the same zone aggregator. Because the data distributions within each cluster are similar, not all clients must participate in training. During each training round, the zone aggregator only needs to randomly select a few clients as representatives, thus reducing communication costs. In real-world scenarios, client data often exhibits the Non-Independent and Identically Distributed (Non-IID) characteristic. If clients are randomly selected to participate in training, it may lead to an imbalanced data distribution, resulting in a global model biased towards specific directions. Therefore, by selecting clients after clustering, each specific data distribution direction can participate in training, reducing the risk of obtaining imbalanced data distributions when selecting clients.
3) **Secure Aggregation (SA)**: Clients add a two-layer mask before returning their post-training model weights. This method ensures that even after aggregating the models, the zone aggregator cannot discern the exact model parameters of individual clients, thwarting any attempts by attackers to infer original data from model weights.

Based on the experimental results, we reduce communication volume by up to 29% while maintaining accuracy. Furthermore, the application of clustering led to even greater improvements in accuracy compared to when not using clustering. The framework reduces communication costs and effectively protects clients' privacy while maintaining training accuracy.

## II. Relate Work

Since the Federated Learning (FL) concept was introduced, many scholars have contributed much research and innovations in this field. Li et al. [3] detailedly reviewed FL's current challenges and evolution and explored its applications. In [4], Liu et al. proposed a hierarchical federated learning structure of Client-Edge-Cloud. This structure's primary goal is to mitigate network congestion's impact, which can lead to inefficient training when there are many participating clients. The foundational architecture of this study primarily consists of three layers, as depicted in Fig 2. These are the server (Core Aggregator), the regional nodes (Zone Aggregator), and the clients (FL Client). The basic workflow among these three entities is as follows:

1) The server sends training tasks to the regional nodes in the second layer.
2) Regional nodes then sample the clients who will participate in the training.
3) Once the clients are determined, the regional nodes send training commands to their respective clients. Upon completing the training, each client sends its post-training model weights back to its regional node for model aggregation.
4) Each regional node returns its aggregated model to the server, which performs the final aggregation to update the global model for that round.

Regarding clustering, [7] proposed the WSCC clustering method, which dynamically clusters clients using affinity propagation and cosine distance. In [8], the FLIS algorithm is introduced, where clients return their trained models to the server, and the server obtains inference results on its small dataset. Clients are then clustered based on the similarity of these inference results. The approach in [9] utilizes Hierarchical Clustering for clustering under FL (referred to as FL+HC). During training, clients are clustered based on the similarity between their local model updates and the global model, and each cluster trains a specialized model.

In terms of security measures in federated learning (FL), the secure aggregation (SA) method proposed in [10] is a

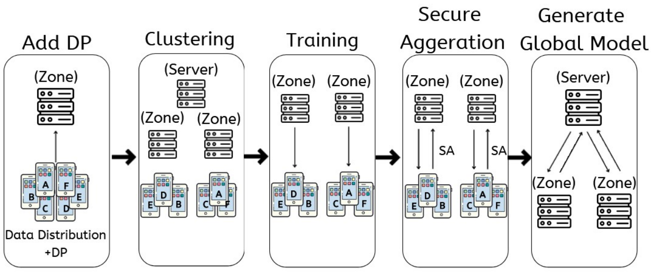

Fig. 2. The flow chart of CPE-HFL.

prevalent approach. It is an encryption protocol incorporating techniques such as Diffie–Hellman key exchange and Shamir's secret sharing. Additionally, it utilizes a Pseudo Random Generator to generate two types of masks. After client training is completed, these two masks are added to the model weights before being transmitted. Consequently, the model weights received by the server are in an encrypted state. During the model aggregation process, the second mask is removed, and then the server retrieves the first mask from each client. This enables the server to decrypt the already-aggregated global model for the new round. In summary, the server can complete updates for the new round of the global model without knowing the actual model weights returned by each client. Moreover, suppose a client departs midway, as long as the remaining clients exceed a certain threshold (complying with the t-out-of-n condition in Shamir's secret sharing). In that case, they can still fully decrypt the global model under the SA protocol. This further enhances the security and stability of the participating clients in the training process.

Regarding differential privacy, the idea is to introduce a small amount of noise into the data, preserving its characteristics while minimizing privacy leak risks. Attackers cannot deduce accurate client information by examining computation results, making this one of the widely-used data protection methods today. Kang Wei *et al.* applied this technique in [11], proposing a Noise-before-Aggregation FL (NbAFL) method. The concept is to add noise to the model on the client side before uploading. However, aggregating models that already have noise affects the convergence speed and accuracy of the model. Hence, this study mainly adds differential privacy to the clients' data distribution for clustering, as expressed in equation 1, and uses secure aggregation during the initial model aggregation phase to ensure model upload security.

$$\Pr[M(D_i) \in S] \leq e^{\epsilon} \Pr[M(D_i') \in S] + \delta \tag{1}$$

## III. Cluster-based Privacy-Enhanced Hierarchical Federated Learning

We propose a cluster-based privacy-enhanced hierarchical federated learning (CPE-HFL), which encompasses secure aggregation, differential privacy, and clustered federated learning.

1) **Data processing**: We use open dataset such as CIFAR-10. Since these datasets are all independently and identically distributed, we generated multi-dimensional arrays of the same number as the data categories using the Dirichlet distribution. We used this array as the basis for partitioning the data, resulting in a proportionally scaled data distribution.
2) **Add noise**: The server sends a training task to the second-layer local area, and then the local area requests the data distribution from all clients below. Clients, in turn, protect data privacy by adding Gaussian noise to the data distribution using differential privacy techniques. Finally, the processed data is sent back to the server through the local area for clustering.
3) **Clustering**: During the clustering phase, we use the data distribution that has already been noised to perform clustering. We employ three clustering algorithms to cluster the clients: K-means, Hierarchical Clustering, and Affinity Propagation. We then select the algorithm with the best silhouette scores from this clustering. Based on the clustering result, all clients are reconnected to the appropriate local area.
4) **Start training**: Once clustering is complete, each zone aggregator will have well-defined groups. At this point, we commence federated learning while incorporating a secure aggregation mechanism. In this context, we have referenced the secure aggregation protocol from [10] and integrated Salvia [12] to deploy SA within our framework. Before each training round, each zone aggregator selects a certain proportion of clients from its client group to participate. Subsequently, the selected clients, in addition to receiving the global model broadcast by the zone aggregator, are required to independently generate public-private key pairs based on the secure aggregation protocol. Then, clients within each group can share and obtain partially encrypted information from other clients using the t-out-of-n mechanism from Shamir's secret sharing and the key agreement algorithm from Diffie-Hellman. Afterward, each client uses this encrypted information to generate two layers of private masks through a Pseudo Random Generator (PRG). Finally, before returning the model, each client adds the two newly generated layers of private masks.
5) **Model Secure Aggregation:** During the model aggregation phase, each zone aggregator receives all the model weights that have undergone double-masking processing. It then combines all the encrypted models to obtain the regional model. According to the Secure Aggregation algorithm, even though the aggregated model parameters are still in an encrypted state, the second mask of each model will be effectively eliminated after summing up all the models. In other words, at this point, the regional aggregated model contains the sum of the first masks of all client's models. The zone aggregator only needs to obtain the first mask information from each participating client to eliminate it and obtain the correct sum of the

client model. Afterward, a weighted average is applied to form the regional model. It is worth noting that during this process, the zone aggregator can only acquire either the first or second mask from each client to decrypt the regional model. For the clients, this ensures that even if one mask is lost, another mask still protects the trained model. Furthermore, during the process, the zone aggregator can only obtain the sum of the models from the participating clients but cannot access the exact individual client models. This safeguards client model privacy and prevents any potential leaks. In the end, all zone aggregators then send their respective regional models back to the server for a second round of aggregation to form the global model for that training round, prepared for broadcasting to the zone aggregators as the initial model for the next training round.

## IV. Experiment

### A. Dataset

Data distribution is typically Non-Independent and Identically Distributed (Non-IID) in real-world federated learning scenarios. This means that the dataset of each participating client may have a different distribution. However, in some commonly used datasets, such CIFAR-10, the number of samples in each category is almost the same. Thus, in our experiment, we must create Non-IID datasets based on the actual data to simulate real-world scenarios.

CIFAR-10 consists of 10 distinct categories: airplanes, cars, birds, cats, deer, dogs, frogs, horses, and ships. Each category contains 6,000 images, totaling 60,000 images with a size of 32x32. Of these, 50,000 images serve as training data and 10,000 as test data. To simulate Non-IID scenarios, we adopted two segmentation approaches:

1) We utilized the Dirichlet method to emulate Non-IID conditions. The Dirichlet distribution is commonly used when the sum of probabilities of multiple components is one. This can replicate situations where each training participant might have more samples from specific categories and fewer from others.
2) Apart from the first method, we also tried an extreme segmentation approach by randomly selecting 2 out of the ten categories. One category's dataset contains only about 80% of One category's total samples in this method, while the other has around 20%, simulating extreme real-world scenarios.

### B. The Experiment setting and method

Under a setup of one server and 50 clients, we use CIFAR-10 as our dataset. CNN, ResNet18, and VGG16 models are employed for each dataset with a batch size 128. In addition, each federated learning round has three local epochs, with a default total of 50 federated learning rounds. The exception is the second Non-IID dataset of CIFAR-10, which, due to its extreme distribution, requires more rounds to converge and thus has 200 federated learning rounds. To observe the accuracy performance of model training at various filtering levels, we have set filtering levels at 100%, 75%, 50%, and 25%, representing the "proportion of clients selected within each group" (with 100% meaning no filtering). For example, if a group contains seven clients post-clustering, 50% implies that four clients (rounded from 3.5) are randomly selected from the 7 for that training round. However, due to secure aggregation constraints, at least two clients must be dispatched from each group for training; otherwise, key exchange in secure aggregation cannot proceed.

Initially, the 50 clients add noise to their individual data distributions and submit them to the server for clustering. In the clustering phase, three clustering algorithms, k-means, hierarchical clustering, and Affinity Propagation, are calculated, yielding corresponding silhouette score values. This score ranges from -1 to 1. A silhouette score close to 1 indicates that data points within a cluster are very similar, and there is a significant difference between clusters, implying excellent clustering results. A score near -1 suggests that cluster data points are not similar, and cluster differences are minimal, indicating suboptimal clustering results. A score near 0 means the similarity within a cluster and differences between clusters are nearly identical, indicating ambiguous clustering results. Hence, the clustering algorithm with the highest score of the three is chosen for that specific experiment.

After clustering all clients, training commences. Before each training round, based on the chosen filtering level for the current experiment (100%, 75%, 50%, 25%), a subset of clients is randomly selected from each cluster to participate in that round. During training, secure aggregation ensures that the server remains unaware of the models returned by individual clients, obtaining only the aggregated model and completing one training round. This process is repeated to examine the communication cost savings at different filtering levels while maintaining data balance without significantly compromising training efficacy.

## V. Experiment result

In our experiments, we used the CIFAR-10 datasets and divided them into Non-IID forms, as mentioned earlier. The explanations for the experimental result tables are as follows:

1) **Fraction**: This refers to the number of clients participating in each round of federated learning from every cluster. A minimum of 2 clients from each cluster is required for training. Also, when the number of selected clients after filtering is not an integer, it will be rounded. Hence, a slight difference might exist between the actual ratio of clients participating in the training and the proportion stated in the table.
2) **Accuracy**: This is the model accuracy at the point of convergence.
3) **Round**: Convergence is considered when the accuracy does not increase by more than 1% over ten rounds.
4) **Traffic**: The communication cost is calculated as twice the number of convergence rounds multiplied by the sum of the server broadcasting the global model to n zones and m participating clients, further multiplied

by M. Here, the factor of two accounts for both the broadcasting of data and its return, while M stands for the size of the model.

5) **Benefits**: Calculated based on the 100% client participation ratio, this indicates the percentage reduction in communication overhead achieved with the current filtering.

### A. *CIFAR-10*

1) **Using Dirichlet to Simulate Non-IID**: From Table I, in experiments using the CNN model, as the proportion of clients participating in training in each group decreases, the number of rounds required for convergence increases compared to the unfiltered scenario. However, the communication overhead before reaching the converging round decreases noticeably, and the accuracy drops at most by 2.74%. For VGG16 and ResNet18, although the savings in communication costs are not more significant than using the CNN model, the accuracy only drops slightly, even at the smallest client participation rate of 25%.
2) **Randomly Selecting two classes from ten**: From Table II, this experiment used data that's even more imbalanced than that from the Dirichlet distribution. Of the ten categories in CIFAR-10, each client only possesses images from 2 of these categories, with one comprising about 80% and the other about 20%. Due to the extreme imbalance, the training accuracy is considerably reduced compared to the dataset from the Dirichlet distribution. However, regarding benefits, most models can reduce communication overhead through filtering, although VGG16's performance could be better. The accuracy drops as the filtering ratio decreases, and the benefits are less pronounced. However, ResNet18 performs exceptionally well, maintaining nearly the same accuracy as the unfiltered 100% scenario while reducing communication overhead.
3) **Comparison between Clustered and Non-clustered**: We use the second Non-IID dataset of CIFAR-10 for the experiment. In Fig. 3, and Fig. 4 and Fig. 5, the red line represents training after clustering using our system, while the blue line represents training by randomly selecting an equivalent number of clients as the clustered scenario. The results show that using our system maintains higher accuracy while reducing the number of participating clients. Compared to the potential data skew of random selection, our system ensures more consistent data after filtering, thereby mitigating Non-IID effects.

TABLE I
USE DIRICHLET TO SIMULATE NON-IID

| Model | Fraction | Accuracy | Round | Traffic | Benefit |
|---|---|---|---|---|---|
| CNN | 100% | 62.2% | 15 | $2 \times 15 \times (50 + 9)$ M | 0.0% |
|  | 75% | 61.7% | 17 | $2 \times 17 \times (37 + 9)$ M | -11.63% |
|  | 50% | 61.8% | 19 | $2 \times 19 \times (24 + 9)$ M | -29.15% |
|  | 25% | 59.5% | 23 | $2 \times 23 \times (18 + 9)$ M | **-29.83%** |
| ResNet18 | 100% | 89.3% | 12 | $2 \times 15 \times (50 + 9)$ M | 0.0% |
|  | 75% | 88.9% | 15 | $2 \times 17 \times (37 + 9)$ M | -2.54% |
|  | 50% | 89.0% | 19 | $2 \times 19 \times (24 + 9)$ M | -11.44% |
|  | 25% | 88.5% | 22 | $2 \times 23 \times (18 + 9)$ M | **-16.1%** |
| VGG16 | 100% | 89.0% | 14 | $2 \times 15 \times (50 + 9)$ M | 0.0% |
|  | 75% | 88.5% | 16 | $2 \times 17 \times (37 + 9)$ M | -10.8% |
|  | 50% | 88.6% | 21 | $2 \times 19 \times (24 + 9)$ M | **-16.1%** |
|  | 25% | 88.8% | 27 | $2 \times 23 \times (18 + 9)$ M | -11.7% |

TABLE II
TAKE TWO OUT OF TEN CLASSES.

| Model | Fraction | Accuracy | Round | Traffic | Benefit |
|---|---|---|---|---|---|
| CNN | 100% | 43.2% | 36 | $2 \times 36 \times (50 + 10)$ M | 0.0% |
|  | 75% | 43.0% | 45 | $2 \times 45 \times (36 + 10)$ M | -4.16% |
|  | 50% | 41.7% | 52 | $2 \times 52 \times (24 + 10)$ M | -18.14% |
|  | 25% | 41.3% | 56 | $2 \times 56 \times (20 + 10)$ M | **-22.22%** |
| ResNet18 | 100% | 65.5% | 60 | $2 \times 60 \times (50 + 10)$ M | 0.0% |
|  | 75% | 64.8% | 73 | $2 \times 73 \times (36 + 10)$ M | -6.72% |
|  | 50% | 65.3% | 76 | $2 \times 76 \times (24 + 10)$ M | **-28.22%** |
|  | 25% | 65.2% | 106 | $2 \times 106 \times (20 + 10)$ M | -11.66% |
| VGG16 | 100% | 43.2% | 103 | $2 \times 103 \times (50 + 10)$ M | 0.0% |
|  | 75% | 43.0% | 127 | $2 \times 127 \times (36 + 10)$ M | **-5.46%** |
|  | 50% | 41.7% | 183 | $2 \times 183 \times (24 + 10)$ M | +0.67% |
|  | 25% | 41.3% | 200 | $2 \times 200 \times (20 + 10)$ M | -2.91% |

## VI. CONCLUSION

This study introduces a "Cluster-based Privacy-Enhanced Hierarchical Federated Learning Framework with Secure Aggregation (CPE-HFL)." We employ clustering to alleviate the impact of Non-IID data on hierarchical federated learning. This helps ensure that training accuracy is not compromised when selecting clients for training, even in data imbalance. Furthermore, we incorporate differential privacy and secure aggregation techniques to safeguard the data distribution used for clustering and the models clients return in the federated learning process. In other words, in the experiment results, we achieved a communication volume reduction of up to 29%, all while preserving accuracy. Notably, accuracy demonstrated more significant improvements when incorporating clustering. This framework can effectively reduce communication expenses and safeguard clients' privacy while upholding training accuracy.

## ACKNOWLEDGEMENTS

This work is sponsored by the National Science and Technology Council (NSTC) under the projects NSTC 112-2622-8-A49-021 and NSTC 110-2222-E-008-008-MY3.

## REFERENCES

[1] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

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

[4] Lumin Liu et al., "Client-edge-cloud hierarchical federated learning," in *IEEE International Conference on Communications (ICC)*.

[5] Ligeng Zhu, Zhijian Liu, and Song Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[6] Bo Zhao et al., "idlg: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.

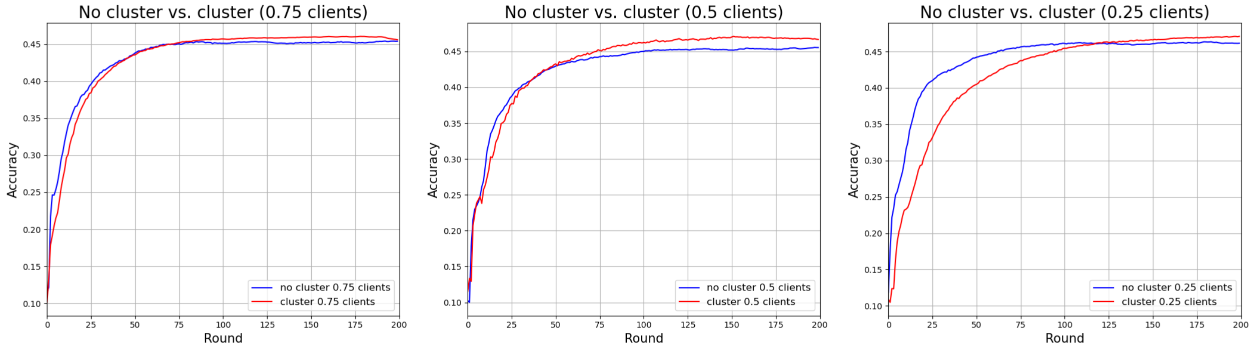

Fig. 3. Comparison CNN between Clustered and Non-clustered

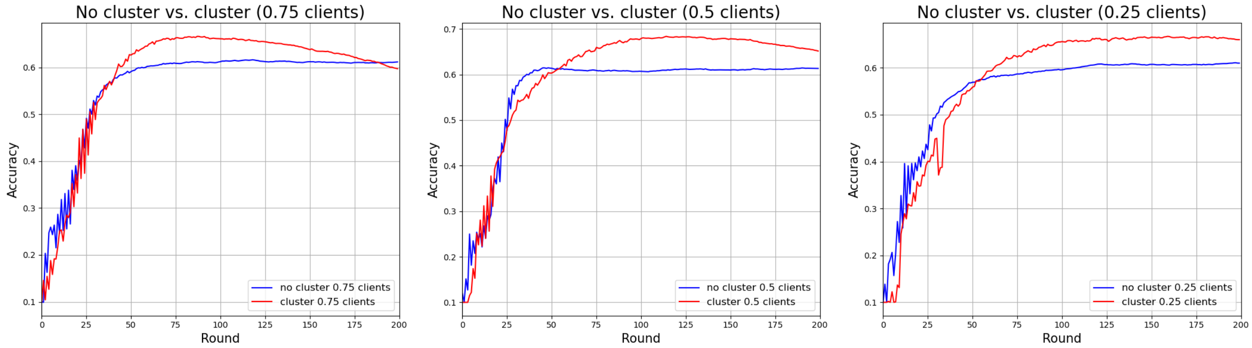

Fig. 4. Comparison ResNet18 between Clustered and Non-clustered

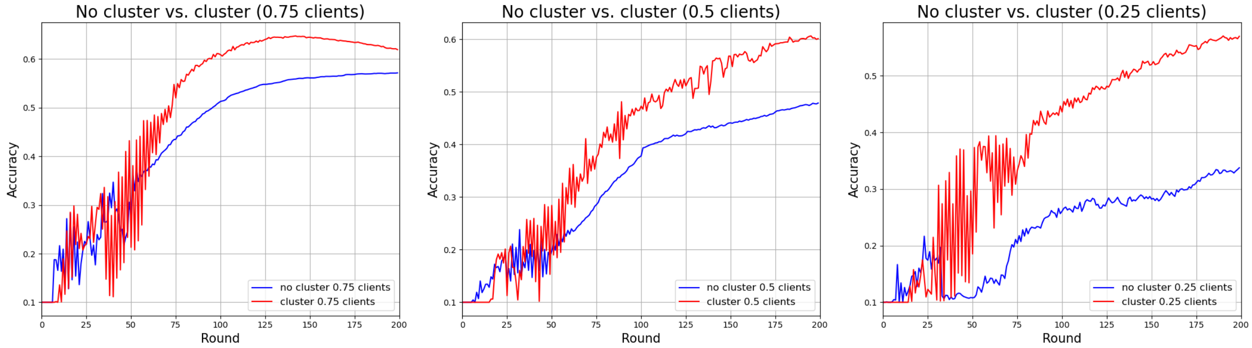

Fig. 5. Comparison VGG16between Clustered and Non-clustered

[7] Pu Tian et al., "Wscc: A weight-similarity-based client clustering approach for non-iid federated learning," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20243–20256, 2022.

[8] Mahdi Morafah et al., "Flis: Clustered federated learning via inference similarity for non-iid data distribution," *arXiv preprint arXiv:2208.09754*, 2022.

[9] Christopher Briggs et al., "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *IEEE International Joint Conference on Neural Networks (IJCNN)*.

[10] Keith Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *ACM SIGSAC Conference on Computer and Communications Security*.

[11] Kang Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, 2020.

[12] Kwing Hei Li et al., "Secure aggregation for federated learning in flower," in *ACM International Workshop on Distributed Machine Learning*.