Stochastic Clustered Federated Learning

Dun Zeng
University of Electronic Science and Technology of China
Cheng Du, China
zengdun@std.uestc.edu.cn

Xiangjing Hu
Harbin Institute of Technology, Shenzhen
Shenzhen, China
xiangjinghu@stu.hit.edu.cn

Shiyu Liu
University of Electronic Science and Technology of China
Cheng Du, China
shyu.liu@foxmail.com

Yue Yu
Peng Cheng Lab
Shenzhen, China
yuy@pcl.ac.cn

Qifan Wang
Meta AI
Menlo Park, USA
wqfcr618@gmail.com

Zenglin Xu*
Harbin Institute of Technology, Shenzhen
Shenzhen, China
xuzenglin@hit.edu.cn

ABSTRACT

Federated learning is a distributed learning framework that takes full advantage of private data samples kept on edge devices. In real-world federated learning systems, these data samples are often decentralized and Non-Independently Identically Distributed (Non-IID), causing divergence and performance degradation in the federated learning process. As a new solution, clustered federated learning groups federated clients with similar data distributions to impair the Non-IID effects and train a better model for every cluster. This paper proposes StoCFL, a novel clustered federated learning approach for generic Non-IID issues. In detail, StoCFL implements a flexible CFL framework that supports an arbitrary proportion of client participation and newly joined clients for a varying FL system, while maintaining a great improvement in model performance. The intensive experiments are conducted by using four basic Non-IID settings and a real-world dataset. The results show that StoCFL could obtain promising cluster results even when the number of clusters is unknown. Based on the client clustering results, models trained with StoCFL outperform baseline approaches in a variety of contexts.

KEYWORDS

Clustered Federated Learning, Non-IID Data, Bi-level Optimization

ACM Reference Format:
Dun Zeng, Xiangjing Hu, Shiyu Liu, Yue Yu, Qifan Wang, and Zenglin Xu. 2023. Stochastic Clustered Federated Learning. In *KDD FL4Data-Mining '23, August 7, 2023, Long Beach, CA, USA.*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Federated earning (FL) [18, 25] allows smart devices or institutions to collaboratively conduct machine learning tasks without violating privacy regulations. In this way, data collected on mobile phones and personal computers could be fully utilized by the framework. Furthermore, these data samples usually contain habits, preferences, and even geographic information. Hence, the distribution of data samples among devices in this decentralized system can be quite heterogeneous. Previous research [11, 13, 18, 27, 28] has revealed that the heterogeneous distribution of data between FL devices can cause divergence or slow convergence in the FL training process, which is referred to as Non-Independently Identically Distributed (Non-IID) issues.

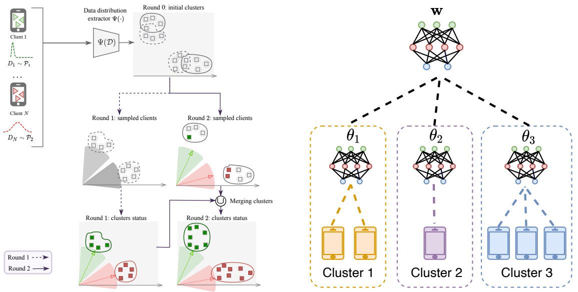
Clustered federated learning (CFL) [1, 4, 5, 19, 20] is an approach to address the Non-IID issues by clustering clients with similar data distributions and learning a personalized model for each cluster. In this case, the Non-IID issues are negligible within the clusters that can be solved easily. Typically, the CFL is built on an assumption,

Assumption 1 (Clustered Federated Learning [19]). There exists a partitioning $C = \{c_1, \dots, c_K\}, \bigcup_{k=1}^{K} c_k = \{1, \dots, N\}$ $(N \geq K \geq 2)$ of the client population, such that every subset of clients $c_k \in C$ satisfies the distribution learning assumption (i.e., the data distribution among these clients is similar).

It is assumed that clients in the same group have a similar data distribution. Hence, FedAvg [18] can thus fit data samples in the same cluster very well as long as the clustering goal is fulfilled. Meanwhile, the convergence study of such a CFL pattern is given in [15]. However, a number of actual limitations are not considered in existing CFL algorithms for real-world applications. For example, some CFL algorithms [4, 19, 21] require all clients to participate in the FL process. However, federated devices are not always online in real-world applications, especially in cross-device settings. Other studies [5, 16, 23] necessitate information about the number of clusters, which is often difficult to get in real-world systems due to privacy constraints. As a result, incorrectly estimating the number of clusters may hamper the FL models' performance. Overall, the application opportunities for CFL approaches are limited due to these severe constraints. Therefore, it is necessary to study a flexible and practical clustered federated learning framework for unknown and Non-IID data in FL scenarios.

[*] Corresponding author

KDD FL4Data-Mining '23, August 7, 2023, Long Beach, CA, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn



(a) Stochastic Client Clustering (b) Bi-level Clustered Federated Learning

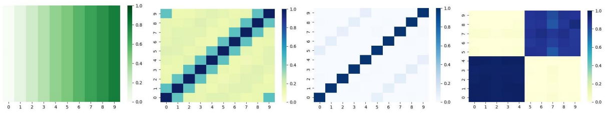
Figure 1: StoCFL consists of two vital components: stochastic client clustering and bi-level clustered federated learning. In Figure 1(a), the distribution extractor $\Psi(\cdot)$ draws local data distribution representation. Then, the server randomly samples a subset of clients for each round and merges clients into clusters based on cosine similarity. Compared with conventional CFL algorithms (in dashed boxes), Figure 1(b) illustrates an example with 3 clusters that StoCFL enables cluster models to improve each other via a global model w.

To this end, we introduce StoCFL, a novel CFL algorithm that does not require the number of clusters to be known in advance and allows an arbitrary number of clients to participate in each FL round. In detail, StoCFL creates a representation of local data distributions and evaluates the distribution similarity of any two clients via cosine similarity. Based on this, we implement stochastic federated client clustering, which solves the client clustering problem in that only a subset of clients participates in each round. Furthermore, we propose a bi-level CFL algorithm that enables a knowledge-sharing scheme to further enhance the model performance. Our experimental results show that stochastic federated client clustering can produce promising cluster results on four Non-IID settings and the real-world dataset FEMNIST, including both cross-device (4,800 clients) and cross-silo settings (20 clients), while the cluster models trained with StoCFL outperform conventional CFL techniques.

2 STOCHASTIC CLUSTERED FEDERATED LEARNING

In this section, we provide the details of StoCFL, where the StoCFL consists of a stochastic federated client clustering algorithm and a bi-level CFL algorithm. We illustrate the StoCFL in the following order. Firstly, we build a data extractor function for representing the local data distribution and empirically prove the effectiveness of the similarity metric of clients' local datasets in Section 2.1. Based on this, we propose a stochastic client clustering algorithm in Section 2.2. Then, we present the bi-level CFL algorithm to train better cluster models in Section 2.3.

Notation: We use $[R]$ to denote the set of integers $\{1, 2, \dots, R\}$, $\|\cdot\|$ to denotes the ℓ_2 norm of vectors, and $|\cdot|$ to denote the size of a set. For FL notation, we let N be the number of all federated clients. We use $\mathcal{P}_k, k \in [K]$ to denote the latent data source distribution, and K is the number of data source distributions. We consider each federated client $i \in [N]$ to have a local data set D_i.



(a) Label distribution skew (b) Feature distribution skew (c) Label concept skew (d) Feature concept skew

Figure 2: Visualization of the cosine similarity matrix. Let s denote the value of x-axis or y-axis in the figures. For the Figure. (a), s indicates the number of the same label between these two clients. For the Figure. (b), we rotated the data with $36 \times s$ degrees. For the Figure. (c), we modified the label $y = (y + s) \bmod 10$. For the Figure. (d), we use gradients calculated based on data samples from MNIST ($s \in [0, 5)$) and Fashion-MNIST ($s \in [5, 10)$). Then, we random initialize a linear model as anchor $\boldsymbol{\psi}$ for the digit recognition task. We calculate the representation $\Psi(\cdot)$ based on these partitioned datasets and observe the cosine similarity.

2.1 Key Observation

The conventional CFL approach is to cluster clients with similar data distribution. Then, it aims to minimize the following objectives for each cluster $k \in [K]$:

$$\begin{aligned} &\min_{\theta_1,\dots,\theta_K} \mathbb{E}_{D^{(k)} \sim \mathcal{P}_k}[\ell(\boldsymbol{\theta}_k; D^{(k)})], \\ &\text{s.t. } C = \{c_1, \dots, c_K\},\ D^{(k)} = \cup_{i \in c_k} D_i, \end{aligned} \tag{1}$$

where C denotes client clustering results, and $D^{(k)}$ denotes the data samples of all clients in cluster k. The federated client clustering results, i.e., the subjective term in Equation (1), determines the performance of CFL to some extent. Therefore, Equation (1) motivates that the federated client clustering is a vital component in CFL.

We design a distribution extractor function $\Psi(D) = \frac{\partial \ell(\boldsymbol{\psi}; D)}{\partial \boldsymbol{\psi}}$, which indicates the updated direction toward the local minimum corresponding for the input dataset D, anchor model $\boldsymbol{\psi}$ and loss function ℓ. We do not optimize the anchor model $\boldsymbol{\psi}$ and maintain the loss function ℓ constant across all datasets in our implementations. As a result, the $\Psi(\cdot)$ output can be viewed as a representation of data distribution corresponding to the input dataset. Based on the findings with Non-IID data [28], we expect datasets with similar data distributions to provide similar $\Psi(\cdot)$ values. Then, we use cosine similarity to evaluate the distribution similarity of the two decentralized datasets, i.e., given any two unknown datasets D_i, D_j, the similarity is determined as: $\cos(\Psi(D_i), \Psi(D_j)) = \frac{\Psi(D_i) \cdot \Psi(D_j)}{\|\Psi(D_i)\| \|\Psi(D_j)\|}$. To better support our assumptions, we implement observation experiments on cosine similarity, as shown in Figure 2, in which we augment MNIST/Fashion-MNIST dataset and partition them with varying levels of augmentation (similar with augmentation described in Appendix A.1). The results reveal a significant difference in cosine similarity values. As a result, we conclude that $\Psi(\cdot)$ could represent a local data distribution, and clients with similar local data distributions (at both label and feature levels) have higher cosine similarity.

2.2 Stochastic Federated Client Clustering

Based on the observations, we implement a stochastic federated client clustering algorithm in this section. In detail, we aim to cluster federated clients by minimizing the following objective:

$$\min_{C} \sum_{i=1}^{\tilde{K}} \sum_{j=i+1}^{\tilde{K}} \cos\left(\Psi(\tilde{D}^{(i)}), \Psi(\tilde{D}^{(j)})\right), \tag{2}$$

where $\Psi(\tilde{D}^{(j)}) \triangleq \frac{1}{|c_j|}\sum_{i\in c_j}\Psi(D_i)$ is the average point for decentralized local datasets of clients in the current j-th cluster in C, $\tilde{K}$ denotes the number of clusters and $\tilde{K} = |C|$. As the larger cosine similarity indicates the closer distance, minimizing the objective is to find the best partition C where the representations of current clusters are far from each other.

At the initialization stage of stochastic federated client clustering, we treat each client as a single cluster. In other words, the server maintains a partition set $C = \{c_1, c_2, \ldots, c_N\}$, where N is the number of clients. For each $c_i \in C$, we have $c_i = \{i\}$ and i denotes the client id. Meanwhile, we have $\Psi(\tilde{D}^{(k)}) = \Psi(D_i)$ for all $k, i \in [N]$ and $\tilde{K} = N = |C|$ at the beginning.

For the federated client clustering process, we greedily decrease the value of Equation (2) by merging the similar clusters sampled at each round. We adjust this merging process via a threshold τ, indicating the minimum cosine similarity that two datasets should be considered similar. In practice, the Equation (2) can be represented by a pair-wise cosine similarity matrix M, where $M_{i,j}$ indicates the cosine similarity between the i-th and the j-th clusters in C.

We summarise the stochastic federated client clustering procedure in Lines 4-13, Algorithm 1. For each federated round, the server would request the local data distribution representation from the sampled clients (Line 6). Then, the server updates the distribution representation of clusters (Line 9). Finally, the server merges any two clusters that satisfy the requirements (Lines 11-13). For each merging procedure, the current number of clusters $\tilde{K}$ is reduced by 1. If the federated server samples all clients at the first round, StoCFL recovers to client-wise agglomerative clustering, with the metric provided by distribution extractor $\Psi(\cdot)$.

2.3 Bi-level Clustered Federated Learning

In this section, we established a bi-level CFL objective to further improve conventional CFL approaches. Looking back to the conventional CFL objective in Equation (1), the cluster models are optimized within the cluster alone, with no inter-cluster knowledge sharing. Although the implicit data distribution may differ between clusters, we argue that there is certain knowledge that can be shared by one cluster model to better the other. Based on the observation, we propose the bi-level CFL objective, which regularizes the local optimization and improves cluster models via a shared global model.

In detail, corresponding with the client clustering procedure, the server maintains a global model denoted by $\boldsymbol{\omega}$ and cluster models $\boldsymbol{\theta}_k, k \in [\tilde{K}]$. Our method solves a bi-level optimization problem for all cluster $k \in [\tilde{K}]$ given by:

$$\begin{aligned} &\min_{\boldsymbol{\theta}_k} f_k(\boldsymbol{\theta}_k) + \frac{\lambda}{2}\|\boldsymbol{\theta}_k - \boldsymbol{\omega}^*\|^2, \\ &s.t.\ \boldsymbol{\omega}^* \in \arg\min_{\boldsymbol{\omega}} G\big(f_1(\boldsymbol{\omega}), \ldots, f_N(\boldsymbol{\omega})\big), \end{aligned} \tag{3}$$

where N is the total number of clients, $f_i(\cdot) = \mathbb{E}[\ell(\cdot; D_i)]$ is empirical loss for the i-th client, and $G(\cdot)$ denotes the global objective function for the global model $\boldsymbol{\omega}$.

At the initialization stage, we make $\boldsymbol{\omega}_0 = \boldsymbol{\theta}_1 = \cdots = \boldsymbol{\theta}_N$ and $\tilde{K} = N$. Meanwhile, if any two clusters are merged in the client clustering, then the server will merge corresponding cluster models. Hence, the real-time number of clusters $\tilde{K}$ is always the same as the number of cluster models.

The pseudocode of bi-level CFL is described in Lines 14-23, Algorithm 1. During the training process, the server broadcasts to sampled clients the global model $\boldsymbol{\omega}$ and corresponding cluster model $\boldsymbol{\theta}_k$. Then, the sampled clients perform several steps of SGD to optimize the cluster model (Line 21) and global model (Line 22) locally before uploading updated models to the server. The server updates the global model by aggregating the models from all sampled clients (Line 17). Then, the server updates the cluster models respectively (Lines 18-19).

2.4 Discussion

This section goes through the critical components that could affect the StoCFL. In particular, we discuss the impact of the global model on the cluster model. Additionally, we explain the clustering and optimization parameters to clarify the position of StoCFL in global FL and personalized FL.

Global model $\boldsymbol{\omega}$. In our bi-level optimization objective, the global model $\boldsymbol{\omega}$ fits the data samples of all clients. Hence, the global model preserves the knowledge (including distribution and feature information) of all clients. As a result, using the regularization term in local cluster model optimization, knowledge from separate clusters might be transferred to others. Additionally, the clustered learning process is separated from the global model optimization process. Hence, StoCFL is free to select the global objective $G(\cdot)$ [10]. Consequently, the cluster model could inherit the convergence benefit (e.g., robustness or fairness).

Relation to clusters C. The performance of cluster models is subjected to the client clustering results. Meanwhile, the client clustering results $\tilde{K}, C$ depend on the merging threshold τ. Particularly, no clusters will be merged if $\tau = 1$. Then, the final number of clusters will be $\tilde{K} = N$, which will degenerate the optimization objective function (3) to the personalized FL algorithm Ditto [10]. If $\tau = -1$, however, all clients will be clustered together. As a result, the optimization objective degenerates to the global FL algorithm FedProx [12]. Hence, by altering the clustering threshold τ, the StoCFL could achieve an effective balance between global FL and personalized FL.

Regularization weight λ. The λ is to adjust the impact of the global model on cluster models. When λ is set to 0, the objective function degenerates into the conventional CFL task that is described in Equation (1). As the λ grows large, it makes the cluster model reach the global objective function $G(\cdot)$. Furthermore, if

Algorithm 1: Stochastic Clustered Federated Learning

```
Input: Client set $S$, where $|S| = N$, initialized cluster
       partition $C = \{c_1, \ldots, c_N\}$, initialized model
       $\omega_0, \theta = [\theta_1, \ldots, \theta_N]$, anchor $\psi$, threshold $\tau$ and
       learning rate $\eta$.
Output: Cluster result $C = \{c_1, \ldots, c_{\tilde{K}}\}$, cluster models
        $\theta_1, \ldots, \theta_{\tilde{K}}$, and global model $\omega$
ServerProcedure:
    $P \leftarrow \emptyset$
    for round $t \in [T]$ do
        Random sample a subset of client $S^t \subseteq S, |S^t| = m$.
        // federated client clustering
        for client id $i \in (S^t \cap \complement_S P)$ in parallel do
            $\Psi(D_i) \leftarrow$ GetDatasetRepresentation($\psi$)
        end
        $P \leftarrow P \cup S^t$
        for $c_k \in C$ do
            $\Psi(\tilde{D}^{(k)}) = \sum_{i \in c_k} \Psi(D_i)$
        end
        Obtain cosine similarity matrix $M$ following
         Equation 2.
        for $M_{i,j}$ in $M$ do
            if $M_{i,j} > \tau$ then
                $c_i \leftarrow c_i \cup c_j, C \leftarrow C / c_j$
            end
        end
        // clustered federated learning
        for client id $i \in S^t$ in parallel do
            Let $k$ denote the index of the cluster and client
             id $i \in c_k$.
            $\omega_t^i, \theta_k^i \leftarrow$ ClientProcedure($\omega_t, \theta_k$)
        end
        $\omega_{t+1} \leftarrow$ Aggregate($[\omega_t^i]$), $i \in S^t$
        for $c_k \in C^t$ and indexed by $k$ do
            $\theta_k \leftarrow$ FedAvg($[\theta_k^i]$), $i \in c_k$
        end
    end
ClientProcedure($\omega, \theta_k$):
    $\theta_k^i \leftarrow \theta_k - \eta\big(\nabla f_i(\theta_k) + \lambda(\theta_k - \omega)\big)$
    $\omega^i \leftarrow \omega - \eta \nabla f_i(\omega)$
    return $\omega^i, \theta_k^i$
```

$\lambda = 0, \tau = -1$, StoCFL recovers to FedAvg. The impacts of λ are further studied in Table 5, Section A.2.

Limitations. The proposed Algorithm 1 induces additional but necessary computation. For the server side, we require to compute a similarity matrix of clients, where the computation complexity is $O(\tilde{K}^2 d)$. It also costs each of the clients to compute the local distribution representation vector once and an additional local training procedure for updating the global model ω. Besides, despite this paper does not directly provide convergence guarantees, it can be obtained from literature [10, 15].

3 EXPERIMENT EVALUATION

In this part, we assess the performance of StoCFL using four basic Non-IID settings and the real-world dataset FEMNIST. Our experiment investigation includes federated client clustering and CFL evaluation. Furthermore, we discuss the effect of StoCFL hyper-parameters. In the end, we demonstrate the application inference ability of StoCFL and study its generalization ability to unseen clients. The experiment is developed using the open-source FL framework [26].

3.1 Experiment Setup

Baselines. We compare the IFCA [5] and CFL [19] under different Non-IID settings. Before that, we first recall the details of the CFL techniques. IFCA [5] takes an input of the assumption number of cluster $\tilde{M}$. Then, the server initializes $\tilde{M}$ different models for each cluster and broadcasts them to clients for each round. The clients will optimize the specific model, which achieves the lowest forward loss on the local dataset. Clients upload the updated model to the server. The server would aggregate the updated $\tilde{M}$ models in the same cluster following FedAvg. CFL [19] monitors the model updates of all clients for each round. In the beginning, all clients are in the same cluster. Then, the server will bi-partition clients into two clusters when certain conditions are satisfied. Particularly, The CFL server would recursively run the above procedures in the same cluster till the partition conditions are no longer satisfied (client clustering finished).

Settings. We compare the baseline algorithms respectively using identical experimental conditions as described in their study. The MNIST task model is a linear classification model with a hidden layer of 2048 units, and the CIFAR10 task model is a convolution neural network model with two convolutional layers followed by two fully connected layers. We execute five runs for each experiment, each with a different random seed. We also provide the average test accuracy and standard deviation. Without loss of generality, we initialize model $\psi = \omega_0$ for the distribution extractor function $\Psi(\cdot)$ in experiments, and the loss function ℓ is the cross-entropy loss for the classification task.

3.2 Comparison Experiment

Experiments in IFCA setting. Following the setting of IFCA, we create the Rotated MNIST and Rotated CIFAR10 datasets. Firstly, We rotated the MNIST data samples by 0, 90, 180, and 270 degrees, resulting in cluster number $K = 4$. Then, we randomly partition datasets into $N = (4800, 2400, 1200)$ clients, and each client has $|D| = (50, 100, 200)$ data samples. For the configuration of MNIST tasks, We run 100 federated communication rounds for the Rotated MNIST setting. Each client performs 5 epochs of local SGD with full batch size and the learning rate $\eta = 0.1$. Besides, the Rotated CIFAR10 is created similarly with 0 and 180 degrees of rotation. We run 200 federated communication rounds for the Rotated CIFAR setting. Each client performs 5 epochs of local SGD with a local batch size is 50 and the learning rate $\eta = 0.1$. For the hyper-parameters of StoCFL, the stochastic client clustering threshold $\tau = 0.5$, and the optimization parameter $\lambda = 0.05$.

We highlight that the original experiments in IFCA report a full client sample rate in MNIST and a client sample rate of 10% in

Table 1: Test accuacies(%)± std.

	Rotated MNIST, K=4						**Rotated CIFAR**, K=2	
N, $\|D\|$	4800, 50		2400, 100		1200, 200		200, 500	
Sample Rate	10%	100%	10%	100%	10%	100%	10%	100%
FedAvg	95.72±0.13	96.10±1.26	95.32±0.16	96.00±1.31	94.96±0.11	95.72±1.60	48.26±0.66	47.98±1.73
FedProx	76.45±0.07	76.47±0.08	77.40±0.09	77.48±0.08	77.82±0.07	77.90±0.09	47.17±0.35	47.15±0.36
Ditto	65.05±0.05	73.60±0.05	73.93±0.11	79.71±0.05	79.38±0.08	83.96±0.03	46.91±0.38	46.61±0.39
IFCA $\tilde{M}$=2	84.42±1.05	84.98±0.28	85.25±0.94	84.64±2.54	86.12±0.78	84.37±4.40	50.62±1.86	49.83±2.79
IFCA $\tilde{M}$=4	91.74±0.02	90.43±2.71	91.39±1.24	90.75±2.48	91.00±1.44	88.69±2.83	50.80±1.69	51.40±2.00
IFCA $\tilde{M}$=6	91.74±0.03	91.65±0.12	91.94±0.13	91.99±0.04	92.15±0.07	92.17±0.04	51.04±0.40	50.63±1.85
StoCFL	**97.00±0.04**	**97.36±0.38**	**96.89±0.02**	**97.4±0.35**	**96.71±0.04**	**97.40±0.38**	**52.84±0.74**	**54.99±0.66**

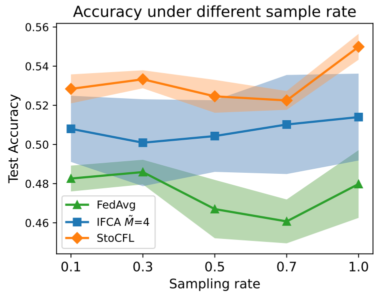


Figure 3: Robustness comparison of different federated learning methods with various sampling rates.

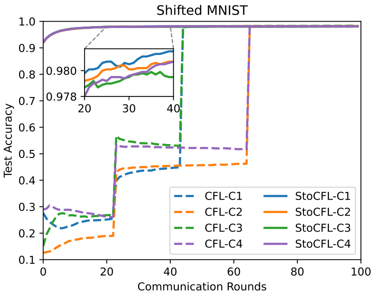


Figure 4: Evolution curves of test accuracy for CFL and StoCFL with increasing communication rounds.

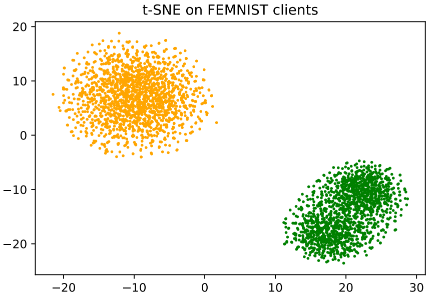


Figure 5: Clustering Results of StoCFL on FEMNIST.

Table 2: Test accuracy(%)± std

	Shifted MNIST	**Shifted CIFAR**
N, $\|D\|$	20, 9600	20, 8000
FedAvg	24.44±0.03	13.01±2.48
IFCA	98.11±0.03	54.05±0.59
CFL	98.20±0.02	55.23±1.02
StoCFL	98.14±0.01	55.42±1.42

CIFAR10. For a fair comparison, we choose different sample rates for both MNIST and CIFAR10 tasks. In addition, we compare to FedAvg, FedProx, and Ditto to demonstrate the improvement of StoCFL. The performance results are shown in Table 1. According to the results, StoCFL outperforms IFCA in most cases. The global baseline performance is worse since the global model tries to fit all data samples from all distributions. More importantly, we observe that IFCA fails to cluster clients from different data distributions in MNIST experiments, where a particular model may dominate another model from the beginning. That is, if a model fits two distributions well in the first few rounds, then this model will dominate another model, making it no chance to fit the expected distribution (no clients will update this model). Hence, we argue that IFCA depends on model initialization to some extent. Meanwhile, previous studies [15, 24] on the heterogeneity of FL observe a similar phenomenon in the IFCA algorithm.

We study the effect of client sample rate on StoCFL with the Rotated CIFAR10 setting, where the results are depicted in Figure 3. The performance curve of StoCFL is stable and better with different settings of the sample rate. Hence, the results reveal that StoCFL is robust and flexible with the proportion of client participation.

Experiments in CFL setting. Following the setting of CFL, we conduct experiments on the Shifted MNIST and the Shifted CIFAR-10 dataset [7]. In this case, datasets are partitioned into $N = 20$ clients, each belonging to one of $K = 4$ clusters. The labels of each client data are modified by randomly shifted labels, i.e., $\tilde{y} = (y + s)\%10, s \in \{0, 3, 6, 9\}$. For a fair comparison, we let all clients participate in StoCFL in this part. Additionally, we also conduct IFCA on this setting for comparison. The performance is shown in Table 2. We obtain a close model performance in the setting of CFL, however, the accuracy curve of StoCFL is better in most training rounds (shown in Figure 4). Besides, the CFL requires full client participation for every round to bi-partition the clients at a proper stage. In contrast, StoCFL supports an arbitrary proportion of client participation, which reveals the flexibility of StoCFL.

Real-world dataset evaluation. We provide the additional experimental results on the real-world dataset FEMNIST [3], which is proposed by LEAF [2]. It is a realistic FL dataset where the data samples on a single client are the handwritten digits or letters from a specific writer. Besides, There are 3,597 clients in FEMNIST, and the number of data samples among these clients is different. Hence, this setting could be considered a hybrid Non-IID scenario. Although there are no clear clusters for FEMNIST clients, the writing style of different people may be clustered. For the model, we use a neural network with two convolutional layers, with a max pooling layer after the convolutional layer, followed by two fully connected layers. We also added a dropout ratio of 0.5 between the convolutional layer and the fully connected layer. The network parameters initialization follows Xavier [6]. We run 100 rounds of FL with 5% clients sampled per round. We also evaluate IFCA and CFL on FEMNIST for comparison. The final results on FEMNIST are reported in Table 3.

The results indicate that StoCFL achieves better performance on FEMNIST, compared with IFCA and CFL. Furthermore, we highlight that CFL consumes more resources as it requires the full participation of clients. For the client clustering results, we observe that StoCFL clusters FEMNIST clients into two main clusters (shown in Figure 5), i.e., FEMNIST consists of two implicit data-generating distributions. Importantly, previous studies [5, 17] on FEMNIST draw the same conclusion, which proves the StoCFL correct. In all, the real-world dataset evaluation results reveal the ability of StoCFL to handle non-synthetic datasets, and further indicate the practical value of StoCFL.

Table 3: Test accuracy(%)± std on FEMNIST.

IFCA	$\tilde{M}$=2	$\tilde{M}$=3	**CFL**	**FedAvg**
	73.11±0.92	69.11±2.49	79.64±0.37	86.27±0.11
StoCFL	τ=0.55	τ=0.60	τ=0.65	
	90.92±0.02	90.71±0.04	90.23±0.20	

3.3 Cluster Inference and Generalization

In this section, we further explain the advantages of StoCFL for practical applications. First, we demonstrate the ability of StoCFL to infer newly joined clients, which reveals that StoCFL could handle a varying FL system. Based on the inference ability, we further study the generalization performance of StoCFL in the FEMNIST setting.

Cluster inference for newly joined clients. StoCFL is flexible with newly joined clients. In other words, the StoCFL server could determine which cluster to join for a new coming client. Furthermore, StoCFL is adaptable in terms of inferring a newly joined client during or after the training process. Consider a newly joined client with a local dataset $\hat{D}$ that reports the local representation $\Psi(\hat{D})$ to the server. Then, the server could then use a few steps to determine the target cluster:

(1) Calculate the candidate cluster with the closest distance

$$\hat{d} = \arg\max \cos(\Psi(\hat{D}), \Psi(\tilde{D}^{(j)})), \ \ j \in [\tilde{K}],$$

and record the candidate cluster id $\hat{c}$.

(2) If $\hat{d} \geq \tau$, then assign the client to the cluster $\hat{c}$. Otherwise, the server assigns the client to a new cluster marked by $\tilde{K}+1$, $\tilde{K} = \tilde{K} + 1$, and let $\theta_{\tilde{K}} = \theta_{\hat{c}}$.

We emphasize that if the client is unable to join any existing cluster, the server should assign a cluster model with the closest cluster distribution. As a result, the cluster model could be learned from good initialization during the training process. Furthermore, once the FL training process is finished, we could use the closest model to predict data samples from new clients.

Table 4: Generalization performance (%) on FEMNIST

Method	FedAvg	CFL	IFCA	StoCFL
Test Accuracy	85.93±0.16	79.78±0.28	74.82±0.49	**91.00±0.05**
Unparticipation	86.70±0.22	73.24±0.99	75.78±0.45	**91.06±0.05**

Generalization to unseen clients. The experiments in this section are carried out in the FEMNIST environment in order to evaluate the generalization performance of cluster models. In particular, 1,079 clients (30%) were selected as test clients who did not participate in the FL process. Only 2,518 clients (70%) are participating in the CFL process in this case. Then, using this configuration, we run StoCFL, CFL, and IFCA to infer the cluster of unparticipated clients using their training data samples. As a result, we report in Table 4 the accuracy of test data samples from participated and unanticipated clients. According to the results, StoCFL achieves better generalization performance while preserving higher test accuracy.

4 CONCLUSION

In this paper, we proposed a novel CFL algorithm StoCFL, which consists of stochastic client clustering and bi-level CFL algorithms. Our study demonstrates that StoCFL could cluster federated clients with different unknown distributions and train better-generalized models. Besides, StoCFL is flexible and robust with real-world applications, especially in a varying FL system. Furthermore, the results of intensive experiments have shown the superiority of the proposed algorithms.

ACKNOWLEDGMENTS

This work was partially supported by the National Key Research and Development Program of China (No. 2018AAA0100204), a key program of fundamental research from Shenzhen Science and Technology Innovation Commission (No. JCYJ20200109113403826), an Open Research Project of Zhejiang Lab (NO.2022RC0AB04), the Major Key Project of PCL (No. PCL2021A06), and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (No. 2022B1212010005).

REFERENCES

[1] Christopher Briggs, Zhong Fan, and Peter Andras. 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–9.

[2] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097* (2018).

[3] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. 2017. EMNIST: Extending MNIST to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2921–2926.

[4] Moming Duan, Duo Liu, Xinyuan Ji, Yu Wu, Liang Liang, Xianzhang Chen, Yujuan Tan, and Ao Ren. 2021. Flexible Clustered Federated Learning for Client-Level Data Distribution Shift. *IEEE Transactions on Parallel and Distributed Systems* (2021).

[5] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems* 33 (2020), 19586–19597.

[6] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[7] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[9] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2021. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079* (2021).

[10] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*. PMLR, 6357–6368.

[11] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*, Inderjit S. Dhillon, Dimitris S. Papailiopoulos, and Vivienne Sze (Eds.). mlsys.org.

[12] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* 2 (2020), 429–450.

[13] Zonghang Li, Yihong He, Hongfang Yu, Jiawen Kang, Xiaoping Li, Zenglin Xu, and Dusit Niyato. 2022. Data Heterogeneity-Robust Federated Learning via Group Client Selection in Industrial IoT. *IEEE Internet Things J.* 9, 18 (2022), 17844–17857.

[14] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems* 30 (2017).

[15] Jie Ma, Guodong Long, Tianyi Zhou, Jing Jiang, and Chengqi Zhang. 2022. On the convergence of clustered federated learning. *arXiv preprint arXiv:2202.06187* (2022).

[16] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. 2020. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619* (2020).

[17] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. 2021. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems* 34 (2021), 15434–15447.

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

[19] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems* 32, 8 (2020), 3710–3722.

[20] Felix Sattler, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. 2020. On the byzantine robustness of clustered federated learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8861–8865.

[21] Morris Stallmann and Anna Wilbik. 2022. Towards Federated Clustering: A Federated Fuzzy c-Means Algorithm (FFCM). *arXiv preprint arXiv:2201.07316* (2022).

[22] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[23] Yi Wang, Mengshuo Jia, Ning Gao, Leandro Von Krannichfeldt, Mingyang Sun, and Gabriela Hug. 2022. Federated Clustering for Electricity Consumption Pattern Extraction. *IEEE Transactions on Smart Grid* 13, 3 (2022), 2425–2439.

[24] Shanshan Wu, Tian Li, Zachary Charles, Yu Xiao, Ziyu Liu, Zheng Xu, and Virginia Smith. 2022. Motley: Benchmarking Heterogeneity and Personalization in Federated Learning. *arXiv preprint arXiv:2206.09262* (2022).

[25] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[26] Dun Zeng, Siqi Liang, Xiangjing Hu, and Zenglin Xu. 2021. FedLab: A Flexible Federated Learning Framework. *arXiv preprint arXiv:2107.11621* (2021).

[27] Shenglai Zeng, Zonghang Li, Hongfang Yu, Yihong He, Zenglin Xu, Dusit Niyato, and Han Yu. 2022. Heterogeneous Federated Learning via Grouped Sequential-to-Parallel Training. In *Database Systems for Advanced Applications - 27th International Conference, DASFAA 2022, Virtual Event, April 11-14, 2022, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 13246)*. Springer, 455–471.

[28] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).

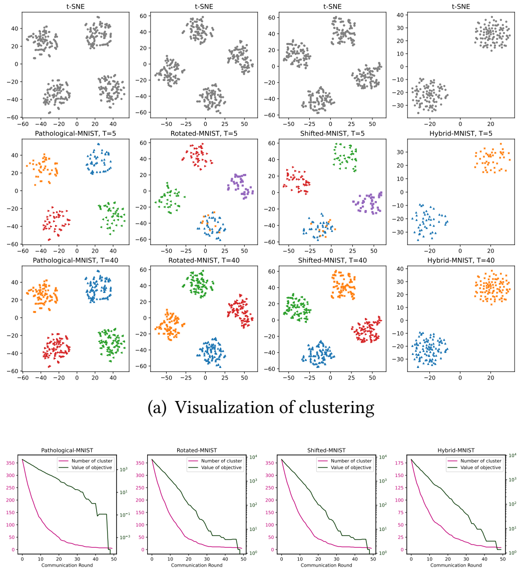
A DISCUSSION AND ADDITIONAL EXPERIMENTS

A.1 Data Partition & Additional Experiments on Client Clustering

We evaluate the stochastic federated client clustering algorithm on the MNIST [8] and Fashion-MNIST [9]. We recall that the MNIST/Fashion-MNIST dataset has 60,000 training samples and 10,000 test samples with ten classes. To simulate an FL environment where the data distributions among clients are different, we augment and partition these datasets as follows.

- **Pathological MNIST** [18]: we sort the data samples by labels and split them into {{0,1,2},{3,4},{5,6},{7,8,9}} (4 clusters). Then we randomly partition the dataset into 100 clients for each cluster, resulting in 400 clients with $K = 4$. We refer to it as *Label distribution skew.*
- **Rotated MNIST** [14]: we augment MNIST by rotating images with 0, 90, 180, and 270 degrees, resulting in $K = 4$ clusters. We randomly partition the Rotated MNIST of each degree into 100 clients. This scenario is *Feature distribution skew.*
- **Shifted MNIST** [19]: we modify the label of each sample by adding shift level s(i.e., $\bar{y}_s = (y + s)\%10, s \in 0, 3, 6, 9)$ and partition dataset of each shift level into 100 clients. Hence, this case shall be *Label concept skew.*
- **Hybrid MNIST**: we partition MNIST and Fashion-MNIST into 100 clients, resulting in 200 clients and 2 clusters. The label across clients is the same ($y \in [10]$), but the feature domain is different. Therefore, this case is *Feature concept skew.*

In experiments, we randomly initialize an anchor model ψ and choose the cross-entropy loss as ℓ for the classification task. Based on that, we obtain all representation vectors using $\Psi(\cdot)$ and display them using t-SNE [22] visualizing. We conducted 50 rounds of the proposed stochastic federated client clustering procedure, where only 10% of clients are randomly sampled each round. The visualization results are shown in Figure 6(a). The graph with grey points depicts the visualization results of distribution representation vectors. Furthermore, clear clusters are shown from the representation vectors captured by t-SNE, which further proves that $\Psi(\cdot)$ could extract and represent the local data distributions. Besides, an additional clustering status curve is depicted in Figure 6(b). The client clustering results demonstrate that our federated client clustering could deal with different Non-IID scenarios.



(a) Visualization of clustering

(b) Clustering status curve

Figure 6: Illustration of client clustering. In Figure (a), the first row is the t-SNE results on all $[\Psi(D_i)], i \in [N]$, where each point denotes the data distribution of the dataset from each client. The other row is the clusters at communication round $T = \{5, 10, 40\}$, which are colored by StoCFL. In Figure (b), we depicted the overall number of clusters and the value of the clustering objective in Equation 2 at each communication round.

A.2 Impact of Hyper-parameters

This section discusses the influence of regularization weight λ on model performance. Then, we illustrate the impacts of clustering threshold τ on clustering results. Based on the results, we provide insights about the tuning of hyper-parameters when deploying StoCFL on applications.

Effect of λ. We demonstrate the effect of λ via experiments on Pathological-MNIST, Rotated-MNIST, Shifted-MNIST, and Hybrid-MNIST settings. We run 50 communication rounds of StoCFL with $\lambda = \{0, 0.01, 0.05, 0.5, 1, 10\}$, and report the test accuracy of the global model and cluster models. The global accuracy is the performance of the final $\boldsymbol{\omega}$ on all augmented test sets, while the contents of other columns are the average performance of cluster models. The results are summarized in Table 5.

We note that $\lambda = 0$ indicates the conventional CFL with correct client clustering results. In comparison, the results with $\lambda > 0$ prove that StoCFL improves the cluster model's performance by introducing useful knowledge from other clusters via the regularization term. For instance, the global accuracy results of Rotated-MNIST and Shifted-MNIST are distinct, while the cluster models are enhanced by the regularization term with $\lambda = 0.05$. Besides, compared with the conventional CFL results ($\lambda = 0$), StoCFL is better.

Furthermore, given the impacts of decentralized data distributions in these four settings are different, the value of λ where the cluster models achieve the best accuracy is not the same. Hence, we conclude that the best λ relies on the real scenario of Non-IID data. In the real-world training process of StoCFL, the λ could be adjusted dynamically during the training process. Besides, we could refer to [10, 12] for the best strategies for choosing λ. We will further study the relation between λ and the Non-IID data in future work.

Table 5: Effect of λ. Test accuracy (%).

MNIST	Global	StoCFL, λ					
		10	1	0.5	0.05	0.01	0
Pathological	92.17	**89.28**	86.89	83.49	37.83	24.12	24.05
Rotated	92.65	62.06	94.88	95.28	**95.86**	94.20	92.25
Shifted	24.46	32.69	92.36	93.85	**95.12**	93.80	92.22
Hybrid	92.50	92.26	92.75	**92.77**	92.69	91.99	90.11

Effect of τ. We demonstrate the effects of τ on client clustering with a hybrid Non-IID setting combination. In this part, we follow the partition strategies described in Section A.1. Firstly, we create Rotate-MNIST by 0,180 degrees of rotation and marked them with R0, R180. Then, we pathologically partition (marked by P1, P2, P3, P4) each of them into 200 clients. In this case, we have 400 clients in total, where the client data distributions are 2 clusters in feature distribution (rotation degree), and 4 clusters in label distribution. From the overall perspective, there are 8 clusters in both label and feature distribution. We conduct the stochastic federated client clustering procedure with different τ on this setting. In our experiments, we observe that the clustering results vary with the value of τ. Then, we depict the representative results via t-SNE in Figure 7 and color the points according to the clustering results.

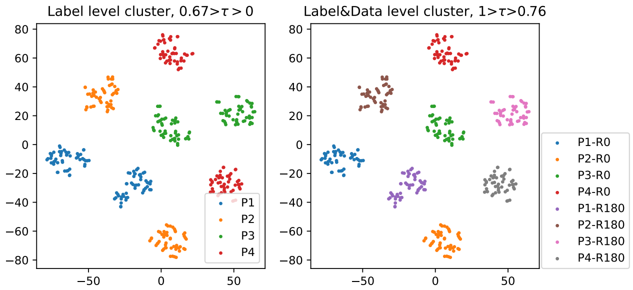


Figure 7: The value of τ decides the clustering focus.

The results indicate that the value τ determines the focus of the clustering algorithm. For instance, $\tau > 0.76$ in this case, the clustering algorithm will cluster clients only when their feature distribution and label distribution are similar. In contrast, $\tau < 0.67$ makes the clustering algorithm focus on the label distribution while ignoring the differences in feature level. Hence, a higher value of τ decides the client clustering in a more fine-grained way. More importantly, we emphasize that StoCFL is robust to the client clustering results as shown in Table 3. For a further relation between the clustering granularity and the threshold τ, we will study it in future work.