

ClusterPredict: Enhancing Federated Clustering by Combining Global and Local Parameters

MingliangNi, Chaochao Sun

School of Computer Science and Technology, Shanghai University of Electric Power, China

nmlnml@mail.shiep.edu.cn, pop-scc@163.com

Abstract— This paper proposes a novel federated learning algorithm and framework for multicentre clustering, which addresses the challenges of non-IID data distribution. The distributed machine learning technique of federated learning, which is currently being employed in an expanding array of domains, represents a novel approach to data management and analysis. The technique permits the training of models on local devices, with only the parameters of the model being updated, rather than the raw data, being transmitted to and from a server. This aggregation of updates allows for the improvement of model training efficiency while ensuring the protection of user privacy. Although federated learning provides an efficient distributed machine learning framework, traditional federated learning methods are prone to inefficiencies and model performance degradation when dealing with non-iid data. In the context of data science, non-independent and identically distributed (non-iid) data refers to a situation where the data samples are not independent from each other and their distributions are not identical. A variety of personalised federated learning methods have been developed to address the challenges posed by non-IID data. Among these, clustered federated learning algorithms represent a particularly robust strategy. However, general clustering federation learning algorithms are unable to effectively address the cases of new clients joining the training set and some clients participating in the training. Consequently, based on clustering federation learning, we also consider the above cases and propose a new multi-centre federation learning algorithm using the FedPredict mechanism, which enables the new mechanism to effectively address the cases of some clients participating in the training set as well as new clients joining the training set. The efficacy of the algorithm is validated through experimentation with disparate datasets.

Keywords— federated learning, non-iid data, distributed, personalization, clustering

I. INTRODUCTION

With the rapid proliferation of smart devices and the accelerated development of Internet of Things (IoT) technologies, the interaction between devices has generated massive amounts of data, providing abundant resources for training machine learning models. Furthermore, deep neural networks (DNNs) have achieved significant breakthroughs across various domains, including natural language processing (NLP), speech recognition, and network attack detection. These achievements are primarily attributed to the availability of big

data, algorithmic innovations, and advancements in computational hardware.

However, the development of these technologies also faces notable challenges. The exponential growth of data poses a risk to user privacy, while data silos restrict data availability and diversity, diminishing the effectiveness and accuracy of machine learning models. Although cloud computing offers virtually unlimited computational resources, the high cost of data transmission and computing resources constrains the training and deployment of large-scale machine learning models. Federated Learning (FL) has emerged as a distributed machine learning paradigm that addresses these challenges by enabling data processing and model training on local devices. Rather than sharing raw data, FL only transmits model updates, which are aggregated by a central server to coordinate the global model. This approach allows for model training on decentralized data while preserving user privacy. Despite these advantages, traditional FL methods still face significant inefficiencies and suboptimal model performance when dealing with non-independent and identically distributed (non-IID) data.

Non-IID data implies that the data distribution among participating devices is heterogeneous. For instance, variations in user behavior, geographic location, and personal interests result in distinct data patterns. Such discrepancies make it challenging to achieve satisfactory performance for all clients with a global model trained in an FL setting. Moreover, due to these data distribution disparities, FL models require more communication rounds to converge, increasing training time and communication costs.

This paper addresses these challenges by proposing a novel federated learning method. Our approach modifies the traditional multi-center federated learning mechanism. Building on the multi-center federated learning framework [3], we leverage the FedPredict algorithm [4] to improve the multi-center federated clustering algorithm, enabling it to operate independently of the participation of all clients in training and better accommodate newly added clients. This adaptation enhances model accuracy and effectively addresses the non-IID client environment.

In the subsequent sections, we will provide a detailed explanation of the mechanism underlying this multi-center federated learning algorithm. To validate the effectiveness of the proposed algorithm, we conduct experiments to analyze its

model accuracy and F1 score, and benchmark it against other algorithms on standard datasets.

II. RELATED WORK

A. Non-iid Data Research

In recent years, researchers have proposed various solutions to address the non-IID data challenge. Smith et al. [5] introduced MOCHA, a novel system-aware optimization framework based on federated multi-task learning, which improves the distributed optimization method COCOA. This work was the first to address the high communication costs, stragglers, and fault tolerance issues in federated learning environments involving multi-task learning. However, while MOCHA provides solutions for these challenges, it is not suitable for non-convex deep learning models. Chen et al. [6] proposed the FedMeta framework, which adopts a meta-learning approach. Unlike traditional FL methods (e.g., FedAvg, FedProx) that directly share global parameters, FedMeta shares a parameterized algorithm that can be adaptively adjusted based on the data on each local device.

In contrast to these approaches, which focus on feature shift non-IID problems (e.g., images collected from different hospitals using different equipment may have significant visual differences, despite similar label distributions), Li et al. [7] proposed the FedBN algorithm. This algorithm first uses a local batch normalization (BN) strategy to balance the differences caused by feature shifts between clients, and then performs model averaging, which can be seen as an improvement over the classic FedAvg algorithm. Although FedBN has significant advantages in handling feature shifts, when the data disparity between clients is too large, relying solely on local BN may not fully capture and leverage these differences. In some extreme cases, over-reliance on local BN may overlook information beneficial to model generalization.

B. Clustering Algorithms

Additionally, clustering algorithms have been widely utilized in the federated learning process. Sattler et al. [8] proposed a clustered federated learning algorithm, CFL, which is a post-processing method that performs clustering operations only after the federated learning process has converged to a stable point. The cosine similarity between weight updates of different clients indicates the similarity of their data distributions, and clients are grouped based on this cosine similarity. This strategy effectively addresses the issue of model performance degradation caused by differences in client data distributions. However, due to the post-processing nature of CFL, the entire framework is overly dependent on the participation of all clients in the training process before reaching the convergence threshold, which increases communication overhead and computational cost.

Subsequently, Long et al. [9] proposed the Federated Expectation-Maximization algorithm (FeSEM), which adopts a different strategy compared to the CFL framework. FeSEM introduces a novel multi-center aggregation mechanism that clusters clients based on their model parameters. During the

optimization process, FeSEM employs a stochastic expectation-maximization algorithm to dynamically adjust the local model parameters of each client. Clients are assigned to the nearest cluster center by calculating the distance between each node and the cluster centers. Although current FL solutions like FeSEM have made some progress in handling non-IID data, they inherit the drawbacks of the KMeans algorithm [10], such as over-reliance on the participation of all clients and inefficiency in accommodating new clients outside the training set.

III. PROPOSED METHOD

A. Problem setting

In classical federated learning, each local device i has a local data distribution D_i . The training process on the local device can be represented as $\min_{W_i} L_s(M_i, D_i, W_i)$, where $L_s(\cdot)$ is a generic loss function. The global model optimization objective can be expressed as $\min_{\{W_i\}_{i=1}^m} \sum_{i=1}^m a_i L_s(M_i, D_i, W_i)$, where m represents the number of devices and $a_i = \frac{|D_i|}{\sum_j |D_j|}$ denotes the proportion of the total sample size contributed by each client. Table 1 presents the main federated learning symbols along with all symbols relevant to this work.

TABLE 1. NOTATION TABLE

Notation	Description
t	Current round
T	Total number of rounds
lt	Last round that a user has trained so far
nt	Rounds since last training
ul	update level of the client
el	Evolution level of the FL process
lw	Local model weight
gw	Global model weight
d_{ik}	the distance between the cluster center and clients
W_i	Clients' local model parameters
$\bar{W}^{(k)}$	Global model parameters of each cluster center
K	The number of clusters
$r_i^{(k)}$	cluster assignment of each client
M	Indexes of combining layers
$W^{i,l}$	Local model parameter of client i and layer l
W_{t+1}^l	Global model parameter of layer l at round $t+1$
sc_k	Selected clients of cluster k

B. Overall framework

This section introduces the framework of the proposed clustered federated learning approach. As illustrated in Figure 1, client devices, which can be either smart terminal devices or computers, collaborate to train the model under the coordination of a central server. First, each client initializes its local model using the global model received from the server. After a fixed number of training rounds, the server clusters the model parameters across clients. We then select a subset of clients from each of the k clusters to form the corresponding list sc_k . All clients in these lists perform local training using their

own data and upload the updated parameters to the server. For each cluster, the server sends the cluster center parameters to clients requesting them, and these clients then perform client evaluation methods.

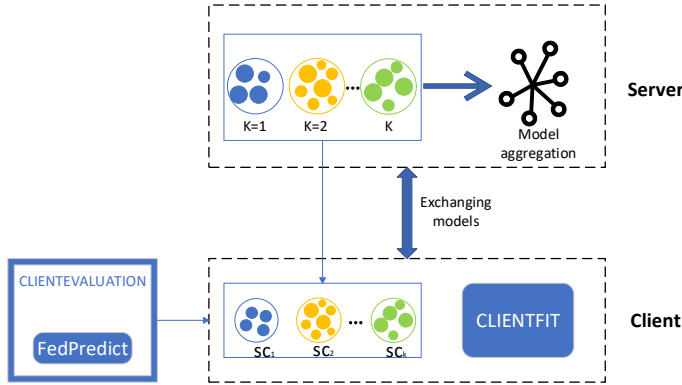


Figure 1. The overall framework of clustering federated learning proposed in this paper

C. Algorithm

Based on the aforementioned framework, we propose an algorithm called ClusterPredict, which combines global and local parameters during the prediction phase of multi-center clustered federated learning. Below, we provide a comprehensive introduction to this algorithm. Before presenting the ClusterPredict algorithm, we first provide the pseudocode for FedPredict in Algorithm 1.

Algorithm 1: FedPredict

Ensure: w_i

```

1: if client never trained then
2:    $g_w \leftarrow 1$  use only the global model parameter
3:    $l_w \leftarrow 0$ 
4: else if client trained in the current round ( $n_t = 0$ ) then
5:    $g_w \leftarrow 0$ 
6:    $l_w \leftarrow 1$  use only the local model parameter
7: else
8:   use Equations 1,2,3
9:    $ul \leftarrow \frac{1}{n_t}$ 
10:   $el \leftarrow \frac{t}{T}$ 
11:   $gw \leftarrow e^{(-ul-el)}$ 
12:   $lw \leftarrow 1 - gw$ 
13: end if
14: for each layer  $l = 1, 2, \dots, L$  do
15:   if  $l \in M$  then
16:     $w^{i,l} \leftarrow gw \times w_{t+1}^l + lw \times w^{i,l}$ 
17:   else
18:     $w^{i,l} \leftarrow w_{t+1}^l$ 
19:   end if
20: end for
21: return  $w_i$  return the combined local model parameter

```

The principle of FedPredict is to improve the prediction phase in federated learning by integrating global and local

model parameters during evaluation. In the standard federated learning process, the global model is sent to some clients for testing during the evaluation phase. However, solely relying on the global model parameters for prediction is ineffective, especially in non-IID scenarios, where each client possesses unique characteristics reflecting its local data distribution, and not all clients have necessarily participated in the training process. FedPredict improves the prediction process by combining the global parameters with the local model parameters of the clients. This allows devices to utilize both global and locally updated parameters for predictions. When a device needs to make predictions on new data, it computes a weighted combination of local and global parameters based on the specified weight ratio.

Next, we introduce the proposed ClusterPredict algorithm. The initialization step of this algorithm involves multiple rounds of pure clustering to select the optimal initial cluster centers. During this process, the number of cluster centers K , the local model parameters W_i of each device, and the global parameters $\tilde{W}^{(k)}$ for each cluster center are determined. Over T rounds, the server executes the E-step, M-step, and broadcasts global parameters to the clients sequentially until the loop concludes. In the E-step, the server calculates the distance between each device and every cluster center and updates each client's cluster assignment based on these distances. In the M-step, the cluster assignments and local model parameters of each client are fixed, and the global parameters of each cluster center are updated using the formula provided in Algorithm 2. Finally, the server iterates through each cluster center and broadcasts the global parameters to all clients within each cluster.

On the client side, each cluster is traversed, and clients are randomly selected from each cluster to form corresponding lists $SC_1, SC_2, \dots, SC_k$. The algorithm then iterates over all clients in these lists, loading the most recently saved local model parameters of the current client. The global parameters received from the corresponding cluster center are then used to train or fine-tune the model using the client's local data. After this step, the updated local model parameters are saved, and the updated parameters are sent back to the server. The time interval from the last time a client participated in training to the current round is then calculated. This reflects how many rounds have elapsed since the client last updated its model parameters, which is crucial for dynamically adjusting the fusion ratio between global and local model parameters during the model evaluation phase. Finally, the client evaluation step is executed by calling the CLIENTEVALUATION method, which utilizes the FedPredict mechanism to assess the contribution of each client to training and combines the parameters, returning them to each client for evaluation. The algorithm ends after all clusters have been traversed.

Algorithm 2: ClusterPredict**Server Side**

Initialize: $K, \{W_i\}_{i=1}^m, \{\{\tilde{W}^{(k)}\}_{k=1}^K\}$

- 1: **for** each round $t = 1, 2, \dots, T$ **do**
- 2: **E-Step:**
- 3: Calculate distance d_{ik} (Eq.5)
- 4: Update cluster assignment $r_i^{(k)}$ using d_{ik} (Eq.6)
- 5: **M-Step:**
- 6: Update $\tilde{W}^{(k)}$ using $r_i^{(k)}$ and W_i (Eq.7)
- 7: **for** each cluster $k = 1, 2, \dots, K$ **do**
- 8: broadcast $\tilde{W}^{(k)}$ to clients
- 9: **end for**
- 10: **end for**

Client Side

- 11: **for** each cluster k **do**
- 12: Randomly select clients from each cluster to form $sc_1, sc_2, \dots, sc_k$
- 13: **for** each selected client sc in $sc_1, sc_2, \dots, sc_k$ **do**
- 14: Load the lastly saved local model parameter W_i
- 15: **def** CLIENTEVALUATION($\tilde{W}^{(k)}, W_i, nt, t, T, M$):
- 16: $W_i = \text{FedPredict}(\tilde{W}^{(k)}, W_i, nt, t, T, M)$
- 17: Evaluate(W_i)
- 18: Update W_i using i, W_i, t
- 19: Save local model W_i
- 20: Return W_i to server
- 21: $nt = t - lt$
- 22: CLIENTEVALUATION($\tilde{W}^{(k)}, W_i, nt, t, T, M$)
- 23: **end for**
- 24: **end for**

IV. EXPERIMENT

To validate the effectiveness of our proposed algorithm (ClusterPredict), we conducted experiments to analyze its model accuracy and F1 score, and compared it with other algorithms on benchmark datasets. The experiments were conducted using the FEMNIST dataset. To verify the effectiveness of the proposed algorithm, we employed a convolutional neural network (CNN) architecture used in previous studies. We compared the ClusterPredict algorithm against three baselines: FedAvg, FedProx, and FedRobust.

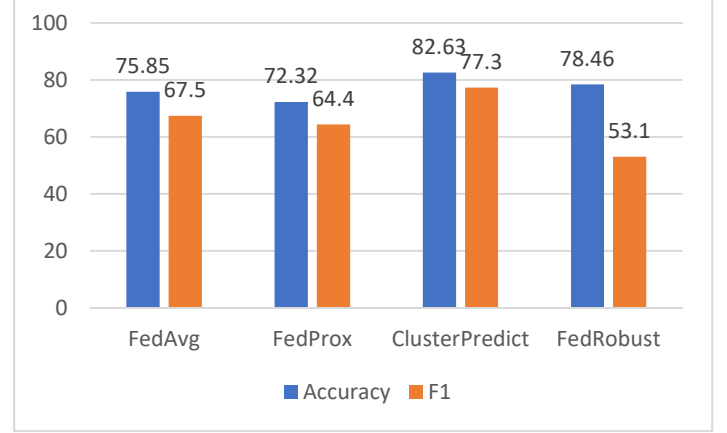
For evaluation metrics, we used classification accuracy and F1 score to assess the performance of the FL methods on individual devices. The reason for using the F1 score is that it combines both precision and recall through their harmonic mean. A high F1 score indicates a good balance between precision and recall, providing a more comprehensive and fair performance evaluation, especially when dealing with imbalanced datasets.

TABLE 2. COMPARISON OF CLUSTERPREDICT AND FESEM ON FEMNIST

Algorithm	Accuracy	F1
FeSEM	87%	70.6
ClusterPredict	91.16%	77.3

As shown in Table 2, ClusterPredict outperforms the FeSEM algorithm on the FEMNIST dataset, achieving improvements of 4.16% in accuracy and 6.7 points in F1 score. This improvement can be partially attributed to its innovative approach of combining global and local parameters during the prediction phase. The global parameters capture the overall characteristics of the dataset, while the local parameters reflect the unique data distribution of each client. This combination

enables the model to leverage general patterns from global data while also adapting to the specificities of local data, resulting in better prediction performance across diverse data distributions. Table 2 only presents a comparison of the accuracy between the two algorithms over selected rounds. Due to the use of partial client sampling in the training process, ClusterPredict converges relatively faster.

**Figure 2.** Comparison with other baseline algorithms on the FEMNIST dataset

Considering the imbalance in the number of samples across devices, we adopted a weighted average metric for device performance. In this approach, each device's weight is proportional to the amount of data it holds. The performance of ClusterPredict clearly surpasses that of the other algorithms, which can be attributed to its distinct strategy of selecting clients for clustering and integrating global and local parameters during the prediction phase. This approach enhances the prediction accuracy of clients that were not selected to participate in training, allowing newly joined clients to quickly adapt to the non-IID environment and participate effectively in training.

Among the four algorithms, FedRobust exhibits the lowest F1 score, which may be due to its anomaly detection mechanism inadvertently filtering out important information while eliminating outliers, particularly in non-IID scenarios.

V. CONCLUSIONS

This study presents a multi-center clustering algorithm, ClusterPredict, which integrates global and local model parameters to enhance the prediction step. The algorithm improves robustness in accommodating new clients by utilizing partial client aggregation and a new client evaluation mechanism, addressing the limitations of K-MEANS-based multi-center clustering algorithms, which overly rely on the participation of all clients. This allows for clustering training even with a subset of clients. The algorithm demonstrates promising performance on the FEMNIST dataset, showing that ClusterPredict maintains a high level of accuracy while performing clustering with selected clients. This is attributed to the new client evaluation mechanism from the FedPredict plugin, which combines global and local model parameters,

ensuring that the average prediction accuracy of the clients remains relatively high.

In future work, we plan to focus on dataset partitioning strategies, employing advanced methods such as stratified sampling or group partitioning to better approximate real-world data distributions. This will help us validate the performance of the algorithm when dealing with more complex datasets.

REFERENCES

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273-1282.
- [2] 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.
- [3] K. V. Sarma, S. Harmon, T. Sanford, H. R. Roth, Z. Xu, J. Tetreault, D. Xu, M. G. Flores, A. G. Raman, R. Kulkarni, et al., "Federated learning improves site performance in multicenter deep learning without data sharing," *Journal of the American Medical Informatics Association*, vol. 28, no. 6, pp. 1259-1264, 2021.
- [4] C. G. S. Capanema, A. M. de Souza, F. A. Silva, L. A. Villas, and A. A. F. Loureiro, "FedPredict: Combining Global and Local Parameters in the Prediction Step of Federated Learning," in *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, 2023, pp. 17-24.
- [5] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [6] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018.
- [7] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.
- [8] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3710-3722, 2020.
- [9] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning: Clients clustering for better personalization," *World Wide Web*, vol. 26, no. 1, pp. 481-500, 2023.
- [10] D. Steinley, "K-means clustering: A half-century synthesis," *British Journal of Mathematical and Statistical Psychology*, vol. 59, no. 1, pp. 1-34, 2006.



Mingliang Ni is a master's student currently studying for his master's degree in the School of Computer Science and Technology at Shanghai University of Electric Power. His research interests include federated learning algorithms and privacy protection. Focusing on communication efficiency improvement by federated learning.



Chaochao Sun received his Ph.D. degree from the School of Computer Science and Technology, Fudan University, in 2020, and he is currently a lecturer at the School of Computer Science and Technology, Shanghai University of Electric Power. His research interests include spectrum sensing, cognitive wireless networks, and federated learning.