FedMHC: Overcoming Dimensionality and Communication Challenges for Personalized Federated Learning Using Model Head Clustering

Haotian Zheng[1,2], Yingchi Mao[1,2], Haowen Xu[1,2], Xiaoming He[3], Benteng Zhang[1,2], Feng Mao[4], Jie Wu[5]
[1]*School of Computer and Information, Hohai University, Nanjing, China*
[2]*Key Laboratory of Big Data for Water Resources, Ministry of Water Resources, Hohai University, Nanjing, China*
[3]*College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China*
[4]*Research and Development Department Suma Technology Co., Ltd. Suzhou, China*
[5]*Center of Networked Computing, Temple University, Philadelphia, PA, U.S.A.*

***Abstract*—In real Internet of Things (IoT) environments, IoT devices vary widely in data types and needs. IoT devices participating in Personalized Federated Learning (PFL) all have their own unique data characteristics, but there exist some similarities. Previous work enhances Personalized Local Models (PLMs)' performance by clustering IoT devices' PLM while ignoring dimensionality and communication volume, resulting in lower Global Model (GM) accuracy and PLMs' performance. To this end, we propose a Personalized Federated Learning method based on Model Head Clustering (FedMHC). Specifically, FedMHC groups IoT devices with similar data characteristics and distributes different GMs to different device groups. FedMHC allows each IoT device to obtain a GM that best fits its local data characteristics and guides PLM training. FedMHC only clusters model head parameters on the server side. Thus, the edge server only needs to transmit the head parameters and a single shared feature extractor parameters during communication with IoT devices. The improvement can effectively address the issues of dimensionality and high communication volume. Experiments on CIFAR-100, Tiny-ImageNet, and AG News datasets demonstrate that FedMHC enhances the model accuracy by 1.79% and 5.9% in pathological heterogeneous scenarios, and by 1.43%, 0.92%, and 0.94% in practical heterogeneous scenarios, compared to the top-performing methods among 9 baselines.**

***Index Terms*—Personalized federated learning, data heterogeneity, model clustering**

I. Introduction

In real Internet of Things (IoT) environments, the increasing number and functional requirements of IoT devices can lead to rising data heterogeneity [1]. Additionally, the continuous increase in model dimensions raises communication pressure, thereby reducing the communication efficiency of Personalized Federated Learning (PFL). Previous work improves the performance of Personalized Local Models (PLMs) by clustering the PLM of IoT devices. However, the above methods cannot adequately address the issues of dimensionality and high communication volume in clustering, resulting in poor accuracy of PLM. *As shown in Challenge 1* of Fig.1, most existing model clustering methods directly cluster all local Model Parameters (MPs). High-dimensional MPs are impacted by the Curse of Dimensionality (CoD), diminishing intra-group compactness and inter-group separability. Unstable or inaccurate clustering outcomes may result from such issues. Furthermore, as shown in *Challenge 2* of Fig.1, unstable or inaccurate clustering results decrease the accuracy of PLM. Therefore, mitigating the impact of data heterogeneity in PFL while reducing CoD and improving communication efficiency remains a challenge. Existing studies address the mentioned issues by employing Improved model clustering strategies.

Corresponding author: yingchimao@hhu.edu.cn

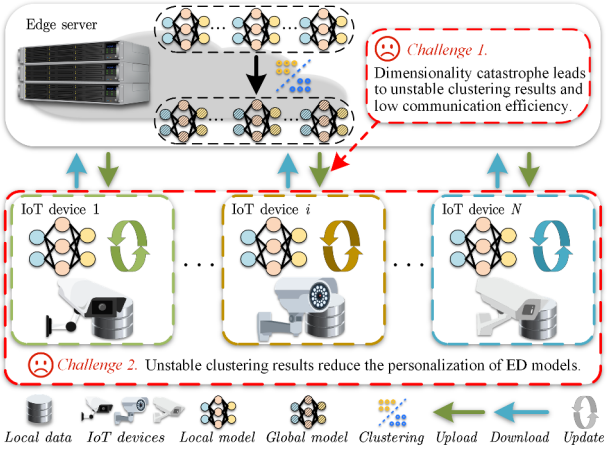

Fig. 1. Challenges of Data Heterogeneity and Dimensionality Catastrophe in Personalized Federated Learning.

In PFL, the heterogeneity of IoT devices' local data can impact model efficiency. Yurochkin *et al.* [2] propose a PFL framework that adjusts to variations in IoT device numbers and distributions using Bayesian nonparametric clustering methods. However, the CoD results in poor clustering outcomes when directly clustering high-dimensional MPs.

979-8-3315-4046-3/24/$31.00 ©2024 IEEE
DOI 10.1109/HPCC64274.2024.00075

FedSplit [3] mitigates the impact of high-dimensional MPs by grouping IoT devices for localized model training and then merging the results. Nevertheless, FedSplit may result in substantial performance disparities among IoT devices. In conclusion, developing a novel approach capable of achieving IoT devices model personalization, overcoming the CoD, and enhancing communication efficiency remains a challenge.

To this end, we propose a method called Personalized Federated Learning Based on Model Head Clustering (FedMHC). FedMHC focuses on clustering the head parameters of the model and reduces parameter dimensions during clustering, mitigates dimensionality problems, and effectively decreases communication volume. During the PLM training, a regularizer [4] is used to train the PLM for each IoT device individually, preventing overfitting and reducing the gap between the PLM and the group's GM. In summary, FedMHC can enhance the personalization of IoT device models, effectively mitigate CoD, and improve communication efficiency. The main contributions are summarized as follows.

- We propose a Personalized Federated Learning method based on Model Head Clustering (FedMHC), which clusters only the head MPs on the edge server to overcome the CoD and high communication volume issues.
- On the IoT device side, FedMHC employs a regularizer to train each PLM independently, thereby reducing the disparity between the PLM and the GM.
- Experiments on CIFAR-100, Tiny-ImageNet, and AG News datasets show that FedMHC improves test accuracy by 1.79% and 5.9% in pathological heterogeneous scenarios [5], [6], and by 1.43%, 0.92%, and 0.94% in practical heterogeneous scenarios, compared to the top-performing methods among 9 baselines.

II. Related Work

A. Model Clustering in Personalized Federated Learning

Clustering strategy. Model clustering in PFL groups similar models to improve overall model performance and efficiency. IFCA [7] and FedSEM [8] both employ the K-means clustering algorithm on MPs at the edge server to build K GMs. Grouping IoT devices reduces the influence of intra-group data heterogeneity, thereby enhancing the precision of intra-group models. However, due to the CoD, directly clustering high-dimensional MPs results in subpar clustering outcomes for both IFCA and FedSEM.

Utilization of clustering results. Following clustering, pFedCAM [9] makes GMs from all groups at the IoT device side to obtain PMs, further refining PMs using clustering results, yielding higher testing accuracy. However, this approach necessitates the edge server to distribute all GMs to each participating IoT device, leading to elevated communication costs and diminished federated training efficiency.

B. Alleviating The Curse of Dimensionality

Dimensionality reduction. To address the CoD in PFL and improve communication efficiency, FedAC [10] integrates global and local knowledge by introducing low-rank

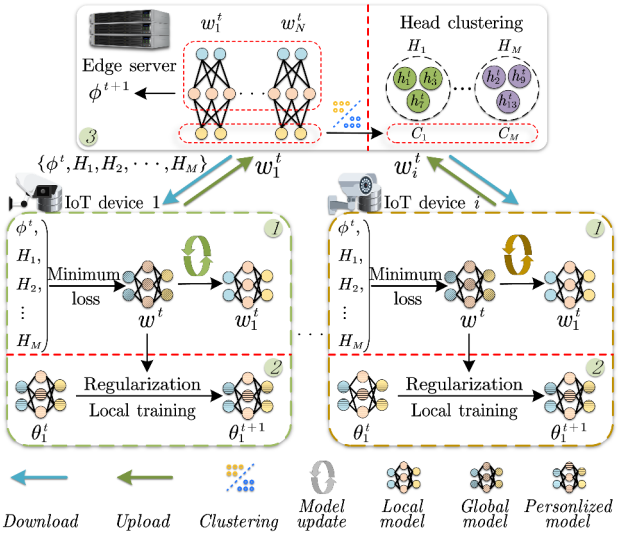

Fig. 2. Clustering of FedMHC Head Parameters and PLM Training.

cosine model similarity measurement for online clustering adjustment, facilitating real-time optimization of the clustering structure. However, handling large-scale heterogeneous datasets remains challenging.

Feature selection improvement. Conducting feature selection prior to clustering to eliminate irrelevant or redundant features can help mitigate the CoD. FedSDG-FS [11] reduces the feature count processed in vertical PFL models via efficient feature selection methods, thereby ameliorating the complexity and dimensionality issues arising from high-dimensional data. However, this method may necessitate intricate encryption operations and entail additional communication overhead.

In brief, PFL methods with model clustering boost accuracy but still face challenges such as dimensionality issues and high communication overhead due to data heterogeneity. Overcoming these hurdles is our primary goal.

III. System Model

A. Model Overview

As shown in Fig.2, the system model consists of N IoT devices and an edge server. The system clusters only the head MPs, reducing parameter dimensionality during clustering to mitigate the CoD and decrease communication overhead. During PLM training, regularizers individually train PLMs for each IoT device, narrowing the gap between PLMs and the GM of their respective groups. Specifically, FedMHC employs the K-means algorithm on the server side to cluster head MPs, thereby constructing multiple GMs to guide the training of PLMs on IoT devices. In a heterogeneous data environment, the K-means clustering algorithm groups IoT devices into M distinct groups, with similar data distributions

within each group. The goal is to minimize intra-group distances and optimize IoT device grouping, as defined by

$$\min \quad \frac{1}{K}\sum_{m=1}^{M}\sum_{i=1}^{K} r_i^m \text{Dist}(h_i, H_m), \tag{1}$$

$$r_i^m = \begin{cases} 1, & \text{if } h_i \in C_m \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where K denotes the number of IoT devices participating in training at a particular iteration, M represents the number of groups after clustering, $H_m(m \in [0, M])$ denotes the aggregated head MPs of the m group, and $Dist(\cdot, \cdot)$ represents the distance function, h_i represents the head parameters of the local model uploaded by IoT device i to the edge server,r_i^m indicates whether IoT device i belongs to the m group after clustering and C_m represents the set of all head MPs of the m group after clustering.

Drawing inspiration from pFedMe [12], a regularization term is incorporated into the PLM training process updates to restrict parameter updates. Consequently, the PFL's objective for each IoT device is to minimize the disparity between the PFL and GM, which is defined as

$$\min \quad f_i(\theta_i) + \lambda r(\theta_i, w_i), \tag{3}$$

where θ_i represents the PLM parameters on IoT device i, $f_i(\cdot)$ is the loss function, λ is a regularization hyperparameter, and $r(\cdot, \cdot)$ represents the regularizer. w_i forms the GM, including the aggregated head MPs of the group on IoT device i and the Shared Feature Extractor Parameters (SFEPs).

B. Problem Statement

During the training process, local training based on private datasets from IoT devices encourages the PLM to fit its private data distribution as closely as possible. At the same time, a regularizer restricts the updates to the PLM to ensure it does not deviate excessively from the GM, thereby transferring the GMs' knowledge to the PLM. Therefore, The overarching objective of FedMHC can be framed as a two-layer optimization problem, optimized through alternating updates on the IoT device and the edge server. Specifically, update w_i and θ_i using local data on each IoT device. On the edge server, cluster and update the head MPs H_m

$$\begin{aligned} \min \quad & \frac{1}{K}\sum_{i=1}^{K}\{F_i(\theta_i) := f_i(\theta_i) + \lambda r(\theta_i, w_i)\} \\ \text{s.t.} \quad & \min \frac{1}{K}\sum_{m=1}^{M}\sum_{i=1}^{K} r_i^m \text{Dist}(h_i, H_m). \end{aligned} \tag{4}$$

IV. Head Parameters Clustering

A. Head Parameters Selection and Blending Mechanism

Each IoT device's model consists of SFEPs and head MPs. The SFEPs transform raw data from a high-dimensional space to a lower-dimensional space, extracting key features. Each IoT device calculates local updates (e.g., gradients) based on the global representation and sends them back to the edge server. The server aggregates these updates to refine the global representation. Each IoT device computes a personalized low-dimensional classifier, referred to as the IoT device's head parameters, to adapt to the unique labels of its local data. IoT devices use the global feature representation as input and train their head model by minimizing the local loss function. This process is typically completed locally on the IoT device side.

The objective is to perform E iterations, the IoT devices receive global SFEPs ϕ^t and a set of head MPs $H = \{ H_1, H_2, ..., H_M \}$ from the edge server during the local model update stage. We propose a loss-based method for group identification, assigning IoT devices to the m-th group based on each model's average loss. m is calculated by

$$m = \ arg \min_{j \in [0,M]} f_i(\phi^t, H_j), \tag{5}$$

where m represents that IoT device belongs to the m-th group, i is the IoT device's identifier, and $f_i(\cdot)$ is the loss function. Global MPs w^t are constructed by combining the head MPs H_m from the relevant group on the IoT device with the global SFEPs ϕ^t, represented by

$$w^t = \{\phi^t, H_m\}, \tag{6}$$

using the global MPs w^t to replace the local MPs w_i^t, serves as the initial parameters for the local model in round t of federated communication. The IoT device then updates the local model using its local data, expressed as

$$w_i^t = w^t, \tag{7}$$

$$w_i^t = w_i^t - \eta \nabla f_i(w_i^t), \tag{8}$$

where $\nabla f_i(\cdot)$ is the stochastic gradient of IoT device i in communication round t and η is the local learning rate. Finally, the IoT device uploads w_i^t to the edge server.

B. Regularization Adjustment

Inspired by pFedMe [12], a novel loss function is employed to update the PLM. This function comprises a supervised learning loss and a regularizer that constrains the model updates, ensuring they do not deviate excessively from the GM. Consequently, the goal of personalized learning in each IoT device is to minimize the gap between the PLM and the GM. During the t-th round of federated communication, the loss function for PLM training is given by

$$F_i(\theta_i^t) = f_i(\theta_i^t) + \frac{\lambda}{2}\|\theta_i^t - w^t\|, \tag{9}$$

where θ_i^t is the PLM parameters of IoT device i. Before training, the parameters from the previous round are used as the initial parameters for the current round, calculated as

$$\theta_i^t = \theta_i^{t'}, \tag{10}$$

where t' indicates that the last time client i participated in federated training was during the t'-th round of federated

Algorithm 1: FedMHC

Input: w_i^t, θ_i^t, η, H_m, ϕ^t, T, E, N, K, M, t.
Output: $\{\theta_1^{T+1}, \theta_2^{T+1}, ...\theta_N^{T+1}\}$.

for $t = 1, 2, \ldots, T$ **do**
  IoT device:
    Define $S \subseteq \{1, 2, \ldots, N\}$
    foreach *IoT device* $i \in S$ *in parallel* **do**
      Download model header parameter set $\{H_1, H_2, ...H_M\}$
      Download feature extraction parameters ϕ^t
      $m = \ arg\min_{j \in [1,M]} f_i(\phi^t, H_j)$
      $w_i^t = \{\phi^t, H_m\}$
      for $e = 1, 2, \ldots, E$ **do**
        $w_i^t = w_i^t - \eta \nabla f_i(w_i^t)$
        $F_i(\theta_i^t) = f_i(\theta_i^t) + \frac{\lambda}{2}\|\theta_i^t - w_i^t\|$
        $\theta_i^t = \theta_i^t - \eta \nabla F_i(\theta_i^t)$
      end
      Upload w_k^t **to Server**
    end
  Server:
    $w_i^t = \{\phi_i^t, h_i^t\}$
    Cluster $\{h_1^t, h_2^t, ...h_K^t\}$ into $C_1, C_2, ...C_M$
    update $r_i^m = \begin{cases} 1, & \text{if } h_i^t \in C_m \\ 0, & \text{otherwise.} \end{cases}$
    for $m = 1, 2, \ldots, M$ **do**
      $H_m = \frac{1}{|C_m|} \sum_{i=1}^{K} r_i^m h_i^t$
      $\phi^{t+1} = \frac{1}{K} \sum_{i=1}^{K} \phi_i^t$
    end
  Share $\{\phi^{t+1}, H_1, H_2, \cdots H_M\}$ **with IoT devices**
end
return $\{\theta_1^{T+1}, \theta_2^{T+1}, ...\theta_N^{T+1}\}$

communication. The IoT device then performs multiple local iterations to update the PLM based on local data

$$\theta_i^t = \theta_i^t - \eta \nabla F_i(\theta_i^t), \tag{11}$$

where $F_i(\cdot)$ is a loss function incorporating an $L2$ norm regularizer. PLM can use the regularizer to control the distance between the PLM and its corresponding GM.

C. Header Parameters Clustering Mechanism

The edge server receives the local MPs w_i^t from the IoT devices. Presently, the prevailing approach involves the direct aggregation of these MPs across the network. Inspired by FedRep [13], local MPs w_i^t are partitioned into two parts

$$w_i^t = \{\phi_i^t, h_i^t\}, \tag{12}$$

where ϕ_i^t is the SFEPs and h_i^t is the head MPs. Using the K-means algorithm to cluster the local model's header parameters. Then, update r_i^m, and perform federated averaging on the header parameters within each group, we can get

$$H_m = \frac{1}{|C_m|} \sum_{i=1}^{K} r_i^m h_i^t, \tag{13}$$

where, $|C_m|$ denotes the count of model header parameters in the m-th group post-clustering. To minimize communication overhead, direct federated aggregation is conducted solely on the feature extraction segment of all local models. When distributing MPs, only a single feature extraction parameter and the header parameters of M models are transmitted. The feature extraction segment aggregation is computed as

$$\phi^{t+1} = \frac{1}{K} \sum_{i=1}^{K} \phi_i^t. \tag{14}$$

Then, the edge server chooses K IoT devices for the subsequent round of federated training. Simultaneously, the edge server dispatches the set of model header parameters $\{ H_1, H_2, ..., H_M \}$ and the parameters of the feature extraction component ϕ^{t+1} to the selected IoT devices.

D. Algorithm Workflow

Fig.2 illustrates the training process of FedMHC during the t-th communication round. In each communication round, FedMHC undergoes three key training stages. The workflow of FedMHC is outlined in **Algorithm 1**. The computational complexity per round is dominated by $O(N{\cdot}E{\cdot}d)$ for the IoT devices and $O(M{\cdot}K + K{\cdot}d{\cdot}M)$ for the edge server.

1) Update local model. Each IoT device receives the SFEPs ϕ^t and the set of head MPs $H = \{ H_1, H_2, ..., H_M \}$ from the edge server, then uses these parameters to instantiate multiple models and select the one with the minimum loss on local data as the GM w^t. The local model w_i^t is initialized with w^t and updated using the local data. After training, IoT devices upload the local MPs w_i^t to the edge server.

2) Update personalized parameters in local models. Each IoT device contains local MPs w_i^t and personalized MPs θ_i^t. After determining the GM parameters w^t, update the personalized MPs θ_i^t using a regularizer. This mechanism effectively diminishes the discrepancy between the PLM θ_i^t and the GM w^t, thus facilitating the transfer of knowledge from the GM to the PLM, helping prevent local overfitting.

3) Cluster local model head parameters. The edge server receives local MPs w_i^t from IoT devices, segments them into SFEPs ϕ_i^t and head MPs h_i^t, and clusters the latter using the K-means. After determining the group assignment r_i^m for each IoT device, federated averaging is applied to all head MPs within the same group, resulting in a new collection of head MPs H. The edge server executes federated averaging on the SFEPs of all local models to derive ϕ^{t+1}.

V. Experimental Evaluation

A. Experimental Setup

Experimental Environment.

• *Hardware Environment.* We employ a simulated IoT environment to experimentally verify the accuracy of PLMs under heterogeneous data conditions. The federated learning system consists of an edge server with an Intel Xeon Platinum 8369B CPU, operating at 2.4 GHz with 8 cores and 16GB of RAM, and twenty IoT devices. These IoT devices are

represented by Docker nodes across various workstations, each powered by an Intel i5-12500H CPU at 2.5 GHz with 4 cores and 4GB of RAM. While the dataset characteristics of each Docker node vary, some similarities persist.

• *Models.* For CIFAR-100, a 4-layer CNN with two convolutional and two fully connected layers is used. A similar 4-layer CNN and a ResNet-18 model are applied to Tiny-ImageNet. For text classification with the AG News, the FastText model with an input layer, a fully connected hidden layer, and a fully connected output layer are utilized.

• *Hyperparameter Settings.* In this experiment, the clustering method sets the number of cluster centers M to 4. For all three datasets, set $batchsize = 10$, and IoT device participation rate $\rho = 1$. For real-world heterogeneous data: set Dirichlet distribution parameters $\beta = 0.1$. For CIFAR-100: $T = 500$, $\eta = 0.005$. For Tiny-ImageNet: $T = 500$, $\eta = 0.005$. For AG News: $T = 2000$, $\eta = 0.1$.

Datasets. The experiment utilized the CIFAR-100 [14], Tiny-ImageNet [15] and the AG News datasets [16], simulating both actual and pathological heterogeneous scenarios.

Baselines. We implemented baselines, including FedAvg [17], FedProx [18], Per-FedAvg [19], CPFL [20], IFCA [7], FedSEM [8], pFedCAM [9], pFedMe [12] and FedRep [13].

Metrics. FedMHC is evaluated using these metrics.

- *Personalized Model Average Accuracy (PMAA).* Average test accuracy of all IoT device-side PLMs, indicating overall performance.
- *Average model parameter distance.* Compute the Euclidean distance between all model parameters within each group and the centroids of the clusters. Sum these distances and then divide by the number of IoT devices.
- *Clustering Iteration Rounds.* K-means iterations for clustering head MPs per communication round.

B. Experimental Results Analysis

Verage Accuracy. Table I displays the PMAA for FedMHC and baseline methods across CIFAR-100, Tiny-ImageNet, and AG News datasets. TINY signifies the use of a 4-layer CNN model on Tiny-ImageNet, while TINY* indicates a ResNet-18 model on Tiny-ImageNet.

In both practical and pathological heterogeneous data scenarios, FedMHC outperforms other benchmark methods. However, the performance gain of FedMHC is less pronounced in practical heterogeneous data setups compared to pathological ones. This is primarily due to the long-tail data distribution among IoT devices in practical scenarios, exacerbating data distribution differences within clusters. On the AG News dataset, FedMHC maintains the highest test accuracy, indicating its effectiveness across different tasks.

Number of Clusters. Experiments on CIFAR-100 and Tiny-ImageNet datasets are conducted to assess FedMHC's performance and its resilience to varying group numbers M. Specifically, when $M = 1$, there's a single unclustered group, making FedMHC behave like pFedMe. Conversely, when $M = N$, each group accommodates only one IoT device, relying solely on local data for training. To prevent overfitting, M is capped at $N/2$. Experiments on FedMHC with M ranging from 1 to $N/2$ are conducted. Comparing test accuracy across different M values helps identify the optimal balance between personalization and generalization in FedMHC. As shown in Fig.3(a,b), FedMHC outperforms baseline methods in test accuracy on both datasets. All methods perform best with $M = 4$, the default parameter used in the experiments.

TABLE I
AVERAGE ACCURACY UNDER PRACTICAL HETEROGENEOUS SCENARIOS AND PATHOLOGICAL HETEROGENEOUS SCENARIOS PMAA

Methods	Practical %				Pathological %	
	CIFAR-100	TINY	TINY*	AG News	CIFAR-100	TINY
FedAvg	31.89	19.46	19.45	79.57	25.98	14.20
FedProx	31.99	19.37	19.27	79.35	25.94	13.85
Per-Fedavg	44.28	25.07	21.81	93.27	56.80	28.06
FedRep	52.39	37.27	39.95	96.28	67.56	40.85
pFedMe	47.34	26.93	33.44	91.41	58.20	27.71
CPFL	40.25	24.53	24.37	93.15	45.90	21.55
FedSEM (M=4)	41.07	25.03	23.58	94.58	56.03	25.01
IFCA (M=4)	43.16	32.73	34.83	94.77	56.42	40.77
pFedCAM (M=4)	46.17	35.66	34.57	95.16	64.12	41.22
FedMHC (M=4)	**53.82**	**38.19**	**40.42**	**97.22**	**69.35**	**47.52**

Cluster Results. This section evaluates clustering efficacy by assessing the average distance between MPs. Smaller distances indicate tighter cohesion within a group, implying superior clustering outcomes. We conduct experiments on the CIFAR-100 dataset, comparing the results of FedMHC with CPFL, FedSEM, IFCA, and pFedCAM over 200 communication rounds. Fig.3(c) shows the fluctuation in the average distance between MPs of FedMHC and other clustering methods across training rounds. As training progresses, the average distance of MPs of FedMHC gradually decreases, indicating strong intra-group cohesion and effective clustering. Other methods using entire MPs for clustering tend to have higher average distances than FedMHC.

Additionally, the K-means algorithm has a certain computational overhead. We observe the iteration rounds used by the K-means algorithm for clustering in FedMHC and FedSEM during each communication round. Fewer iterations indicate reduced computational resources consumed. Specifically, experiments are conducted on the CIFAR-100. Fig.3(d) shows that in the initial training stages, FedSEM and clustering all MPs yield unstable results and also require more iterations. Conversely, FedMHC and clustering-only head MPs converge faster with fewer iterations.

Communication Cost. We evaluate FedMHC's communication efficiency by comparing the communication volume per IoT device for various methods on CIFAR-100 and Tiny-ImageNet datasets. Table II shows that FedAvg has the lowest communication cost, and FedMHC's communication cost is not significantly higher than FedAvg's. However, FedMHC's test accuracy is much higher than FedAvg's, being 2.23 times that of FedAvg. Furthermore, FedMHC's communication cost is much lower than that of FedSEM, IFCA, and pFedCAM, which is around 40% of their costs. FedMHC incurs minimal communication costs while achieving superior test accuracy compared to other clustering methods.

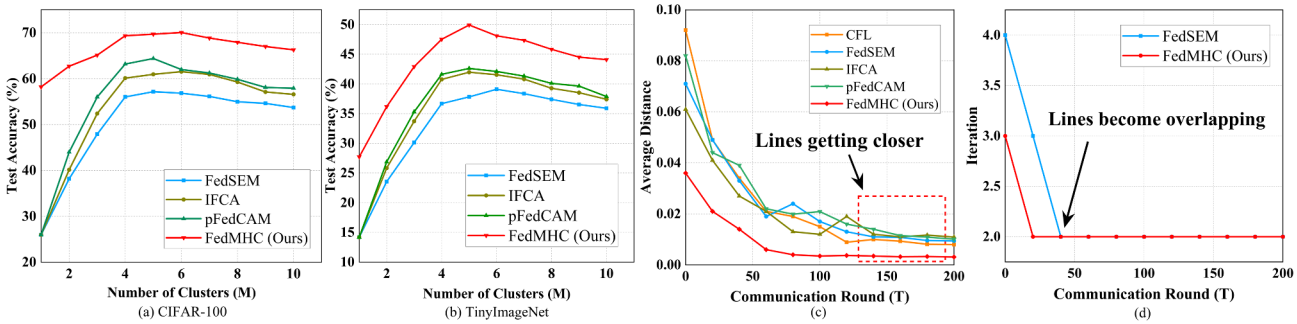

Fig. 3. The test accuracy of FedMHC and other clustering methods at different numbers of clusters on CIFAR-100 and Tiny-ImageNet (a, b), the variation in average model parameter distance (c), and the change in iteration rounds using K-means clustering (d).

TABLE II
THE COMMUNICATION VOLUME PER SINGLE IOT DEVICE IN A SINGLE ROUND (MB)

Methods	CIFAR-100		Tiny-ImageNet	
	CNN	ResNet-18	CNN	ResNet-18
FedAvg	7.06	85.66	43.44	86.06
FedSEM (M=4)	17.65	214.15	108.6	215.15
IFCA (M=4)	17.65	214.15	108.6	215.15
pFedCAM (M=4)	17.65	214.15	108.6	215.15
FedMHC (M=4)	7.66	86.26	44.61	87.23

VI. CONCLUSIONS

In pathological heterogeneous scenarios, model clustering in PFL improves accuracy but still faces dimensionality and communication overhead issues. To this end, we propose FedMHC, which clusters only the head MPs on the edge server. During PLM training, a regularizer prevents overfitting, aligning PLMs with their respective group's GMs. The edge server transmits only head MPs and SFEPs, thereby reducing communication volume. Results on CIFAR-100, Tiny-ImageNet, and AG News datasets show FedMHC improves test accuracy by 1.79% and 5.9% in pathological heterogeneous scenarios, and by 1.43%, 0.92%, and 0.94% in practical heterogeneous scenarios, compared to best baselines. In summary, FedMHC effectively mitigates CoD and reduces communication pressure.

ACKNOWLEDGEMENTS

This work was partially supported by the Key Research and Development Program of China (Grant 2022YFC3005401), the Yunnan Province Key Research and Development Program (Grant 202203AA080009), and the Suzhou Innovation Consortium Project - Suzhou Advanced Computing Core Equipment and Key Technology Innovation Consortium.

REFERENCES

[1] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous Federated Learning: State-of-the-art and Research Challenges," *arXiv*, 2023. DOI: 10.48550/arXiv.2307.10616.

[2] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, T. N. Hoang, and Y. Khazaeni, "Bayesian Nonparametric Federated Learning of Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 7252–7261.

[3] R. Pathak and M. J. Wainwright, "FedSplit: An Algorithmic Framework for Fast Federated Optimization," *arXiv preprint arXiv:2005.05238*, 2020.

[4] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proceedings of the International Conference on Machine Learning*, 2021, pp. 6357–6368.

[5] A. Rauniyar, D. H. Hagos, D. Jha, J. E. Hakegard, U. Bagci, D. B. Rawat, and V. Vlassov, "Federated learning for medical applications: A taxonomy, current trends, challenges, and future research directions," *IEEE Internet of Things Journal*, 2023.

[6] W. Oh and G. N. Nadkarni, "Federated learning in health care using structured medical data," *Advances in Kidney Disease And Health*, vol. 30, no. 1, pp. 4–16, 2023.

[7] A. Ghosh, J. Chung, D. Yin, and K. Ramachandran, "An Efficient Framework for Clustered Federated Learning," in *Advances in Neural Information Processing Systems*, 2020, pp. 19586–19597.

[8] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-Center Federated Learning: Clients Clustering for Better Personalization," in *World Wide Web*, 2023, pp. 481–500.

[9] Z. Yang, Y. Liu, S. Zhang, and K. Zhou, "Personalized Federated Learning with Model Interpolation Among Client Clusters and Its Application in Smart Home," in *World Wide Web*, 2023, pp. 1–26.

[10] Y. Zhang, H. Chen, Z. Lin, Z. Chen, and J. Zhao, "FedAC: An Adaptive Clustered Federated Learning Framework for Heterogeneous Data," *arXiv preprint arXiv:2403.16460*, 2024.

[11] A. Li, H. Peng, L. Zhang, J. Huang, Q. Guo, H. Yu, and Y. Liu, "FedSDG-FS: Efficient and Secure Feature Selection for Vertical Federated Learning," *arXiv preprint arXiv:2302.10417*, 2023.

[12] T. Dinh, N. Tran, and J. Nguyen, "Personalized Federated Learning with Moreau Envelopes," in *Advances in Neural Information Processing Systems*, 2020, pp. 21394–21405.

[13] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting Shared Representations for Personalized Federated Learning," in *International Conference on Machine Learning*, 2021, pp. 2089–2099.

[14] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," 2009.

[15] P. Chrabaszcz, I. Loshchilov, and F. Hutter, "A Downsampled Variant of ImageNet as an Alternative to the CIFAR Datasets," *arXiv preprint arXiv:1707.08819*, 2017.

[16] X. Zhang, J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification," in *Advances in Neural Information Processing Systems*, 2015.

[17] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Artificial Intelligence and Statistics*, 2017.

[18] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," in *Proceedings of Machine Learning and Systems*, 2020, pp. 429–450.

[19] A. Nichol, J. Achiam, and J. Schulman, "On First-Order Meta-Learning Algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[20] 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*, 2020, pp. 3710–3722.