PFL-LDG: Privacy-preserving Federated Learning via Lightweight Device Grouping

1st Zhilu Wang
School of Computer Science
Nanjing University of Information Science and Technology
Nanjing, China
202212200002@nuist.edu.cn

2nd Qi Liu
School of Software
Nanjing University of Information Science and Technology
Nanjing, China
qi.liu@nuist.edu.cn

3rd Xiaodong Liu
School of Computing
Edinburgh Napier University
Edinburgh, UK
x.liu@napier.ac.uk

***Abstract*—The rapid growth of private data from distributed edge networks, driven by the proliferation of IoT sensors, wearable devices, and smartphones, offers significant opportunities for AI applications. However, traditional distributed machine learning methods struggle to address data privacy concerns effectively. Federated learning (FL) has appeared as a popular, innovative paradigm for distributed machine learning that enables collaborative training of models across multiple data silos while preserving privacy. Yet, in large-scale and complex edge networks, the convergence performance of existing FL methods deteriorates when dealing with highly heterogeneous data. This paper introduces PFL-LDG, a similarity-based lightweight privacy-protected grouping FL method that mitigates the impact of non-IID data on FL model performance in data-heterogeneous scenarios. Unlike conventional FL, PFL-LDG clusters devices based on data distribution similarity, reducing inefficiency and straggler issues while supplying personalized FL models for edge devices and enhancing FL accuracy. The paper's main contribution is the proposal of a novel similarity-based lightweight privacy-protected grouping FL framework, focusing on improving privacy protection and training efficiency in heterogeneous edge resource-constrained FL systems.**

***Index Terms*—Federated Learning, Non-IID, Personalization, Lightweight Encryption, Similarity-based Clustering, Privacy**

I. Introduction

With the advent of 5G technology, the extensive adoption of edge devices, such as IoT sensors, wearable devices, and smartphones, has led to a rapid surge in private data originating from distributed data sources at the edge network. This vast amount of data offers immense opportunities for AI applications, prompting various organizations to harness big data for optimizing AI processes and performance. However, most data is inherently sensitive and exists as data silos. Traditional distributed machine learning methods [1] have not effectively tackled data privacy concerns, and with the recent introduction of data privacy protection laws, such as the General Data Protection Regulation (GDPR) [2], the demand for privacy-preserving AI solutions has surged.

Considering the data privacy challenges in distributed machine learning, federated learning (FL) [3] has gained popularity in recent years. FL is an innovative distributed machine learning paradigm that enables collaborative training of machine learning models across multiple data silos while preserving privacy. Current FL settings assume that data holders (i.e., clients) form a federation, with edge devices acting as clients, and collaboratively train FL models under the coordination of a central parameter server (i.e., FL server). Training data is stored locally on clients, and raw data is not directly shared during the process. Most existing FL training methods are derived from Google's federated averaging (FedAvg) [3] algorithm.

However, in large-scale and complex edge networks, edge devices demonstrate data heterogeneity [4]. The convergence performance of the FedAvg method deteriorates when dealing with highly heterogeneous data, which is evidenced by the substantial reduction in the accuracy of the FL model when learning non-independent and identically distributed (non-IID) data [5, 6]. This reduction in FL model performance is attributed to client drift [7, 8], a consequence of multiple rounds of local training and global aggregation for FL clients holding non-IID data. In situations of data heterogeneity, common FL methods cannot obtain personalized FL models, since all clients share a single global model for FL training, which is unable to adapt to diverse local data distributions.

To enhance the accuracy of FL in data-heterogeneous scenarios, researchers have introduced various similarity-based personalized federated learning (PFL) approaches [9], including multi-task learning (MTL) [10-12], model interpolation [13-15], and clustering [16-17], aiming to mitigate the weight divergence issue. However, most methods addressing non-IID FL present considerable drawbacks, as they lead to significant computation and communication overhead. Furthermore, certain techniques expose clients' raw data distribution features to the server for comparison, potentially resulting in user privacy breaches.

Presently, mainstream strategies for enhancing data privacy and security in FL involve integrating differential privacy (DP) [20, 21], secure multi-party computation (MPC) [22, 23], and homomorphic encryption (HE) [24] technologies. Research [25] indicates that combining FL with these privacy-preserving techniques can provide robust security. Nonetheless, the use of DP can reduce data utility, while HE and MPC incur high computation and communication costs [26]. These challenges, related to computational overhead and privacy protection, hinder the extensive deployment of FL models in data-heterogeneous edge computing scenarios. As a result, developing fast and accurate personalized FL methods without exposing user privacy information has become a significant bottleneck in the design of heterogeneous IoT edge intelligent systems.

To tackle these challenges, this paper introduces a similarity-based lightweight privacy-protected grouping FL method named PFL-LDG, which can mitigate the impact of non-IID data in data-heterogeneous scenarios on FL model performance. In contrast to conventional FL, which randomly selects edge devices for model gradient aggregation, the proposed non-IID FL method clusters devices based on the similarity of their data distribution features. Given that mobile devices have limited network connectivity, it is impractical for FL to perform model updates and aggregation in parallel on all participating devices. Moreover, the accuracy of FL training relies on capturing all unique data distributions rather than all clients. In data-heterogeneous cases, selecting one client from each group for training does not affect the overall group's accuracy. Hence, after device grouping, each FL training round can choose a small subset of devices from a group for model gradient aggregation. This approach reduces inefficiency and straggler issues caused by improper random scheduling, while offering personalized FL models for edge devices and enhancing the accuracy of FL in data-heterogeneous scenarios.

This paper systematically focuses on improving privacy protection and training efficiency in heterogeneous edge resource-constrained FL systems. The main contribution of this paper lies in proposing a novel similarity-based lightweight privacy-protected grouping FL framework.

The remainder of this paper is organized as follows. Section II introduces related work on similarity-based PFL. Section III provides a detailed description of the group-based FL method (i.e., PFL-LDG). Section IV presents the experimental results, and Section V concludes the paper. By addressing the challenges of data heterogeneity and privacy concerns in FL, this paper paves the way for more efficient and secure AI applications in edge computing environments.

II. Related Work

Recent works have concentrated on PFL, aiming to tackle performance issues caused by client drift in non-IID data learning within FL. In contrast to strategies that involve training a single global FL model, these approaches construct personalized models by adjusting the aggregation process of the FL model. Similarity-based methods strive for personalization by modeling the relationships among clients in FL. Numerous researchers have extensively explored similarity-based PFL methods, with multi-task learning and model interpolation considering pairwise client relationships, while clustering focuses on group-level client relationships.

Multi-Task Learning (MTL): MTL's objective is to train a model capable of concurrently performing multiple related tasks. This enhances generalization by exploiting domain-specific knowledge across tasks. By considering each FL client as a task in MTL, the server can learn and discern the relationships among clients based on their heterogeneous local data. MOCHA [10] expands distributed MTL to the FL context, learning personalized models for each client; however, all clients must participate in each round of FL model training. MOCHA is only applicable to convex models and has limited generalization in deep learning. Consequently, [11] proposed the VIRTUAL federated MTL algorithm using Bayesian methods for variational inference, which can handle non-convex models. However, this results in higher computational costs for large-scale IoT edge FL. FedAMP [12] is an attention mechanism-based method that fosters pairwise collaboration among FL clients with similar data distributions, maintaining personalized cloud models for each client on the server. In large-scale edge FL, this leads to substantial computational and communication overhead for the server.

Model Interpolation: [13] proposed a novel paradigm for learning PFL models by combining global and local models, with the server balancing the degree of generalization and personalization for both client local models and global models. APFL [14] introduces extra parameters for each client, adaptively adjusting the weights of local and global models during FL training, allowing each client to reach an optimal level of personalization. HeteroFL [15] adaptively builds personalized local models for each client using a single global model, resolving performance issues stemming from data heterogeneity. Model interpolation methods rely on a single global model as the foundation for personalization, necessitating the optimal combination of local and global models or the best interpolation, which generates additional computation and communication between clients and servers. This may heighten computational and communication overhead, particularly in resource-constrained large-scale FL. Furthermore, sharing local information for model interpolation may subject clients to privacy risks.

Clustering: In scenarios with distinct client or data distributions, using a client-server architecture to train a federated global model is not the optimal choice. Instead, it is more suitable for PFL to first group clients, then train an FL model for each client group, ultimately obtaining multiple models. In [16], the server employs an optimal bipartition algorithm based on the cosine similarity of client gradient updates to divide FL clients into different clusters. Since multiple communication rounds are needed to separate all inconsistent clients, the proposed method incurs higher computational and communication costs, limiting its practical feasibility in resource-constrained scenarios. The framework designed by [17] initially trains a global FL model for a certain

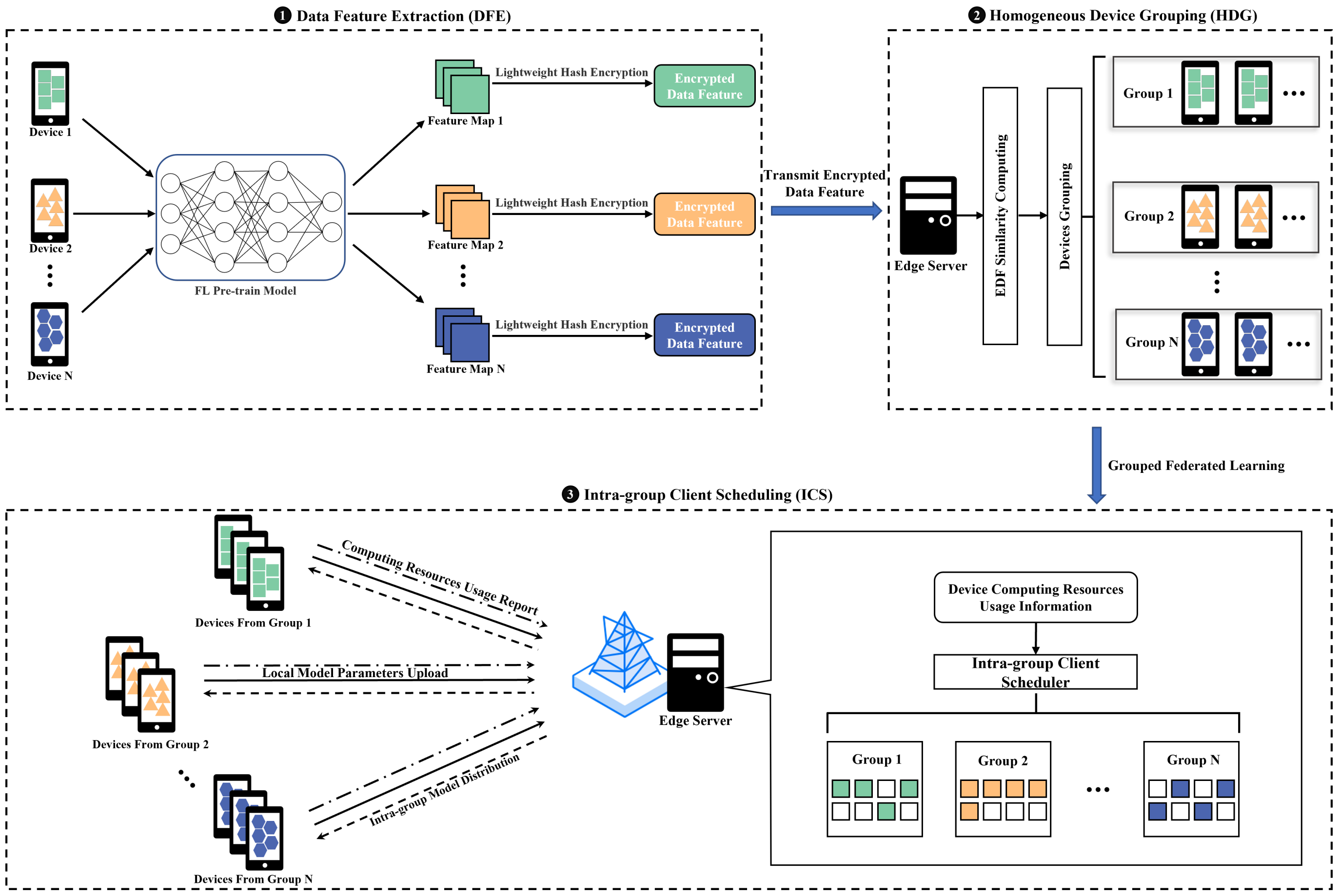
❶ Data Feature Extraction (DFE)
Device 1
Device 2
Device N
FL Pre-train Model
Feature Map 1
Feature Map 2
Feature Map N
Lightweight Hash Encryption
Encrypted Data Feature
Transmit Encrypted Data Feature
❷ Homogeneous Device Grouping (HDG)
Edge Server
EDF Similarity Computing
Devices Grouping
Group 1
Group 2
Group N
Grouped Federated Learning
❸ Intra-group Client Scheduling (ICS)
Devices From Group 1
Devices From Group 2
Devices From Group N
Computing Resources Usage Report
Local Model Parameters Upload
Intra-group Model Distribution
Edge Server
Device Computing Resources Usage Information
Intra-group Client Scheduler

Fig. 1. An overview of PFL-LDG.

number of rounds and performs one-time clustering using all clients' gradient updates. Afterward, it conducts FL training for each cluster separately, ultimately generating multiple FL models. This method is suitable for non-IID settings with data heterogeneity, but requires extensive computation when calculating pairwise distances between all clients needed for clustering. PFA [18] enables clients with similar data distributions to collaborate in a grouping manner, ultimately obtaining personalized FL models. However, PFA does not provide formal privacy protection guarantees when extracting data distribution features. Moreover, when using the FedAvg algorithm to aggregate clients within a group during group training, it does not account for resource-constrained edge scenarios, potentially resulting in dropouts that impact the efficiency of PFL. HACCS [19] is a heterogeneity-aware clustered client selection system that extracts device data distribution histograms and employs differential privacy techniques to safeguard user privacy before submitting them to the server for clustering. However, applying differential privacy directly to data distribution features diminishes data utility, leading to suboptimal clustering results and ultimately decreasing the accuracy of the PFL models obtained from clustered FL.

III. Methodology

A. An Overview of PFL-LDG

This paper assumes that IoT devices located in different places within a computation-resource-constrained edge network hold non-IID data. Therefore, a new Privacy-preserving Federated Learning via Lightweight Device Grouping (PFL-LDG) is proposed to achieve efficient PFL in heterogeneous data settings for computation-resource-constrained edge networks. PFL-LDG is a framework that implements PFL through clustering. Figure 1 describes the basic modules of the workflow in PFL-LDG, which include the Data Feature Extraction module (DFE), Homogeneous Device Grouping module (HDG), and Intra-group Client Scheduling module (ICS).

- **DFE:** A lightweight privacy-preserving data distribution feature extraction module, which initially extracts Data Distribution Features (also known as feature maps) from each device's raw data. It then employs a lightweight encryption function PHOTON-beetle-hash [27] to encrypt these data distribution features, while leveraging hashing to minimize the required data storage space. This module preprocesses device data distribution features before officially commencing FL, avoiding high computational costs for PFL processes in computation-resource-constrained situations. Subsequently, the devices transmit the encrypted data distribution features to the edge server.

- **HDG:** The Homogeneous Device Grouping Module, in which the server organizes devices based on the received hash summaries. The edge server utilizes hamming distance to gauge the similarity between the extracted Encrypted Data Features (EDF), generating a similarity matrix representing client data distribution similarities. Upon completing the similarity computation process, devices with comparable data distributions are allocated to the same group.
- **ICS:** The grouping FL training process formally begins as the server notifies devices in each group and orchestrates the FL process within each group. Devices exclusively share gradient updates with other devices in the same group. The edge server gathers computation resource usage reports from devices during each training round, subsequently selecting devices with adequate computing power within each group and aggregating the gradient information sent by the chosen devices in each group to iteratively train an Intra-Group model (i.e., the global model for the group). Ultimately, the resulting Intra-Group models constitute PFL models.

B. Data Feature Extraction Module

Considering the large volume of data samples collected in edge computing and the limited computational resource budget for IoT devices to extract their data distribution features in resource-constrained scenarios, it is essential to note that directly collecting and transmitting data distribution features by IoT devices increases the risk of privacy leakage. This contradicts the initial intention of FL, which aims to enhance the privacy protection level of distributed machine learning. To achieve efficient PFL in resource-constrained edge networks with heterogeneous data, the data feature extraction module is crucial, as obtaining data distribution features is a prerequisite for device grouping in the proposed similarity-based grouping FL.

The data feature extraction module designed in this article primarily consists of three steps: server model distribution, device pre-training, and encrypted feature extraction. The edge server broadcasts the pre-trained FL model to all IoT devices within the domain, and then the devices use their own data to train their local models. Finally, the device data distribution features are extracted from the pre-training process and undergo lightweight encryption.

Specifically, the data feature extraction module in the proposed framework, before device grouping and formal group FL, performs the first stage of preprocessing for the data distribution of devices. Firstly, the edge server broadcasts the global model, all devices use their own data to train their local models, and extract the ReLU layer feature map from the CNN during the pre-training process. Then, the device employs a lightweight encryption hash function to encrypt the ReLU layer feature map and uploads the encrypted data feature (EDF) to the edge server after obtaining it.

The data feature extraction module retrieves device data distribution features before formally initiating FL and employs a lightweight encrypted hash function to obtain the encrypted data feature. The lightweight encrypted hash function enhances the privacy protection level for users on different devices and avoids introducing excessive computational costs for edge devices in resource-constrained scenarios. The data dimension reduction feature supported by the hash function minimizes the storage space occupied by data distribution features and reduces the overhead of communication between devices and servers.

C. Homogeneous Device Grouping Module

Considering real-world environments, edge devices often possess highly heterogeneous data, and non-IID data presents challenges to the speed and convergence of FL. Moreover, in resource-constrained edge scenarios, IoT devices have limited network connections, making parallel model updates and aggregation across all participating devices impractical. To address these challenges, this article introduces an HDG (Homogeneous Device Grouping) module, which primarily focuses on grouping homogeneous devices by the edge server using a similarity-based clustering method. This approach mitigates the impact of non-IID data on FL model performance in cases of highly heterogeneous data. The HDG module mainly comprises three processes: edge server collects hash summaries, calculates the similarity of device data distributions, and groups devices based on the similarity matrix.

Specifically, the homogeneous device grouping module in the proposed framework conducts a second stage of preprocessing for devices before formally initiating group FL. Firstly, the edge server collects the encrypted data features sent by all devices within the domain. Then, the hamming distance is employed to measure the similarity of data distributions among different devices, and the calculated results are saved in a similarity matrix representing the degree of data distribution similarity between clients. Finally, the Iterative Self Organizing Data Analysis Techniques Algorithm(ISODATA) clustering method is utilized to assign devices with similar data distributions to the same group, ensuring that all devices within each group are homogeneous.

D. Intra-group Client Scheduling Module

Considering that improper scheduling can lead to slow learning efficiency and cause straggler problems, especially in computation-resource-constrained edge environments. The accuracy of FL training depends on capturing all unique data distributions, not all clients. Therefore, based on the grouped devices, one or more devices can be randomly selected within each homogeneous group for FL training. However, in computation-resource-constrained scenarios, grouping FL still

Algorithm 1 Implementation of PFL-LDG.

```
Input: $C$,$D$,$K$,$G$,$S$,$P$,$R$
Output: Each intra-group model $W_k$ trained by PFL-LDG;
    Initialisation : Initialize homogeneous device grouping
    result set $G$ and encrypted data distribution feature hash
    value set $S$;
 1: The edge server distributes an initial global federated
    model to all devices, as well as the PHOTON-Peele-Hash
    function, and sends computation resource information
    requests;
 2: for $i = 1$ to $N$ do
 3:     for $j = 1$ to $P$ do
 4:         The device uses its own data training pre-training
            model;
 5:     end for
 6:     The device extracts ReLU layer feature map $v_i$;
 7:     $h_i$ = PHOTON-Beetle-Hash $(v_i)$;
 8:     $S \leftarrow h_i$;
 9: end for
10: The edge server, based on the received encrypted data
    distribution feature set $S$, uses the ISODATA algorithm
    to obtain the grouping result $G$;
11: for $k = 1$ to $K$ do
12:     for $l = 1$ to $R$ do
13:         The server selects devices with computational capa-
            bilities within the homogeneous group $G_k$ to partici-
            pate in the $l^{th}$ round of training based on the reported
            resource usage from the devices;
14:         The server distributes the intra-group model $W_k^l$
            to each client $c_l$ participating in the $l^{th}$ round of
            training;
15:         for each client $c_l \in G_k$ parallelly do
16:             Client $c_l$ trains the received model locally, as
                $$W_i^{l+1} = W_i^l - \frac{\alpha}{|D_i|}\sum_{j=1}^{|D_i|} \nabla f\left(W_i^l,\ x_{i,j}, y_{i,j}\right) \quad (1)$$
17:             Client $c_l$ uploads the local model $W_i^{l+1}$ to the
                edge server;
18:         end for
19:         The edge server aggregates the uploaded models
            from the selected clients within the group into a new
            intra-group model as
            $$W_k^{l+1} = \frac{\sum\limits_{c_l \in G_k} |D_i| W_i^{l+1}}{\sum\limits_{c_l \in G_k} |D_i|} \quad (2)$$
20:     end for
21: end for
22: return Each intra-group model $W_k$ trained by PFL-LDG.
```

requires reasonable lightweight scheduling for clients within each homogeneous group.

The primary function of the intra-group client scheduling module proposed in this paper involves the edge server collecting computation resource usage reports sent by clients. Subsequently, the server selects one or more devices with adequate resources (i.e., more memory and CPU processing power) for FL training. The edge server identifies the homogeneous group to which a device belongs and coordinates the FL process within each homogeneous group. Each device shares gradient updates only with other devices in the same group, resulting in distinct intra-group global models. Upon receiving the server's request, devices that have already been grouped send computation resource usage reports. Each group then commences FL. During this process, devices within each group monitor and periodically report their computation resource usage, including CPU and memory, to the server. This information enables the edge server to gauge each device's capabilities and schedule devices within the group accordingly.

The specific steps of grouping FL are as follows:

1) Initialization: The edge server sends computation resource information requests to all devices. What's more, Initialize homogeneous device grouping result set and encrypted data distribution feature hash value set.
2) Intra-group Model Distribution: Utilizing the computation resource usage reports submitted by each device, the server selects one or more devices with adequate computation capabilities within each homogeneous group. The server then sends the group-specific intra-group model to the chosen devices within each group. These devices participate in the current round of training and aggregation, while the remaining devices do not partake in the current training round.
3) Local Model Training: Each participant in the homogeneous client group receives the intra-group model. Participants update local model parameters based on local data and devices. The goal for client c_i in grouping FL is to obtain the optimal parameter W_i^t by minimizing the loss function $L\left(W_i^t\right)$ in the t^{th} iteration. Upon completing local training, participants send the updated model parameters back to the server.

$$W_i^t = \arg\min_{W_i^t} L\left(W_i^t\right) \tag{3}$$

4) Intra-group Model Aggregation: The edge server collects and aggregates the updated local model parameters from all participants within each homogeneous group. It then sends the updated intra-group model parameters W_G^t to all devices within the group to minimize the loss function $L\left(W_G^t\right)$, updating the group-specific global model.

$$L\left(W_G^t\right) = \frac{1}{N}\sum_{n=1}^{N} L\left(W_i^t\right) \tag{4}$$

Moreover, the server assesses the updated homogeneous group-specific intra-group models to determine if they have achieved the desired accuracy or convergence level. If the intra-group model has not yet converged, the process returns to step 2 and continues with subsequent grouping FL iterations for that group until the intra-group loss functions of all homogeneous groups attain optimal accuracy. The server shares the final intra-group models with all participating edge devices within each group. These devices can then use the updated models for prediction or further fine-tuning based on their specific use cases.

TABLE I
MAIN NOTATION IN PFL-LDG

Notation	Meaning
C	The collection of devices, i.e., $C = \{c_1, c_2, ..., c_n\}$
c_i	The i^{th} device
D_i	The private dataset held by device c_i
$\|D_i\|$	The size of the dataset
$x_{i,j}$	The j^{th} data sample on device c_i
$y_{i,j}$	The corresponding label of the j^{th} data sample on device c_i
K	The number of groups
G	The grouping result set of homogeneous devices
P	Local model training rounds
v_i	The vectorization result of extracting the ReLU layer feature map during pre-training FL on device c_i
h_i	The hash value obtained using a lightweight encryption hash function
S	The hash value set of encrypted data distribution features h_i
M	The similarity matrix used to compare the similarity of data distributions
R	Intra-group model training rounds

Algorithm 1 presents a comprehensive description of the PFL-LDG processing procedure. Table 1 provides a summary of the primary notations used throughout this paper. Throughout the grouping FL process, devices within each group monitor and periodically report their computation resource usage, such as CPU and memory, to the server. The server ensures that the grouping FL learning process remains efficient and adapts to each device's computation resource constraints. Additionally, devices only share gradient updates with other devices within their group, enhancing privacy and minimizing the impact of heterogeneous data on model performance.

IV. EXPERIMENTS

A. Experimental Setup

1) Datasets

The experiments utilize the MNIST [28] and CIFAR-10 [29] datasets, each containing data from 10 distinct categories. A non-IID data distribution is created by manually assigning training and testing samples from different categories to clients, simulating real-world data heterogeneity scenarios where the data categories within each client may vary. Specifically, each dataset is divided into five categories, with every 10 clients belonging to one category. Table 2 illustrates the classification of the two datasets, including the number of clients per category and the allocation of training and testing samples.

TABLE II
CLASSIFICATION OF DATASET

Dataset	Number of Categories	Number of clients for each category	Training samples	Testing samples
MNIST	5	10	5*10*800	5*10*200
CIFAR-10	5	10	5*10*800	5*10*200

2) Models

The model architecture employed is MobileNetV2 [30], a lightweight convolutional neural network specifically designed for mobile devices, making it well-suited for edge computing. Considering the computing resource constraints of FL clients in resource-constrained scenarios, the smallest variant, MobileNetV2-0.25, is used.

3) Comparative Methods

The comparison methods chosen for this paper are FedAvg [3] and HACCS [19], a clustering FL method that protects user data distribution features using differential privacy techniques.

- FedAvg: Without sharing raw data, multiple clients collaborate to train a shared global model, which can test the performance of all clients within the context of this paper.
- HACCS: HACCS is a clustering FL method that accounts for data heterogeneity. Its core concept involves clustering similar clients into distinct groups for local training within each group. This approach reduces the weight divergence problem between groups, minimizes communication overhead, and achieves higher-performance personalized FL models.

4) Implement Details

The proposed method is implemented using Pytorch [31]. Experiments are conducted on a Windows 10 workstation with an Intel(R) Xeon(R) Scalable Processors CPU@3.7GHz and 16GB RAM. One NVIDIA Tesla V100S PCIe 32 GB GPU is utilized for training and testing. The learning rate for FL is set to 0.01, and the momentum is set to 0.5. Local training for device grouping is carried out for 30 rounds, and intra-group FL training takes place for 100 rounds following the completion of grouping.

B. Experimental Results

In FL, employing clustering can enhance the model's accuracy. Table 3 presents the results of the proposed PFL-LDG, HACCS, and FedAvg when training MobileNetV2 on the MNIST dataset under non-IID settings, while the same Table 4 also displays the outcomes for the CIFAR-10 dataset.

In the context of simulated edge data heterogeneity with non-IID settings, the FedAvg algorithm fails to capture the distinct data distribution of FL clients, leading to a substantial performance loss in the final global model. Both PFL-LDG and the personalized FL algorithm HACCS outperform

TABLE III
MNIST RESULTS FOR ALL METHODS ON MOBILENET

Method	Each Intra-group Model Accuracy (%)					Average(%)
	Group1	Group2	Group3	Group4	Group5	
FedAvg	N/A	N/A	N/A	N/A	N/A	87.57
HACCS	96.53	96.79	97.12	97.25	96.87	96.91
PFL-LDG	**98.86**	**98.32**	**99.07**	**98.77**	**98.29**	98.66

TABLE IV
CIFAR-10 RESULTS FOR ALL METHODS ON MOBILENET

Method	Each Intra-group Model Accuracy (%)					Average(%)
	Group1	Group2	Group3	Group4	Group5	
FedAvg	N/A	N/A	N/A	N/A	N/A	56.19
HACCS	78.46	78.92	76.27	78.71	77.36	77.94
PFL-LDG	**82.11**	**82.79**	**79.84**	**81.67**	**80.43**	**81.37**

FedAvg significantly, which relies on a single shared global model. As PFL-LDG selects clients based on resource information during the grouped FL process and HACCS only randomly chooses one device in each round after clustering is completed, PFL-LDG achieves superior performance compared to HACCS, which focuses on personalization for individual clients.

V. CONCLUSION

In this paper, PFL-LDG is proposed as a privacy-preserving lightweight grouping federated learning algorithm, which designed for edge resource-constrained conditions. PFL-LDG employs a lightweight encrypted hash function during the grouping phase, which offers privacy protection for the data distribution features necessary for device grouping. This method circumvents the low data utility issue associated with DP techniques and avoids the prohibitive computational costs introduced by techniques such as MPC and HE. Additionally, the grouping strategy substantially mitigates the influence of non-IID data on FL, enhancing the training efficiency of heterogeneous edge resource-constrained FL systems.

Experimental results comparing the proposed method to the FL baseline methods and advanced methods that use DP techniques to ensure user privacy illustrate the personalized performance of the proposed approach. As for future work, the authors will continue to explore the server's scheduling process for selecting clients within groups after grouping completion, further optimizing the efficiency and performance of the grouping FL.

ACKNOWLEDGMENT

This work has received funding from National Key Research and Development Program of China under Grant No.2021YFE0116900, National Natural Science Foundation of China (No. 42275157 and 41975142), Major Program of the National Social Science Fund of China (Grant No. 17ZDA092), and basic Research Programs (Natural Science Foundation) of Jiangsu Province (BK20191398).

REFERENCES

[1] J. Liu et al., "From distributed machine learning to federated learning: A survey," Knowledge and Information Systems, vol. 64, no. 4, pp. 885-917, 2022.
[2] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," A Practical Guide, 1st Ed., Cham: Springer International Publishing, pp. 10-5555, 2017.
[3] B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in Proc. AISTATS, PMLR, 2017.
[4] B. Luo et al., "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in Proc. IEEE INFOCOM 2022, IEEE, 2022.
[5] X. Li et al., "On the convergence of fedavg on non-IID data," arXiv preprint arXiv:1907.02189, 2019.
[6] H. Wang et al., "Optimizing federated learning on non-IID data with reinforcement learning," in Proc. IEEE INFOCOM 2020, IEEE, 2020.
[7] S. P. Karimireddy et al., "Scaffold: Stochastic controlled averaging for federated learning," in Proc. ICML, PMLR, 2020.
[8] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local SGD on identical and heterogeneous data," in Proc. AISTATS, PMLR, 2020.
[9] A. Z. Tan et al., "Towards personalized federated learning," IEEE Transactions on Neural Networks and Learning Systems, 2022.
[10] V. Smith et al., "Federated multi-task learning," in Proc. NIPS, vol. 30, 2017, pp. 4427-4437.
[11] L. Corinzia, A. Beuret, and J. M. Buhmann, "Variational federated multi-task learning," arXiv:1906.06268, 2019.
[12] Y. Huang et al., "Personalized cross-silo federated learning on non-IID data," in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 9, 2021, pp. 7865-7873.
[13] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," arXiv:2002.05516, 2020.
[14] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," arXiv:2003.13461, 2020.
[15] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," in Proc. ICLR, 2021, pp. 1-24.
[16] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 8, pp. 3710-3722, Aug. 2021.
[17] C. Briggs, Z. Fan, and P. Andras, "F ederated learning with hierarchical clustering of local updates to improve training on non-IID data," in Proc. IJCNN, Jul. 2020, pp. 1-9.
[18] B. Liu, Y. Guo, and X. Chen, "PFA: Privacy-preserving federated adaptation for effective model personalization," in Proc. Web Conf. 2021, 2021.
[19] J. Wolfrath et al., "Haccs: Heterogeneity-aware clustered client selection for accelerated federated learning," in Proc. IEEE IPDPS, 2022.
[20] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," IEEE Trans. Inf. Forensics Security, vol. 15, pp. 3454-3469, 2020.
[21] L. Zhang et al., "A robust game-theoretical federated learning framework with joint differential privacy," IEEE Trans. Knowl. Data Eng., vol. 35, no. 4, pp. 3333-3346, 2022.
[22] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in Proc. ACM SIGSAC Conf. Comput. Commun. Security, 2017.
[23] K. Mandal, G. Gong, and C. Liu, "Nike-based fast privacy-preserving high-dimensional data aggregation for mobile devices," IEEE T Depend Secure, 2018, pp. 142-149.
[24] L. Zhang et al., "Homomorphic encryption-based privacy-preserving federated learning in iot-enabled healthcare system," IEEE Trans. Netw. Sci. Eng., 2022.
[25] M. Song et al., "Analyzing user-level privacy attack against federated learning," IEEE J. Sel. Areas Commun., vol. 38, no. 10, pp. 2430-2444, 2020.
[26] Y. Jiang et al., "Anonymous and Efficient Authentication Scheme for Privacy-Preserving Distributed Learning," IEEE Trans. Inf. Forensics Security, vol. 17, pp. 2227-2240, 2022.

[27] Z. Bao et al., ”PHOTON-beetle authenticated encryption and hash family. A submission to the NIST lightweight cryptography standardization process (2021),” 2022.

[28] Y. LeCun et al., ”Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[29] A. Krizhevsky, ”Learning multiple layers of features from tiny images,” University of Toronto, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf

[30] M. Sandler et al., ”Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proc. IEEE CVPR, 2018.

[31] Paszke et al., ”Pytorch: An imperative style, high-performance deep learning library,” in Adv. Neur. Inf. Process. Syst. 32, 2019.