Communication-Efficient Personalized Federated Meta-Learning in Edge Networks

Feng Yu, Hui Lin, Xiaoding Wang, Sahil Garg, *Member, IEEE*, Georges Kaddoum, Satinder Singh, *Member, IEEE*, and Mohammad Mehedi Hassan, *Senior Member, IEEE*

***Abstract*—Due to the privacy breach risks and data aggregation of traditional centralized machine learning (ML) approaches, applications, data and computing power are being pushed from centralized data centers to network edge nodes. Federated Learning (FL) is an emerging privacy-preserving distributed ML paradigm suitable for edge network applications, which is able to address the above two issues of traditional ML. However, the current FL methods cannot flexibly deal with the challenges of model personalization and communication overhead in the network applications. Inspired by the mixture of global and local models, we proposed a Communication-Efficient Personalized Federated Meta-Learning algorithm to obtain a novel personalized model by introducing the personalization parameter. We can improve model accuracy and accelerate its convergence by adjusting the size of the personalized parameter. Further, the local model to be uploaded is transformed into the latent space through autoencoder, thereby reducing the amount of communication data, and further reducing communication overhead. And local and task-global differential privacy are applied to provide privacy protection for model generation. Simulation experiments demonstrate that our method can obtain better personalized models at a lower communication overhead for edge network applications, while compared with several other algorithms.**

***Index Terms*—Edge networks, federated meta learning, representation learning, autoencoder, differential privacy.**

I. Introduction

With the rapid development of edge networks and mobile Internet of Things, a large number of intelligent terminals have entered people's lives. As time goes, smart terminals will generate "massive" data. According to the "Data Age 2025" white paper released by IDC [1], the global data volume is expected to grow to 175 ZB by 2025, which is more than ten times the 16.1 ZB data generated in 2016. Therefore, the generated massive data are distributed in various devices at the edge or data centers of different organizations. The nodes located at the edge of the network drive the migration of applications, data, and computing power from centralized data centers to these edge nodes. Therefore, data storage and computing resources must be as close as possible to the demand side, reducing the data to be moved, traffic, and distance traveled, lowering latency and transmission costs. It is critical to place computing resources and data storage at the edge of the network for the efficient functioning of edge networks.

The problems of traditional machine learning (ML) technology in edge network applications, such as data silos, privacy leakage and data security risks, regulatory requirements and engineering obstacles, can be solved by federated learning (FL) [2]. However, FL encounters several challenges, which are grouped into following four main aspects [3]. (1) Communication. To establish a common goal and model structure in the federated network, the model or parameters have to be transferred between the client and server, which can create a significant communication overhead. (2) System heterogeneity. Computing and communication capabilities vary by network connectivity, hardware, energy, and storage. (3) Statistical heterogeneity. How devices collect or generate data vary widely, with samples collected in distinct situations. And these heterogeneous samples are usually known as non-IID data [4]. (4) Privacy. During FL training, there is a risk that model updates may unintentionally leak sensitive information to third-party or central servers [5].

In addition, the issue with FL is that its optimization goal is to obtain a global model, which can be regarded as an "average" model. However, in federated setting with a high degree of non-IID data, this global model may not be able to adapt effectively to all samples. Recently, many works have begun to exploit various personalization techniques [6] to obtain "personalized models" to solve this problem. However, these methods cannot achieve flexible personalization and may ignore the communication bottleneck challenge, which is worth considering in the network applications. Per-FedAvg algorithm [7], combining Model-Agnostic Meta Learning (MAML) [8] with FedAvg [9] can quickly obtain personalized models adapted to the data of devices.

Manuscript received 23 September 2022; revised 27 February 2023; accepted 26 March 2023. Date of publication 31 March 2023; date of current version 6 July 2023. This work was supported by the King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP2023R18. The associate editor coordinating the review of this article and approving it for publication was A. Mourad. *(Corresponding authors: Hui Lin; Xiaoding Wang.)*

Feng Yu, Hui Lin, and Xiaoding Wang are with the College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China, and also with the Engineering Research Center of Cyber Security and Education Informatization, Fujian Province University, Fuzhou 350117, Fujian, China (e-mail: fzhiy270@163.com; linhui@fjnu.edu.cn; wangdin1982@fjnu.edu.cn).

Sahil Garg is with the École de Technologie supérieure, Montreal, QC H3C 1K3, Canada (e-mail: garg.sahil1990@gmail.com).

Georges Kaddoum is with École de Technologie supérieure, Montreal, QC H3C 1K3, Canada, and also with the Cyber Security Systems and Applied AI Research Center, Lebanese American University, Beirut, Lebanon (e-mail: georges.kaddoum@etsmtl.ca).

Satinder Singh is with Ultra Communications, Montreal, QC H4T 1V7, Canada (e-mail: Satinder.Singh@ultra-tcs.com).

Mohammad Mehedi Hassan is with the Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: mmhassan@ksu.edu.sa).

Digital Object Identifier 10.1109/TNSM.2023.3263831

Although it provides a solution for personalized federated learning, it does not allow flexibility in obtaining personalized models, and there is some unnecessary communication overhead over heterogeneous data. Motivated by the idea of mixture of the local model and global model presented in new formulation of FL [10], we can solve the aforementioned issues by introducing a personalization coefficient to the FML problem in [11]. And the size of the personalization coefficient is adjusted according to the relevance of the device's data, i.e., the higher the relevance, the larger the coefficient, and vice versa, which can reduce the communication overhead to meet the needs of devices with limited resources in edge networks to participate in the learning process. Therefore, we propose a Communication-Efficient Personalized Federated Meta-Learning algorithm (CE-PFML) to deal with the above challenges. Specifically, we introduce personalized parameter α_i for client i to update the global model with the optimal local model of client i. And the size of α_i can represent the degree of influence of the optimal local model on the global model.

Further, we introduce representation learning to reduce communication overhead, which is achieved by extracting efficient and low-latitude local updates for communication, to address the communication bottleneck challenge in FL setting. Moreover, we introduce differential privacy (DP) for meta learning [12], [13] (i.e., Task-Global DP and Local DP) to ensure the privacy of the federated system.

The main contributions of this paper can be summarized as follows:

- Inspired by the idea of mixing the global model and the local models, we propose a CE-PFML algorithm, where the personalization coefficient α_i is introduced into the FML objective, while compared to Per-FedAvg, to personalize federated model and obtain high quality personalized model. Simultantly, we can flexibly accelerate the convergence of the model by adjusting the size of the personalized parameter α_i for client i. As the higher the degree of data correlation, the closer α_i is set to 1, and the greater the impact on other participants, the better meta-model can be obtained in fewer communication rounds.
- Further, the local model to be uploaded is transformed into the latent space by introducing autoencoder, thereby reducing the amount of communication data, and then reducing communication overhead. And Local and Task-Global DP is applied to provide privacy protection for model generation.
- Simulation experiments demonstrate that CE-PFML is more effective and efficient for edge network applications, while compared with several other algorithms.

Outline of the Paper: The rest of this paper is organized as follows. Section II briefly reviews the related work. Section III introduces Federated Meta-Learning in edge networks. We present details of proposed method in Section IV. Sections V and VI presents the theoretical analysis, performance evaluation and our analyzation, respectively. And we summarize this paper in Section VII. Table I shows related abbreviations of term and their meanings in this paper.

TABLE I
ABBREVIATIONS OF TERM AND THEIR MEANINGS

Abbreviation	Meaning
IID	data distributed on same device and assuming independently from same distribution
FL	Federated Learning
FedAvg	FederatedAveraging
MAML	Model-Agnostic Meta Learning
FML	Federated meta-learning
DP	Differential Privacy
AE	Autoencoder
MLP	Multi-Layer Perceptron
CNNs	Convolutional Neural Networks

II. Related Work

In this section, we overview and discuss the related works and efforts that apply to personalized solutions and communication efficiency in FL.

A. Personalized Federated Learning

There are mainly two strategies for personalized federated learning (PFL) [14], one strategy for PFL is *Global Model Personalization*, whose aim is to improve the generalization performance of global models under data heterogeneity in order to improve the performance of subsequent personalization on local data. Another strategy is *Learning Personalized Models*. It aims to create personalized models by modifying the FL model aggregation process.

1) Global Model Personalization: This strategy can be divided into *Data-based* approaches and *Model-based* approaches.

- *Data-based* approaches. In [4], the authors proposed to improve training on non-IID data by creating a small subset of data that is globally shared among all edge devices. And experiments show that accuracy can be significantly increased (∼30%) with a small globally shared data. In [15], the authors proposed the FedHome algorithm that Generative Convolutional Autoencoder (GCAE) was designed to improve the model by generating a locally augmented class-balanced dataset to achieve accurate and personalised health detection in FL. In [16], the authors proposed a federated learning approach for continuous authentication, which utilizes part of clients' training set to train a warmup global model to solve the non-IID problem in FL. In [17], the author proposed FAVOR, an experience-driven control framework that intelligently selects clients to participate in FL to offset the bias introduced by non-iid data and speed up convergence. The Deep Reinforcement Learning (DRL)-based client selection mechanism is designed to improve maximum accuracy while minimizing the number of communication rounds. In [18], the author proposed a Tier-based Federated Learning system (TiFL), which divides clients into different levels according to their training performance, and selects clients from the same level in each training round to mitigate the stragglers problem caused by the heterogeneity of resources and data volume. Further, to address the heterogeneity caused by non-IID data and resources, an adaptive tier selection method, which updates the tiering in real time based on the observed training performance and accuracy over time, is proposed. Recently, in [19], the authors proposed FedAUR, an approach for adaptive upgrade of clients resources in FL. The client selection and resource allocation problem is formulated as an optimization problem by designing a method to measure the performance of locally generated models against the aggregated global model and a selection scheme based on the importance of client parameters and their device resources. It aims to discover and train the maximum number of samples with the highest quality in each round to achieve the goal of desired performance. In [20], the authors proposed a solution to the client selection problem by using clients' weights to select a compatible subset with minimal weight differences to aggregate the initial global model, while also dealing with the dynamic evolution of the learning environment without sacrificing clients' privacy. In [21], the authors proposed on-demand FL architecture that allows the devices to run a ML model anytime and anywhere using containerization technology through lightweight containers, thereby providing the system with the ability to deploy and select clients in real-time.

- *Model-based* approaches. In [22], the authors introduced an approximation term for the local sub-problem to adjust the impact of local updates taking into account the dissimilarity between the global FL model and the local model to obtain personalized model. In a nutshell, training strongly adaptive models to solve new tasks with a few samples is the goal of Meta Learning. One of the most popular meta-learning algorithms recently is MAML [8], its goal is to train the model's initial parameters so that the optimal results can be obtained after one or several gradient updates based on few data in a new task. Further, in [7], the authors pointed out that the typical FL algorithm FedAvg is essentially a MAML algorithm. And MAML is divided into an outer loop and an inner loop. The inner loop corresponds to the local update of the participants in FedAvg, and the outer loop corresponds to It is based on the global update of FedAvg, and the two kinds of updates are single-step or multi-step gradient descent based on local data and single-step gradient update based on global parameters. At a higher level, the purpose of MAML is to find a suitable parameter that enables it to get a better result with as few updates as possible when fine-tuning a new task; and for personalized FL, we hope that the obtained global model can also get a good personalized model after fine-tuning on the local data, so the two algorithms are intrinsically interoperable [7]. In [23], the authors proposed FedMeta. Each client is treated as a task and to train a well-initialized global model rather than a globally optimal model is the goal. FedMeta uses a shared meta-learner to replace the shared global model in FL, which can well adapt different meta-learning systems to FL systems. At the same time, the framework shares parameterized algorithms in a more flexible way while protecting client privacy by not collecting data on the server. Per-FedAvg, as a personalized variant of the joint averaging algorithm, is proposed in [24]. It leverages meta-learning algorithms to find shared global models that can quickly adapt to different clients, performing well on each client in just a few steps. In [25], the authors developed a generic framework based on transfer learning (TL) and knowledge distillation that allows for FL when each client has not only its own private data but also a uniquely designed model. Before the FL training, TL is first carried out based on a public dataset, and then each client fine-tunes this model on its own private data.

2) Learning Personalized Models: This strategy can be divided into *Architecture-based* approaches and *Similarity-based* approaches.

- *Architecture-based* approaches. In [26], the authors proposed FedPer, a base + personalization layer approach for federated training. In this setting, personalized layers are kept private for each client to learn personalized representations, while the base layers are shared with the server to learn general features. In [27], the authors proposed LG-FedAvg, each client learns a compact local representation, and all clients learn a global model collaboratively. The way that the global model only acts on the compact local representations reduces the amount of communication.
- *Similarity-based* approaches. In [10], a formulation different from FedAvg is proposed, it aims to look for a trade-off between global and local models. Each client takes into account its own local data features and strives to learn a mixture of the local models and the global model, while compared with FedAvg.

B. Model Compression

There are several approaches, i.e., Sparsification, Quantization, Knowledge Distillation and Low-rank factorization, that focus on improving the communication efficiency in FL through better representation of the data.

1) Sparsification: Sparsification is a technique that regenerates the matrices independently for each client in each round by using sparse matrices to characterize locally updated models based on a preset sparse pa. In [28], the authors proposed a sparse ternary compression (STC) framework based on non-IID, unbalanced and small-scale batch local data. STC extends the current uplink and downlink compression methods of top-K gradient sparsification through sparsification, ternaryization, error accumulation and optimal Golomb coding, which can reduce the communication frequency while reducing the amount of data transmitted in each communication round. In [29], the authors integrated local computation and gradient sparseness, and proposed a flexible Top-K local SGD algorithm with a dynamic batch size (FT-LSGD-DB), which achieves flexible compression by allowing participants to perform gradient sparsification with different "K" values.

2) Quantization: Quantization techniques were originally used for data compression. In the FL setting, the gradient is calculated locally by quantization, and the gradient is quantized to a low-precision value instead of directly uploading the original gradient, which reduces the communication cost and the number of communication bits each round, but this will reduce the accuracy and increase the overall energy consumption of calculation. In [30], the authors introduce quantitative techniques into FL to learn recursive neural network models provided by edge data producers for time series prediction to improve the efficiency of data exchange between edge servers and cloud nodes.

3) Knowledge Distillation: Knowledge distillation can be used in FL to alleviate communication challenge by training a smaller, more compact model to mimic the behavior of a larger, more complex model. In [31], the authors proposed a data-free knowledge distillation approach to address heterogeneous FL. The generator learns the feature of the global data distilled from the global model aggregated by the server, and then provides clients for the information to improve the performance of local learning.

4) Low-Rank Factorization: Methods based on low-rank factorization techniques use matrix or tensor factorization to estimate the most informative parameters in deep CNNs. In [32], the authors proposed a heterogeneous federation model compression framework, FedHM, which distributes heterogeneous low-rank models to clients and then aggregates them into a full-rank model. FedHM significantly reduces communication costs by using low-rank models.

C. Discussion

Although these efforts provide personalized solutions for FL or improve the communication efficiency in FL to some extent. However, these methods cannot achieve flexible personalization and there is some unnecessary communication overhead over heterogeneous data, which contributes to inefficiencies in communication. Therefore, we propose a communication-efficient federated meta-learning algorithm to solve these issues by a modified formulation of FL. Further, autoencoder is introduced to reduce communication overhead.

III. Federated Meta-Learning in Edge Networks

A. MAML and FedAvg

In this section, we briefly recap MAML and its learning procedure, the algorithmic logic of MAML [7] and FedAvg [9]. They are the preliminary knowledge of federated meta-learning.

The meta-learning method combined with FL is usually MAML, and its essence is to quickly obtain personalized models through good initial training and fine-tuning. The fine-tuning technique is usually based on the partial layers of the source model pre-trained on the source data to fine-tune to obtain the target model with stronger generalization ability to the target data set. MAML consists of two layers of learners (or models), meta-learner/model and base-learner/model. The training process of MAML is as follows: (1) First, the same model with the same random parameter set is distributed to the meta-learner and the base-learner. In a meta-model update iteration of meta training, all base-learners independently and randomly extract tasks sampled from isolated classes in meta-training and novel tasks to test in meta-testing. (2) Model fine-tuning, each base-learner applies the samples of the support set to optimize the base-model through a vanilla gradient descent, and then determines a descent strategy to minimize the loss of the optimized model on the query set. (3) The global gradient is obtained by calculating the average of all local gradients. The meta-model is updated by the meta-learner through a global gradient descent, and then as the updated base-model it is sent to base-learners. Repeat steps (1)-(3) until the meta-model converges. We can get excellent performance by deploying the trained meta-model on the samples in the meta-testing stage. By reviewing the application of MAML in FL, we find that MAML, like FL, has a two-layer model architecture, which makes MAML a natural fit for FL.

The similarity of FedAvg and MAML algorithms and architectures is introduced as follows, which will fully demonstrate the intrinsic fit of MAML and FL. A batch of tasks for training are randomly sampled in each round. For each task, an *inner-loop update* and aggregating the gradients of each sampled task via the *outer-loop update* are performed in MAML algorithm. In each epoch, a randomly sampled clients set among all clients as participants are used in FL algorithm. Each participant runs the optimization process for multiple epochs over its local datasets for its weight, then the local updates is sent to the server. Next, the current global model is updated by aggregating updates. Therefore, MAML and FL are actually the same algorithm while all clients have the same weight.

B. System Components

The system model of federated meta-learning in edge networks is illustrated in Figure 1. The basic principles of this system can be divided into system components and system workflow. We describe the system elements and their corresponding roles as follows.

- *Clients:* are the distributed devices with non-IID and heterogeneous data sizes at the edge or data centers of different organizations, such as the desktop, laptop, phone, bank and hospital.
- *Participants*[1]*:* are essentially the selected clients to participate in the learning process, they are subscribed to a certain federated meta-learning application. They are responsible for fine-tuning the local model over their own samples and then performing several rounds of mini-batch SGD to update the local base-model.
- *Edge Server:* is a type of server (also known as meta-learner) that is located at the network edge, closer to the clients. It is responsible for coordinating models or parameters communication between the clients. It also perform other tasks, such as model aggregation, and model distribution.
- *Novel Clients:* are "unused" clients in the deployment phase of FML for classification tasks, whose datasets are never used during meta-model training. Typically, their datasets are imbalanced and their distribution differs from that of the participants. This means that they are not eligible to participate in training. So they subscribe to the

[1]In our paper, participants and base-learners, the edge server and meta-learner are equivalent, and we use them as needed.

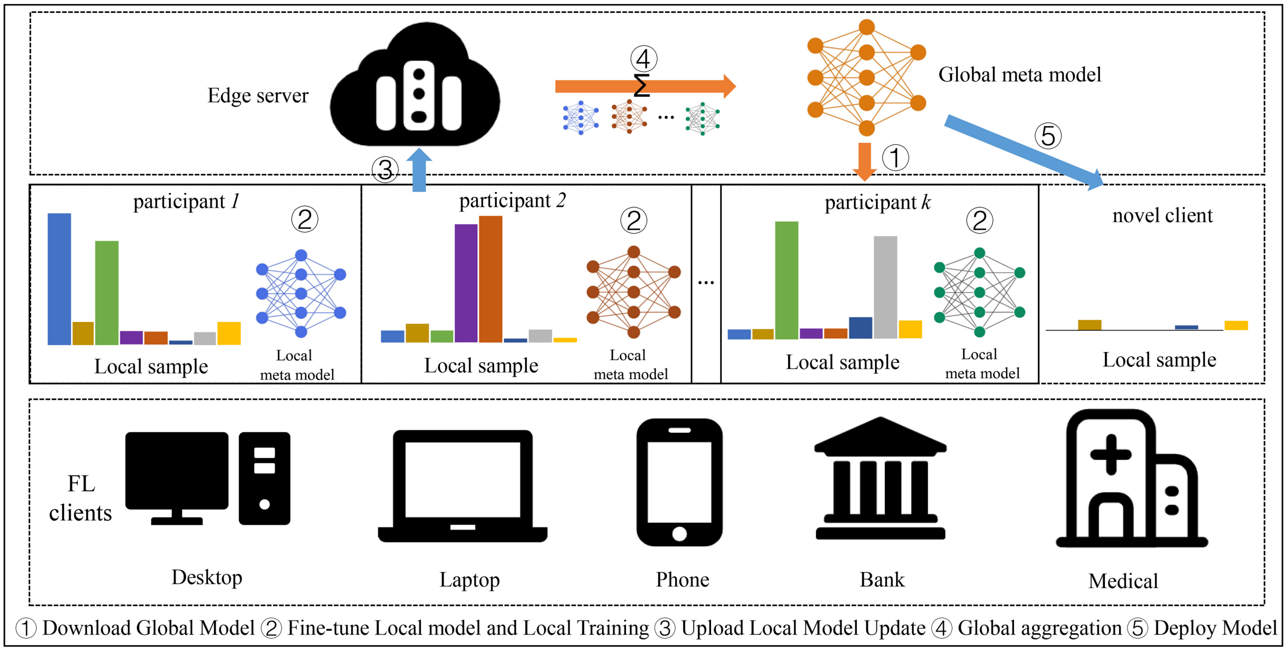


Fig. 1. The system model of federated meta-learning in edge networks.

FML service, hoping to use the optimal meta-model to make classification predictions.

C. System Workflow

In this section, we detail the procedure of our proposed method, shown in Figure 1, as follows.

1) *Initialization:* After receiving requests of using FML service from several clients, the system start to perform parameters initialization based on preset automated programs. Several clients are randomly selected as participants for training.
2) *Download global model:* Meta-learner sends the current meta-model to the base-learners;
3) *Fine-tune local model and local training:* To obtain the optimal base-model, each base-learner trains received model over its own samples for a predetermined duration with the use of an optimizer, such as SGD;
4) *Upload local model update:* Once the local training is finished. All base-learners upload optimal base-model and personalized parameter to meta-learner;
5) *Global aggregation:* Meta-learner receives base-models, and aggregates them with personalized parameters to generate the meta-model.

We repeat step 1) to step 4) until the program has reached the preset number of communication rounds. Finally, the final optimal meta-model will be deployed to the novel clients to be tested.

D. Threat Models

For any privacy-preserving work, we first consider the threat mode, i.e., potential adversaries and information to be protected.

1) *Potential attackers:* For a single task-owner, the attackers are either the receivers of the meta-model (namely, base-learners) or the receiver of base-model updates (the meta-learner). And here we consider an honest but curious meta-learner, that is, an aggregator that does not violate contracted algorithms but may try to obtain the private information of participants from model updates through inference attacks.
2) *Information to be protected:* Here we consider to protect the information in each sample as well as the information in the overall dataset at the same time.

TABLE II
SUMMARY OF NOTATIONS

Notation	Description
N	number of set of all edge devices (clients)
K	number of participants
i	client number
D_i	dataset belonging to client i
p_i	distribution of D_i
X	set of the data inputs
Y	set of the data labels
MM	meta-model
MM^*	optimal MM, equivalent to $x*$
R	number of outer-loop update
τ	number of inner-loop update
α	step size
α_i	personalized parameter of participant i
β	meta-learning rate
s_i	a sample in the dataset
ε	parameter for measuring the degree of privacy protection in DP
σ	probability of ε-DP failure
q	communication rounds of *Preparation Phase*
w	communication model size of each round in uncompressed state
w_{de}	communication model size of each round in compressed state
e	autoencoder error of participant i
C	compression rate of the autoencoder

IV. Proposed Method: CE-PFML

A. Federated Meta Learning

We introduce the problem description and standard algorithm in Federated Meta Learning (FML, a base algorithm of CE-PFML) in this section. And all of the notions are summarized in Table II.

1) FML Problem Description: We consider a C/S architecture model. In this model, a central server connects to N clients, each of which processes its own dataset $D_i = \{x_i^j, y_i^j\}_{j=1}^{D_i}$ and no entity else can access the dataset except itself. For each data sample, $(x_i^j, y_i^j) \in X \times Y$ follows an unknown distribution p_i, the former represents the data and the latter represents the label. Assuming that θ (such as weights) is the model parameter of the deep neural network. For client *i*, we assume that $L_i(\theta; x, y)$ is the loss function of the model $\theta \in \mathbb{R}^d$ based on the input data *x* and corresponding label *y*. FML attempts to look for a good initialization model, known as the optimal meta-model, to quickly obtain a model that performs well on different client devices through several gradient descent steps. More specifically, the learning objective of FML can be formulated as follows:

$$\min_{\theta \in \mathbb{R}^d} f(\theta) \overset{def}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(-\alpha \nabla f_i(\theta) + \theta). \tag{1}$$

We typically take $f_i(\theta) \overset{def}{=} \mathbb{E}[L_i(\theta; x, y)]$ for a machine learning problem, where the expected loss function over the data distribution of the client i is denoted by f_i, and α is the step size.

2) FML Standard Algorithm: Similar to FL, the vanilla FML algorithm also solves the problem through two iterative steps, namely the global aggregation and local update, as follows:

- *Local update:* At the global round $r \in [0, R)$, the participants (*K* clients that are randomly and uniformly selected) first obtain the global model from the server. And in order to update received model based on its own loss $F_i(\theta) \overset{def}{=} f_i(-\alpha \nabla f_i(\theta) + \theta)$, each participant $i \in n_r$ performs τ steps of SGD locally (also known as mini-batch SGD), formulated as follows:

$$\theta_i^{r,t+1} = \theta_i^{r,t} - \beta \tilde{\nabla} F_i\left(\theta_i^{r,t}\right), for\ 0 \leq t \leq \tau - 1 \tag{2}$$

  where θ_t^r denotes client *i*'s local model in the *t*-th step during the *r*-th round's local update with $\theta_i^{r,0} = \theta^r$, and the meta learning rate is denoted by β. The stochastic gradient, denoted by $\tilde{\nabla} F_i(\theta)$, used in (2) can be calculated as

$$\tilde{\nabla} F_i(\theta) \overset{def}{=} \left(I - \alpha \tilde{\nabla}^2 f_i(\theta, D_i'')\right) \tilde{\nabla} f_i\left(-\alpha \tilde{\nabla} f_i(\theta, D_i) + \theta, D_i'\right). \tag{3}$$

  where D_i, D_i', D_i'' are independent batches of distribution p_i, and for a batch of data D of distribution p_i, $\tilde{\nabla}^2 f_i(\theta, D)$ and $\tilde{\nabla} f_i(\theta, D)$ are the unbiased estimates of $\nabla^2 f_i(\theta)$ and $\nabla f_i(\theta)$ respectively, i.e.,

$$\tilde{\nabla} f_i(\theta, D) \overset{def}{=} \frac{1}{|D|} \sum_{(x,y) \in D} \nabla L_i(\theta; x, y) \tag{4}$$

$$\tilde{\nabla}^2 f_i(\theta, D) \overset{def}{=} \frac{1}{|D|} \sum_{(x,y) \in D} \nabla^2 L_i(\theta; x, y). \tag{5}$$

  As illustrated in [11], computation cost of the gradient $\nabla f_i(\theta)$ and the Hessian $\nabla^2 f_i(\theta)$ at every round is often high. So we can reduce the computation overhead by unbiased estimation of the equations (4) and (5).
- *Global aggregation:* When the local model update is completed, each participant sends the central server with the local model $\theta_i^r = \theta_i^{r,\tau-1}$. Then, the global model is updated by

$$\theta^{r+1} = \frac{1}{K} \sum_{i \in n_r} \theta_i^r. \tag{6}$$

B. CE-PFML

Inspired by the model mix thinking and the data compression of representation learning [7], [33], we propose Communication Efficient Personalized Federated Meta Learning (CE-PFML) to deal with high communication cost and model personalization. Here the introduction of the personalized parameter α_i makes the final meta-model more adaptive for the sample of client *i*, and can make the meta-model flexible. And the introduction of local representation learning reduces the amount of communication, which reduces the communication cost. The objective of CE-PFML is formulated as follows:

$$\min_{x \in \mathbb{R}^d} f(x, \alpha_{i=1\ldots n}, x_{i=1\ldots n}) \overset{def}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(\alpha_i x_i + (1 - \alpha_i) x). \tag{7}$$

where $\alpha_i \in (0, 1)$ is the personalized parameter of participant *i*. It is worth noting that when $\alpha_i = 0$, the learning objective and solution of CE-PFML is equal to the equation (1). For all $i \in [1, 2, \ldots, n]$, x_i is the minimized solution of f_i, namely x_i is the optimal base (local in FL) model of client *i*.

Different from the FML solving the problem presented in equation (1), CE-PFML realizes model personalization through the idea of mixture of local models and global model, and then solves equation (7). Specifically, it is realized via α here. Theoretically, the smaller α_i is, the smaller the influence of x_i over the meta-model (denoted as *MM*) is; otherwise, the larger α_i is, the greater the influence is. Therefore, CE-PFML enables more flexible and effective model personalization, while compared to Per-FedAvg. The ultimate goal of CE-PFML is to find a model (denoted as MM^*) that can generalize well on each novel client, formulated as follows:

$$MM^* = x^* = arg \min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(\alpha_i x_i + (1 - \alpha_i) x). \tag{8}$$

That is, the problem of equation (6) is converted into that of equation (8), following equation (2) to (5). And the final model deployed on novel client i is MM^* (namely x^*). As the subscriber of FML service, illustrated in Section III-B, the novel client hope to make classification predictions through optimal meta-model x^*. But its datasets are imbalanced and their distribution differs from that of participants. So x^* should be adapted to the local data distribution of the novel client by performing several steps of SGD over its training data based on x^*, and then make classification predictions on its test data.

The optimal base-model in step (2) of Section III-C satisfies the following equation,

$$x_i = arg \min_{x \in \mathbb{R}^d} f_i(x) \tag{9}$$

Algorithm 1 CE-PFML

Input: α, β
Meta-learner executes:
0: initializes model x^0 and sends it to all clients
1: **for** each round $r \in (0, R-1)$ **do**: //Outer-Loop update
2: $S_r \leftarrow$ (K clients of a random set)
3: **for** each client $i \in S_r$ **in parallel do**:
4: $g_i =$ **InnerLoop**(x^r, i, β);
5: After receiving α_i and x_i from client i, meta-learner
6: aggregate them by (6) to solve (8).
7: Optimal meta-model x^* is obtained after the learning process.
8: return x^*

InnerLoop(x^r, i, β):
9: **for** each inner-loop $t = 1, 2, \ldots, \tau$ **do**:
10: If marking parameter of distinguish whether $(\varepsilon, \sigma)-$DP is True:
11: Update base-model by (2) over its own samples with noise to solve (9).
12: else:
13: Update base-model by (2) over its own samples to solve (9).
14: return $\alpha_i, x^{r,\tau}$ to meta-learner

which indicates that the base-learner learn an converged based-model with the goal of minimizing the loss function over the local samples.

The pseudo code of CE-PFML is shown in Algorithm 1. Differential privacy noise is added in Line 10 and line 11, whose description is illustrated as following section. The specific process of CE-PFML is illustrated in Section III-C.

C. Global DP and Local DP in Meta-Learning Setting

In addition, to deal with the potential attacker mentioned in Section III-B, we introduce differential privacy (DP) for meta-learning [13], [34], [35], specifically, **Global DP** and **Local DP**. We assume that a training set $D = \{s_1, \ldots, s_i, \ldots, s_n\}$, where s_i is a sample in the training set.

1) *Global DP:* For any two datasets D, D' with at most one distinct element, $(\varepsilon, \delta)-DP$ is achieved by a randomized mechanism *M*, only if for all measurable sets $S \subseteq$ *Range*(*M*) we have:

$$\mathbb{P}[M(D) \in S] \leq e^{\varepsilon}\mathbb{P}\left[M\left(D'\right) \in S\right] + \delta. \tag{10}$$

If this holds for D, D' with at most *k* distinct elements, (ε, δ) *k-group DP* is achieved.

2) *Local DP:* For any two possible training samples $s, s' \in X \times Y$ and measurable sets $S \subseteq X \times Y$, $(\varepsilon, \delta)-$*local DP* is achieved by a randomized mechanism *M* only if the following formulation is satisfied:

$$\mathbb{P}[M(s) \in S] \leq e^{\varepsilon}\mathbb{P}\left[M\left(s'\right) \in S\right] + \delta. \tag{11}$$

Global DP guarantees the difficulty of inferring whether a particular sample exists in the training set by observing *M*(*D*). It assumes that a trusted aggregator running *M* with direct access to the dataset *D*, and the final output is privacy-protected. However, Local DP assumes more strictly that the aggregator is untrustworthy, and thus needs to apply a random mechanism separately over each sample *s* before training.

However, we cannot directly use the Global DP and Local DP as simply defined above due to the existence of a hierarchy of agents and statistical queries in meta learning. For each query, we can modify the procedure to satisfy either Local DP or Global DP. Therefore, we can get the following four options that satisfy the standard DP definition.

1) *Global DP:* The distribution of the global model θ^r will not leak information about any particular local model $\theta_{r,\tau}$;
2) *Local DP:* Guarantees that the meta-learner cannot obtain any private information from any local model $\theta_{r,\tau}$.
3) *Task-Global DP:* The distribution of the local model $\theta_{r,\tau}$ will no leak information about any particular sample $s_{r,i}$;
4) *Task-Local DP:* Guarantees that the task-owner cannot obtain any private information from any sample $s_{r,i}$.

D. Representation Learning Enhanced CE-PFML

To further improve the communication efficiency of CE-PFML, autoencoder (AE) [33], as a type of representation learning technique, is introduced. The autoencoder is composed of an encoder and a decoder. The encoder converts the input data into a hidden space through a deterministic mapping while the function of the decoder is to remap the space to the output data as close as possible to the input. In our work, after *q* rounds of learning process (called *Preparation Phase*), the server trains the AE using the models of the previous participants (called *Training Phase*). Once the training is completed, the encoder is sent to the participant, and then the participant can use the encoder to compress the locally trained model into a low-dimensional hidden space, upload it to the server to complete the model compression, and then server can decode the hidden space to obtain approximate local model (called *Compression Phase*). Whereas the AE is lossy compression, likewise, model compression after differential privacy noise addition is also lossy. As proved in [36], the same $\mathcal{O}(1/R)$ convergence rate of FedAvg under noise-free communication can be maintained as long as the variance of the error in the uplink and downlink decreases by $\mathcal{O}(1/r^2)$ and it is zero-mean. This means, when the following formula (18) is satisfied, after the training of AE is completed, the FL system switches to *Compression Phase*, otherwise, the FL system switches to *Training Phase*. In this way, the purpose of reducing communication overhead is achieved. The local models, which are inputs to the AE training process, are with differential privacy noise in this section.

The encoding and decoding process can be described as follows:

$$w_{en}^{i,r} = f_{en}\left(w^{i,r}\right) \tag{12}$$

$$w_{de}^{i,r} = f_{de}\left(w_{en}^{i,r}\right). \tag{13}$$

where $w^{i,r}$ is the local model of the participant $i \in n_r$ in the $r \in [0, R)$ communication round. f_{en} and f_{de} are the encoder and decoder, respectively. $w_{en}^{i,r}$ is the model compressed by the participant and sent to the server. $w_{de}^{i,r}$ is the output model decompressed by the server. For autoencoder training, global aggregation without autoencoder compression can be formulated as follows:

$$w^{r+1} = \frac{1}{\sum_{i \in n_r} D_i} \sum_{i \in n_r} D_i w^{i,r}. \tag{14}$$

and the new global model after using autoencoder is calculated by:

$$\hat{w}^{r+1} = \frac{1}{\sum_{i \in n_r} D_i} \sum_{i \in n_r} D_i w^{i,r}_{de}. \tag{15}$$

To approximate $w^{i,r}_{de}$ close to $w^{i,r}$, we train an autoencoder using the L_2-norm loss function for all participants $i \in n_r$ as follows:

$$L_2\Big(w^i, w^i_{de}\Big) = \Big\|w^i - w^i_{de}\Big\|^2 = \Big\|w^i - f_{de}\Big(f_{en}\Big(w^i\Big)\Big)\Big\|^2. \tag{16}$$

where $e^i = w^i - w^i_{de}$ represents the AE error of participant i in *Training Phase*. Further, we can define the AE error of participant i for each round r as follows:

$$e^{i,r} = w^{i,r} - w^{i,r}_{de}. \tag{17}$$

As described above, the convergence of autoencoder training in FL systems is guaranteed when the following conditions is satisfied:

$$\mathbb{E}\Big[e^i\Big] = 0, \mathbb{E}\Big\|e^i\Big\|^2 \leq e^r_{th} \leq \eta_r^2 \sim \mathcal{O}\bigg(\frac{1}{r^2}\bigg) \tag{18}$$

where η_r is the learning rate and $e^r_{th} \sim \mathcal{O}(\frac{1}{r^2})$ is a pre-fixed function [37]. To approximate the statistical values, we define:

$$\mathbb{E}\Big[e^i\Big] \simeq \frac{1}{q} \sum_{l=r-q}^{r} e^{i,l}, \mathbb{E}\Big\|e^i\Big\|^2 \simeq \frac{1}{q} \sum_{l=r-q}^{r} \mathbb{E}\Big\|e^{i,l}\Big\|^2. \tag{19}$$

For simplicity, in each communication round, we set the number of iterations for AE training to 100, once the performance of the AE model is within the acceptable error range of FL, the training process is completed.

V. Theoretical Analysis

Privacy Protection: For the outer-loop update, the samples are the model updates and the aggregator is the meta-learner, while for the inner-loop update, the samples are the records owned by base-learner and the aggregator is the base-leaner. Therefore, *Global DP* is implemented by the meta-learner, *Local DP* and *Task-Global DP* is implemented by the task-owner (namely base-learner for the within-task procedure), and *Task-Local DP* is implemented by the record-owner. Processing through *Task-Global DP* and *Task-Local DP* also protect the meta-model of subsequent iterators, which protect future task-owners as well. Therefore, we can implement *Task-Global* and *Sample-level* privacy through above four basic options.

Convergence Guarantee: We present the main theoretical results that our method converges to the global optimum at the rate of $\mathcal{O}(1/R)$, which is the same convergence performance as FedAvg, as follows. To simplify the analysis, we assume $K = N$, K clients are selected to participate inn FL and each participant run SGD for E epochs in this section. We employ the following assumptions that have also been commonly made in the literature [36], [38] as follows:

1) *L-Smooth:* $\forall v, w, F_i(v) \leq F_i(w) + (v-w)^T \nabla F_i(w) + \frac{L}{2}||v-w||^2$.
2) *μ-Strongly Convex:* $\forall v, w, F_i(v) \geq F_i(w) + (v - w)^T \nabla F_i(w) + \frac{\mu}{2}||v-w||^2$.
3) *Uniformly Bounded Gradient and Variance for Gradient:* $\mathbb{E}||\nabla F_i(w,\xi)||^2 \leq G^2$, and $\mathbb{E}||\nabla F_i(w,\xi) - \nabla F_i(w)||^2 \leq \delta_i^2$, for mini-batch data ξ at participant $i \in [K]$.

Theorem: Let the above assumptions 1) to 3) hold and L, μ, δ_i, G be defined therein. Choose $\phi = \frac{L}{\mu}$, $\gamma = \max\{8\phi, E\}$. Set the learning rate $\eta_r = \frac{2}{\mu(\gamma+r)}$. If the AE error scales such that:

$$\mathbb{E}\Big[e^i\Big] = 0, e^r_{th} \leq \eta_r^2 = \frac{4}{\mu^2(\gamma+r)^2} \sim \mathcal{O}\bigg(\frac{1}{r^2}\bigg), \forall i \in K. \tag{20}$$

Then, the convergence of our method with non-IID datasets and full clients participation satisfies:

$$\mathbb{E}\big[F\big(\hat{w}^R\big) - F(w^*)\big] \leq \frac{2LB}{\mu^2(\gamma+R)} + \frac{\gamma L}{2(\gamma+R)}\Big[\big\|w^0 - w^*\big\|^2\Big]. \tag{21}$$

where $B = \sum_{i=1}^{N} \frac{\delta_i^2}{N^2} + 6L\Gamma + 8(E-1)G^2 + \mathbb{E}||e_{th}||^2$, $F(w^*)$ is the minimum values of $F(w)$ and Γ is used to quantify the degree of non-IID [38].

Proof: Using the smoothness of F, we can formalize the gap as follows:

$$\mathbb{E}\big[F\big(\hat{w^r}\big) - F(w^*)\big] \leq \frac{L}{2}\mathbb{E}\|\hat{w}^r - w^*\|^2. \tag{22}$$

Using the results in [36], similar to [37], to handle the issue that the uplink errors from different participants are non independent, we bound the uplink error term as follows:

$$\begin{aligned}\mathbb{E}\big\|w^r - \hat{w^r}\big\|^2 = \mathbb{E}\|e^r\|^2 &= \frac{1}{N^2}\mathbb{E}\bigg\|\sum_{i \in N} e^{i,r}\bigg\|^2 \\ &\leq \frac{1}{N^2}\bigg[N^2\bigg\|\max_{i \in N}\mathbb{E}\Big[e^{i,r}\Big]\bigg\|^2\bigg] \leq e^r_{th}.\end{aligned}$$

The gap is given through [36] as follows:

$$\begin{aligned}\mathbb{E}\Big\|\hat{w}^{r+1} - w^*\Big\|^2 &\leq (1-\eta_r\mu)\mathbb{E}\|\hat{w}^r - w^*\|^2 + e^r_{th} \\ &\quad + \eta_r^2\bigg[\sum_{i=1}^{N}\frac{\sigma_i^2}{N^2} + 6L\Gamma + 8(E-1)G^2\bigg].\end{aligned}$$

Denote $\Delta_r = \mathbb{E}||\hat{w}^{r+1} - w^*||^2$. If we set $e^r_{th} \leq \eta_r^2$, we always have $\Delta_{r+1} \leq (1-\eta_t\mu)\Delta_r + \eta_r^2 B$. Set $\eta_r = \frac{\beta}{r+\gamma}$ for some $\gamma \geq 0$ and $\beta \geq \frac{1}{\mu}$ such that $\mu_0 \leq \min\{\frac{1}{\mu}, \frac{1}{4L}\} = \frac{1}{4L}$ and $\eta_r \leq 2\eta_{r+E}$. Next, it is easy to verify for $r = 1$ and prove that $\Delta_r \leq \frac{v}{\gamma+r}$, where $v = \max\{\frac{\beta^2 B}{\beta\mu-1}, (\gamma+1)\ \Delta_0\}$ as follows:

$$\begin{aligned}\Delta_{r+1} &\leq (1-\eta\mu)\Delta_r + \eta_r^2 B = \bigg(1 - \frac{\beta\mu}{r+\gamma}\bigg)\frac{v}{r+\gamma} + \frac{\beta^2 B}{(r+\gamma)^2} \\ &\leq \frac{v}{r+\gamma+1}.\end{aligned}$$

Using Δ_r in equation (22) and setting $r = R$, we can easily verify the assertion. ■

Communication Cost: There are R communication rounds of federated meta learning process. To measure the communication overhead, we define the compression rate of the autoencoder $C = \frac{|w|}{|w_{de}|}$, where $|w|$ and $|w_{de}|$ represent the communication model size of each round in the uncompressed and compressed states. We first present the communication volume of FedAvg and its variants as follows:

$$V_{FedAvg} = R * |w|. \tag{23}$$

Combined with the previous analysis, the communication volume of our method CE-PFML is calculated as follows:

$$V_{ours} = q|w| + (R - q)|w_{de}| = \left[\frac{R}{C} + \left(1 - \frac{q}{C}\right)\right]|w| \tag{24}$$

In CIFAR10 experiments, we set $R = 100$, $C = 128$, it is easy to obtain $V_{ours} \simeq [1 + (1 - \frac{q}{C})]|w| \simeq 2|w|$ and only if $\frac{q}{C}$ is close to zero. Theoretically, the communication cost of FedAvg and CE-PFML are $\mathcal{O}(R * |w|)$ and $\mathcal{O}(|w|)$, respectively. Therefore, CE-PFML effectively reduces communication overhead through autoencoder compression and improves communication efficiency.

VI. Performance Evaluation

A. Experimental Setup

The performance evaluation of the proposed CE-PFML algorithm is carried out on the machine of Ubuntu system, the graphics card is GeForce RTX 3090, and the PyTorch deep learning library is used. First, we study the performance of the resulting personalized meta-model; second, we test the communication overhead by tuning personalized parameters and leveraging local representation techniques.

We evaluate the empirical performance of CE-PFML on different models, tasks, and real-world federated dataset. MNIST [39], FEMNIST [40], and CIFAR10 [41] are used for experiments. The MNIST dataset is a widely used dataset for handwritten digit recognition, usually used for performance evaluation of image classification in the field of computer vision. There are 10 digit categories in this dataset, ranging from digit 0 to digit 9. The MNIST dataset contains 70,000 grayscale images with a resolution of 28 * 28, 60,000 of which are used for model training and the remaining 10,000 images are used for validation. The FEMNIST dataset is known as Federated-MNIST, and is a member of the benchmark dataset LEAF [40], which is dedicated to FL. It consists of 62,400 handwritten character images, belonging to 3,400 writers. The writers are grouped into non-overlapping subsets, where 2,800 writers are used for training and the remaining 600 writers are used for testing. Each image in the dataset is a 28x28 grayscale image of a handwritten character, and the characters include both digits and upper- and lower-case letters. The CIFAR10 dataset consists of ten 32x32 colour images of airplane, bird, cat, dog, etc., with 6000 images for each category. There are 50,000 training images and 10,000 test images. According to the complexity of the dataset and the actual performance of the model, we use a MLP model with two fully connected layers and ReLU activation for MNIST, and a CNN model composed of two 5 × 5 convolutional layers with ReLU activation, followed by two 2 × 2 pooling layers and three fully connected layers, for FEMNIST and CIFAR10, respectively.

TABLE III
Comparison of Average Test Accuracy of Different Algorithms Given α_i While Other Parameters Are Same, i.e., $\alpha = 0.01, \beta = 0.01$

Dataset	Parameters	FedAvg	Per-FedAvg	CE-PFML
MNIST	$\alpha_i = 0$	89.21%	**94.75%**	94.53%
	$\alpha_i = 0.2$	89.16%	94.68%	**95.75%**
	$\alpha_i = 0.5$	89.22%	94.70%	**95.80%**
	$\alpha_i = 0.8$	89.19%	94.71%	**96.36%**
FEMNIST	$\alpha_i = 0$	82.20%	**83.32%**	83.30%
	$\alpha_i = 0.2$	81.88%	83.30%	**85.43%**
	$\alpha_i = 0.5$	82.11%	83.27%	84.86%
	$\alpha_i = 0.8$	82.23%	83.29%	**85.29%**
CIFAR10	$\alpha_i = 0$	41.46%	**45.23%**	44.85%
	$\alpha_i = 0.2$	41.52%	43.04%	**47.88%**
	$\alpha_i = 0.5$	41.37%	**46.12%**	45.86%
	$\alpha_i = 0.8$	41.56%	46.63%	**48.95%**

B. Numerical Results

In the experiment, FedAvg [9], Per-FedAvg [11] algorithm is used as the baseline. Considering the limited resources in the network applications, we set $\tau = 1$. We take $K = 100$ clients in the network, and $R = 1000$.

Different α_i is set to verify the effectiveness of personalization in CE-PFML. At the same time, we ensure that other parameters are the same, and the corresponding accuracy of the algorithm is shown in Table III. The test accuracy of CE-PFML shows an overall increasing trend as α_i increases in the above three datasets, except when $\alpha_i = 0.5$, while the performance of FedAvg and Per-FedAvg is almost constant. We can assert that CE-PFML is can slightly improve the federated model's performance by adjusting the size of α_i. Combining with the experiment in Figure 2, we can claim that a larger α_i indicates a faster convergence of CE-PFML.

Then, the convergence of gradient descent and its dependence on α is tested. We refer to [42] and make a numerical analysis using a simple logistic regression to explore the effects of α_i on convergence. We assume that each client performs the following regularized logistic regression:

$$f_i(x) \overset{def}{=} \frac{1}{D_i}\sum_{j=1}^{D_i}\left[log\left(1 + exp\left(-a_{i,j}^T x\right)\right)\right] + \frac{\lambda}{2}||x||^2. \tag{25}$$

where λ is a parameter for regularization. It is clear that f_i is $\lambda-$smooth and $\lambda-$strongly convex. Here we set $\alpha_i = \alpha$ for each client i. The related experience result is shown in Figure 2, where loss is calculated by $f(x) - f^*$ and squared averaged distance is $\frac{1}{n}\sum_{i=1}^{n}||x_i - x_i^*||^2$. For the MNIST, FEMNIST and CIFAR10 datasets, the squared averaged distance and loss decreased as α increased, and the magnitude of the squared averaged distance and loss corresponding to different α decreased with the global number of rounds of SGD are almost consistent. It demonstrates that larger values of α can get better convergence, because we rely more on local optima, converging to MM^* faster.

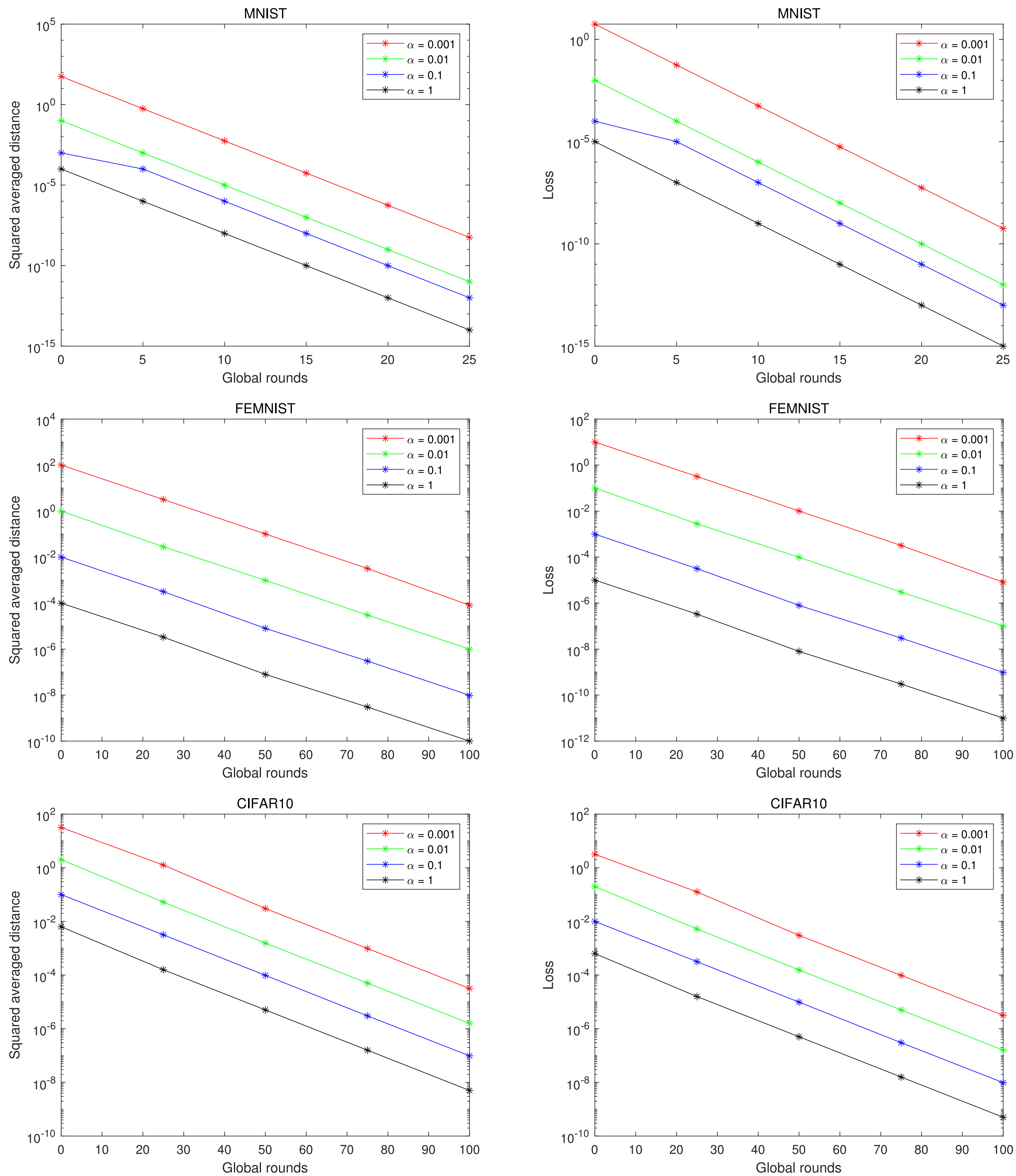


Fig. 2. Squared averaged distance and loss vs. number of global rounds of SGD for logistic regression with *l*2 regularizer.

In Figure 3, we set $\alpha = 0.01, \beta = 0.01, \alpha_i = 0.8$, the test accuracy of CE-PFML is higher than that of FedAvg and Per-FedAvg, and its performance is more stable than the latter two, which can be seen from the two subgraphs. As mentioned above, we set the size of α_i according to the correlation degree of different participants' data. If the data correlation is high, the CE-PFML model will perform better than other methods. In FEMNIST experiment as shown in the figure on the right, the performance of the CE-PFML model is unstable when the data heterogeneity of different participants is higher.

For simplicity, we randomly assign 10 equal parts of IID CIFAR10 data to participants, we try to compress 77% of parameters in over 80% of the communication rounds (the compression rate is 83%), while the accuracy of our model can reach 97% of the accuracy of the uncompressed model, just like the performance of model of the green line in Figure 4. By

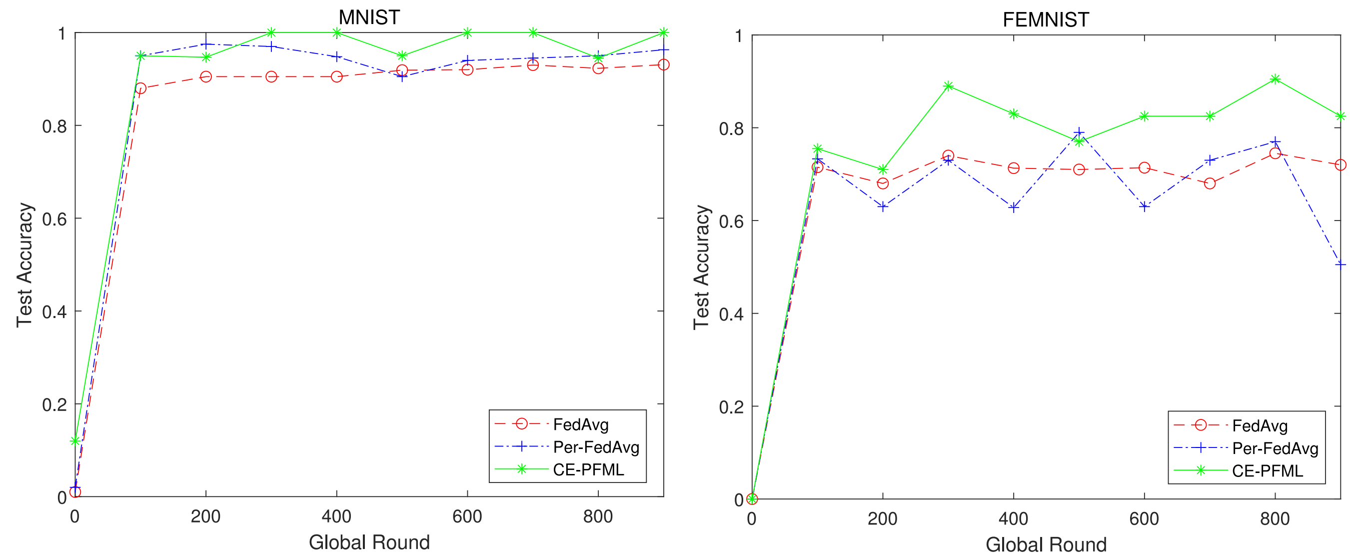


Fig. 3. Test Accuracy vs. number of global rounds for MNIST and FEMNIST. $\alpha_i = 0.8$.

TABLE IV
COMPARISON OF DIFFERENT ALGORITHMS FOR CIFAR10

Algorithms	Accuracy	Communication Cost
FedAvg	63%	$\mathcal{O}(R * \lvert w \rvert)$
Per-FedAvg	62%	$\mathcal{O}(R * \lvert w \rvert)$
pFedMe	59%	$\mathcal{O}(R * \lvert w \rvert)$
CE-PFML	61%	$\mathcal{O}(\lvert w \rvert)$

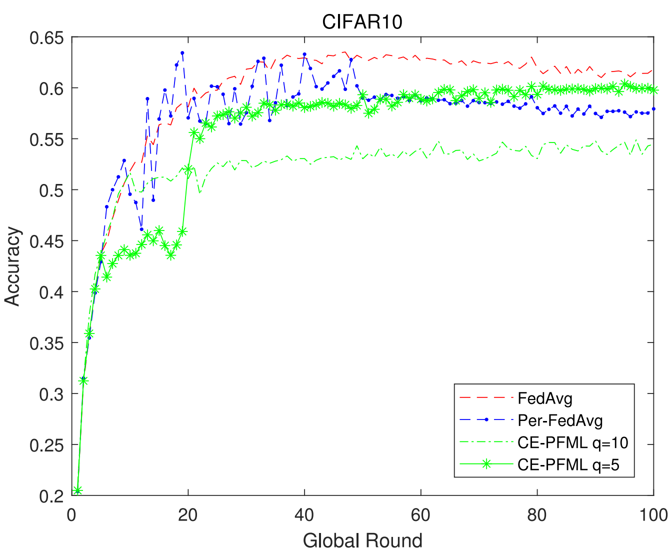


Fig. 4. Accuracy vs. number of global rounds for CIFAR10. $\alpha_i = 0.8$.

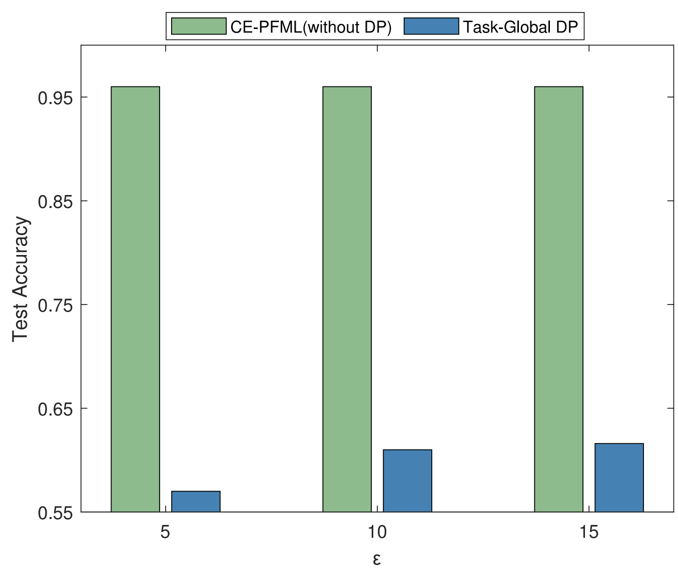


Fig. 5. Test Accuracy vs. ε in CE-PFML with or without Task-Global DP. $\alpha_i = 0.8$.

setting $q = 10$, we can found that *Preparation Phase* is longer than $q = 5$, which is not conducive to the performance of the final model. It can be verified from model performance of both green line in Figure 4. Moreover, as illustrated in Table IV, the communication cost of our method is greatly reduced by compression compared to FedAvg, Per-FedAvg, pFedMe [43], while maintaining the accuracy of model.

To verify the effectiveness of DP applied to CE-PFML, we use the Omniglot [44] dataset for few-shot image classification, specifically 5-ways-m-shots (i.e., $m = 5$, 10, 15). In Figure 5 and Figure 6, 5-ways-5-shots are used to compare the test accuracy performance of CE-PFML without DP, Task-Global DP version and Local DP version respectively. Obviously, the test accuracy drops a lot after using DP, but it is beneficial to both the sample-owner and the task-owner, because it can largely avoid malicious adversaries' access to private information. The results demonstrates that CE-PFML without DP has nothing to do with ε, further, as ε increases (i.e., 5, 10, 15), the accuracy of Task-Global DP and Local DP all show an increasing trend, and the accuracy seems to be close to the optimal (63% and 61%) when $\varepsilon = 10$. In addition, we also did a set of experiments with m-shots (m = 5, 10, 15), as shown in Figure 7, The results show that with the increase of m, the accuracy corresponding to DP version is improved, indicating that the more samples are added, the negative impact of noise on the accuracy can be reduced to a certain extent.

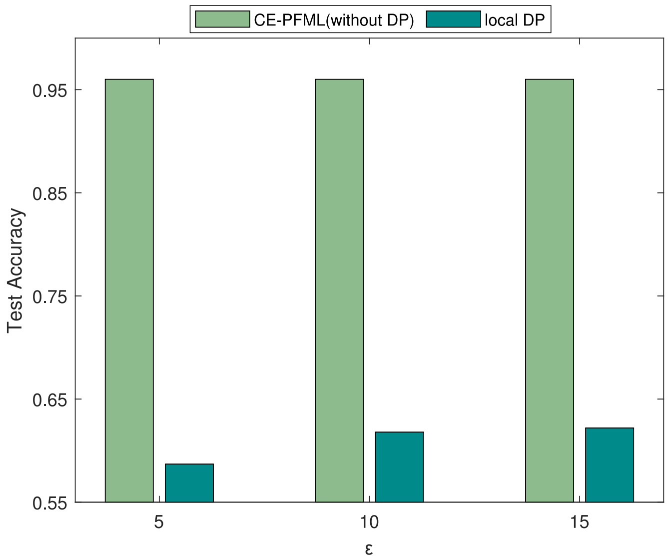


Fig. 6. Test Accuracy vs. ε in CE-PFML with or without Local DP. $\alpha_i = 0.8$.

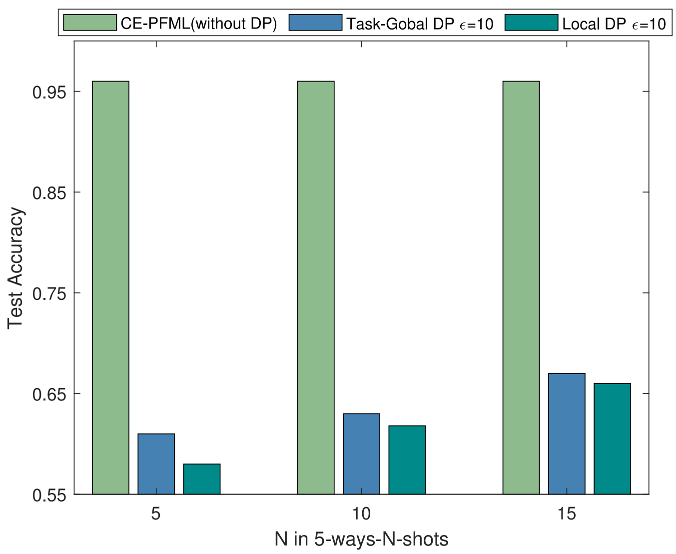


Fig. 7. Test Accuracy of CE-PFML without DP or with DP vs. N in 5-ways-N-shots. $\alpha_i = 0.8$.

VII. Conclusion

In this paper, there is the model personalization challenge in FL due to the goal of an "average" global model, as well as communication bottlenecks in edge networks. To handle the above issues, motivated by the mixture of global and local models, we describe the system model of federated meta-learning in edge networks, and propose the CE-PFML algorithm, which can obtain a novel personalized model to improve the accuracy and flexibly accelerate the convergence of the model by adjusting the size of the personalized coefficient. Further, the local model to be uploaded is transformed into the latent space through autoencoder, thereby reducing the amount of communication data, thereby reducing communication overhead, and Task-Global DP and Local DP are applied to provide privacy protection for model generation. Simulation experiments demonstrate that our method is more effective and efficient compared with several other algorithms. Therefore, CE-PFML can be used to obtain a novel personalized model to improve efficiency and reduce communication costs for different edge network applications.

References

[1] "Data age 2025: The datasphere and data-readiness from edge to core." 2021. [Online]. Available: https://www.i-scoop.eu/big-data-action-value-context/data-age-2025-datasphere/
[2] "Federated learning: Collaborative machine learning without centralized training data." 2017. [Online]. Available: http://ai.googleblog.com/2017/04/federated-learning-collaborative.html
[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.
[4] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," Jun. 2018, *arXiv:1806.00582*.
[5] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 1175–1191.
[6] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent IoT applications: A cloud-edge based framework," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 35–44, 2020.
[7] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," Sep. 2019, *arXiv:1909.12488*.
[8] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," Jul. 2017, *arXiv:1703.03400*.
[9] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," Feb. 2017, *arXiv:1602.05629*.
[10] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," Feb. 2021, *arXiv:2002.05516*.
[11] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A Meta-learning approach," Oct. 2020, *arXiv:2002.07948*.
[12] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.
[13] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, "Differentially private meta-learning," in *Proc. 8th Int. Conf. Learn. Represent.*, Apr. 2020, pp. 1–18.
[14] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 28, 2022, doi: 10.1109/TNNLS.2022.3160699.
[15] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "FedHome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2818–2832, Aug. 2022.
[16] M. Wazzeh, H. Ould-Slimane, C. Talhi, A. Mourad, and M. Guizani, "Privacy-preserving continuous authentication for mobile and IoT systems using Warmup-based federated learning," *IEEE Netw.*, early access, Aug. 8, 2022, doi: 10.1109/MNET.121.2200099.
[17] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *Proc. IEEE Conf. Comput. Commun.*, Toronto, ON, Canada, Jul. 2020, pp. 1698–1707.
[18] Z. Chai et al., "TiFL: A tier-based federated learning system," in *Proc. 29th Int. Symp. High-Perform. Parallel Distribut. Comput.*, 2020, pp. 125–136.
[19] S. AbdulRahman, H. Ould-Slimane, R. Chowdhury, A. Mourad, C. Talhi, and M. Guizani, "Adaptive upgrade of client resources for improving the quality of federated learning model," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4677–4687, Mar. 2023.
[20] M. Arafeh, H. Ould-Slimane, H. Otrok, A. Mourad, C. Talhi, and E. Damiani, "Data independent warmup scheme for non-IID federated learning," *Inf. Sci.*, vol. 623, pp. 342–360, Apr. 2023.
[21] M. Chahoud, S. Otoum, and A. Mourad, "On the feasibility of federated learning towards on-demand client deployment at the edge," *Inf. Process. Manage.*, vol. 60, no. 1, Jan. 2023, Art. no. 103150.
[22] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, Mar. 2020, pp. 429–450.
[23] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," Dec. 2019, *arXiv:1802.07876*.

[24] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3557–3568.

[25] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," Oct. 2019, *arXiv:1910.03581*.

[26] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," Dec. 2019, *arXiv:1912.00818*.

[27] P. P. Liang et al., "Think locally, act globally: Federated learning with local and global representations," Jul. 2020, *arXiv:2001.01523*.

[28] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[29] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.

[30] N. Tonellotto, A. Gotta, F. M. Nardini, D. Gadler, and F. Silvestri, "Neural network quantization in federated learning at the edge," *Inf. Sci.*, vol. 575, pp. 417–436, Oct. 2021.

[31] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," Jun. 2021, *arXiv:2105.10056*.

[32] D. Yao et al., "FedHM: Efficient federated learning for heterogeneous models via low-rank factorization," May 2022, *arXiv:2111.14655*.

[33] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[34] C. Dwork, "Differential privacy," in *Proc. 33rd Int. Colloq. Automata, Lang. Program.*, Venice, Italy, 2006, pp. 1–12.

[35] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2013.

[36] X. Wei and C. Shen, "Federated learning over noisy channels: Convergence analysis and design examples," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 1253–1268, Jun. 2022.

[37] M. Beitollahi and N. Lu, "FLAC: Federated learning with autoencoder compression and convergence guarantee," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 4589–4594.

[38] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int. Conf. Learn. Represent.*, Mar. 2020, pp. 1–26.

[39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradientbased learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[40] S. Caldas et al., "LEAF: A benchmark for federated settings," Dec. 2019, *arXiv:1812.01097*.

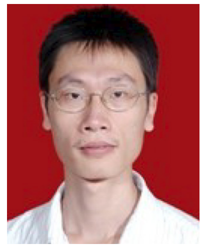
[41] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009. [Online]. Available: https://www.cs.toronto.edu/ kriz/learningfeatures-2009-TR.pdf

[42] E. Gasanov, A. Khaled, S. Horváth, and P. Richtárik, "FedMix: A simple and communication-efficient alternative to local methods in federated learning," in *Proc. 5th Workshop Meta-Learn. Conf. Neural Inf. Process. Syst.*, Dec. 2021, pp. 1–57.

[43] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with Moreau envelopes," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 1–12.

[44] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proc. Annu. Meeting Cogn. Sci. Soc.*, vol. 33, no. 33, 2011, pp. 1–7.

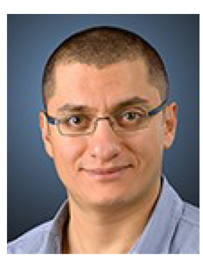
Feng Yu received the bachelor's degree in computer science and technology from the Hunan University of Science and Technology, China, in 2020. He is currently pursuing the master's degree with the College of Computer and Cyber Security, Fujian Normal University, Fuzhou, China. His research interests include federated learning, edge computing, and privacy protection.



Hui Lin received the Ph.D. degree in computing system architecture from the College of Computer Science, Xidian University, China, in 2013. He is currently a Professor with the College of Computer and Cyber Security, Fujian Normal University, Fuzhou, China, where he is also an M.E. Supervisor with the College of Computer and Cyber Security. He has published more than 50 papers in international journals and conferences. His research interests include mobile cloud computing systems, blockchain, and network security.

Xiaoding Wang received the Ph.D. degree from the College of Mathematics and Informatics from Fujian Normal University, Fuzhou, China in 2016, where he is an Associate Professor with the College of Computer and Cyber Security. His main research interests include network optimization and fault tolerance.

Sahil Garg (Member, IEEE) received the Ph.D. degree from the Thapar Institute of Engineering and Technology, Patiala, India, in 2018. He worked as a Research Professional with Resilient Machine Learning Institute, Montreal, QC, Canada. He has more than 80 publications in high-ranked journals and conferences, including more than 40 IEEE transactions/journal papers. He has many research contributions in the area of machine learning, big data analytics, security and privacy, the Internet of Things, and cloud computing. He received the IEEE ICC Best Paper Award in 2018, Kansas City, Missouri. He serves as the Managing Editor for *Human-Centric Computing and Information Sciences*. He is also an Associate Editor of the *Applied Soft Computing*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and *International Journal of Communication Systems*. In addition, he also serves as a Workshops and a Symposia Officer of the IEEE ComSoc Emerging Technology Initiative on Aerial Communications. He has guest-edited a number of Special Issues in top-cited journals, including the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE INTERNET OF THINGS JOURNAL, the IEEE NETWORK, and *Future Generation Computer Systems*. He serves/served as the Workshop Chair/Publicity Co-Chair for several IEEE/ACM conferences, including IEEE INFOCOM, IEEE GLOBECOM, IEEE ICC, and ACM MobiCom. He is a member of ACM.



Georges Kaddoum received the Ph.D. degree (Hons.) in signal processing and telecommunications from the National Institute of Applied Sciences, Toulouse, France, in 2008. He published over 200 journal and conference papers and two pending patents. He is the recipient of the "Research Excellence Award of the Université du Quebec, 2018" and the "Research Excellence Award-Emerging Researcher" from ÉTS in 2019. Additionally, he is a co-recipient of the Best Papers Awards of the IEEE PIMRC 2017 and the IEEE WiMob 2014. Moreover, he received the "Exemplary Reviewer Award" from IEEE TRANSACTIONS ON COMMUNICATIONS in 2015 and 2017. He is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and the IEEE COMMUNICATIONS LETTERS. He held the ÉTS Research Chair in physical-layer security for wireless networks.

Satinder Singh (Member, IEEE) received the B.Eng. degree in wireless communications, digital signal processing from École de technologie supérieur in 2003. He is the Data Science Director of REMI Resilient Machine Learning Institute and the Director Advanced System with Ultra Electronics Communication. At Ultra communication, his earlier work involved development of wireless communications hardware and waveforms for tactical backhaul network presently deployed with U.S. Army TRILOS Program. Since taking leadership of REMI, his interests are aligned with AI/ML development for modern tactical network: situational and spectrum awareness, development of AI/ML empowered waveforms, AI/ML assisted Auto-PACE, and Ease of Use and end user acceptance of machine learning algorithms.

Mohammad Mehedi Hassan (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Kyung Hee University, Seoul, South Korea, in February 2011. He is currently a Full Professor with the Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He has authored and coauthored around over 300 publications, including refereed IEEE/ACM/Springer/Elsevier journals, conference papers, books, and book chapters. His research interests include edge/cloud computing, the Internet of Things, cyber security, deep learning, artificial intelligence, body sensor networks, 5G networks, and social networks.