

A Federated Meta-Learning Aided Intelligent Edge Framework by Using the Parameter Optimization Approach

Xiaofeng Zhu

Xinyu University

Qiaosong Fan

South China University of Technology

Jiaqiang Peng

Xinyu University

Yuwen Qian

`admon@njust.edu.cn`

Nanjing University of Information Science and Technology

Research Article

Keywords: Edge intelligence, privacy protecting, federated learning, meta learning, parameter optimization

Posted Date: June 13th, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-6819153/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at EURASIP Journal on Wireless Communications and Networking on October 15th, 2025. See the published version at <https://doi.org/10.1186/s13638-025-02511-7>.

A Federated Meta-Learning Aided Intelligent Edge Framework by Using the Parameter Optimization Approach

Zhu Xiaofeng¹, Fan Qiaosong², Peng Jiaqiang^{1*}, Qian Yuwen^{3*}

¹School of Mechatronics Engineering, Xinyu Key Laboratory of Power System and Intelligent Control, Xinyu University, Xinyu, 338004, China.

²Shien-ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou, 510641, China.

^{3*}School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China.

*Corresponding author(s). E-mail(s): Jiaqiang_Peng@163.com;
admon@njjust.edu.cn;

Contributing authors: Xiaofeng_Zhu@163.com; Qiaosong_Fan@163.com;

Abstract

Edge intelligence can enable fast intelligent services by integrating edge computing with machine learning, thereby facilitating intelligent information processing for Internet of Things (IoT) devices on the edge. However, intelligent data processing at the edge may expose IoT devices to the risk of private information leakage. To mitigate this issue, we propose a federal meta-learning-aided data processing framework to cope with complex tasks in edge IoT networks. Unfortunately, communications between edge IoT devices and edge servers in federated frameworks incur significant overhead. To address this challenge, we propose a parameter optimization algorithm that alleviates communication costs between edge IoT devices and edge servers, thereby reducing classification errors induced by parameter optimization. Moreover, the convergence of the federated meta-learning method is derived, which theoretically confirms the feasibility of the proposed approach. Simulation results demonstrate that the error minimization-based quantization compression optimization algorithm can substantially enhance communication efficiency while incurring only negligible precision losses.

Keywords: Edge intelligence; privacy protecting; federated learning; meta learning; parameter optimization;

1 Introduction

The rapid advancement of artificial intelligence (AI) has significantly accelerated the development of machine learning (ML) and deep learning techniques. Leveraging ML methods, which excel in robust data processing and feature extraction, within network edge devices enables more efficient handling of massive datasets [1]. In addition, the data collected by network edge devices also provides an important opportunity for the development of ML. In recent years, various ML models capable of supporting complex tasks and processing heterogeneous data have been steadily refined and matured [2]. Consequently, a critical research challenge lies in integrating ML with edge computing to design effective edge-intelligent data processing frameworks. However, the deployment of edge intelligence at the network edge raises important security concerns.

To safeguard data privacy, the federated learning model is investigated to provide a secure solution by enabling distributed model training across multiple network edge terminals [3, 4]. The federated learning was initially proposed by McMahan et al., who demonstrated the efficacy and feasibility of collaborative training a global model in a decentralized manner without requiring data owners to exchange their respective datasets [5]. In contrast to traditional distributed learning approaches that rely on data centers [6, 7], federated learning circumvents centralized data processing, achieving a decentralized architecture. Notably, when dealing with non-independent and identically distributed (Non-IID) data, federated learning algorithms often experience a decline in accuracy and slower convergence rates [8]. To address these issues, the construction of a globally shared data set across all edge devices was proposed, where even a limited amount of shared data can significantly improve classification performance [9]. Similarly, a hybrid learning mechanism and a heuristic algorithm were developed in [10] to address the selection of clients and data during training, thus increasing the number of clients participating in federated learning. This approach ensures that edge terminals upload only a small amount of independent and identically distributed (IID) local data to the central server, which helps improve the accuracy of the aggregated model. However, the aforementioned approaches of sharing local data are impractical in practical systems, as not all edge terminals are willing to share their local data.

In this context, meta-learning, focusing on training models on prior tasks to acquire well-initialized model parameters, was subsequently exploited for the classification of data from novel tasks [11]. This approach not only facilitates rapid adaptation to new tasks but also mitigates the issue of overfitting, enabling effective learning from limited samples. The core objective of meta-learning is to leverage prior experience to efficiently learn new tasks [12]. Initially, meta-learning was first articulated in [13] and [14], which is often characterized as "learning to learn." Therefore, meta-learning has evolved into three categories: metric-based meta-learning, model-based

meta-learning, and optimization-based meta-learning [15]. The foundation of metric-based meta-learning lies in learning a feature space network that captures the essence of each class within a dataset [16, 17]. By evaluating task-specific training data against these abstracted classes in the feature space, quantifiable similarity metrics are derived, enabling the model to classify task data based on this information.

Conversely, model-based meta-learning strategies incorporate an additional "memory module" designed to store critical samples from the training data, thereby augmenting neural network learning [18]. A quintessential example is the memory-augmented neural network (MANN), which utilizes content-addressable memory access and an enhanced neural network architecture to adeptly handle classification tasks with limited sample sizes [19]. In contrast, optimization-based meta-learning methods focus on deriving a sensitive initialization of parameters through initial training phases, followed by fine-tuning these parameters on new task datasets to achieve rapid adaptation. The model-agnostic meta-learning (MAML) algorithm, introduced by Finn et al., exemplifies this approach [20]. To address the computational inefficiencies associated with second-order derivatives in MAML, reptile, a first-order meta-learning algorithm was proposed to circumvent the need to solve Hessian matrices, thus significantly improving computational efficiency [21]. Subsequent advances have introduced proximal regularization terms to further accelerate convergence and boost classification accuracy [22, 23]. With the rapid progress of machine learning, the growing complexity of neural network models, characterized by a large number of parameters, imposes substantial demands on the storage and computational capacities of devices.

To mitigate the computational burden associated with transmitting network models, model compression techniques can be applied to streamline the transmitted parameters. Currently, model compression is primarily achieved through methods such as network pruning, model quantization, knowledge distillation, and low-rank decomposition [24]. Network pruning seeks to reduce computational requirements during the training process by removing unimportant structures and parameters within the neural network [25]. Depending on the granularity of the pruning process, network pruning can be further categorized into structured and unstructured pruning. Model quantization, conversely, compresses the model by decreasing the number of bits required to represent each weight parameter [26]. Then, vector quantization is applied for compression in deep convolutional networks by focusing specifically on densely connected layers, which require the greatest storage compression [27]. Although model quantization can significantly reduce communication overhead, it often entails a trade-off in accuracy. To address this issue, knowledge distillation is employed to transfer knowledge from a pre-trained complex "teacher network" to a simpler "student network", enabling the student network to perform nearly as well as the teacher in real-world applications. According to this idea, convolutional operations are leveraged to convey the teacher's knowledge and translate it to the students. The results demonstrate that the student network trained via this factor transfer method outperforms the student network trained using the conventional knowledge transfer method [28]. Additionally, a knowledge transfer approach is proposed to align the probability distributions of

the data in the feature space, thereby facilitating the distillation of knowledge from a complex network to a simpler one [29].

In this paper, we have proposed a federated meta-learning framework that integrates federated learning with meta-learning, enabling the global model to be compressed locally for rapid and accurate classification across diverse tasks. The main contribution of this paper is listed below.

- We propose an intelligent edge framework for IoT networks by combining federated and meta learning, where model parameters for different tasks can be optimized for fast and accurate classification
- We design a meta-learning aided federated learning algorithm to achieve edge intelligence by using the global model parameters, which are obtained by optimizing local data from edge IoT devices with the meta-learning approach. As a result, an effective balance between communication overhead and model performance can be achieved.
- The convergence of the federated meta-learning method is derived by mathematical analysis, which theoretically confirms the feasibility of the proposed federated meta-learning framework.

The rest of this article is organized as follows. In Section II, we describe the federated learning method for data processing at edge IoT networks. In Section III, we present the proposed federated meta-learning framework. In Section IV, we derive the convergence of the proposed algorithm. In Section V, we present the numerical results.

2 Methods/Experimental

This study introduces a meta-learning-aided federated learning approach for IoT systems to train a global model parameter, enabling each IoT device to achieve fast and accurate classification in local data based on this global parameter. In the approach, federated learning enables model training to be performed in a distributed manner across multiple edge terminals and achieves user privacy protection by exchanging only the intermediate computation results of the edge terminals. Furthermore, meta-learning aims to improve the generalization ability of the global model parameters. As a result, it can be fine-tuned in the local data of each edge terminal to achieve fast and accurate classification. For the convergence of the federated meta-learning method, it is theoretically derived and analyzed to prove the theoretical feasibility of the method.

2.1 Workflow Overview

In this paper, we first design a meta-learning aided federated architecture for edge IoT system for fast edge computing to solve the problem of performance degradation when training NON-IID data by federated learning. The core of our approach is the meta-learning-aided federated architecture, where each IoT device divides the local training into internal and external updates based on the meta learning method and then updates the local model parameters through external updates. After the local training is completed, it is uploaded to the edge server for global aggregation updating. The global model parameters with generalization ability are obtained after global

updating iterations, which are fine-tuned based on local data of each IoT device to achieve fast and accurate classification results.

Therefore, the goal of this model is to obtain a good global model parameter trained by a meta-learning module under the framework of federated learning, so that it can be adapted quickly after fine-tuning in the local data of each edge terminal. In simulation experiments, the effectiveness of the method is verified by comparing it with the federated learning method.

2.2 Key Features

Cooperative intelligent edge computing: By cooperatively training the global model and obtaining its parameters, each edge IoT device can provide edge computing service by enabling fast and accurate classification of the local data.

Privacy protection The proposed meta-learning-aided federated learning framework can enable model training to be performed in a distributed manner across IoT devices. To protect the privacy of each IoT device, only the intermediate computation results of the edge IoT devices are exchanged during the model training.

Generalization ability adaptability By optimizing the parameters with the meta-learning method, the obtained classification model can exhibit a strong generalization capability. Especially, the proposed meta-learning-aided federated learning framework is well-suited for federated learning scenarios with non-IID data.

The effectiveness of the method is verified by comparing and analyzing the performance of different methods based on the MNIST dataset and the more complex CAFIR-10 dataset at IID and NON-IID in simulation experiments.

3 System Model

As illustrated in Fig. 1, the proposed system with federated learning architecture is composed of an edge server and N IoT devices as devices S . In each iteration, a random subset of M devices is selected for global aggregation and model updates. Each device trains a local model using their local respective dataset, denoted as $D_1, D_2, D_3, \dots, D_i, \dots, D_M$ where $i \in S$. The locally updated model parameters are then transmitted back to the central server, which performs global aggregation to consolidate the updates. This iterative procedure enables decentralized collaborative training across edge devices while preserving data privacy by eliminating the direct exchange of raw local datasets.

3.1 Federated Learning Process

During the t -th round of global updates, the central server distributes the globally aggregated model parameters θ_{t-1} , resulting from the previous round, to all participating devices. If it is the initial round of global updates, the central server instead sends the randomly initialized model parameters θ_0 to each device. Then, a random subset of M edge devices is selected to participate in local training. Each selected device performs local training based on the model parameters received from the central server. Specifically, local training involves updating the model using the device's

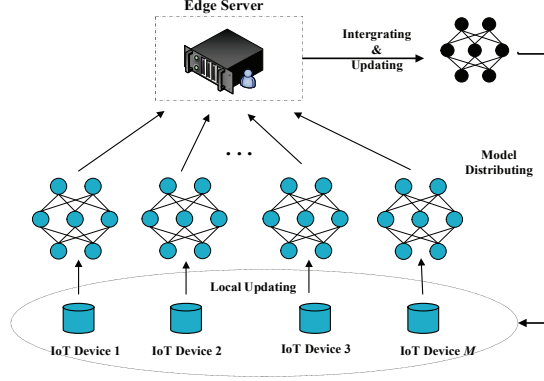


Fig. 1 A typical federated learning architecture for the IoT network at the edge, where there is one edge server and M IoT devices.

private data, which is stored on the device. For each update, mini-batch stochastic gradient descent (SGD) is utilized. A mini-batch of size B is uniformly sampled from the local training dataset. Furthermore, the model parameters are iteratively updated during the gradient descent process to minimize the loss function defined for the local dataset. Thus, the local update for device i can be given by

$$\theta_t^i = \theta_{t-1}^i - \eta \nabla F(\theta_{t-1}^i, b), \quad (1)$$

and

$$\theta_{t-1}^i \leftarrow \theta_t^i, \quad (2)$$

where θ_t^i represents the model parameters obtained by device i after the current local update and θ_{t-1}^i is the model parameters obtained by device i after the previous local update.

If this is the first local update, θ_{t-1}^i corresponds to the global model parameters θ_{t-1} from the previous global update. The symbol η denotes the learning rate, while $\nabla F(\theta_{t-1}^i, b)$ represents the gradient of the loss function $F(\theta_{t-1}^i, b)$ evaluated at θ_{t-1}^i using the mini-batch data b . In (2), the parameter θ_t^i is updated to θ_{t-1}^i , thereby iteratively updating the newly sampled mini-batch data.

In general, the loss function quantifies the difference between the output obtained from the input data, model parameters, and the actual data output, serving as a metric to evaluate the model's performance. In this context, the loss function is given by

$$F(\theta, D) = \frac{1}{|D|} \sum_{(x^j, y^j) \in D} l(\theta, (x^j, y^j)), \quad (3)$$

where D represents the training dataset and $|D|$ denotes the size of the dataset. Moreover, the function $l(\theta, (x, y))$ represents the cross-entropy loss function, where $(x^j, y^j) \in D$ denotes the j -th data point in the dataset D . x^j is the matrix that

represents the image after grayscale processing, while y^j denotes the image category. In addition, θ is a model parameter.

During the t -th round of global updates, after completing their local training updates, each device uploads its local model parameters to the central server for global aggregation, given by

$$\theta_t = \sum_{i \in M} \omega_i \theta_t^i, \quad (4)$$

where θ_t represents the global model parameters after the t -th global update aggregation, and θ_t^i denotes the local model parameters of device i after performing E rounds of local updates. ω_i represents the weight of the local model of i in the global model aggregation update, which corresponds to the proportion of the device's data size relative to the total data size from all participating devices, given by

$$\omega_i = \frac{|D_i|}{\sum_{i \in M} |D_i|}, \quad (5)$$

where $|D_i|$ represents the size of the device's local dataset, and $\sum_{i \in M} |D_i|$ denotes the total size of the datasets from all devices participating in the global update.

After T rounds of global update iterations, the final global aggregated model parameters θ are obtained at the central server. Thus, the optimal objective of federated learning, denoted as θ^* , can be given by

$$\theta^* = \arg_{\theta} \min \sum_{i \in M} \omega_i F(\theta_i). \quad (6)$$

In federated learning, when device data is NON-IID, both the convergence speed of the FedAvg algorithm and the classification accuracy degrade significantly. As a result, the globally aggregated model parameters fail to achieve satisfactory performance on the local datasets of individual devices. To address this limitation, we enhance the Federated Learning algorithm so that the trained model parameters, denoted as θ , can achieve superior performance on the local task data after K steps of gradient updates. Mathematically, the problem can be formulated as

$$\begin{aligned} \min_{\theta, \{\varphi_i\}_{i=1}^N} & \frac{1}{N} \sum_{i=1}^N F(\varphi_i^K(\theta), D_i^t), \\ \text{s.t.} & \varphi_i^K(\theta) = \text{SGD}_K^\alpha(\theta, F(\varphi_i(\theta), D_i^{tr})), \forall i \end{aligned} \quad (7)$$

where θ represents the final model parameters obtained after global updates, $\{\varphi_i\}_{i=1}^N$ denotes the local model parameters of the N global devices, D_i^t denotes the local test set data of i , D_i^{tr} means the local training set data of i , $\varphi_i^K(\theta)$ denotes the model parameters of i after performing K steps of SGD based on the local training set data starting from θ , and α is the learning rate.

3.2 Meta-Learning Aided Federated Learning Approach

As discussed in Section II, the performance of the federated learning architecture significantly declines when device data exhibits NON-IID. Specifically, the model parameters derived from global updates fail to achieve rapid and accurate classification across the heterogeneous data distributed among global devices. Thus, in this section, we adopt a meta-learning approach to enhance the generalization capability of the global model, as illustrated in Fig. 2.

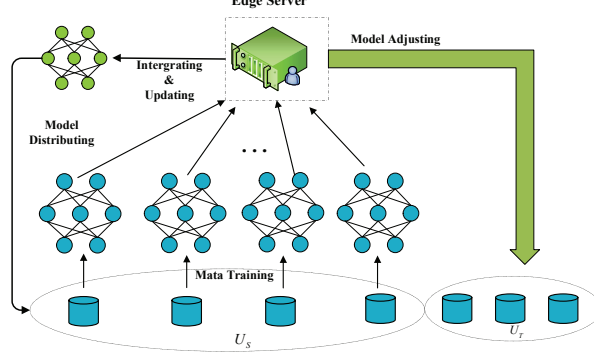


Fig. 2 The framework of the proposed meta-learning aided federated approach.

The core principle of meta-learning lies in optimizing the model parameters that enable rapid adaptation to new tasks. Specifically, these parameters allow the classification model to achieve robust performance on novel tasks using limited data, after just one or a few gradient update steps. In this paper, we build the meta-learning framework in two phases: meta-training and meta-testing.

We let the initial model parameters be denoted as ϕ , and we assume that a set of meta-training tasks $\{T_i\}_{i=1}^N$ are sampled from a known data distribution. Each task $\{T_i\}_i$ is divided into two disjoint sets: a support set D_i^S and a query set D_i^Q . With D_i^S and ϕ , the model parameters ϕ' can be optimized via gradient descent, which corresponds to the internal update, given by

$$\phi'_i = \phi - \alpha \nabla_{\phi} F(\phi, D_i^S), \quad (8)$$

where α is the learning rate for internal updates. The loss function $F_i(\phi'_i, D_i^Q)$ for each task $\{T_i\}_i$ is computed in the query set D_i^Q . Thus, the meta-parameters ϕ are updated based on the loss value corresponding to the outer update. Then, the gradient for the outer update can be given by

$$\phi = \phi - \beta \nabla_{\phi} \sum_{i=1}^N F_i(\phi'_i, D_i^Q), \quad (9)$$

where $\nabla_{\phi} \sum_{i=1}^N F_i(\phi'_i, D_i^Q)$ represents all task losses cumulative and gradient values based on the model meta-parameter ϕ , and β is the learning rate for the external update.

We let T_j represent meta-testing. Then, ϕ , obtained from the meta-training, is updated using one or several steps of gradient descent. This allows for improved classification performance on T_j , as expressed by

$$\phi'_j = \phi - \alpha \nabla_{\phi} F(\phi, T_j). \quad (10)$$

3.3 Parameters Optimization Algorithm

Building on the previous analysis, it is evident that the parameters optimized by the meta-learning method can exhibit strong generalization capabilities, leading to particularly well-suited for federated learning scenarios with non-IID data. In this context, IoT devices can perform meta-training on their local data and subsequently upload the model parameters to an edge server for global aggregation. By iteratively repeating this process, a robust global model can be collaboratively constructed. In addition, each IoT device can compress the aggregated global model locally, thereby adapting it to its specific data distribution. This method can effectively integrate the strengths of meta-learning and federated learning, enabling rapid and accurate classification across heterogeneous local datasets from diverse terminal devices.

In the proposed meta-learning aided federated edge framework, there are two primary phases, i.e., federated meta-training and federated meta-testing. In the federated meta-training phase, two IoT device sets are created, which are the federated meta-training set, denoted as U_S , and the federated meta-testing device set, denoted as U_T . The global model parameters obtained from the federated meta-training with U_S are compressed in U_T , leading to improved classification performance. This indicates that the global model parameters exhibit good adaptability to the global device data. During the meta-training process, the edge server first distributes the global model parameters from the previous round to U_S . If it is the first update, the server instead distributes the initialized global parameters. Then, M IoT devices are randomly selected from U_S to perform E local updates. After completing the local training, the updated local model parameters are uploaded to the central server, where they are aggregated and used to update the global model parameters. This global aggregation process is repeated over T iterations, resulting in the final global model parameters from federated meta-training. In the federated meta-training process, U_T performs one or more steps of gradient descent based on the final global model parameters. As a result, the model parameters, after optimization, can achieve better classification performance with U_T .

The detailed process is presented in Algorithm 1.

Algorithm 1 MAML-based Parameters Compress and Optimization Algorithm

Require: Initial global model parameter θ_0 , number of global update rounds T , number of local updates E , learning rates α and β , number of meta-training devices $M \subseteq U_S$, support set D_i^S , query set D_i^Q

Ensure: Federated meta-trained global model parameters θ_T

- 1: Initialize the global model parameter θ_0
 - 2: Select a group of global devices as federated meta-training devices U_S
 - 3: **for** $t = 1$ to T **do**
 - 4: Distribute global model parameter θ_{t-1} to all devices in U_S
 - 5: Randomly select M devices from U_S
 - 6: **for** each device $i \in M$ **do**
 - 7: Use θ_{t-1} as the initial parameter for local training
 - 8: Set $\theta_{t-1}^{(i)} \leftarrow \theta_{t-1}$
 - 9: **for** $e = 1$ to E **do**
 - 10: Compute adapted parameters with $\phi_t^{(i)}(\theta) \leftarrow \theta_{t-1}^{(i)} - \alpha \nabla F(\theta_{t-1}^{(i)}, D_i^S)$
 - 11: Update local model with $\theta_t^{(i)} \leftarrow \theta_{t-1}^{(i)} - \beta \nabla F(\phi_t^{(i)}(\theta), D_i^Q)$
 - 12: $\theta_{t-1}^{(i)} \leftarrow \theta_t^{(i)}$
 - 13: **end for**
 - 14: **end for**
 - 15: Aggregate updated models at the central server with $\theta_t \leftarrow \frac{1}{|M|} \sum_{i \in M} \theta_t^{(i)}$
 - 16: Obtain updated global model θ_t
 - 17: **end for**
 - 18: Obtain final federated meta-trained global model θ_T
 - 19: Randomly select a group of devices as federated meta-test devices U_T
 - 20: Use θ_T as initial parameters for each test device $j \in U_T$
 - 21: **for** each test device $j \in U_T$ **do**
 - 22: Perform K steps of local adaptation:
$$\theta_j^* \leftarrow \theta_T - \alpha \nabla F(\theta_T, D_j^S)$$
 - 23: $\theta_T \leftarrow \theta_j^*$
 - 24: **end for**
-

Algorithm 1 is detailed as follows. In Algorithm 1, we define the following parameters, including that p is the proportion of the support set for each device's data, T is the total number of global update training rounds, E is the number of local updates, and M is the number of devices participating in the aggregation of federated meta-training in each round. In addition, the learning rate of the internal update is set as α and the learning rate of the external update is set as β .

Distribute the global data and split each device's local data into two subsets: the support set D_i^S and a query set D_i^Q . For the t -th global update of the federated meta-learning process, the global model parameters θ_{t-1} are sent to U_S , followed by the random selection of M devices from U_S for training. Each selected device $i \in M$

initializes local model parameters for training using the global parameters received, such that $\theta_{t-1}^i \leftarrow \theta_{t-1}$.

In particular, the local training update is listed below.

Internal Update: an internal update with a learning rate of α is performed based on the support data set of θ_{t-1}^i and device i to obtain the parameter ϕ_t^i .

External Update: an external update with a learning rate of β is performed based on the query set data of ϕ_t^i to obtain the parameter θ_t^i .

Iteration Cycle: the value of θ_t^i is assigned to θ_{t-1}^i to carry out the iteration cycle.

Each device performs updates to complete the local training and transmits the updated parameters to the central server for global aggregation updates. After T rounds of global update, the model parameters θ_T can be obtained. These parameters are then adapted to local data by performing K gradient descent steps during the federated meta-test phase for device U_T .

4 Convergence Analysis

In this section, we first explore the relevant properties of the objective function in federated meta-learning, focusing on the characteristics of the loss function. Then, we investigate the impact of task similarity across different devices on the convergence performance of federated meta-learning. Finally, we analyze the convergence of the federated meta-learning algorithm.

4.1 Analysis of Loss Function and Objective Function

To characterize the convergence of the federated meta-learning algorithm, we first present the properties of the loss function, denoted by $F(\theta)$. With these properties, the properties of the objective function, denoted by $G(\theta)$, can be obtained.

4.1.1 Loss Function Selection

In the internal update phase of federated meta-learning, the loss function $F_i(\theta)$ for the i -th device is defined as

$$F_i(\theta, D_i^S) = \frac{1}{|D_i^S|} \sum_{i \in U_S} F_i(\theta, D_i^S) \quad (11)$$

The expected loss function of the internal update of all devices is given by

$$F_w(\theta) = \sum_{i \in U_S} \omega_i F_i(\theta). \quad (12)$$

In the external update phase of federated meta-learning, the loss function for the i -th device $F_i(\phi_i)$ is

$$F_i(\phi_i, D_i^Q) = \frac{1}{|D_i^Q|} \sum_{i \in U_S} F_i(\phi_i, D_i^Q) \quad (13)$$

Since ϕ is a function related to θ , it can be denoted as $G_i(\theta)$. The expected loss of U_S external updates is

$$G(\theta) = \sum_{i \in U_S} \omega_i G_i(\theta), \quad (14)$$

where ω_i denote the ratio of data volume of device i to the total data volume of all devices.

4.1.2 Properties of Objective Function

We have the following properties for the objective function.

Property A: $\langle \nabla F_i(\theta) - \nabla F_i(\theta'), \theta - \theta' \rangle \geq \mu \|\theta - \theta'\|$

Property B: $\|\nabla F_i(\theta) - \nabla F_i(\theta')\| \leq H \|\theta - \theta'\|$ and there exists a constant B , satisfying $\|\nabla F_i(\theta)\| \leq B$, where B represents the batch size of samples.

Property C: $\|\nabla^2 F_i(\theta) - \nabla^2 F(\theta')\| \leq \rho \|\theta - \theta'\|$

Property D: Given the constants δ_i and σ_i , it holds that $\|\nabla F_i(\theta) - \nabla F_w(\theta)\| \leq \delta_i$, $\|\nabla^2 F_i(\theta) - \nabla^2 F_w(\theta)\| \leq \sigma_i$

With the properties A and B, it can be ensured that the loss function is smooth and bounded. On the other hand, property C guarantees higher-order smoothness. In addition, Property D qualifies the degree of similarity among devices, where the constants δ_i and σ_i are used as the upper bounds on the gradient and Hessian variance of the device loss functions. Therefore, a smaller value of these upper-bound constants indicates higher data similarity among devices. These constants can be adjusted over a wide range to capture more diverse data scenarios.

With the property of $G(\theta)$, we present the properties of Algorithm 1 as Theorem 1.

Theorem 1 $G(\theta)$ is μ' -strongly convex and H' -smooth when

$$\alpha \leq \min \left\{ \frac{\mu}{2\mu H + \rho B}, \frac{1}{\mu} \right\}, \quad (15)$$

where

$$\mu' = \mu(1 - \alpha H)^2 - \alpha \rho B, \quad (16)$$

and

$$H' = H(1 - \alpha \mu)^2 + \alpha \rho B. \quad (17)$$

Proof See Appendix A. □

From Theorem 1, when the internal learning rate α is relatively small, the objective function $G(\theta)$ is smooth and bounded.

4.2 Task Similarity for Convergence

In this subsection, we investigate the impact of task similarity among different devices on the convergence of federated meta-learning. In particular, we present Theorem 2 to illustrate the convergence of federated meta-learning under the task similarity.

Theorem 2 *Given the loss function gradient and the bounded variance of the Hessian in the case of heterogeneity of data from devices, the upper bounds on the variance of the gradient of $G(\theta)$ exist.*

Proof See Appendix B. □

4.3 Convergence of the Proposed Algorithm

According to [30], we consider a virtual sequence $z_{[n]}^t$, which performs global aggregation at each local update, where $[n]$ represents the interval $[(n-1)E, nE]$, $t \in [(n-1)E, nE]$. Thus, we have

$$z_{[n]}^{t+1} = z_{[n]}^t - \beta \nabla G(z_{[n]}^t). \quad (18)$$

When $t = (n-1)E$, it follows that $z_{[n]}^{(n-1)E} = \theta^{(n-1)E}$. Let θ^t denote the global aggregation parameter obtained after $\frac{t}{E}$ global updates in federated meta-training, corresponding to the cumulative t -th local training update across all devices. Furthermore, θ_i^t represents the local parameter obtained after performing the t -th local update. Then, we can evaluate the convergence of Algorithm 1 under the sequence $z_{[n]}^t$, presented by Theorem 3.

Theorem 3 *When the learning rates α and β satisfy $\alpha \leq \min\left\{\frac{\mu}{2\mu H + \rho B}, \frac{1}{\mu}\right\}$, $\beta < \min\left\{\frac{1}{2\mu'}, \frac{2}{H'}\right\}$, and let $\delta = \sum_{i \in U_S} \omega_i \delta_i$, $\sigma = \sum_{i \in U_S} \omega_i \sigma_i$, we have*

$$G(\theta^{TE}) - G(\theta^*) \leq \xi^{TE} [G(\theta^0) - G(\theta^*)] + \frac{B(1-\alpha\mu)}{1-\xi^E} h(E), \quad (19)$$

where

$$\xi = 1 - 2\beta\mu' \left(1 - \frac{H'\beta}{2}\right), \quad (20)$$

$$h(x) \triangleq \frac{\alpha'}{\beta H'} \left[(1 + \beta H')^x - 1\right] - \alpha' x, \quad (21)$$

and

$$\alpha' = \beta[\delta + \alpha C(H\delta + B\sigma + \tau)]. \quad (22)$$

where θ^{TE} represents the global model parameters obtained in federated meta-learning after each device performs E local updates and T global update iterations, and TE denotes the total number of local updates accumulated by each device.

Proof See Appendix C. □

The error term $\frac{B(1-\alpha\mu)}{1-\xi^E} h(E)$ shows that $h(E)$ increases with increasing δ and σ , highlighting the impact of the dissimilarity of data of IoT devices and the number of local updates before each aggregation on the overall error. Specifically, for each

device, given a fixed total number of training steps TE , the convergence error decreases as task similarity increases when E is large. However, the error increases with the number of local updates E before each global aggregation. Therefore, by adjusting the number of local update steps E per communication round, a trade-off can be achieved between communication overhead and local computational cost, depending on the degree of data similarity between device tasks. **From Theorem 3, Algorithm 1 is convergent.**

5 Experimental Results and Discussion

The federated meta-learning method is experimentally evaluated on both IID and non-IID data and compared with FedAvg [5], which is a federated averaging method. To do that, we first detail the experimental setup and parameters. Then, we present a comparison of the loss values of the global parameters obtained through training with the proposed and FedAvg methods on a new task, utilizing the MNIST and CIFAR-10 datasets. In this paper, the experiments are conducted in the PyCharm environment using Python 3.8, with the MNIST and CIFAR-10 datasets configured for both IID and non-IID settings. The convolutional neural networks (CNN) used in the experiments consist of two convolutional layers, followed by a fully connected layer. The specific parameter settings are listed in Table 1.

Table 1 Specific experimental parameter settings

Parameter	Parameter Description	Value
T	Global update rounds	100
E	Number of local updates	5
N	Number of federal meta-training devices	50
M	Number of sampled during local meta-training	20
K	Number of federal dollar test devices	10
A	Number of global devices	80
α	Internal update of learning rate	0.02
β	External update of learning rate	0.3
p	Percentage of support data set	5%, 50%, 80%

In our experiments, the global update rounds are set to $T = 100$, the local meta-training steps are $E = 5$, and the total number of global devices is $A = 80$. The number of federated meta-testing devices is $K = 10$, while the number of federated meta-training devices is $N = 50$. For each training round, $M = 20$ devices are sampled from the federated meta-training set. The proportion of data in the support set for each device’s dataset is varied and compared at $p = 5\%$, $p = 50\%$, and $p = 80\%$. The internal learning rate of the proposed system, which is the same as that of FedAvg, is set to $\alpha = 0.02$, while the external learning rate for the proposed system is set to $\beta = 0.3$.

5.1 Experimental Results under IID Data

We investigate the loss values of the global model parameters obtained through federated meta-training using the proposed system and FedAvg on the MNIST and CIFAR-10 datasets, when the data is IID. The loss values are evaluated after a few steps of gradient descent fine-tuning on federated meta-testing devices. In this experimental setup, FedAvg is trained using the support data set of the devices, while both the proposed system and FedAvg perform fine-tuning updates with the support data set of the federated meta-testing devices. The proportion of support data set p in the training data is varied to further explore the impact of the training sample size on performance differences. To further analyze the impact of training sample size on performance variations, we vary the proportion of the support data set, denoted as p , within the training data. The detailed results are presented in Fig. 3 and Fig. 4.

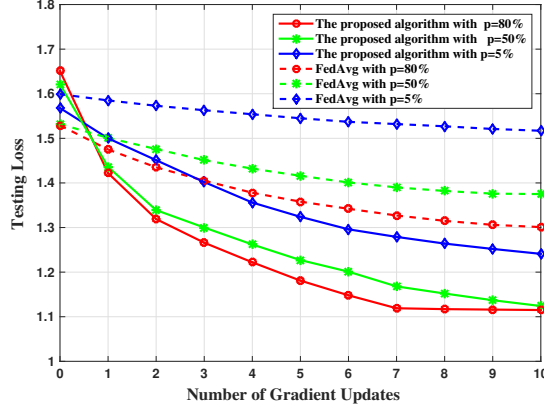


Fig. 3 Loss value testing for the IID dataset MNIST.

Fig. 3 presents the loss values during the fine-tuning process of the proposed approach and FedAvg on the IID MNIST dataset. When the proportion of support data set is $p = 5\%$, the proposed system achieves a loss of 0.185 after 9 steps of gradient descent, with little further decrease. In comparison, FedAvg reaches a loss of 0.397 after 8 steps, with only a slight reduction thereafter. When the proportion of support data set is $p = 50\%$, the proposed system achieves a loss of 0.181 after 9 steps of gradient descent, with little further decrease, while FedAvg attains a loss of 0.238 after 10 steps, continuing to show a downward trend. When the proportion of support data set is $p = 80\%$, the proposed system reaches a loss of 0.097 after 9 steps of gradient descent, with minimal further decrease, whereas FedAvg reaches a loss of 0.238 after 10 steps and still exhibits a downward trend.

Fig. 4 plots the loss values during the fine-tuning process of the proposed system and FedAvg on the IID CIFAR-10 dataset. When the proportion of support data set is $p = 5\%$, the proposed system reaches a loss of 1.241 after 10 steps of gradient descent, continuing to show a noticeable downward trend. However, FedAvg reaches

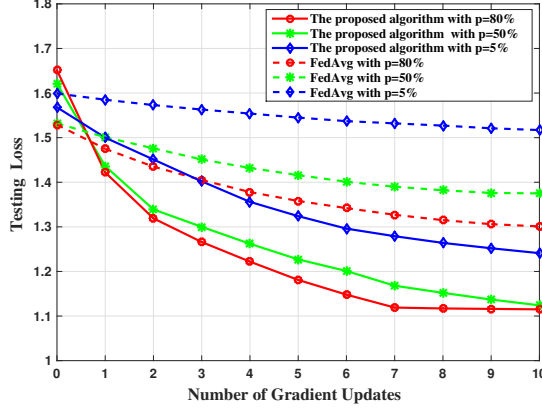


Fig. 4 Loss value testing for the IID dataset CAFIR 10.

a loss of 1.517 after 10 steps, with a weaker downward trend. In comparison, with the same number of updates, the proposed system achieves a lower loss and a stronger downward trend. When the proportion of support data set is $p = 50\%$, the proposed system reaches a loss of 1.124 after 10 steps, continuing to show a downward trend, while FedAvg reaches a loss of 1.376 after 9 steps, with minimal further decrease. When the proportion of support data set is $p = 80\%$, the proposed system reaches a loss of 1.119 after 7 steps, with little further decrease, while FedAvg reaches a loss of 0.301 after 10 steps and continues to show a slight downward trend, but the decrease is marginal.

From the above experimental results, it is evident that, under IID data distribution, the overall loss for CIFAR-10 is higher than that for MNIST for both the proposed system and FedAvg. However, regardless of whether the dataset is MNIST or the more complex CIFAR-10, the global model parameters obtained through the proposed system exhibit a smaller loss with fewer gradient descent steps compared to those trained with FedAvg. This indicates that the proposed system method provides better fast classification performance. Additionally, when the support data set is reduced, this performance gap becomes more pronounced.

5.2 Experimental Results Under NON-IID Data

We further investigate the loss values after fine-tuning the global model parameters, obtained through federated meta-training, on the federated meta-testing devices with a few steps of gradient descent. This is done for the proposed system and FedAvg on the MNIST and CIFAR-10 datasets under the NON-IID setting. As illustrated in Fig. 5 and Fig. 6.

Fig. 5 shows the loss values during the fine-tuning process of the proposed system and FedAvg when training on the NON-IID MNIST dataset. When the support set size is $p = 5\%$, the proposed system reaches a loss value of 0.258 after 10 steps of gradient descent, with no further significant decrease. On the other hand, FedAvg reaches a minimum loss value of 0.472 after 5 steps of gradient descent, after which the

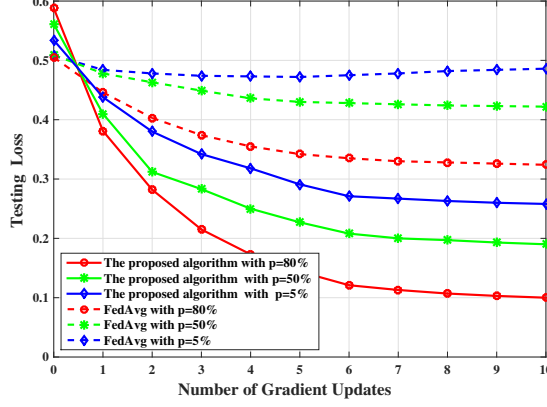


Fig. 5 Loss value testing for the Non-IID dataset MNIST.

loss value increases, indicating overfitting. When the support set size is $p = 50\%$, the proposed system reaches a loss value of 0.190 after 10 steps of gradient descent, with no further significant decrease. FedAvg, after 10 steps of gradient descent, reaches a loss value of 0.422 and shows little further decrease. In comparison, with the same number of updates, the proposed system achieves a lower loss value. When the support set size is $p = 80\%$, the proposed system reaches a loss value of 0.100 after 10 steps of gradient descent, with no further significant decrease, while FedAvg reaches a loss value of 0.324 after 8 steps of gradient descent, with no further significant decrease.

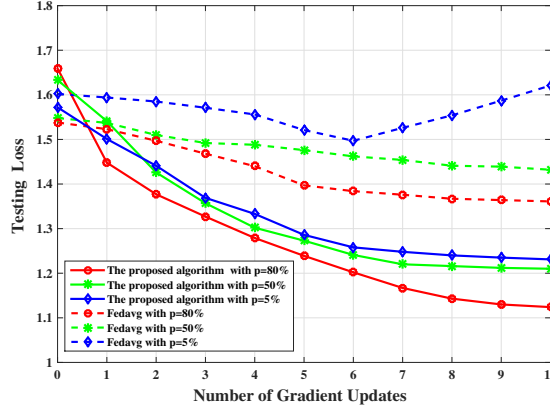


Fig. 6 Loss value testing for the Non-IID dataset CAFIR 10.

Fig. 6 shows the loss values during the fine-tuning process for the proposed system and FedAvg when training on the NON-IID CIFAR-10 dataset. When the support set size is $p = 5\%$, the proposed system reaches a loss value of 0.231 after 10 steps of gradient descent, with no further significant decrease. FedAvg reaches a minimum loss

value of 1.497 after 6 steps of gradient descent, after which the loss value increases, indicating overfitting. When the support set size is $p = 50\%$, the proposed system reaches a loss value of 1.210 after 10 steps of gradient descent, with no further significant decrease. FedAvg reaches a loss value of 1.432 after 10 steps of gradient descent and shows a slight decreasing trend. In comparison, with the same number of updates, the proposed system achieves a lower loss value. When the support set size is $p = 80\%$, the proposed system reaches a loss value of 1.124 after 10 steps of gradient descent, with no further significant decrease, while FedAvg reaches a loss value of 1.361 after 10 steps of gradient descent, with no further significant decrease. Also, the proposed system achieves a lower loss value with the same number of updates.

From the above experimental results, it can be observed that under NON-IID data conditions, especially when the support set contains fewer samples ($p = 5\%$), the FedAvg method suffers from overfitting. This problem is particularly pronounced on the CIFAR-10 dataset. In contrast, the proposed system method effectively mitigates this issue. Overall, the training loss values on CIFAR-10 are consistently higher than those on MNIST, regardless of the method employed. However, for both MNIST and the more complex CIFAR-10 dataset, the global model parameters trained by the proposed system achieve smaller loss values with fewer gradient descent steps on new tasks, compared to the FedAvg method. This indicates that the proposed system method offers improved fast classification performance. Therefore, irrespective of whether the data is IID or NON-IID, the global model parameters trained with the proposed system method exhibit superior fast classification capabilities.

5.3 Discussion

In this paper, we implemented and evaluated a meta-learning-aided federated learning framework for edge IoT systems. Our experimental results demonstrate that the proposed method significantly outperforms traditional approaches, such as FedAvg, particularly under both IID and NON-IID data distributions. With fewer gradient steps and less data (e.g., $p = 5\%$), the proposed system achieves lower loss and faster convergence, rendering it highly suitable for resource-constrained edge scenarios. Although data heterogeneity typically undermines global model performance, the proposed approach effectively mitigates this issue by optimizing model parameters to enable rapid adaptation. This enhancement is attributable to the meta-learning mechanism, which facilitates superior generalization across diverse local tasks.

Based on these findings, we recommend integrating meta-learning into federated learning frameworks to enhance robustness and reduce communication costs. Even under limited support data, the framework can maintain strong performance. Increasing the proportion of available data (e.g., $p = 80\%$) further boosts model accuracy. Moreover, the proposed approach improves learning efficiency by adapting to local data distributions when they are balanced, while preserving performance under high data heterogeneity. In addition, the framework exhibits scalability with complex datasets such as CIFAR-10 and remains effective under constrained communication settings.

6 Conclusion

In this paper, we propose a fast edge computing system model based on the federated meta-learning framework, which can address the issue of performance degradation of training global model parameters with non-IID data distributions. In the proposed framework, the training process is performed locally at the edge terminal, and the trained local model is uploaded to be globally aggregated by a central server, thereby yielding improved global model parameters. Furthermore, we propose a federated meta-learning framework where an initialization parameter optimization strategy is developed to alleviate the communication costs between edge devices and central servers. Then, the convergence of the federated meta-learning method is derived by mathematical analysis, which theoretically confirms the feasibility of the proposed federated meta-learning framework. Simulation results show that the global model parameters obtained by the training of the federated meta-learning exhibit enhanced data adaptability compared to those from traditional federated learning.

Appendix A Proof of Theorem 1

Due to $\phi_i = \theta - \alpha \nabla_{\theta} F_i(\theta)$, $G_i(\theta) = F_i(\phi_i(\theta))$, we have

$$\begin{aligned} \nabla G_i(\theta) &= \nabla F_i(\phi_i(\theta)) = \nabla F_i(\phi_i(\theta)) \cdot (1 - \alpha \nabla^2 F_i(\theta)) \\ &= \nabla F_i(\phi_i(\theta)) - \alpha \nabla F_i(\phi_i(\theta)) \nabla^2 F_i(\theta). \end{aligned} \quad (\text{A1})$$

From (A1), we can obtain

$$\begin{aligned} &\|\nabla G_i(\theta) - \nabla G_i(\theta')\| \\ &= \left\| [I - \alpha \nabla^2 F_i(\theta)] [\nabla F_i(\phi_i) - \nabla F_i(\phi'_i)] - \alpha \nabla F_i(\phi'_i) [\nabla^2 F_i(\theta) - \nabla^2 F_i(\theta')] \right\|, \end{aligned} \quad (\text{A2})$$

where $\phi'_i = \theta' - \alpha \nabla_{\theta} F_i(\theta)$. By combining (A2) with properties A, B, and C, we can obtain

$$\begin{aligned} \|\nabla G_i(\theta) - \nabla G_i(\theta')\| &\geq (1 - \alpha H) \|\nabla F_i(\phi_i) - \nabla F_i(\phi'_i)\| \\ &\quad - \alpha \|\nabla F_i(\phi'_i)\| \|\nabla^2 F_i(\theta) - \nabla^2 F_i(\theta')\| \\ &\geq \mu(1 - \alpha H) \|\phi_i - \phi'_i\| - \alpha \rho B \|\theta - \theta'\|. \end{aligned} \quad (\text{A3})$$

With Property A and Property B, it reveals that

$$\mu \frac{\|\nabla F_i(\theta) - \nabla F_i(\theta')\|}{\|\theta - \theta'\|} = \nabla^2 F_i(\theta) \leq H, \quad (\text{A4})$$

and

$$1 - \alpha H \leq \nabla \phi_i = 1 - \alpha \nabla^2 F_i(\theta) \leq 1 - \alpha \mu. \quad (\text{A5})$$

Let $\nabla \phi_i = \frac{\|\phi_i - \phi'_i\|}{\|\theta - \theta'\|}$, and then we have

$$(1 - \alpha H) \|\theta - \theta'\| \leq \|\phi_i - \phi'_i\| \leq (1 - \alpha \mu) \|\theta - \theta'\|. \quad (\text{A6})$$

Combining (A5) and (A6), we can obtain

$$\|\nabla G_i(\theta) - \nabla G_i(\theta')\| \geq \mu' \|\theta - \theta'\|, \quad (\text{A7})$$

where $\mu' = \mu(1 - \alpha H)^2 - \alpha \rho B > 0$.

Similarly, it can be obtained from (A4) and (A6) that

$$\begin{aligned} \|\nabla G_i(\theta) - \nabla G_i(\theta')\| &\leq (1 - \alpha \mu) \|\nabla F_i(\phi_i) - \nabla F_i(\phi'_i)\| \\ &\quad + \alpha \|\nabla F_i(\phi'_i)\| \|\nabla^2 F_i(\theta) - \nabla^2 F_i(\theta')\| \\ &\leq H(1 - \alpha \mu) \|\phi_i - \phi'_i\| + \alpha \rho B \|\theta - \theta'\| \\ &\leq H' \|\theta - \theta'\|, \end{aligned} \quad (\text{A8})$$

where $H' = H(1 - \alpha \mu)^2 + \alpha \rho B$.

Accordingly, (15) holds. As $\alpha \leq \min \left\{ \frac{\mu}{2\mu H + \rho B}, \frac{1}{\mu} \right\}$, $G(\theta)$ is μ' -strongly convex and H' -smooth.

Appendix B Proof of Theorem 2

Given $\phi_i = \theta - \alpha \nabla_{\theta} L(\theta, D_i^S)$, $\|\nabla F_i(\theta)\| \leq B$. According to Taylor's theorem, we can obtain

$$\begin{aligned} \nabla G_i(\theta) &= \nabla F_i(\phi_i) \\ &= \nabla F_i(\theta) + \nabla^2 F_i(\theta) (\phi_i - \theta) + O(\|\phi_i - \theta\|^2) \\ &= \nabla F_i(\theta) - \alpha \nabla^2 F_i(\theta) \nabla F_i(\theta) + O(\alpha^2 B^2). \end{aligned} \quad (\text{B9})$$

With (B9), the upper bounds on the gradient and Hessian matrix product are then given by

$$\begin{aligned} &\left\| \nabla^2 F_i(\theta) \nabla F_i(\theta) - \sum_{i \in U_S} \omega_i \nabla^2 F_i(\theta) \nabla F_i(\theta) \right\| \\ &\leq \|\nabla^2 F_i(\theta)\| \|\nabla F_i(\theta) - \nabla F_w(\theta)\| + \|\nabla F_w(\theta)\| \|\nabla^2 F_i(\theta) - \nabla^2 F_w(\theta)\| \\ &\quad + \sum_{i \in U_S} \omega_i \|\nabla F_i(\theta) - \nabla F_w(\theta)\| \|\nabla^2 F_i(\theta) - \nabla^2 F_w(\theta)\| l e H \delta_i + B \sigma_i + \tau, \end{aligned} \quad (\text{B10})$$

where $\tau = \sum_{i \in U_S} \omega_i \delta_i \sigma_i$. Similarly, it holds that

$$\left\| [\nabla^2 F_i(\theta)]^2 \nabla F_i(\theta) - \alpha^2 \sum_{i \in U_S} \omega_i [\nabla^2 F_i(\theta)]^2 \nabla F_i(\theta) \right\| \leq H \delta'_i + B \sigma_i + \tau', \quad (\text{B11})$$

where $\delta'_i = H \delta_i + B \sigma_i + \tau$, $\tau' = \sum_{i \in U_S} \omega_i \delta'_i \sigma_i$. According to (B10) and (B11), when α is small, there exists a constant C that satisfies

$$\begin{aligned}
& \|\nabla G_i(\theta) - \nabla G(\theta)\| \\
&= \left\| [I - \alpha \nabla^2 F_i(\theta)] \nabla F_i(\phi_i) - \sum_{i \in U_S} \omega_i [I - \alpha \nabla^2 F_i(\theta)] \nabla F_i(\phi_i) \right\| \\
&\leq \|\nabla F_i(\theta) - \nabla F_w(\theta)\| + O(\alpha^2 B^2) + 2\alpha \left\| \nabla^2 F_i(\theta) \nabla F_i(\theta) - \sum_{i \in U_S} \omega_i \nabla^2 F_i(\theta) \nabla F_i(\theta) \right\| \\
&\quad + \alpha^2 \left\| [\nabla^2 F_i(\theta)]^2 \nabla F_i(\theta) - \sum_{i \in U_S} \omega_i [\nabla^2 F_i(\theta)]^2 \nabla F_i(\theta) \right\| \\
&\leq \delta_i + \alpha C (H\delta_i + B\sigma_i + \tau)
\end{aligned} \tag{B12}$$

Given a constant C , it holds that

$$\|\nabla G_i(\theta) - \nabla G(\theta)\| \leq \delta_i + \alpha C (H\delta_i + B\sigma_i + \tau), \tag{B13}$$

where $\tau = \sum_{i \in \mathcal{S}} \omega_i \delta_i \sigma_i$.

It completes the proof of Theorem 2.

Appendix C Proof of Theorem 3

To analyze the gap between $z_{[n]}^t$ and θ^t within the interval $[n]$, we first investigate the difference between $z_{[n]}^t$ and the local parameter θ_i^t , which is obtained after device i completes t local updates, starting from

$$\begin{aligned}
& \left\| \nabla^2 F_i(\theta) \nabla F_i(\theta) - \sum_{i \in U_S} \omega_i \nabla^2 F_i(\theta) \nabla F_i(\theta) \right\| \\
&= \left\| \nabla^2 F_i(\theta) \nabla F_i(\theta) - \nabla^2 F_w(\theta) \nabla F_w(\theta) + \nabla^2 F_w(\theta) \nabla F_w(\theta) - \sum_{i \in \mathcal{S}} \omega_i \nabla^2 F_i(\theta) \nabla F_i(\theta) \right\| \\
&\leq \|\nabla^2 F_i(\theta)\| \|\nabla F_i(\theta) - \nabla F_w(\theta)\| + \|\nabla F_w(\theta)\| \|\nabla^2 F_i(\theta) - \nabla^2 F_w(\theta)\| \\
&\quad + \sum_{i \in U_S} \omega_i \|\nabla F_i(\theta) - \nabla F_w(\theta)\| \|\nabla^2 F_i(\theta) - \nabla^2 F_w(\theta)\| \\
&\leq H\delta_i + B\sigma_i + \tau
\end{aligned} \tag{C14}$$

Accordingly, we have

$$\|\theta_i^t - z_{[n]}^t\| \leq g(t - (n-1)T_0), \tag{C15}$$

where

$$g(x) \triangleq \frac{\delta_i + \alpha C (H\delta_i + B\sigma_i + \tau)}{H'} [(1 + \beta H')^x - 1]. \quad (\text{C16})$$

Thus, for $t \in [(n-1)E, nE]$, we have

$$\begin{aligned} \|\theta^{t+1} - z_{[n]}^{t+1}\| &= \left\| \sum_{i \in U_S} \omega_i \theta_i^{t+1} - z_{[n]}^{t+1} \right\| = \left\| \theta^t - \beta \sum_{i \in U_S} \omega_i \nabla G_i(\theta_i^t) - z_{[n]}^t + \beta \nabla G(z_{[n]}^t) \right\| \\ &\leq \|\theta^t - z_{[n]}^t\| + \beta H' \sum_{i \in S} \omega_i \|\theta_i^t - z_{[n]}^t\| = \|\theta^t - z_{[n]}^t\| + \alpha' [(1 + \beta H')^{t-(n-1)T_0} - 1], \end{aligned} \quad (\text{C17})$$

where $\alpha' = \beta[\delta + \alpha C(H\delta + B\sigma + \tau)]$. Since $\theta^{(n-1)E} = z_{[n]}^{(n-1)E}$, it holds that

$$\begin{aligned} \|\theta^t - z_{[n]}^t\| &\leq \sum_{j=1}^{t-(n-1)E} \left\{ \alpha' [(1 + \beta H')^j - 1] \right\} \\ &= \frac{\alpha'}{\beta H'} [(1 + \beta H')^{t-(n-1)E} - 1] - \alpha' [t - (n-1)E] \\ &\triangleq h(t - (n-1)T_0). \end{aligned} \quad (\text{C18})$$

In the following, we evaluate the convergence performance of the virtual sequence $z_{[n]}^t$ within the interval $[n]$. Since $G(\theta)$ is H' -smooth, we have

$$\begin{aligned} G(z_{[n]}^{t+1}) - G(z_{[n]}^t) &\leq \langle \nabla G(z_{[n]}^t), z_{[n]}^{t+1} - z_{[n]}^t \rangle + \frac{H'}{2} \|z_{[n]}^{t+1} - z_{[n]}^t\|^2 \\ &\leq -\beta \left(1 - \frac{H'\beta}{2}\right) \|\nabla G(z_{[n]}^t)\|^2. \end{aligned} \quad (\text{C19})$$

Additionally, since $G(\theta)$ is μ' -strongly convex, we have

$$G(z_{[n]}^t) \leq G(\theta^*) + \frac{1}{2\mu'} \|\nabla G(z_{[n]}^t)\|^2, \quad (\text{C20})$$

where θ^* is the globally optimal solution. Combining C19 and C20, we can obtain

$$\begin{aligned} G(z_{[n]}^{t+1}) - G(z_{[n]}^t) &\leq -2\beta\mu' \left(1 - \frac{H'\beta}{2}\right) [G(z_{[n]}^t) - G(\theta^*)], \\ G(z_{[n]}^{t+1}) - G(\theta^*) &\leq \left[1 - 2\beta\mu' \left(1 - \frac{H'\beta}{2}\right)\right] [G(z_{[n]}^t) - G(\theta^*)] \\ &= \xi [G(z_{[n]}^t) - G(\theta^*)], \end{aligned} \quad (\text{C21})$$

where $\xi = 1 - 2\beta\mu' \left(1 - \frac{H'\beta}{2}\right) \in (0, 1)$, $\beta < \min \left\{ \frac{1}{2\mu'}, \frac{2}{H'} \right\}$. Thus, it holds that

$$\begin{aligned}
G\left(z_{[n]}^{nE}\right) - G\left(\theta^*\right) &\leq \xi \left[G\left(z_{[n]}^{nE-1}\right) - G\left(\theta^*\right) \right] \\
&\leq \xi^2 \left[G\left(v_{[n]}^{nE-2}\right) - G\left(\theta^*\right) \right] \\
&\dots \\
&\leq \xi^{T_0} \left[G\left(v_{[n]}^{(n-1)E}\right) - G\left(\theta^*\right) \right] \\
&= \xi^{T_0} \left[G\left(v_{[n-1]}^{(n-1)E}\right) - G\left(\theta^*\right) \right] \\
&+ \xi^{T_0} \left[G\left(v_{[n]}^{(n-1)E}\right) - G\left(v_{[n-1]}^{(n-1)E}\right) \right].
\end{aligned} \tag{C22}$$

Since

$$\begin{aligned}
\|G(\theta)\| &= \left\| \sum_{i \in U_S} \omega_i \left[(I - \alpha \nabla^2 F_i(\theta)) \nabla F_i(\phi_i) \right] \right\| \leq \sum_{i \in S} \omega_i \|I - \alpha \nabla^2 F_i(\theta)\| \|\nabla F_i(\phi_i)\| \\
&\leq (1 - \alpha\mu)B,
\end{aligned} \tag{C23}$$

According to the median theorem, it holds that

$$\|G(\theta) - G(\theta')\| \leq (1 - \alpha\mu)B \|\theta - \theta'\|. \tag{C24}$$

According to (C18), we have

$$\begin{aligned}
G\left(z_{[n]}^{(n-1)E}\right) - G\left(z_{[n-1]}^{(n-1)E}\right) &= G\left(\theta^{(n-1)E}\right) - G\left(z_{[n-1]}^{(n-1)E}\right) \\
&\leq B(1 - \alpha\mu) \left\| \theta^{(n-1)E} - z_{[n-1]}^{(n-1)E} \right\| \\
&\leq B(1 - \alpha\mu)h(T_0).
\end{aligned} \tag{C25}$$

Substituting (C25) into (C23), yield

$$G\left(z_{[n]}^{nE}\right) - G\left(\theta^*\right) \leq \xi^E \left[G\left(v_{[n-1]}^{(n-1)E}\right) - G\left(\theta^*\right) \right] + \xi^E B(1 - \alpha\mu)h(E). \tag{C26}$$

Then, we have

$$\begin{aligned}
&G\left(z_{[N]}^{NT_0}\right) - G\left(\theta^*\right) \\
&\leq \xi^E \left[G\left(v_{[N-1]}^{(N-1)E}\right) - G\left(\theta^*\right) \right] + \xi^E B(1 - \alpha\mu)h(E) \\
&\dots \\
&= \xi^{NE} \left[G\left(\theta^0\right) - G\left(\theta^*\right) \right] + \sum_{j=1}^{N-1} \xi^{jE} B(1 - \alpha\mu)h(T_0)
\end{aligned} \tag{C27}$$

According to (C27), we can obtain

$$\begin{aligned}
G(\theta^{TE}) - G(\theta^*) &= G\left(z_{[N+1]}^{NE}\right) - G(\theta^*) \\
&\leq \xi^{NE} [G(\theta^0) - G(\theta^*)] + \sum_{j=1}^{N-1} \xi^{jE} B(1 - \alpha\mu)h(E) \\
&\quad + B(1 - \alpha\mu)h(E) \\
&= \xi^{TE} [G(\theta^0) - G(\theta^*)] + \frac{1 - \xi^{TE}}{1 - \xi^E} B(1 - \alpha\mu)h(E) \\
&\leq \xi^{TE} [G(\theta^0) - G(\theta^*)] + \frac{B(1 - \alpha\mu)}{1 - \xi^E} h(E).
\end{aligned} \tag{C28}$$

Therefore, (20) holds. It completes the proof of Theorem 3.

Availability of data and materials

Data available upon request.

Abbreviations

IoT:	Internet of Things
AI:	Artificial intelligence
FedAvg:	Federated Averaging Algorithm
ML:	Machine learning
Non-IID:	Non-independent and identically distributed
IID:	Independent and identically distributed
MANN:	Memory-augmented neural network
MAML:	Model-agnostic meta-learning
SGD:	Stochastic gradient descent
CNN:	Convolutional neural networks

Funding

This work was supported by the Hainan Province Science and Technology Special Fund under Grant ZDYF2024GXJS292.

Author information

Authors and Affiliations

Xiaofeng Zhu and Jiaqiang Peng are with the School of Mechatronics Engineering, Xinyu Key Laboratory of Power System and Intelligent Control, Xinyu 338004, China. Qiaosong Fan is with the Shien-ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou 510641, China. Yuwen Qian is with the

School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China.

Contributions

ZX and FQ have contributed to the idea development, conducting experiments and evaluation including preparing the draft of the manuscript. PJ contributed to the idea and review of the manuscript. PJ and QY is the corresponding author. All authors read and approved the final manuscript.

Corresponding author

Correspondence to Jiaqiang Peng and Yuwen Qian.

Ethics declarations

Competing interests

The authors declare that they have no competing interests.

Additional information

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

- [1] Shrestha, A., Mahmood, A.: Review of deep learning algorithms and architectures. *IEEE Access* **7**, 53040–53065 (2019)
- [2] Cao, L.: Deep learning applications. *IEEE Intelligent Systems* **37**(3), 3–5 (2022)
- [3] Xu, Z.: Federated learning in the age of foundation models. In: 2024 2nd International Conference on Federated Learning Technologies and Applications (FLTA), pp. 2–2 (2024)
- [4] AbdulRahman, S., Tout, H., Ould-Slimane, H., *et al.*: A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal* **8**(7), 5476–5497 (2020)
- [5] McMahan, B., Moore, E., Ramage, D., *et al.*: Communication-efficient learning of deep networks from decentralized data. In: *Artificial Intelligence and Statistics*, pp. 1273–1282 (2017). PMLR

- [6] Kairouz, P., McMahan, H.B., Avent, B., *et al.*: Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* **14**(1-2), 1–210 (2021)
- [7] Li, T., Sahu, A.K., Talwalkar, A., *et al.*: Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* **37**(3), 50–60 (2020)
- [8] Zhou, Z.: Research on efficient federated learning algorithm for communication based on model compression. PhD thesis, Wuhan University of Engineering, Hubei (2023)
- [9] Zhao, Y., Li, M., Lai, L., *et al.*: Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018)
- [10] Yoshida, N., Nishio, T., Morikura, M., *et al.*: Hybrid-fl for wireless networks: Cooperative learning mechanism using non-iid data. In: *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–7 (2020). IEEE
- [11] Zhang, C.-Y., Si, S.-J., Wang, J.-Z., *et al.*: A review of federated meta-learning. *Big Data* **9**(2), 122–146 (2023) <https://doi.org/10.11959/j.issn.2096-0271.2022051>
- [12] Zhou, P., Yuan, X.-T., Xu, W., *et al.*: Efficient meta learning via minibatch proximal update. *Advances in Neural Information Processing Systems* **32** (2019)
- [13] Schmidhuber, J.: Evolutionary principles in self-referential learning. PhD thesis, Technische Universität München (1987)
- [14] Bengio, Y., Bengio, S., Cloutier, J.: Learning a synaptic learning rule. In: *International Joint Conference on Neural Networks (IJCNN)* (1990)
- [15] Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A.: Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**(9), 5149–5169 (2022)
- [16] Michelon, G.K., Assunção, W.K.G., Grünbacher, P., Egyed, A.: Analysis and propagation of feature revisions in preprocessor-based software product lines. In: *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 284–295 (2023)
- [17] Luo, Q., Yu, T., Lan, C., Huang, Y., Wang, Z., Pan, Z.: A generalizable method for practical non-intrusive load monitoring via metric-based meta-learning. *IEEE Transactions on Smart Grid* **15**(1), 1103–1115 (2024)
- [18] Xu, Z., Chen, X., Cao, L.: Fast task adaptation based on the combination of model-based and gradient-based meta learning. *IEEE Transactions on Cybernetics* **52**(6), 5209–5218 (2022)

- [19] Santoro, A., Bartunov, S., Botvinick, M., *et al.*: Meta-learning with memory-augmented neural networks. In: International Conference on Machine Learning, pp. 1842–1850 (2016). PMLR
- [20] Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning, pp. 1126–1135 (2017). PMLR
- [21] Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999 (2018)
- [22] Denevi, G., Ciliberto, C., Stamos, D., et al.: Learning to learn around a common mean. *Advances in Neural Information Processing Systems* **31** (2018)
- [23] Balcan, M.-F.F., Khodak, M., Talwalkar, A., *et al.*: Provable guarantees for gradient-based meta-learning. In: International Conference on Machine Learning, pp. 424–433 (2019). PMLR
- [24] Boggust, A., Sivaraman, V., Assogba, Y., Ren, D., Moritz, D., Hohman, F.: Compress and compare: Interactively evaluating efficiency and behavior across ml model compression experiments. *IEEE Transactions on Visualization and Computer Graphics* **31**(1), 809–819 (2025)
- [25] Fu, S., Li, Z., Liu, K., Din, S., Imran, M., Yang, X.: Model compression for iot applications in industry 4.0 via multiscale knowledge transfer. *IEEE Transactions on Industrial Informatics* **16**(9), 6013–6022 (2020)
- [26] Qin, Z., Feng, G., Liu, Y., Yum, T.P., Wang, F., Wang, J.: Efficient federated learning in wireless networks with incremental model quantization and uploading. *IEEE Transactions on Network Science and Engineering*, 1–14 (2025)
- [27] Gong, Y., Liu, L., Yang, M., et al.: Compressing deep convolutional networks using vector quantization. arXiv preprint arXiv:1412.6115 (2014)
- [28] Kim, J., Park, S.W., Kwak, N.: Paraphrasing complex network: Network compression via factor transfer. *Advances in neural information processing systems* **31** (2018)
- [29] Passalis, N., Tefas, A.: Learning deep representations with probabilistic knowledge transfer. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 268–284 (2018)
- [30] Wang, S., Tuor, T., Saloniidis, T., *et al.*: Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications* **37**(6), 1205–1221 (2019)

Figure Legends

Fig. 1 Federated learning architecture for edge IoT with one server and M devices. The figure illustrates a federated learning system architecture deployed in an edge computing environment. The proposed system with federated learning architecture is composed of an edge server and N IoT devices as devices S . In each iteration, a random subset of M devices is selected for global aggregation and model updates. Each device trains a local model using their local respective dataset, denoted as $D_1, D_2, D_3, \dots, D_i, \dots, D_M$ where $i \in S$.

Fig. 2 The framework of the proposed meta-learning aided federated approach. The figure presents a meta-learning aided federated learning framework designed to improve model generalization under Non-IID data conditions. It illustrates the two-phase process: meta-training, where the model learns fast-adapting parameters using task-specific support and query sets; and meta-testing, where the model rapidly adapts to new tasks with minimal data. This approach enhances performance in heterogeneous edge environments by enabling more effective global model updates.

Fig. 3 Loss value testing for the IID dataset MNIST. This figure compares the loss curves of the proposed method and FedAvg on the IID MNIST dataset, under varying proportions of support data proportions ($p = 5\%, 50\%, 80\%$). The proposed approach converges faster and achieves lower final loss in all cases. At $p = 5\%$, it reaches a loss of 0.185 after 9 steps, while FedAvg reaches 0.397 after 8 steps. As p increases, performance improves further, with the lowest loss of 0.097 achieved at $p = 80\%$.

Fig. 4 Loss value testing for the IID dataset CAFIR 10. This figure compares the loss curves of the proposed method and FedAvg on the IID CIFAR-10 dataset under varying support data proportions ($p = 5\%, 50\%, 80\%$). The proposed system consistently achieves lower loss and shows stronger convergence. At $p = 5\%$, it reaches a loss of 1.241 after 10 steps with a clear downward trend, outperforming FedAvg (loss: 1.517). Performance improves further as p increases, demonstrating the effectiveness of the approach in handling image classification tasks.

Fig. 5 Loss value testing for the Non-IID dataset MNIST. This figure compares the fine-tuning loss of the proposed system and FedAvg on the Non-IID MNIST dataset at different support set proportions ($p = 5\%, 50\%, 80\%$). The proposed method consistently achieves lower loss and better stability. At $p = 5\%$, it reaches a loss of 0.258 after 10 steps, while FedAvg shows signs of overfitting with a loss of 0.472. Performance improves further with higher p , demonstrating the method's effectiveness under data heterogeneity.

Fig. 6 Loss value testing for the Non-IID dataset CAFIR 10. This figure compares the fine-tuning loss of the proposed system and FedAvg on the Non-IID CIFAR-10 dataset under varying support set sizes ($p = 5\%, 50\%, 80\%$). The proposed method achieves significantly lower loss in all cases. At $p = 5\%$, it reaches a

loss of 0.231 after 10 steps, while FedAvg shows overfitting (loss: 1.497). As p increases, the performance gap widens, demonstrating improved robustness and convergence under data heterogeneity.