FCBAFL: An Energy-Conserving Federated Learning Approach in Industrial Internet of Things

Bin Qiu[1,2*], Duan Li[1,2], Xian Li[3], and Hailin Xiao[4]
[1] School of Computer Science and Engineering, Guilin University of technology, Guilin, 541004, China
[2] Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin University of technology Guilin, 541004, China
[3] School of Electronics and Information Engineering, Shenzhen University, Shenzhen, 518060, China
[4] School of Computer Science and Information Engineering, Hubei University, Wuhan, 430062, China
[e-mail: qiubin1122@126.com; 15340669724@163.com; xianli@szu.edu.cn; xhl_xiaohailin@163.com]
[*]Corresponding author: Bin Qiu

Received April 1, 2024; revised July 25, 2024; accepted August 18, 2024; published September 30, 2024

Abstract

Federated learning (FL) has been proposed as an emerging distributed machine learning framework, which lowers the risk of privacy leakage by training models without uploading original data. Therefore, it has been widely utilized in the Industrial Internet of Things (IIoT). Despite this, FL still faces challenges including the non-independent identically distributed (Non-IID) data and heterogeneity of devices, which may cause difficulties in model convergence. To address these issues, a local surrogate function is initially constructed for each device to ensure a smooth decline in global loss. Subsequently, aiming to minimize the system energy consumption, an FL approach for joint CPU frequency control and bandwidth allocation, called FCBAFL is proposed. Specifically, the maximum delay of a single round is first treated as a uniform delay constraint, and a limited-memory Broyden-Fletcher-Goldfarb-Shanno bounded (L-BFGS-B) algorithm is employed to find the optimal bandwidth allocation with a fixed CPU frequency. Following that, the result is utilized to derive the optimal CPU frequency. Numerical simulation results show that the proposed FCBAFL algorithm exhibits more excellent convergence compared with baseline algorithm, and outperforms other schemes in declining the energy consumption.

Keywords: Federated learning (FL), industrial internet of things (IIoT), heterogeneity, frequency control, bandwidth allocation.

1. Introduction

Traditional industrial manufacturing tends to become more and more intelligent and informatized due to the swift progress of both the internet of things (IoT) and artificial intelligence (AI) [1-2]. As a result, the Industrial Internet of Things (IIoT) has emerged. There are billions of diverse industrial devices connected to the edge of the network [3]. Moreover, a large number of industrial tasks rely on real-time monitoring and decision support, whereby various sensors and controllers generate massive amounts of data to support them [4-5]. Since these data usually contain private information [6], it is essential to protect such important information from leakage during data processing. For traditional centralized machine learning (ML) approach, it may result in privacy leakage because it collects all the data into a data center for uniform handling [7-8]. For example, some IIoT applications involve location information of transportation vehicles, and unauthorized access or leakage of such location data may pose a threat to the security and privacy of supply chains [9]. Moreover, massive industrial equipment data need to be transmitted over wireless network, which incurs unacceptable communication traffic [10].

As a distributed ML paradigm, FL is expected to alleviate concerns about data privacy leakage and huge overhead posed by centralized ML [11]. This technology deploys ML models on multiple devices [12-13] and allows them to cooperatively construct a shared learning model as well as maintaining all training data on edge devices [14]. FL only needs to send the model parameters to the data center, which significantly lessens cost while well protecting data privacy. FL has been quickly applied and developed [15-24]. On one hand, the advances in memory elements have provided favorable conditions for large amounts of data collecting and storing. On the other hand, these devices are typically equipped with high-performance chips, which makes it easy to complete the model training tasks in FL.

Despite the above advantages of FL, there are a couple of challenges that need to tackle. In practice, the distribution of data on devices is usually non-independent identically distributed (Non-IID) data [15]. Many researchers have devoted a lot of work to address this [16-18], an example is the widely used FedAvg utilizes a mean stochastic gradient descent (SGD) strategy [16], which shows excellent performance on heterogeneous data, but still has limitations due to the lack of convergence analysis. Based on the derived upper bound on the expected weight divergence, Zhao et al. [17] introduced a federated averaging approach aiming at mitigating the distribution divergence presented in Non-IID data. You et al. [18] proposed the federated gradient scheduling, an enhanced method utilizing historical gradient sampling. It gathers and samples user gradients to solve the Non-IID problem, and also utilizes differential privacy techniques to enhance privacy protection. Nevertheless, the above methods proposed for the Non-IID problem ignore the massive cost in FL.

In reality, FL requires intense exchanges of model updates between the server and clients during model training [19]. Unfortunately, in recent years, as deep learning models are getting increasingly huge, which often contain up to billions or even trillions of learnable parameters, thus incurring huge communication and computing overheads [20-21]. Motivated by this, many researchers have devoted their research work to energy conservation in wireless FL networks [22-24]. Alishahi et al. [22] adopted a low-complexity bisection algorithm and jointly considered the various resources of the devices for minimizing the overall energy consumption of the wireless FL network. Kim et al. [23] proposed a joint dataset and computation management scheme by integrating learning efficiency and global energy consumption. For the sake of lowering the cost of communication caused by frequent model interactions in FL, Malan et al. [24] proposed a novel approach of FL and gradual tier freezing,

which reduces the transmission cost while guaranteeing the performance of training. However, most of them ignore the differences in the performance of various devices.

To solve the above problems for FL in IIoT system, a novel solution is proposed in this article. Aiming at the heterogeneous data, we assume that the loss function is smooth and strongly convex for theoretical analysis. Considering local computation and model transmission, we start with investigating the computation delay and energy consumption of heterogeneous devices. Then, the ultimate objective of this article is defined as an energy minimization problem under delay constraint. Owing to the nonconvexity of the original problem, we decompose it into two simple sub-problems, which are described as the CPU frequency control subproblem and the bandwidth allocation subproblem, respectively. Moreover, we devise an iterative method to solve the problem. In the end, the performance of the proposed scheme is evaluated by a wide range of experiments. In summary, the contributions of this article are mainly in the following aspects:

(1) We propose a novel FL algorithm to address the challenge for the Non-IID data and heterogeneity of devices in IIoT scenario. Specifically, a local proxy function is initially constructed for each device to ensure a smooth decline in global loss, and a hyperparameter η is introduced to trade-off between local and global gradient estimation, and the linear convergence of the method is proved theoretically.

(2) To cope with the high energy consumption problem generated by model training in FL, a joint CPU frequency control and bandwidth allocation approach, called FCBAFL is proposed. Considering the synchronous communication and limited bandwidth resource, the ultimate goal of this article is characterized as minimizing energy consumption under delay constraint. Due to the nonconvexity of the original problem, the proposed scheme splits it into two subproblems to solve.

(3) Extensive numerical simulations show that FCBAFL has higher and stable convergence compared with baseline algorithm on unbalanced MNIST dataset. Moreover, the proposed algorithm can decrease the energy consumption of IIoT system to a certain extent while satisfying various delay constraints.

The remainder of the article is organized as follows. Section 2 presents the system model and problem formulation. Section 3 provides the solution to the problem. Section 4 offers experimental numerical results that demonstrate the superiority of the proposed method. Section 5 summarizes the work of this article.

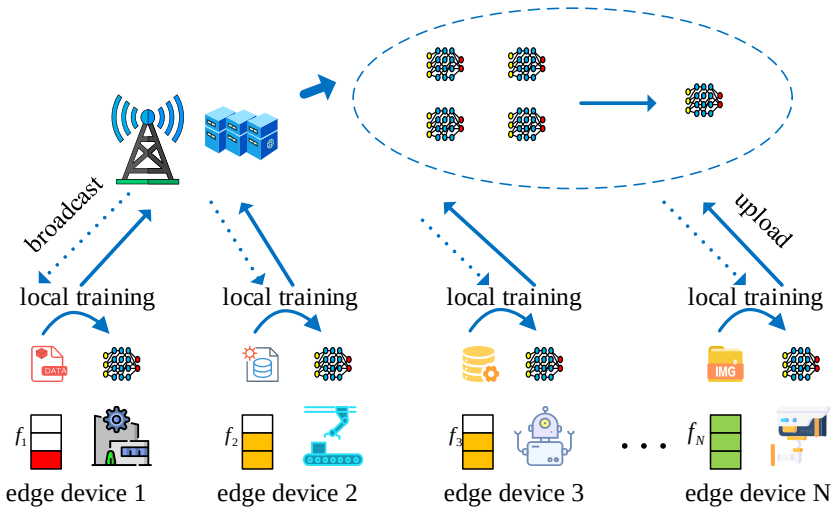


Fig. 1. System model.

2. System Model

We consider a framework of FL system in an IIoT scenario, which includes a central server and multiple edge devices, as shown in **Fig. 1**. The central server is located in the center of the region and provides basic model communication and aggregation services. The set of edge devices is defined as $\mathsf{N}=\{1,2,...,N\}$, where N denotes the number of edge devices. The edge devices generate local models by local training, the dataset produced on device n is expressed as $D_n=\{(x_i,y_i)\}_{i=1}^{d_n}$, where x_i is sample i-th input sample and y_i denotes its output label. And d_n represents the data size of D_n. Therefore, the local loss function on device n's dataset can be expressed as

$$F_n(\omega)=\frac{\sum_{(x_i,y_i)\in D_n} f_n(\omega,x_i,y_i)}{d_n} \tag{1}$$

where ω and $f_n(\omega,x_i,y_i)$ represent the model parameter and loss function, respectively. The purpose of FL is to find an optimal model parameter $\omega^*\in R^d$ to minimize the global loss function, denoted as

$$\omega^*=\arg\min_{\omega} F(\omega)\overset{\Delta}{=}\arg\min_{\omega}\frac{\sum_{n\in\mathsf{N}} d_n F_n(\omega)}{\sum_{n\in\mathsf{N}} d_n} \tag{2}$$

2.1 FL Process

To facilitate the analysis, this article assumes that local loss function is smooth and strongly convex.

Assumption 1: $F_n(\cdot)$ is L-smooth, $\forall\omega,\omega'\in R^d$

$$F_n(\omega)\le F_n(\omega')+<\nabla F_n(\omega'),\omega-\omega'>+\frac{L}{2}\|\omega-\omega'\|^2 \tag{3}$$

Assumption 2: $F_n(\cdot)$ is μ-strong convex (since the Hessian of $F_n(\cdot)$ can be positive semi-definite), $\forall\omega,\omega'\in R^d$

$$F_n(\omega)\ge F_n(\omega')+<\nabla F_n(\omega'),\omega-\omega'>+\frac{\mu}{2}\|\omega-\omega'\|^2 \tag{4}$$

The assumptions are similar to the l_2-regularized linear regression model $f_n(\omega)=\frac{1}{2}(<x_i,\omega>-y_i)^2+\frac{\mu}{2}\|\omega\|^2$, and let $\rho=L/\mu$ be the condition number of $F_n(\cdot)'s$ Hessian matrix. In each round of iteration, central server interacts with the edge devices in the following process.

2.1.1 Local model updates

Device n first receives the global model parameter and gradient information (which are given in (10) and (11) below, respectively) from the previous global round to minimize the proxy loss function, which is denoted as

$$\min L_n^t(\omega) = F_n(\omega) + < \eta \nabla \bar{F}^{t-1} - \nabla F_n(\omega^{t-1}), \omega > \tag{5}$$

Instead of $\nabla F(\omega^{t-1})$, we utilize $\nabla \bar{F}^{t-1}$ as the estimate of the global gradient, due to the fact that the latter can be obtained from the edge device information via the server, while the former is unrealistic. Moreover, we have

$$\nabla L_n^t(\omega) = \nabla F_n(\omega) + \eta \nabla \nabla \bar{F}^{t-1} - \nabla F_n(\omega^{t-1}) \tag{6}$$

where a variable hyperparameter η is introduced to achieve weighted estimation of local and global gradient. The key of this algorithm is that each device can solve (5) and obtain an approximate solution ω_n^t that satisfies

$$\| \nabla L_n^t(\omega_n^t) \| \leq \theta \| \nabla L_n^t(\omega^{t-1}) \|, \forall n \tag{7}$$

Here, $\theta \in [0,1]$ denotes the local accuracy. Note that $L_n^t(\omega)$ is also L-smooth and μ-strongly convex because it has the same Hessian matrix as $F_n(\cdot)$. Therefore, the objective in (5) can be solved by gradient descent algorithm. It has been shown that this can achieve linear convergence [25], as follows

$$L_n^t(x_k) - L_n^t(x^*) \leq c(1-h)^k (L_n^t(x_0) - L_n^t(x^*)) \tag{8}$$

where x_k and x^* represent the model of the k-th local iteration and optimal solution of (5), respectively. Both c and h are constants, whose values depend on ρ. According to [14], the problem in (5) can be solved eventually while satisfying (7) in Q_l rounds, which is represented as

$$Q_l = \frac{2}{h} \log \frac{c\rho}{\theta} \tag{9}$$

2.1.2 Global model updates

The local model ω_n^t and gradient $\nabla F_n(\omega_n^t)$ at each edge device are sent to the central server for aggregating a new global model and gradient, which are denoted respectively as

$$\omega^t = \frac{\sum_{n \in \mathbf{N}} d_n \omega_n^t}{\sum_{n \in \mathbf{N}} d_n} \tag{10}$$

$$\nabla F^t = \frac{\sum_{n \in \mathbf{N}} d_n \nabla F_n(\omega_n^t)}{\sum_{n \in \mathbf{N}} d_n} \tag{11}$$

The central server then broadcasts them to all devices. This process is repeated in Q_g rounds until the global model in (1) can reach convergence for an arbitrary small constant $\varepsilon > 0$ and when (12) is satisfied

$$F(\omega^t) - F(\omega^*) \leq \varepsilon, \forall t \geq Q_g \tag{12}$$

where ω^* is the optimal solution to (1). And the number of global rounds required for the global model to reach this convergence condition is

$$Q_g = \frac{1}{\theta}\log\frac{F(\omega^0)-F(\omega^*)}{\varepsilon} \tag{13}$$

the proof of this conclusion can also be found in [14].

2.2 Computation Model and Communicaton Model

In the local model updating phase, the computation delay and energy consumption generated by a single local training of device n are respectively

$$T_n^{cp} = \frac{c_n d_n}{f_n} \tag{14}$$

$$E_n^{cp} = c_n d_n f_n^2 \sigma \tag{15}$$

where c_n and f_n denote the number of CPU cycles required for device n to compute a data unit and its operating frequency, respectively. Let σ denote the effective capacitance coefficient of the devices' computation module [26].

The communication process includes two stages, uplink and downlink, for local model uploading and global model broadcasting, respectively. Since the power and bandwidth of central server are much larger than that of the devices, the downlink broadcasting delay can be ignored compared with the uplink transmission delay. All devices upload models based on the orthogonal frequency division multiple access protocol after completing local training. According to Shannon formula, the data transmission rate of device n is denoted as

$$R_n = b_n B\log_2\left(1+\frac{p_n h_n}{b_n B N_0}\right) \tag{16}$$

where B represents the total bandwidth of the central server, $b_n \in (0,1)$ is the proportion of bandwidth allocated by the central server for device n in each round of communication. p_n and h_n denote the transmission power and channel gain of device n, respectively. And N_0 is the noise power spectral density. Then, the transmission delay and energy consumption incurred by device n during a communication round are respectively

$$T_n^{co} = \frac{s_n}{R_n} \tag{17}$$

$$E_n^{co} = p_n T_n^{co} = \frac{p_n s_n}{b_n B\log_2\left(1+\frac{p_n h_n}{b_n B N_0}\right)} \tag{18}$$

where s_n represents the data size of the model parameter ω_n and gradient $\nabla F_n(\omega_n)$ uploaded by device n.

2.3 Unified Model

Note that the global model can only be updated after all local models have been received, which can be presented as

$$T_g = \max_{n\in\mathbf{N}}(T_n^{co} + Q_l T_n^{cp}) = \max_{n\in\mathbf{N}}\left\{\frac{s_n}{b_n B\log_2\left(1+\frac{p_n h_n}{b_n B N_0}\right)} + \frac{Q_l c_n d_n}{f_n}\right\} \tag{19}$$

We should note that T_g must satisfy the QoS requirement Q_T, which is expressed as

$$\max_{n\in\mathbf{N}}\left\{\frac{s_n}{b_n B\log_2\left(1+\frac{p_n h_n}{b_n B N_0}\right)} + \frac{Q_l c_n d_n}{f_n}\right\} \le Q_T \tag{20}$$

and it can be transformed into

$$\frac{s_n}{b_n B\log_2\left(1+\frac{p_n h_n}{b_n B N_0}\right)} + \frac{Q_l c_n d_n}{f_n} \le Q_T, \forall n \in \mathbf{N} \tag{21}$$

Similarly, the energy consumption of all devices in a single FL round can be calculated as

$$E_g = \sum_{n\in\mathbf{N}}(E_n^{co} + Q_l E_n^{cp}) = \sum_{n\in\mathbf{N}}\left[\frac{p_n s_n}{b_n B\log_2\left(1+\frac{p_n h_n}{b_n B N_0}\right)} + Q_l c_n d_n f_n^2 \sigma\right] \tag{22}$$

To facilitate the description of the problem below, let $\boldsymbol{b} = \{b_1, b_2, ..., b_N\}$ denote the set of bandwidth proportion allocated by the center server for all devices in a round of FL communication. And let $\boldsymbol{f} = \{f_1, f_2, ..., f_N\}$ denote the set of CPU frequencies of all devices.

2.4 Problem Formulation

Apparently, the goal of minimizing system latency and energy consumption cannot be satisfied at the same time. For example, to decrease the training delay, edge device needs to operate at high frequency, which also leads to higher energy consumption. Thus, the problem we will address is minimizing the energy consumption at the same time meeting the delay constraint, which is given by

$$\boldsymbol{P}_0 : \min_{\boldsymbol{b},\boldsymbol{f}} \sum_{n\in\mathbf{N}}\left[\frac{p_n s_n}{b_n B\log_2\left(1+\frac{p_n h_n}{b_n B N_0}\right)} + Q_l c_n d_n f_n^2 \sigma\right] \tag{23}$$

$$s.t. \quad b_n \in (0,1), \forall n \in \mathsf{N} \tag{23a}$$

$$f_n^{\min} \le f_n \le f_n^{\max}, \forall n \in \mathsf{N} \tag{23b}$$

$$p_n^{\min} \le p_n \le p_n^{\max}, \forall n \in \mathsf{N} \tag{23c}$$

$$\sum_{n \in \mathsf{N}} b_n = 1 \tag{23d}$$

$$\frac{s_n}{b_n B \log_2\left(1+\frac{p_n h_n}{b_n B N_0}\right)} + \frac{Q_l c_n d_n}{f_n} \le Q_T, \forall n \in \mathsf{N} \tag{23e}$$

where constraint (23a) represents the proportion of bandwidth allocated by the central server for the device n in one round of communication. The ranges of CPU frequency and transmission power of each device are indicated by constraints (23b) and (23c), respectively. Constraint (23d) represents that the bandwidth of the central server is fully exploited in each round of communication. Constraint (23e) indicates the QoS requirement, i.e., delay constraint.

3. Solutions to Problem

In this section, we present an FL approach based on CPU frequency control and bandwidth allocation aiming at the issue of minimizing energy consumption under delay constraint.

Apparently, $\boldsymbol{P}_0$ is challenging because the variables b_n and f_n are coupled with each other in (23e). Thus, we decompose it into two subproblems: the energy consumption minimization for global model aggregation and local model training, which are respectively expressed as

$$\boldsymbol{P}_1 : \min_{\boldsymbol{b}} \sum_{n \in \mathsf{N}} \frac{p_n s_n}{b_n B \log_2\left(1+\frac{p_n h_n}{b_n B N_0}\right)} \tag{24}$$

$$s.t. \quad (23a), (23c), (23d), (23e) \tag{24a}$$

$$\boldsymbol{P}_2 : \min_{\boldsymbol{f}} \sum_{n \in \mathsf{N}} Q_l c_n d_n f_n^2 \sigma \tag{25}$$

$$s.t. \quad (23b), (23e) \tag{25a}$$

For simplicity of the problem, the transmit power p_n of each device takes a uniform constant value p_n^*. And the size of model parameters s_n is constant. Obviously, $\boldsymbol{P}_0$ is a problem of bandwidth allocation under multiple bounds by fixing the CPU frequency in $\boldsymbol{P}_0$, which can be transformed into

$$\boldsymbol{P}_1' : \min_{\boldsymbol{b}} \sum_{n \in \mathsf{N}} \frac{p_n^* s_n}{b_n B \log_2\left(1+\frac{p_n^* h_n}{b_n B N_0}\right)} \tag{26}$$

$$\frac{s_n}{b_n B \log_2\left(1+\frac{p_n^* h_n}{b_n B N_0}\right)} + \frac{Q_l c_n d_n}{f_n^*} \le Q_T \tag{26a}$$

Then, a bounded storage quasi-newton approach called L-BFGS-B is utilized to solve it [27]. It is the extension of the L-BFGS algorithm, where the scale matrix only stores the information of the most recent iterations, and the stored information of the matrix is updated after an iteration is completed, which greatly reduces the computation memory. However, it is only suitable for solving unconstrained optimization problems. Targeting this limitation, L-BFGS-B employs strategies such as backtracking and line search with a limited maximum step size to solve constrained optimization problems. Thus, it has the advantages of reasonable memory requirement, small iteration cost and fast computation.

For the convenience of description, we transform $\boldsymbol{P}_1^{'}$ into a function with $\boldsymbol{b}$ as the independent variable, denoted by $f(\boldsymbol{b})$. The specific flow of the L-BGFS-B algorithm is as follows:

step 1: Set the initial value $\boldsymbol{b}^0$ and integer m that determines the number of finite memory correction stores, define the initial finite memory matrix and let $k=0$.

step 2: Calculate the gradient using the chain derivation rule.

$$\nabla f(\boldsymbol{b}) = \sum_{n\in \mathsf{N}} \frac{\partial f}{\partial b_n} \tag{27}$$

step 3: Calculate the search direction by the direct method.

$$p_k = -(H_k \nabla f(\boldsymbol{b}^k)) \tag{28}$$

where H_k denotes the Hessian matrix of the function in the k-th iteration.

step 4: A line search is performed along the direction of p_k, the step mediation factor α is computed, and the parameter is updated to find the minimum of the function.

$$\boldsymbol{b}^{k+1} = \boldsymbol{b}^k + \alpha_k p_k \tag{29}$$

step 5: Update the H_k and check for convergence.

The optimal solution $\boldsymbol{b}^*$ of $\boldsymbol{P}_1^{'}$ can be obtained by repeating the above steps until one of the following three conditions is met:

(i). The maximum number of iterations is reached.

(ii). The reduction of the objective function becomes smaller.

(iii): The paradigm of the projected gradient is sufficiently small.

Similarly, according to (31a), $\boldsymbol{P}_2$ can be converted into

$$\boldsymbol{P}_2^{'}: \min_{f} \sum_{n\in \mathsf{N}} Q_l c_n d_n f_n^2 \sigma \tag{30}$$

$$\max\{f_n^{\min}, f_n^*\} \le f_n \le f_n^{\max}, \forall n \in \mathsf{N} \tag{30a}$$

where

$$f_n^* = \frac{Q_l c_n d_n}{Q_T - \dfrac{s_n}{b_n^* B \log_2\left(1+\dfrac{p_n^* h_n}{b_n^* B N_0}\right)}} \tag{31}$$

here, $f_n = \max\{f_n^{\min}, \tilde{f}_n\}$ is known as the closed-form solution of $\boldsymbol{P}_2^{'}$, which is referred to as the optimal CPU frequency in this article. Its proof is simple: since f_n is always positive and the objective function in $\boldsymbol{P}_2^{'}$ is monotonically increasing with respect to f_n. Hence, f_n

should be the minimum in its feasible domain (30a). In conjunction with the previous section, the detailed flow of the FCBAFL algorithm is shown in Algorithm 1.

Algorithm 1 FCBAFL

Input: $\omega^0, \theta \in [0,1], \eta > 0$
for t=1 to Q_g do
  initialize $\boldsymbol{b}^0, m = 0, k = 0$
  while not converge
    calculate $\nabla f(\boldsymbol{b})$ and p_k based on (27), (28)
    calculate α_k, update $\boldsymbol{b}^k$ based on (29) and H_k
    k++
  end while
  for n=1 to N do
    central server broadcasts global model to edge devices
    calculate f based on (30a), (31)
    local training until satisfying (7) in Q_l iterations
  end for
  edge devices send local models to central server
  local models are aggregated into a new global model based on (16), (17)
end for

4. Simulation to Results

In the simulation experiments, we implement the FCBAFL algorithm in PyTorch. The IIoT scenario considered in this article contains a central server and $N = 100$. The bandwidth B of the central server is 1MHz. Each device's channel gain is generated as Rayleigh fading and the path-loss model is $128.1+37.6 \log_{10} d[km]$ [28]. The noise power spectral density is -174dBm/MHz. The transmit power p_n of each device is a uniform constant, i.e., p_n^*=0.5w, and the effective capacitance coefficient σ=10^{-23}w. The amount of data on each device d_n follows a uniform distribution of [5,10] MB, and they are randomly divided, with 75% for training and 25% for testing. The model parameters and gradient size s_n uploaded by each device are about 5×10^5bits. c_n is the number of CPU cycles needed by the device to compute a unit of data, follows a uniform distribution of [10, 30] cycles/bit, and f_n is the CPU frequency of device, which follows a uniform distribution of [1, 2] GHz. In the following experiments, we first evaluate the FCBAFL proposed in this paper in comparison with the classical FedAvg [16] algorithm based on MNIST dataset [29].

4.1 FL Performance Comparison

Fig. 2 and **Fig. 3** validate the effect of different parameters on the testing accuracy, both algorithms use uniform values for the number of local iterations $Q_l = 20$ and the number of

global communication rounds $Q_g = 500$. **Fig. 2** demonstrates the effect of different batchsize and η on the convergence speed of both algorithms, while **Fig. 3** further validates the effect of η on FCBAFL by fixing the batchsize=∞.

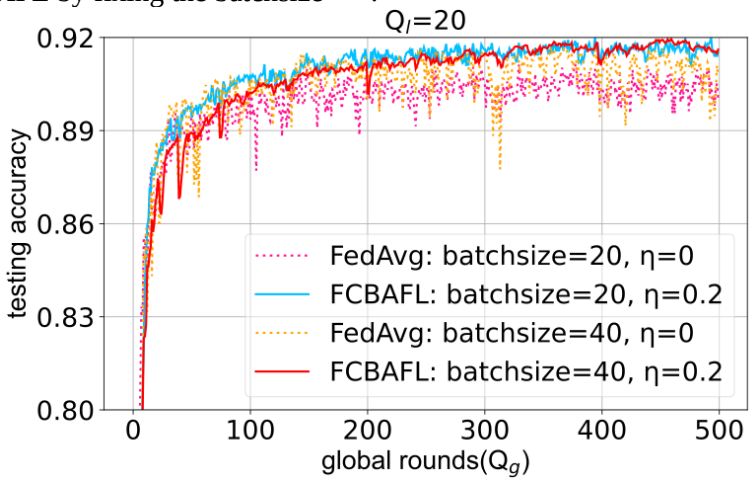


Fig. 2. Comparison of testing accuracy.

In **Fig. 2**, we first analyze the effect of batchsize on the convergence performance by observing two dashed lines and two solid lines, respectively. It can be seen that the smaller the batchsize is, the faster the convergence of both algorithms will be. This is because when the amount of data is constant, training with a smaller batchsize requires more frequent model updates to adapt to changes in the training data more quickly. And when the batchsize is fixed, e.g., batchsize=20 or 40, FCBAFL has better performance than FedAvg. This is because FCBAFL introduces the hyperparameter η, which enables a better trade-off between the local gradient and the global gradient. We also found that an increase in the value of η also makes FCBAFL converge faster, this is because a larger η implies that the update of the local gradient is closer to the global gradient, and equation (6) illustrates this phenomenon well.

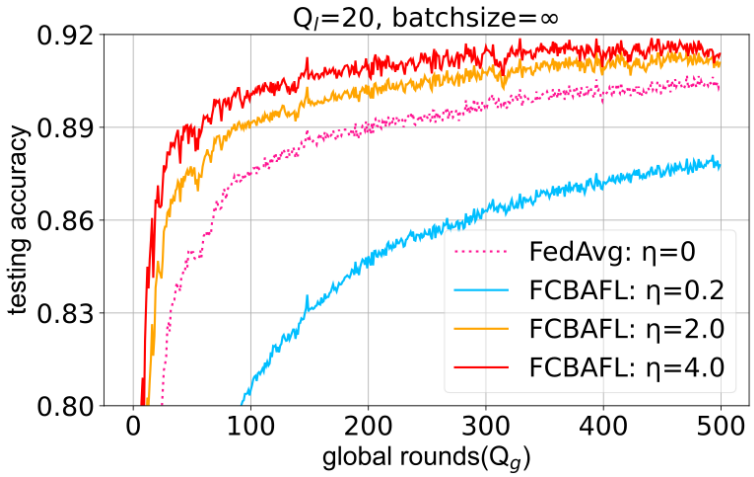


Fig. 3. Comparison of testing accuracy.

In **Fig. 3**, both methods use full-batch training, i.e., fixing batchsize=∞ again verifies the effect of the hyperparameter η on FCBAFL, especially when η takes a smaller value such as 0.2, FCBAFL does not perform as well as FedAvg, but we can still improve the situation by increasing the value of η. We also notice another phenomenon, compared with **Fig. 2**, the four curves in **Fig. 3** oscillate significantly less. This is because the small number of data samples in the small batch training leads to noisy gradient estimation, and the local gradient is prone to deviate from the global gradient thus causing instability in the training process. While the full-batch training has more data samples in each batch thus the training process is smoother.

Fig. 4 and **Fig. 5** depict the effect of different parameters on the training loss of both FL algorithms when using the same comparison settings as **Fig. 2** and **Fig. 3**. Similarly, **Fig. 4** demonstrates the effect of different batchsize and η on the convergence speed of both algorithms, while **Fig. 5** further validates the effect of η on FCBAFL by fixing the batchsize=∞.

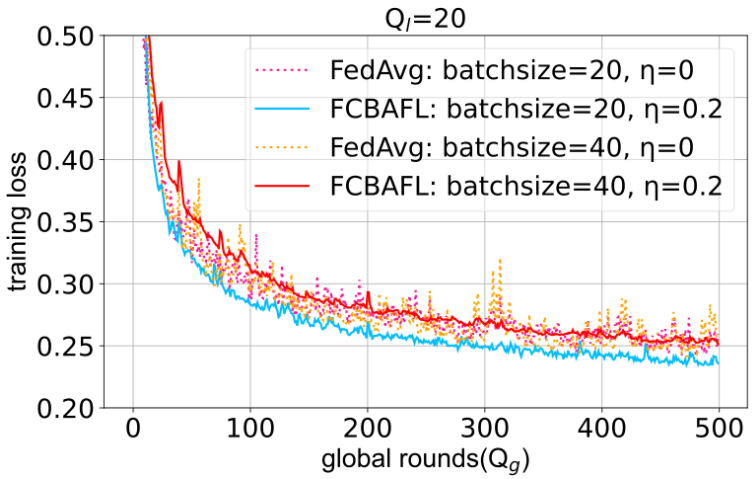


Fig. 4. Comparison of training loss.

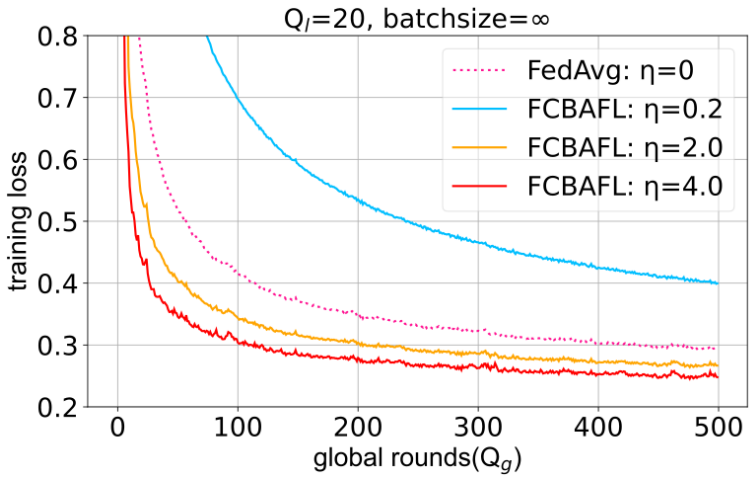


Fig. 5. Comparison of training loss.

Fig. 4 and **Fig. 5** present the same results as **Fig. 2** and **Fig. 3**. The introduction of the hyperparameter η leads to a better convergence performance of FCBAFL than FedAvg.

4.2 Energy Consumption Comparison

We further demonstrate the advantages of the FCBAFL method proposed in reducing resource consumption in **Fig. 6**-**Fig. 8**. **Fig. 6** compares the energy consumption generated by several frequency control strategies under different delay constraints when the number of devices is constant. **Fig. 7** depicts the variation of energy consumption produced by various frequency control strategies with the number of devices. **Fig. 8** verifies the variation of energy consumption produced by different bandwidth allocation strategies with the number of devices.

Since FCBAFL combines frequency control and bandwidth allocation, both the optimal frequency (OF) strategy and the optimal bandwidth allocation (OA) strategy in the following experiments aim at verifying the advantages of FCBAFL in reducing energy consumption. In this regard, we perform three experiments.

(1). When the number of devices is fixed, adopting the strategy of average bandwidth allocation, all devices train at different CPU frequencies under different delay constraints.

The three CPU frequencies are as follows:

(a). Random Frequency (RF): All devices take random values within the feasible range of CPU frequencies, i.e., $f_n \in [f_n^{\min}, f_n^{\max}], \forall n \in \mathbf{N}$.

(b). Average Frequency (AF): All devices perform the training task at an average frequency, i.e., $f_n = (f_n^{\min} + f_n^{\max})/2, \forall n \in \mathbf{N}$.

(c). Optimal Frequency (OF): All devices use the optimal CPU frequency according to equation (31), i.e., $f_n = \max\{f_n^{\min}, f_n^*\}, \forall n \in \mathbf{N}$.

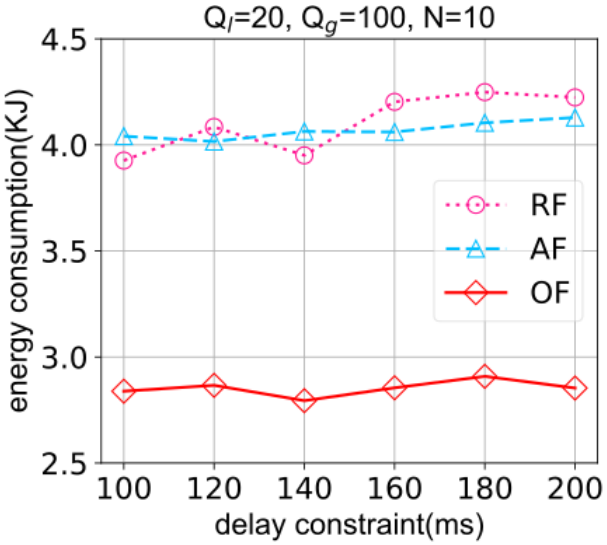


Fig. 6. Comparison of energy consumption generated by various CPU frequencies under different delay constraints.

Fig. 6 reflects the impact of various CPU frequencies on the energy consumption under different delay constraints. It can be seen that RF is inferior to AF in most cases due to its randomness and OF always produces lower energy consumption. In detail, OF reduces the

total energy consumption by about 18.3% and 17.9% compared with RF and AF, respectively.

(2). Adopt the strategy of average bandwidth allocation, all devices train at different CPU frequencies under different number of devices.

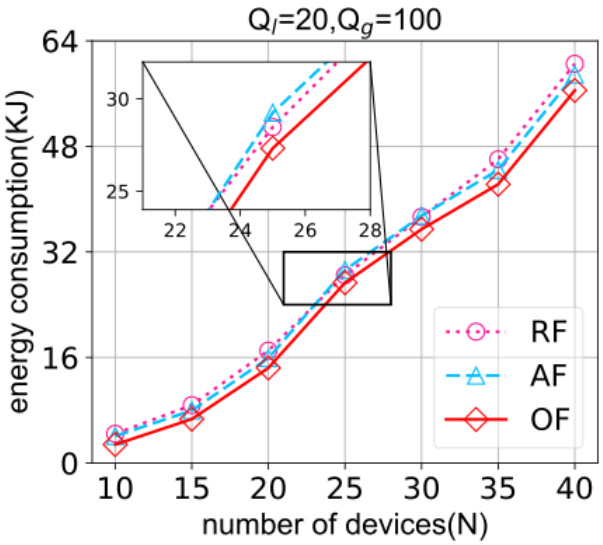


Fig. 7. Comparison of energy consumption generated by various CPU frequencies under different number of devices.

Fig. 7 depicts the effect of various CPU frequencies on the energy consumption under different number of devices. OF still performs best in all three strategies. The other two frequencies produce almost the same energy consumption, which is due to the fact that in practical circumstances, the communication energy consumption is usually greater compared with the training energy consumption. The strategy of average allocation is applied uniformly, so the difference among three schemes is very small. However, compared with RF and AF, OF still reduces the total energy consumption by 10.1% and 7.8%, respectively.

(3). All devices train at the optimal CPU frequency, different bandwidth allocation strategies are adopted under different numbers of devices.

The four bandwidth allocation strategies are as follows:

(a). Random Allocation (RA): The central server randomly allocates bandwidth to the devices, i.e., $b_n \in (0,1), \forall n \in \mathbf{N}$.

(b). Average Allocation (AA): The central server allocates the same bandwidth to all devices, i.e., $b_n = 1/N, \forall n \in \mathbf{N}$.

(c). SNR-Based Allocation (SA) [30]: The central server allocates bandwidth according to the signal-to-noise-ratio(SNR) of device, i.e., $b_n = (p_n h_n / N_0) / \sum_{n \in \mathbf{N}} (p_n h_n / N_0), \forall n \in \mathbf{N}$.

(d). Optimal Allocation (OA): The optimal solution $\boldsymbol{b}^*$ which is obtained by the L-BFGS-B algorithm.

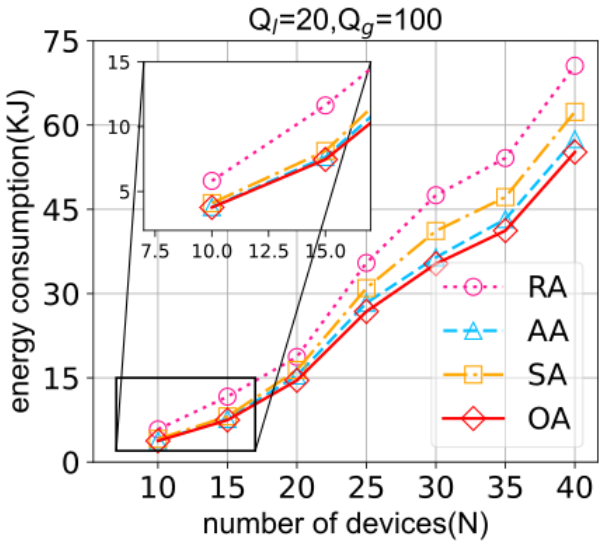


Fig. 8. Comparison of energy consumption generated by various bandwidth allocation strategies under different number of devices.

Fig. 8 illustrates the impact of different bandwidth allocation policies on the energy consumption under different number of devices. All four curves are trending upward. However, RA strategy does not consider the device's performance and channel condition at all, resulting in low resource utilization and thus increases energy consumption. AA strategy allocates the bandwidth equally, which may lead to overuse of resource by some devices and lack of resource by others. SA strategy allocates bandwidth according to the device's channel condition, but it does not consider the local training process, so the overall energy consumption is still very high. OA strategy optimizes the objective function by considering the delay constraint generated by local training and model transmission, thus achieving the optimal allocation and generating the lowest energy consumption [31]. Compared with RA, AA and SA, OA strategy reduces about 18.9%, 7.9% and 2.6% of the total energy consumption, respectively.

5. Conclusion

In this article, we propose an iterative FL approach for the wireless IIoT scenario, which is called FCBAFL. Its excellent convergence is certified theoretically and analytically by designing a local proxy function for each edge device to solve the local problem. Next, we present an objective of minimizing energy consumption while satisfying delay constraint based on the consideration of synchronous communication. And we describe it as a joint optimization problem of CPU frequency control and bandwidth allocation, which is subsequently split into two simple subproblems to be solved separately. Simulation results demonstrate that FCBAFL performs better convergence than baseline algorithm and reduces the energy consumption to a certain extent.

Acknowledgements

This work was supported by the Guangxi Key Research and Development Program under Grants AB23075175 and AB23026034, National Natural Science Foundation of China under Grant 62341117, Guangxi Natural Science Foundation under Grant 2024GXNSFAA010458, and the Innovation Project of Guangxi Graduate Education under Grant YCSW2024362.

References

[1] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proceedings of the IEEE*, vol.107, no.8, pp.1738-1762, Aug. 2019. Article(CrossRef Link)

[2] S. K. Jagatheesaperumal, M. Rahouti, K. Ahmad, A. Al-Fuqaha, and M. Guizani, "The Duo of Artificial Intelligence and Big Data for Industry 4.0: Applications, Techniques, Challenges, and Future Research Directions," *IEEE Internet of Things Journal*, vol.9, no.15, pp.12861-12885, Aug. 2022. Article(CrossRef Link)

[3] L. Yang, X. Chen, S. M. Perlaza, and J. Zhang, "Special Issue on Artificial-Intelligence-Powered Edge Computing for Internet of Things," *IEEE Internet of Things Journal*, vol.7, no.10, pp.9224-9226, Oct. 2020. Article(CrossRef Link)

[4] M. A. Rahman, M. S. Hossain, A. J. Showail, N. A. Alrajeh, and A. Ghoneim, "AI-Enabled IIoT for Live Smart City Event Monitoring," *IEEE Internet of Things Journal*, vol.10, no.4, pp.2872-2880, Feb. 2023. Article(CrossRef Link)

[5] Y. Chi, Y. Dong, Z. J. Wang, F. R. Yu, and V. C. M. Leung, "Knowledge-Based Fault Diagnosis in Industrial Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol.9, no.15, pp.12886-12900, Aug. 2022. Article(CrossRef Link)

[6] X. Deng, J. Li, C. Ma et al., "Low-Latency Federated Learning With DNN Partition in Distributed Industrial IoT Networks," *IEEE Journal on Selected Areas in Communications*, vol.41, no.3, pp.755-775, Mar. 2023. Article(CrossRef Link)

[7] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards Secure Industrial IoT: Blockchain System With Credit-Based Consensus Mechanism," *IEEE Transactions on Industrial Informatics*, vol.15, no.6, pp.3680-3689, Jun. 2019. Article(CrossRef Link)

[8] T. Hafeez, L. Xu, and G. Mcardle, "Edge Intelligence for Data Handling and Predictive Maintenance in IIOT," *IEEE Access*, vol.9, pp.49355-49371, Mar. 2021. Article(CrossRef Link)

[9] P. Zhang, P. Gan, G. S. Aujla, and R. S. Batth, "Reinforcement Learning for Edge Device Selection Using Social Attribute Perception in Industry 4.0," *IEEE Internet of Things Journal*, vol.10, no.4, pp.2784-2792, Feb. 2023. Article(CrossRef Link)

[10] Z. Tong, J. Cai, J. Mei, K. Li, and K. Li, "Dynamic Energy-Saving Offloading Strategy Guided by Lyapunov Optimization for IoT Devices," *IEEE Internet of Things Journal*, vol.9, no.20, pp.19903-19915, Oct. 2022. Article(CrossRef Link)

[11] Y. Liu, Z. Ma, Y. Yang, X. Liu, J. Ma, and K. Ren, "RevFRF: Enabling Cross-Domain Random Forest Training With Revocable Federated Learning," *IEEE Transactions on Dependable and Secure Computing*, vol.19, no.6, pp.3671-3685, Nov.-Dec. 2022. Article(CrossRef Link)

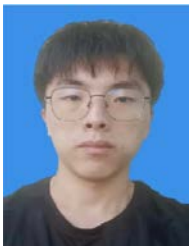
[12] K. Jung, I. Baek, S. Kim, and Y. D. Chung, "LAFD: Local-Differentially Private and Asynchronous Federated Learning With Direct Feedback Alignment," *IEEE Access*, vol.11, pp.86754-86769, 2023. Article(CrossRef Link)

[13] X. Cao, G. Sun, H. Yu, and M. Guizani, "PerFED-GAN: Personalized Federated Learning via Generative Adversarial Networks," *IEEE Internet of Things Journal*, vol.10, no.5, pp.3749-3762, Mar. 2023. Article(CrossRef Link)

[14] C. T. Dinh, N. H. Tran, M. N. H. Nguyen et al., "Federated Learning Over Wireless Networks: Convergence Analysis and Resource Allocation," *IEEE/ACM Transactions on Networking*, vol.29, no.1, pp.398-409, Feb. 2021. Article(CrossRef Link)

[15] J. Lu, H. Liu, R. Jia, J. Wang, L. Sun, and S. Wan, "Toward Personalized Federated Learning Via Group Collaboration in IIoT," *IEEE Transactions on Industrial Informatics*, vol.19, no.8, pp.8923-8932, Aug. 2023. Article(CrossRef Link)
[16] J. Mills, J. Hu, and G. Min, "Communication-Efficient Federated Learning for Wireless Edge Intelligence in IoT," *IEEE Internet of Things Journal*, vol.7, no.7, pp.5986-5994, Jul. 2020. Article(CrossRef Link)
[17] Z. Zhao, C. Feng, W. Hong et al., "Federated Learning With Non-IID Data in Wireless Networks," *IEEE Transactions on Wireless Communications*, vol.21, no.3, pp.1927-1942, Mar. 2022. Article(CrossRef Link)
[18] X. You, X. Liu, N. Jiang, J. Cai, and Z. Ying, "Reschedule Gradients: Temporal Non-IID Resilient Federated Learning," *IEEE Internet of Things Journal*, vol.10, no.1, pp.747-762, Jan. 2023. Article(CrossRef Link)
[19] S. Park, and W. Choi, "Regulated Subspace Projection Based Local Model Update Compression for Communication-Efficient Federated Learning," *IEEE Journal on Selected Areas in Communications*, vol.41, no.4, pp.964-976, Apr. 2023. Article(CrossRef Link)
[20] S. Ji, W. Jiang, A. Walid, and X. Li, "Dynamic Sampling and Selective Masking for Communication-Efficient Federated Learning," *IEEE Intelligent Systems*, vol.37, no.2, pp.27-34, Mar.-Apr. 2022. Article(CrossRef Link)
[21] J. Shu, W. Zhang, Y. Zhou, Z. Cheng, and L. T. Yang, "FLAS: Computation and Communication Efficient Federated Learning via Adaptive Sampling," *IEEE Transactions on Network Science and Engineering*, vol.9, no.4, pp.2003-2014, Jul.-Aug. 2022. Article(CrossRef Link)
[22] M. Alishahi, P. Fortier, W. Hao, X. Li, and M. Zeng, "Energy Minimization for Wireless-Powered Federated Learning Network With NOMA," *IEEE Wireless Communications Letters*, vol.12, no.5, pp.833-837, May 2023. Article(CrossRef Link)
[23] J. Kim, D. Kim, J. Lee, and J. Hwang, "A Novel Joint Dataset and Computation Management Scheme for Energy-Efficient Federated Learning in Mobile Edge Computing," *IEEE Wireless Communications Letters*, vol.11, no.5, pp.898-902, May 2022. Article(CrossRef Link)
[24] E. Malan, V. Peluso, A. Calimera, and E. Macii, "Communication-Efficient Federated Learning With Gradual Layer Freezing," *IEEE Embedded Systems Letters*, vol.15, no.1, pp.25-28, Mar. 2023. Article(CrossRef Link)
[25] Y. Nesterov, Lectures on Convex Optimization (2nd. ed.), Springer Publishing Company, SOIA, vol.137, 2018. Article(CrossRef Link)
[26] X. Ji, J. Tian, H. Zhang, D. Wu, and T. Li, "Joint Device Selection and Bandwidth Allocation for Cost-Efficient Federated Learning in Industrial Internet of Things," *IEEE Internet of Things Journal*, vol.10, no.10, pp.9148-9160, May 2023. Article(CrossRef Link)
[27] L. Cantos, M. Awais, and Y. H. Kim, "Max-Min Rate Optimization for Uplink IRS-NOMA With Receive Beamforming," *IEEE Wireless Communications Letters*, vol.11, no.12, pp.2512-2516, Dec. 2022. Article(CrossRef Link)
[28] B. Zhong, Z. Zhang, D. Zhang, K. Long, and H. Cao, "Partial Relay Selection in Decode and Forward Cooperative Cognitive Radio Networks over Rayleigh Fading Channels," *KSII Transactions on Internet and Information Systems*, vol.8, no.11, pp.3967-3983, 2014. Article(CrossRef Link)
[29] L. Cui, J. Ma, Y. Zhou, and S. Yu, "Boosting Accuracy of Differentially Private Federated Learning in Industrial IoT With Sparse Responses," *IEEE Transactions on Industrial Informatics*, vol.19, no.1, pp.910-920, Jan. 2023. Article(CrossRef Link)
[30] X. Wang, S. Lv, X. Wang, and Z. Zhang, "Greedy Heuristic Resource Allocation Algorithm for Device-to-Device Aided Cellular Systems with System Level Simulations," *KSII Transactions on Internet and Information Systems*, vol.12, no.4, pp.1415-1435, 2018. Article(CrossRef Link)
[31] B. Qiu, X. Chang, X. Li, H. Xiao and Z. Zhang, "Federated Learning-Based Channel Estimation for RIS-Aided Communication Systems," *IEEE Wireless Communications Letters*, vol.13, no.8, Aug. 2024. Article(CrossRef Link)

Bin Qiu received the B.S. and M.S. degrees from the School of Computer and Electronic Information, Guangxi University, Nanning, China, in 2010 and 2013, respectively, and the Ph.D. degree from the Guilin University of Electronic Technology, Guilin, China, in 2020. He is currently an associate professor with the School of Computer Science and Engineering, Guilin University of Technology, Guilin, China. His research interests include vehicular communications and resource allocation in wireless communications.



Duan Li received the B.S. degree from the Shanxi University, Taiyuan, China, in 2022. He is currently working toward the M.S. degree in Computer Science and Engineering from the Guilin University of Technology, Guilin, China. His research interest includes computation offloading and resource allocation in industrial internet of things.

Xian Li received his B.S. degree and M.S. degree in the School of Computer and Electronic Information in Guangxi University, Nanning, China, in 2010 and 2013, respectively. He received his Ph.D. degree from Southeast University, Nanjing, China, in 2019. He is now a research associate with the College of Electronics and Information Engineering, Shenzhen University, China. He received the Best Paper Award of IEEE/CIC ICCC 2021. His research interests mainly include optimizations in mobile edge computing and wireless powered communication networks.

Hailin Xiao received the B.S. degree from Wuhan University, Wuhan, China, in 2004, the M.S. degree Guangxi Normal University, Guilin, China, in 1998, and the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2007. He is currently a Professor with the School of Computer Science and Information Engineering, Hubei University, Wuhan. He was a Research Fellow with the Joint Research Institute for Signal and Image Processing, School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, U.K., form January 2011 to February 2012. He was also a Research Fellow with the School of Electronics and Computer Science, University of Southampton, Southampton, U.K., from March 2016 to March 2017. He has published two books and over 200 papers in refereed journals and conference proceedings. His research interests include MIMO wireless communications, cooperative communications, and vehicular communication. Dr. Xiao has received Guangxi Natural Science Foundation for Distinguished Young Scholars, a Distinguished Professor of the Qianjiang Scholars, and a Distinguished Professor of the Chutian Scholars, China, in 2014, 2018, and 2020, respectively.