IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 November 2022, accepted 5 December 2022, date of publication 8 December 2022, date of current version 28 February 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3227561

RESEARCH ARTICLE

Semi-Asynchronous Hierarchical Federated Learning Over Mobile Edge Networks

QIMEI CHEN, (Member, IEEE), ZEHUA YOU, JING WU, YUNPENG LIU, AND HAO JIANG
School of Electronic Information, Wuhan University, Wuhan 430072, China

Corresponding author: Jing Wu (wujing@whu.edu.cn)

This work was supported in part by the International Cooperation Project of Hubei Province under Grant 2022EHB014, in part by the Fundamental Research Funds for the Central Universities under Grant 2042022gf0019, in part by the Young Elite Scientists Sponsorship Program by CAST 2019, in part by the Wuhan Knowledge Innovation Special Basic Research Project under Grant 2022020801010110, and in part by the National Natural Science Foundation of China Enterprise Innovation Development Key Project under Grant U19B2004.

ABSTRACT Mobile edge network has been recognized as a promising technology for future wireless communications. However, mobile edge networks usually gathering large amounts of data, which makes it difficult to explore data science efficiently. Currently, federated learning has been proposed as an appealing approach to allow users to cooperatively reap the benefits from trained participants. In this paper, we propose a novel *Semi-Asynchronous Hierarchical Federated Learning* (SAHFL) framework for mobile edge networks that enables elastic edge to cloud model aggregation from data sensing. We further formulate a joint edge node association and resource allocation problem under the proposed SAHFL framework to prevent personalities of heterogeneous devices and achieve communication-efficiency. To deal with our proposed *Mixed integer nonlinear programming* (MINLP) problem, we introduce a distributed *Alternating Direction Method of Multipliers* (ADMM)-*Block Coordinate Update* (BCU) algorithm. With this algorithm, a tradeoff between training accuracy and transmission latency has been derived. Numerical results demonstrate the advantages of the proposed algorithm in terms of training overhead and model performance.

INDEX TERMS Edge association, federated learning, mobile edge network, resource allocation, semi-asynchronous.

I. INTRODUCTION

With the improvement of sensing and computing capability of mobile edge networks, the explosive growth of devices has generated a large amount of data [1]. The full utilization of these data will greatly facilitate the mobile edge network to provide secure and efficient needs for devices. However, since the traditional centralized data training method would increase the communication load and affect the data security, it is impractical for the mobile edge networks with large amount of data. Therefore, a new distributed machine learning paradigm named Federated Learning (FL) [2] is emerged that allows the device to complete the training process without uploading their raw data to the central server.

Currently, FL has been widely studied to deal with the data science in terminal devices [3], [4] and foster new applications such as medical diagnosis [5] and autonomous vehicles [6]. The FL technology allows participant devices to collaboratively build a shared model while preserving privacy data locally [7]. Particularly, the prevalent FL algorithm, namely federated averaging, allows each device to train a model locally with its own dataset, and then transmits the model parameters to the central controller for a global aggregation [2]. However, FL efficiency is severely degraded by limited communication resources. Furthermore, the participant devices in mobile edge networks usually have heterogeneous resources, which lead to *non-independent-identically distributed* (non-IID) private data during the communication [8], [9]. The existence of non-IID data creates the need for customized services for individual terminals. Learning a common model proposed by the traditional FL

The associate editor coordinating the review of this manuscript and approving it for publication was Tiago Cruz.

algorithm may produce mediocre performance on some terminals with large data imbalances. Intuitively, FL presents a great potential for mobile edge networks to facilitate the large data management. However, directly applying FL to mobile edge networks still faces three major deficiencies: 1) limited wireless resources; 2) high latency; 3) obliterated data diversity.

According to [10] and [11], *Federated Learning training at Edge networks* (FEL) has been regarded as a solution to facilitate the above limitations through bringing model training closer to the data produced locally. Compared with the conventional cloud centric FL approaches, the implementation of FEL can provide higher wireless resources utilization since less information is required to be transmitted to the cloud. In addition, FEL has a much lower transmission latency and higher privacy than the conventional FL by making decisions at the edge nodes. In [12], the authors develop an importance aware joint data selection and resource allocation algorithm to maximize the resource and learning efficiencies. Meanwhile, the authors in [13] propose an adaptive federated learning mechanism in resource constrained edge computing systems. Along the FEL, the authors in [14] propose a novel *Hierarchical Federated Edge Learning* (HFEL) framework, where edge servers deployed with base stations fixedly and can upload edge aggregation model to the cloud. The above HFEL enables great potentials in low latency and high energy efficiency.

Besides, due to the heterogeneity of devices, some authors propose to improve the efficiency of FL algorithm by changing the FL aggregation method. The existing federated learning methods mainly utilize the synchronous model aggregation mechanism, where the central server needs to wait for the slowest device to complete the training in each communication round [15], [16]. In the synchronous FL method, the edge server aggregates local models of all devices or a subset of pre-selected devices. In [17], the authors proposed a joint device association and wireless resource allocation scheme under IID and non-IID datasets, respectively. The authors in [18] proposed a novel device selection and resource allocation scheme under wireless resource fruitful unlicensed spectrum (NR-U) networks. However, in this case, the computing resources of those unselected devices are wasted. Besides, for heterogeneous data, the transmission latency of each synchronous model aggregation mechanism is unacceptable for time-sensitive devices. In this way, several works have proposed asynchronous model aggregation methods, where only one participant device would update the global model each time [19], [20], [21]. Meanwhile, when one device uploads its model, the others continue to complete their training. The authors in [22] proposed a novel asynchronous FL mechanism to coordinate the heterogeneity of devices, communication environments, and learning tasks. Nevertheless, the training round under asynchronous methods is higher than synchronous methods. Moreover, due to the asynchrony, gradient staleness may be difficult to control [20]. Therefore, the authors in [23] design an n-softsync aggregation model that can significantly reduce training time by combines the benefits of both synchronous and asynchronous aggregations.

Inspired by the above analyses, we aim to leverage a novel *Semi-Asynchronous Hierarchical Federated Learning* (SAHFL) framework that can provide secure and efficient services to mobile edge networks. Specifically, the proposed SAHFL framework consists of both edge and cloud layers, where each edge node aggregates all of homogeneous local models and the cloud layer aggregates parts of heterogeneous edge models. These selected nodes would update the global model once the selected slowest node finishes training, which combines the merits of both synchronous and asynchronous aggregations. For further performance enhancement, we formulate a joint edge node association and resource allocation optimization problem to prevent heterogeneous edge node personalities as well as ensure communication-efficient of the whole system. The objective function is a *Mixed Integer NonLinear Programming* (MINLP) problem, which has been solved by a distributed *Alternating Direction Method of Multipliers* (ADMM)-*Block Coordinate Update* (BCU) algorithm. It is shown that the proposed algorithm can achieve near optimal with low computational complexity. In addition, to protect the data diversity contribution required by edge nodes, we design an elastic edge update method before edge nodes broadcast the cloud model to devices.

Overall, the main contributions of this work can be listed as follows.

- We propose a novel SAHFL framework by applying the synchronous aggregation model for local-edge and the semi-asynchronous aggregation model for edge-cloud to provide secure and efficient services for mobile edge networks.
- To reserve the personalities of heterogeneous edge nodes, we introduce an elastic edge model update method based on the distance between the global model and the edge model.
- We formulate a joint edge node association and resource allocation problem to achieve communication-efficiency by achieving a tradeoff between training accuracy and transmission latency. A distributed ADMM-BCU algorithm has been used to solve the MINLP problem.
- Under CIFAR-10 dataset, we found that our framework has a good performance in training accuracy and loss. The proposed algorithm can reduce the device latency, and the elastic edge model update method can well protect the personalized level of edge models.

The rest of this paper is organized as follows. Section II introduces the system model and the SAHFL learning mechanism. In Section III, we formulate the communication-efficient problem. A joint edge node association and resource allocation strategy is presented in Section IV. Section V presents the numerical results, followed by the conclusions in Section VI.

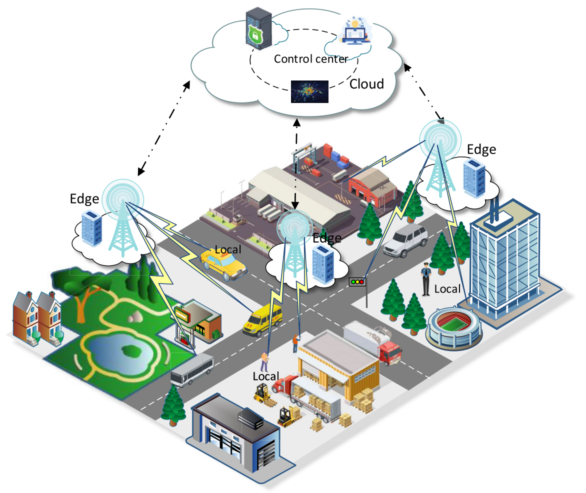


FIGURE 1. Illustration of the SAHFL based mobile edge networks.

II. SYSTEM MODEL

In this work, we aim to design a novel SAHFL framework for mobile edge networks that contains three layers, namely the cloud layer, the edge layer, and the local layer, as shown in Fig. 1. Here, we consider the devices have heterogeneous data structures, namely the local datasets are non-iid. We let homogeneous devices with similar data size, network bandwidth, and QoS gather in the same edge node. Hence, the edge nodes are heterogeneous. A shared *Deep Neural Network* (DNN) model is distributed over the local devices, which has been trained collaboratively across the devices under their datasets. Different from conventional FLs, the proposed SAHFL framework allows devices train their data locally, homogeneous devices report their computed parameters to the same edge node synchronously, and heterogeneous edge nodes upload their models to the cloud node semi-asynchronously, which can preserve data privacy as well as improve communication efficiency. In the proposed framework, we assume there has a set of K edge nodes $\mathcal{K} = \{1, \ldots, K\}$. Any edge node k consists of a set of N_k local devices, denoted as $\mathcal{N}_k = \{L_{k,1}, \ldots, L_{k,N_k}\}$. Under edge node $k \in \mathcal{K}$, local device $n \in \mathcal{N}_k$ owns a local data set $D_{k,n} = \{(\boldsymbol{x}_{j,k,n}, y_{j,k,n}) : j = 1, \ldots, |D_{k,n}|\}$, where $\boldsymbol{x}_{j,k,n}$ is the j-th input training data sample, $y_{j,k,n}$ is the j-th corresponding output, and $|D_{k,n}|$ denotes the cardinality of the data set $D_{k,n}$. For simplicity, we assume the SAHFL algorithm with a single output. However, this work can be extended to the multiple outputs case. In what follows, we would introduce each part of the proposed SAHFL framework at the t-th iteration.

A. EDGE AGGREGATION

The edge aggregation stage contains three processes, including local model computation, local model transmission, and edge model aggregation. In detail, local model first trained by local data, then local models respectively transmit to their associated edge nodes for edge aggregation. The detailed processes are as follows.

1) LOCAL MODEL COMPUTATION

Without loss of generality, we consider a supervised machine learning task on device $n \in \mathcal{N}_k$ associated with edge node $k \in \mathcal{K}$, which has a learning model of $\boldsymbol{w}_{k,n}$. We further define $f_n(\boldsymbol{x}_{j,k,n}, y_{j,k,n}, \boldsymbol{w}_{k,n})$ as the loss function of data sample j that quantifies the prediction error between data sample $\boldsymbol{x}_{j,k,n}$ and output $y_{j,k,n}$. In this work, we mainly focus on the logistic regression model for the loss function, i.e., $f_n(\boldsymbol{x}_{j,k,n}, y_{j,k,n}, \boldsymbol{w}_{k,n}) = -\log\left(1 + \exp\left(-y_{j,k,n}\boldsymbol{x}_{j,k,n}^{\mathrm{T}}\boldsymbol{w}_{k,n}\right)\right)$. Hence, the loss function of device $n \in \mathcal{N}_k$ associated with edge node $k \in \mathcal{K}$ on dataset $D_{k,n}$ can be defined as

$$F_{k,n}(\boldsymbol{w}_{k,n}) = \frac{1}{|D_{k,n}|}\sum_{j=1}^{|D_{k,n}|} f_{k,n}(\boldsymbol{x}_{j,k,n}, y_{j,k,n}, \boldsymbol{w}_{k,n}),$$
$$\forall k \in \mathcal{K}, n \in \mathcal{N}_k. \qquad (1)$$

The local update model of device $n \in \mathcal{N}_k$ in edge node $k \in \mathcal{K}$ can be achieved by

$$\boldsymbol{w}_{k,n}^{t} = \boldsymbol{w}_{k,n}^{t-1} - \eta \nabla F_{k,n}(\boldsymbol{w}_{k,n}^{t-1}),\ \forall k \in \mathcal{K}, n \in \mathcal{N}_k, \qquad (2)$$

where η is a predefined learning rate.

Define $C_{k,n}$ as the number of CPU cycles for local device $n \in \mathcal{N}_k$ associated with edge node $k \in \mathcal{K}$ to process one sample data. Assuming each sample data has the same size, the total CPU cycles to run one local iteration is $C_{k,n}|D_{k,n}|$. We further let $f_{k,n}$ be the computation frequency of device $n \in \mathcal{N}_k$ in edge node $k \in \mathcal{K}$. In this way, the related local gradient calculation latency in one round can be formulated as

$$T_{k,n}^{c} = \frac{C_{k,n}|D_{k,n}|}{f_{k,n}},\ \forall k \in \mathcal{K}, n \in \mathcal{N}_k. \qquad (3)$$

2) LOCAL MODEL TRANSMISSION

We adopt the *Orthogonal-Frequency-Division Multiple Access* (OFDMA) technique for local uplink transmissions. Define $B_{k,n}$ as the bandwidth allocated to device $n \in \mathcal{N}_k$. Therefore, we have $\sum_{n=1}^{\mathcal{N}_k} B_{k,n} = B_k$, where B_k is the bandwidth allocated to edge node $k \in \mathcal{K}$ for the transmission between edge node $k \in \mathcal{K}$ and the associated local devices. Meanwhile, we have $\sum_{k=1}^{K} B_k \leq B_e$, where B_e is the total bandwidth allocated for the communication between edge nodes to the local devices. Therefore, the achievable local uplink data rate from device $n \in \mathcal{N}_k$ to edge node $k \in \mathcal{K}$ can be formulated as

$$r_{k,n}^{u} = B_{k,n}\log_2\left(1 + \frac{P_{k,n}g_{k,n}}{B_{k,n}N_0}\right),\ \forall k \in \mathcal{K}, n \in \mathcal{N}_k, \qquad (4)$$

where $P_{k,n}$ is the uplink transmission power of device $n \in \mathcal{N}_k$ in edge node $k \in \mathcal{K}$, $g_{k,n}$ denotes the channel gain between local device $n \in \mathcal{N}_k$ and edge node $k \in \mathcal{K}$, and N_0 means the noise power.

Similarly, the achievable downlink data rate for device $n \in \mathcal{N}_k$ associated with edge node $k \in \mathcal{K}$ can be expressed as

$$r_{k,n}^{d} = B_k\log_2\left(1 + \frac{P_k g_{k,n}}{B_k N_0}\right),\ \forall k \in \mathcal{K}, n \in \mathcal{N}_k, \qquad (5)$$

where P_k is the downlink transmission power of edge node $k \in \mathcal{K}$.

In this work, we use the same training model for the whole communication system. Therefore, the number of model parameters in each level of model transfer has the same size. Denote Z as the data size of the model parameter bits. The local gradient upload latency of device $n \in \mathcal{N}_k$ in edge node $k \in \mathcal{K}$ can be expressed as

$$T_{k,n}^{u} = \frac{Z}{r_{k,n}^{u}} = \frac{Z}{B_{k,n}\log_2\left(1+\frac{P_{k,n}g_{k,n}}{B_{k,n}N_0}\right)}, \ \forall k \in \mathcal{K}, n \in \mathcal{N}_k. \tag{6}$$

Correspondingly, the edge model download latency of device $n \in \mathcal{N}_k$ in edge node $k \in \mathcal{K}$ can be formulated as

$$T_{k,n}^{d} = \frac{Z}{r_{k,n}^{d}} = \frac{Z}{B_k\log_2\left(1+\frac{P_k g_{k,n}}{B_k N_0}\right)}, \ \forall k \in \mathcal{K}, n \in \mathcal{N}_k. \tag{7}$$

3) EDGE MODEL AGGREGATION

In this work, each edge node can receive the updated model parameters from its associated homogeneous devices. Since the devices under one edge node usually have a similar type, we adopt the synchronous aggregation method to average these updated models. It means that the edge node would wait for the slowest node to complete training in each round and collect all the connected devices' updated model parameters. Therefore, the edge model aggregating equation for edge node $k \in \mathcal{K}$ can be formulated as

$$\boldsymbol{w}_k^t = \frac{\sum_{n=1}^{\mathcal{N}_k}|D_{k,n}|\boldsymbol{w}_{k,n}^t}{|D_k|}, \ \forall k \in \mathcal{K}, \tag{8}$$

where $|D_k| = \sum_{n=1}^{\mathcal{N}_k}|D_{k,n}|$ is the total number of data in edge node $k \in \mathcal{K}$.

We omit edge model aggregation time due to its strong computing capability. Similarly, due to the advantages of bandwidth and transmission power when edge devices broadcast, the edge model download latency can also be neglected Hence, the computation and communication latency between each edge $k \in \mathcal{K}$ and the related local devices can be derived as

$$T_k^{edge} = \max_{n\in\mathcal{N}_k}\left(T_{k,n}^{c} + T_{k,n}^{u}\right), \ \forall k \in \mathcal{K}. \tag{9}$$

B. CLOUD AGGREGATION

Similarly, the cloud aggregation stage contains two processes, i.e., edge model transmission and cloud model aggregation. Particularly, the selected edge nodes upload their updated model parameters to the cloud for aggregation. The detailed processes are as follows.

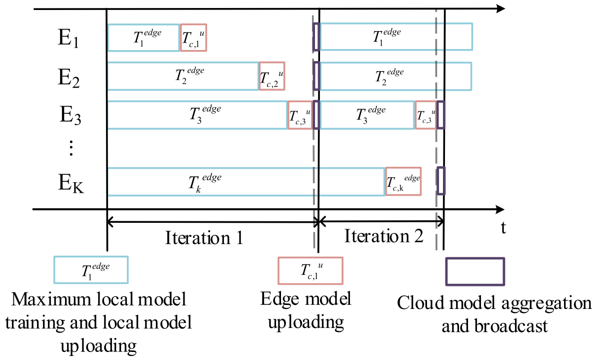


FIGURE 2. The proposed SAHFL process.

1) EDGE MODEL TRANSMISSION

Edge nodes would upload their model parameters to the cloud after edge model aggregations. To ensure uninterrupted transmission from edge to cloud, we also adopt the OFDMA technique. Hence, the uplink data rate for edge node $k \in \mathcal{K}$ can be expressed as

$$r_{c,k}^{u} = B_{c,k}\log_2\left(1+\frac{P_{c,k}g_{c,k}}{B_{c,k}N_0}\right), \ \forall k \in \mathcal{K}, \tag{10}$$

where $B_{c,k}$ is the bandwidth allocated to edge node $k \in \mathcal{K}$ transmits to the cloud node, $P_{c,k}$ is the uplink transmission power of edge node $k \in \mathcal{K}$ to the cloud node, and $g_{c,k}$ denotes the channel gain between edge node $k \in \mathcal{K}$ and the cloud node.

Correspondingly, the downlink data rate from the cloud node to edge node $k \in \mathcal{K}$ can be formulated as

$$r_{c,k}^{d} = B_c\log_2\left(1+\frac{P_c g_{c,k}}{B_c N_0}\right), \ \forall k \in \mathcal{K}, \tag{11}$$

where P_c is the downlink transmission power of the cloud node, B_c is the total bandwidth for the transmission between the edge nodes and the cloud. As we would discuss later, only parts of the edge nodes can be selected in each round. Therefore, we have the constraint of

$$\sum_{k=1}^{K}\alpha_k B_{c,k} \le B_c, \tag{12}$$

where $\alpha_k \in \{0, 1\}$. Here, $\alpha_k = 1, \forall k \in \mathcal{K}$ indicates edge node k has been selected, and $\alpha_k = 0, \forall k \in \mathcal{K}$ otherwise.

In this way, the upload latency from edge node $k \in \mathcal{K}$ to the cloud node can be written as

$$T_{c,k}^{u} = \frac{Z}{r_{c,k}^{u}} = \frac{Z}{B_{c,k}\log_2\left(1+\frac{P_{c,k}g_{c,k}}{B_{c,k}N_0}\right)}, \ \forall k \in \mathcal{K}. \tag{13}$$

Similarly, the downlink latency from the cloud node to edge node $k \in \mathcal{K}$ can be expressed as

$$T_{c,k}^{d} = \frac{Z}{r_{c,k}^{d}} = \frac{Z}{B_c\log_2\left(1+\frac{P_c g_{c,k}}{B_c N_0}\right)}, \ \forall k \in \mathcal{K}. \tag{14}$$

2) CLOUD MODEL AGGREGATION

Since these edge nodes correspond to heterogeneous local datasets, their model updated periods various. If we adopt the synchronous aggregation model, the latency for faster training nodes is unacceptable. On the contrary, the asynchronous method has shorter round latency, however, it requires several times of training rounds than the synchronous method. Therefore, in this work, we propose a flexible semi-asynchronous aggregation method by combining the merits of both synchronous and asynchronous methods. As shown in Fig. 2, the cloud node would select $|\mathcal{S}^t| = \sum_{k=1}^{K} \alpha_k$ edge nodes with the fastest training round for model aggregation, where the set of selected edge nodes is denoted as $\mathcal{S}^t$. Slow nodes would wait for the next communication round to upload their models. Hence, under the semi-asynchronous aggregation method, we can achieve a balance between training accuracy and communication latency. The semi-asynchronous aggregation method can be written as

$$\boldsymbol{w}_c^t = \boldsymbol{w}_c^{t-1} + \sum_{k \in \mathcal{S}^t} \frac{|D_k|}{\sum_{k=1}^{K} |D_k|} (\boldsymbol{w}_k^t - \boldsymbol{w}_c^{t-1}). \tag{15}$$

Also, we ignore the cloud model aggregation latency due to its strong computing capability. Therefore, the cloud-edge communication latency can be derived as

$$T_k^{cloud} = T_{c,k}^u + T_{c,k}^d, \ \forall k \in \mathcal{K}. \tag{16}$$

Towards this end, the one-round latency for edge node $k \in \mathcal{K}$ is given by

$$T_k = T_k^{edge} + T_k^{cloud}, \ \forall k \in \mathcal{K}. \tag{17}$$

C. EDGE UPDATE MODEL

From Eq. (2), the local updated models are determined by their own characteristics. Since the non-iid devices that connected with one edge node have a similar characteristic, the edge aggregation models are heterogeneous. Therefore, if we directly use the cloud model to update the edge models, the personalities among edge models would be eliminated. Meanwhile, the accuracy of the cloud model would be decreased. Hence, we introduce a new edge update model based on [25], which defines a weight distance formula to represent the difference among different weight relatives as

$$dist\left(\boldsymbol{w}_k^t, \boldsymbol{w}_c^t\right) = \frac{\left|\left|\boldsymbol{w}_k^t - \boldsymbol{w}_c^t\right|\right|}{\left|\left|\boldsymbol{w}_c^t\right|\right|}, \forall k \in \mathcal{K}. \tag{18}$$

Intuitively, the larger of $dist\ (\boldsymbol{w}_k, \boldsymbol{w}_c)$, the greater of the model difference.

Typically, deep learning networks that consist of multiple layers and each layer contains various amounts of weights can be adopted here. For simplicity, we use a small dataset to obtain the layer with the most obvious characteristics, which has been denoted as $\mathcal{L} = \{\ell_1, \ell_2, \cdots\}$. Thereafter, we introduce a parameter ε_k to measure the difference between the cloud model and edge model k, which can be formulated as

$$\varepsilon_k = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} dist\left(\boldsymbol{w}_k^{t,\ell}, \boldsymbol{w}_c^{t,\ell}\right), \forall k \in \mathcal{K}, \tag{19}$$

where $\boldsymbol{w}_k^{t,\ell}$ and $\boldsymbol{w}_c^{t,\ell}$ represent the weight of the ℓ-th layer of edge model $\boldsymbol{w}_k^t$ and cloud model $\boldsymbol{w}_c^t$. Meanwhile, $|\mathcal{L}|$ is the cardinality of $\mathcal{L}$.

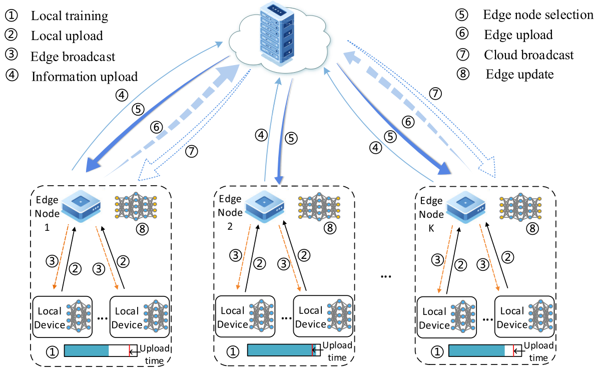


FIGURE 3. Learning procedure of the proposed SAHFL model.

From Eq. (19), ε_k increases with $dist\ (\boldsymbol{w}_k, \boldsymbol{w}_c)$. To keep the personalities, the edge updated model can be derived by

$$\boldsymbol{w}_k^t \leftarrow \varepsilon_k \boldsymbol{w}_c^t + (1 - \varepsilon_k)\boldsymbol{w}_k^t, \forall k \in \mathcal{K}. \tag{20}$$

D. LEARNING PROCEDURE OF THE SAHFL MODEL

Based on the definition of SAHFL model, the training procedure of the SAHFL model at the t-th iteration proceeds as follows, which is also shown in Fig. 3.

1) *Local model training and update*: Devices in mobile edge network train their learning model and calculate their local gradient as $\nabla F_{k,n}(\boldsymbol{w}_{k,n}^t), \forall k \in \mathcal{K}, n \in \mathcal{N}_k$. After receiving $\boldsymbol{w}_k^t, \forall k \in \mathcal{S}^t$, devices in the selected edge nodes update their learning model based on Eq. (2).
2) *Local model upload*: Local devices upload their concrete models to related edge nodes based on the local-edge bandwidth allocation scheme.
3) *Edge model aggregation*: After receiving local models, each edge node computes the average edge model based on Eq. (8). Since device types among edge nodes are heterogeneous, their edge model aggregations are semi-asynchronous.
4) *Edge node selection and resource management*: Based on the reports of edge nodes, the cloud node selects a subset of edge nodes $\mathcal{S}^t$ with the fastest training round and determines the bandwidth allocation.
5) *Selected edge model upload*: The selected edge nodes upload their aggregated models to the cloud node.
6) *Cloud model aggregation and broadcast*: The cloud server aggregates the uploaded edge models, and then broadcasts the current aggregated model $\boldsymbol{w}_c^t$ to the selected edge nodes.
7) *Edge model update and broadcast*: The selected edge servers broadcast the updated model $\boldsymbol{w}_k^t$ to local related devices.

The procedure starts from $t = 1$ and repeats the above steps until convergence.

E. CONVERGENCE ANALYSIS

Before delving into the convergence analysis, we introduce the following assumptions on loss functions and gradient estimates according to [24]. We first assume that $F(\boldsymbol{w}^*)$ is the optimal global FL model obtained by collecting the local models of all selected devices in each iteration by the SAHFL algorithm. Other assumptions are as follows.

Assumption 1: (Smoothness). We assume that F is L-smooth with $L > 0$, namely, $\|\nabla F(\boldsymbol{w}_2) - \nabla F(\boldsymbol{w}_1)\| \leq L\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|$, $\forall \boldsymbol{w}_1, \boldsymbol{w}_2$, where $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ represent the training model in training round 1 and training round 2.

Assumption 2: (Strong Convexity). We also assume that F is strongly convex with $\delta \geq 0$, namely, $F(\boldsymbol{w}_2) - F(\boldsymbol{w}_1) \geq \langle \nabla F(\boldsymbol{w}_1), \boldsymbol{w}_2 - \boldsymbol{w}_1 \rangle + \frac{\delta}{2}\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|^2$, $\forall \boldsymbol{w}_1, \boldsymbol{w}_2$. By minimizing both sides of inequality with respect to $\boldsymbol{w}_2$, we have $F(\boldsymbol{w}^*) \geq F(\boldsymbol{w}) - \frac{1}{2\delta}\|\nabla F(\boldsymbol{w})\|^2$.

Based on Assumptions 1 and 2, we can introduce the following Theorem to present the convergence of the SAHFL algorithm.

Theorem 1: Given the optimal global FL model $F(\boldsymbol{w}^*)$ and the learning rate $0 \leq \eta \leq \frac{1}{L}$, the upper bound of $\mathbb{E}[F(\boldsymbol{w}_c^{t+1}) - F(\boldsymbol{w}^*)]$ can be given by

$$\mathbb{E}[F(\boldsymbol{w}_c^{t+1}) - F(\boldsymbol{w}^*)] \leq (1 - \delta\eta)\mathbb{E}[F(\boldsymbol{w}_c^{t}) - F(\boldsymbol{w}^*)]. \quad (21)$$

III. PROBLEM FORMULATION

As discussed earlier, there exists a tradeoff between the training accuracy and the transmission latency. Therefore, our goal in this work is to find a balance between them to provide safety and communication-efficiency services for the SAHFL based mobile edge network framework.

According to Eq. (2), the local model *Gradient-Norm-Value* (GNV) influences the local model updating, which measures the data importance. The GNV of local device $n \in \mathcal{N}_k$ in edge node $k \in \mathcal{K}$ can be expressed as

$$\begin{aligned} \boldsymbol{g}_{k,n}^{\boldsymbol{w},t} &= \nabla F_{k,n}(\boldsymbol{w}_{k,n}^t) \\ &= \sum_{D_{k,n}} \frac{\partial f_{k,n}(\boldsymbol{x}_{j,k,n}, y_{j,k,n}, \boldsymbol{w}_{k,n}^t)}{\partial \boldsymbol{w}_{k,n}^t}, \forall k \in \mathcal{K}, n \in \mathcal{N}_k. \end{aligned} \quad (22)$$

Without loss of generality, we leverage the norm of GNV to present the importance, which can be written as

$$\sigma_{k,n}^t = \left|\left|\boldsymbol{g}_{k,n}^{\boldsymbol{w},t}\right|\right|^2, \forall k \in \mathcal{K}, n \in \mathcal{N}_k. \quad (23)$$

Since an edge node connects homogeneous local devices, the GNVs among these local devices are approximately equal. Moreover, local devices in one edge node also have similar training duration, hence, all of these training models (GNV) would be uploaded. In this way, the GNV of edge node $k \in \mathcal{K}$ can be defined as

$$\sigma_k^t = \sum_{n=1}^{\mathcal{N}_k} \sigma_{k,n}^t, \forall k \in \mathcal{K}. \quad (24)$$

On the contrary, the cloud node associates with heterogeneous edge nodes, the GNVs among them various. Intuitively, edge nodes with significant gradients have more contributions on model updating and convergence. Therefore, the cloud would preferentially select impactive edge nodes to upload their information for cloud model aggregation. Then, the GNV of the cloud model can be written as

$$\sigma^t = \sum_{k=1}^{K} \alpha_k^t \sigma_k^t. \quad (25)$$

For easy of expression, we remove the iteration t in the following.

Now, we are ready to describe the problem formulation. The goal of this work is to maximize communication-efficient via joint edge node selection and resource allocation scheduling for an SAHFL based mobile edge network. To accelerate the learning process, it is desirable to select more edge nodes with larger data importance. However, to shorten the communication and computation latency, it is better to upload as fewer edge nodes as possible. As a result, the objective function that represents the tradeoff between GNVs and transmission latency can be formulated as

$$\min_{\boldsymbol{\alpha}, \boldsymbol{B_{k,n}}, \boldsymbol{B_{c,k}}} \left(-\rho \sum_{k=1}^{K} \alpha_k \sigma_k + (1-\rho) \max_{k \in K} \alpha_k T_k \right), \quad (26)$$

subject to

$$\sum_{k=1}^{K} \sum_{n=1}^{\mathcal{N}_k} B_{k,n} \leq B_e, \quad (26a)$$

$$\sum_{k=1}^{K} \alpha_k B_{c,k} \leq B_c, \quad (26b)$$

$$\alpha_k \in \{0, 1\}, \forall k \in \mathcal{K}, \quad (26c)$$

where $B_c = B - B_e$, B is the total bandwidth, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_K]^{\mathrm{T}}$, $\boldsymbol{B_{k,n}} = [B_{k,1}, B_{k,2}, \cdots, B_{k,N_k}]^{\mathrm{T}}$, $\boldsymbol{B_{c,k}} = [B_{c,1}, B_{c,2}, \cdots, B_{c,K}]^{\mathrm{T}}$, and $\rho \in [0, 1]$ is the weight factor that controls the tradeoff between data importance and transmission latency.

Obviously, (26) is a MINLP problem, which is NP-hard. In the following, we would introduce an ADMM-BCU method to find the joint edge node selection and resource allocation strategy.

IV. JOINT EDGE NODE SELECTION AND RESOURCE ALLOCATION

As known to us, all of the steps in the learning procedure are independent with the optimal scheduling decision. Denote $T_k' = \max_{n \in \mathcal{N}_k}(T_{k,n}^c + T_{k,n}^u) + T_{c,k}^u$, the original problem (26) can be rewritten as

$$\min_{\boldsymbol{\alpha}, \boldsymbol{B_{k,n}}, \boldsymbol{B_{c,k}}} \max_{k \in K} \left\{ -\rho \sum_{k=1}^{K} \alpha_k \sigma_k + (1-\rho)\alpha_k T_k' \right\}, \quad (27)$$

subject to (26a), (26b), and (26c).

To solve the above min-max problem, we first denote $X = \max_{n \in \mathcal{N}_k} \left(T_{k,n}^c + T_{k,n}^u \right)$ and $Y = \max_{k \in K}(-\rho \sum_{k=1}^{K} \alpha_k \sigma_k + (1-\rho)\alpha_k(X + T_{c,k}^u))$. By applying the parametric method [26],

(27) can be transformed into

$$\min_{\boldsymbol{\alpha},\boldsymbol{B_{k,n}},\boldsymbol{B_{c,k}},X,Y} Y, \tag{28}$$

subject to (26a), (26b), (26c), and

$$T^{c}_{k,n} + T^{u}_{k,n} \leq X, \tag{28a}$$

$$-\rho\sum_{k=1}^{K}\alpha_k\sigma_k + (1-\rho)\alpha_k(X+T^{u}_{c,k}) \leq Y. \tag{28b}$$

Nevertheless, (28) is still a unsolvable MINLP problem. We then introduce an auxiliary variable $\tilde{\boldsymbol{\alpha}}$ to deal with the binary vector $\boldsymbol{\alpha}$ [27]. Thereafter, (28) can be reformulated as

$$\min_{\boldsymbol{\alpha},\boldsymbol{B_{k,n}},\boldsymbol{B_{c,k}},X,Y} Y, \tag{29}$$

subject to (26a), (26b), (26c), (28a), (28b), and

$$\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}} = \mathbf{0}, \tag{29a}$$

$$\alpha_k(1-\tilde{\alpha}_k) = 0, \tag{29b}$$

$$0 \leq \alpha_k \leq 1. \tag{29c}$$

Problem (27) is transferred as a convex problem with equality constraints now. Hereinafter, we introduce the Augmented Lagrangian (AL) method to solve problem (29) through penalizing and dualizing the equality constraints (29a) and (29b) as

$$\min_{\boldsymbol{\alpha},\tilde{\boldsymbol{\alpha}},\boldsymbol{B_{k,n}},\boldsymbol{B_{c,k}},X,Y} Y + \frac{1}{2\nu}\sum_{k=1}^{K}\left[\alpha_k\left(1-\tilde{\alpha}_k\right)+\nu\lambda_k\right]^2 + \frac{1}{2\nu}\sum_{k=1}^{K}\left(\alpha_k-\tilde{\alpha}_k+\nu\tilde{\lambda}_k\right)^2, \tag{30}$$

subject to (26a), (26b), (28a), (28b), and (29c).

Here, ν is the non-negative penalty parameter, and $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_K]$ and $\tilde{\boldsymbol{\lambda}} = [\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_K]$ denote the dual variable vectors correspond to constraints (29a) and (29b), respectively.

Nevertheless, (30) is still a coupled problem due to the multiply variables of Y, $\boldsymbol{B_{k,n}}$, and $\boldsymbol{B_{c,k}}$. Therefore, in this work, we propose a distributed ADMM-BCU algorithm that can iteratively approach a near optimal stable solution with low computational complexity. Specifically, during each iteration, (30) is decomposed into edge node selection and resource allocation subproblems, which aim to solve the blocks of $\{\boldsymbol{\alpha}, X, Y\}$ and $\{\tilde{\alpha}, \boldsymbol{B_{k,n}}, \boldsymbol{B_{c,k}}\}$, respectively.

a: THE OPTIMAL EDGE NODE SELECTION $\{\alpha, X, Y\}$

Under the fixed resource allocations block $\left\{\tilde{\boldsymbol{\alpha}}, \boldsymbol{B_{k,n}}, \boldsymbol{B_{c,k}}\right\}$, the edge node selection optimization subproblem over the variable block $\{\boldsymbol{\alpha}, X, Y\}$ can be rewritten as

$$\min_{\boldsymbol{\alpha},X,Y} Y + \frac{1}{2\nu}\sum_{k=1}^{K}\left[\alpha_k\left(1-\tilde{\alpha}_k\right)+\nu\lambda_k\right]^2 + \frac{1}{2\nu}\sum_{k=1}^{K}\left(\alpha_k-\tilde{\alpha}_k+\nu\tilde{\lambda}_k\right)^2, \tag{31}$$

subject to (26b), (28a), (28b), and (29c).

Obviously, (31) is a convex problem, which can be solved by standard tools, such as CVX.

In what follows, we provide the closed form expression of the optimal edge node selection by introducing Lemma 1.

Lemma 1: The optimum edge node selection $\boldsymbol{\alpha}^*$ can be expressed as

$$\alpha_k^* = \frac{\nu M_k + \nu\kappa_k\left(\rho\sigma_k - (1-\rho)\left(X+T^{u}_{c,k}\right)\right)}{(1-\tilde{\alpha}_k)^2+1}, \forall k \in \mathcal{K}, \tag{32}$$

where $M_k = \psi_k - \xi_k - \lambda_k\left(1-\tilde{\alpha}_k\right) - \tilde{\lambda}_k - \phi_k B_{c,k} + \frac{1}{\nu}\tilde{\alpha}_k$. $\boldsymbol{\phi}$, $\boldsymbol{\zeta}$, $\boldsymbol{\kappa}$, $\boldsymbol{\psi}$, and $\boldsymbol{\xi}$ are Lagrangian multipliers correspond to constraints (26b), (28a), (28b), and (29c), which can be found by one-dimensional search methods based on the complementary slackness conditions.

Proof: To find the optimal $\alpha_k, \forall k \in \mathcal{K}$, we apply for the Lagrangian dual method, which can rearrange Eq. (31) with respect to $\alpha_k, \forall k \in \mathcal{K}$ as

$$\begin{aligned}
&L(\boldsymbol{\alpha_k}, \boldsymbol{\phi_k}, \boldsymbol{\zeta_k}, \boldsymbol{\kappa_k}, \boldsymbol{\psi_k})\\
&\quad= \frac{Y}{\sum_{k=1}^{K}\sigma_k\alpha_k} + \frac{1}{2\nu}\sum_{k=1}^{K}\left[\alpha_k\left(1-\tilde{\alpha}_k\right)+\nu\lambda_k\right]^2\\
&\quad+\frac{1}{2\nu}\sum_{k=1}^{K}\left(\alpha_k-\tilde{\alpha}_k+\nu\tilde{\lambda}_k\right)^2\\
&\quad+\phi_k\left(\sum_{k=1}^{K}\alpha_k B_{c,k} - B_c\right) - \psi_k\alpha_k - \xi_k\left(1-\alpha_k\right)\\
&\quad+\zeta_k\left(T^{c}_{k,n}+T^{u}_{k,n}-X\right)\\
&\quad+\kappa_k\left(-\rho\sum_{k=1}^{K}\alpha_k\sigma_k\left(1-\rho\right)\alpha_k\left(X+T^{u}_{c,k}\right)-Y\right).
\end{aligned} \tag{33}$$

Calculate the first-order partial derivatives with respect to $\alpha_k, \forall k \in \mathcal{K}$, we derive that

$$\begin{aligned}
&\frac{\partial L(\alpha_k, \phi_k, \zeta_k, \kappa_k, \psi_k)}{\partial \alpha_k}\\
&\quad= \frac{1}{\nu}\left[\alpha_k\left(1-\tilde{\alpha}_k\right)^2+\alpha_k-\tilde{\alpha}_k\right]\\
&\quad+\lambda_k\left(1-\tilde{\alpha}_k\right)\\
&\quad+\tilde{\lambda}_k+\phi_k B_{c,k}-\psi_k+\xi_k\\
&\quad+\kappa_k\left(-\rho\sigma_k+(1-\rho)\left[X+T^{u}_{c,k}\right]\right), \forall k \in \mathcal{K}.
\end{aligned} \tag{34}$$

Setting $\frac{\partial L(\alpha_k,\mu_k,\phi_k,\zeta_k)}{\partial \alpha_k} = 0$, we have

$$\alpha_k^* = \frac{\nu M_k + \nu\kappa_k\left(\rho\sigma_k - (1-\rho)(X+T^{u}_{c,k})\right)}{(1-\tilde{\alpha}_k)^2+1}, \tag{35}$$

where $M_k = \psi_k - \xi_k - \lambda_k\left(1-\tilde{\alpha}_k\right) - \tilde{\lambda}_k - \phi_k B_{c,k} + \frac{1}{\nu}\tilde{\alpha}_k$. This ends the proof. □

From Lemma 1, we find that the edge node selection is mainly determined by the edge node importance σ_k and the uplink transmission latency from edge k to the cloud $T^{u}_{c,k}$.

Intuitively, the cloud preferentially selects the edge node with either a larger edge node importance or a smaller uplink transmission latency that can improve the communication-efficiency.

b: THE OPTIMAL BANDWIDTH ALLOCATION $\left\{\tilde{\alpha}, \boldsymbol{B_{k,n}}, \boldsymbol{B_{c,k}}\right\}$

Similarly, under the fixed edge node selection block $\{\boldsymbol{\alpha}, X, Y\}$, the resource allocation optimization subproblem over the block $\left\{\tilde{\boldsymbol{\alpha}}, \boldsymbol{B_{k,n}}, \boldsymbol{B_{c,k}}\right\}$ can be rearranged as

$$\min_{\tilde{\boldsymbol{\alpha}}, \boldsymbol{B_{k,n}}, \boldsymbol{B_{c,k}}} \frac{1}{2\nu} \sum_{k=1}^{K} \left[\alpha_k \left(1-\tilde{\alpha}_k\right)+\nu\lambda_k\right]^2 + \frac{1}{2\nu} \sum_{k=1}^{K} \left(\alpha_k - \tilde{\alpha}_k + \nu\tilde{\lambda}_k\right)^2, \tag{36}$$

subject to (26a), (26b), and

$$\frac{Z}{B_{k,n} \log_2\left(1+\frac{P_{k,n} g_{k,n}}{B_{k,n} N_0}\right)} + \frac{C_{k,n} N_{k,n}}{f_{k,n}} \leq X, \tag{36a}$$

$$-\rho \sum_{k=1}^{K} \alpha_k \sigma_k + (1-\rho)\alpha_k \cdot \left(X + \frac{Z}{B_{c,k} \log_2\left(1+\frac{P_{c,k} g_{c,k}}{B_{c,k} N_0}\right)}\right) \leq Y. \tag{36b}$$

Also, it is easy to observe that (36) is a convex problem. For ease of analyses, we write this problem under the Lagrangian dual formulation, where (36) can be rearranged as (37), shown at the bottom of the next page, where $\boldsymbol{\beta}$, $\boldsymbol{\varkappa}$, $\boldsymbol{\varphi}$, and $\boldsymbol{\tau}$ are the Lagrangian multipliers corresponding to constraints (26a), (26b), (36a), and (36b), respectively.

By taking $\frac{\partial L(\tilde{\alpha}_k, B_{k,n}, B_{c,k}, \beta_k, \varkappa_k, \varphi_k, \tau_k)}{\partial B_{k,n}} = 0$, the optimal local-edge uplink bandwidth allocation $B^*_{k,n}$ can be derived by (38), as shown at the bottom of the next page. From Eq. (38), the optimal local-edge bandwidth allocation $B^*_{k,n}$ is mainly influenced by the related channel conditions $\frac{P_{k,n} g_{k,n}}{N_0}$.

Alternatively, by taking $\frac{\partial L(\tilde{\alpha}_k, B_{k,n}, B_{c,k}, \beta_k, \varkappa_k, \varphi_k, \tau_k)}{\partial B_{c,k}} = 0$, the optimal edge-cloud uplink bandwidth allocation $B^*_{c,k}$ can be obtained by (39), as shown at the bottom of the next page. Obviously, the optimal edge-cloud uplink bandwidth allocation $B^*_{c,k}$ has a similar rule with $B^*_{k,n}$.

Thereafter, by setting $\frac{\partial L(\tilde{\alpha}_k, B_{k,n}, B_{c,k}, \beta_k, \varkappa_k, \varphi_k, \tau_k)}{\tilde{\alpha}_k} = 0$, we can obtain the optimal auxiliary variable $\tilde{\alpha}^*_k$ as

$$\tilde{\alpha}^*_k = \frac{\alpha_k \left(1+\alpha_k+\nu\lambda_k\right)+\nu\tilde{\lambda}_k}{1+\alpha_k^2}, \ \forall k \in \mathcal{K}. \tag{40}$$

The detailed procedure for the joint edge node selection and resource allocation scheduling is presented in Algorithm 1.

In Algorithm 1, $\epsilon^{(\jmath)}$ means the successive divergence of the objective function at the $\jmath$-th iteration [27], which can be defined as

$$\epsilon^{(\jmath)} = F^{(\jmath)} - F^{(\jmath-1)}, \tag{41}$$

where $F = Y + \frac{1}{2\nu} \sum_{k=1}^{K} \left[\alpha_k \left(1-\tilde{\alpha}_k\right)+\nu\lambda_k\right]^2 + \frac{1}{2\nu} \sum_{k=1}^{K} \left(\alpha_k - \tilde{\alpha}_k + \nu\tilde{\lambda}_k\right)^2$.

Algorithm 1 Joint Edge Node Selection and Resource Allocation Strategy

```
Initialize the gradient norm value $\sigma_k, \forall k \in \mathcal{K}$.
Set the minimum successive divergence threshold of the objective function $\epsilon^{\min}$ and the maximum iteration number $R^{\max}$.
Set the iteration number $\jmath = 0$.
Initialize the auxiliary variables $\boldsymbol{\alpha}^{\tilde{(\jmath)}}$, $\boldsymbol{\lambda}^{(\jmath)}$, $\boldsymbol{\lambda}^{\tilde{(\jmath)}}$.
While $\epsilon^{(\jmath)} \geq \epsilon^{\min}$ and $\jmath \leq R^{\max}$ do
    Calculate the optimal device selection decision $\alpha_k^{*(\jmath)}$according to (35).
    Calculate the optimal bandwidth $B^{*(\jmath)}_{k,n}$, $B^{*(\jmath)}_{c,k}$ according to (38) and (39).
    Obtain $\tilde{\alpha}_k^{*(\jmath)}$ according to (40).
    Update $\boldsymbol{\lambda}^{(\jmath)}$ and $\boldsymbol{\lambda}^{\tilde{(\jmath)}}$ according to
        $\boldsymbol{\lambda}^{(\jmath)} = \boldsymbol{\lambda}^{(\jmath-1)} + \frac{1}{\nu}\boldsymbol{\alpha}^{*(\jmath-1)}\left(1-\tilde{\boldsymbol{\alpha}}^{*(\jmath-1)}\right),$   (42)
        $\tilde{\boldsymbol{\lambda}}^{(\jmath)} \quad \tilde{\boldsymbol{\lambda}}^{(\jmath-1)} + \frac{1}{\nu}\left(\boldsymbol{\alpha}^{*(\jmath-1)} - \tilde{\boldsymbol{\alpha}}^{*(\jmath-1)}\right).$   (43)
    Update $\epsilon^{(\jmath)}$ according to (41).
    Set $\jmath = \jmath + 1$.
End while
```

V. NUMERICAL RESULTS

In this section, we conduct experiments to evaluate the theoretical analyses and test the performance of the proposed algorithm.

A. EXPERIMENT SETTINGS

CNN model settings: For exposition, we consider the learning task of training image classifiers, which are implemented on a *Convolutional Neural Network* (CNN) model, namely VGGNet 16 [28]. The corresponding training dataset is CIFAR-10, which contains 50000 training images and 10000 testing images with 10 categories. To simulate the distributions of heterogeneous data based mobile devices, all data samples are first sorted by digital labels, and then divided into 100 shards of size 500 and each local device is assigned with 5 shards. The batch size of each local device is set as 50 and the average quantitative bit number of each parameter is set as 16 bits. In addition, we adopt the *Stochastic Gradient Descent* (SGD) optimizer, and the learning rate for the CNN model is set as 0.1. The computation frequency of each local device is randomly set between 2 GHz to 4 GHz.

Wireless communication settings: We consider a hierarchical SAHFL communication network consists of one cloud node and 10 edge nodes. Each edge node connects with two local devices. Both edge nodes and local devices are uniformly distributed under the coverage of the cloud node. The total bandwidth is set as 20 MHz. Moreover, the uplink transmission powers of each local device and edge node are set as 10 dBm and 24 dBm, respectively. Also, the downlink transmission powers of each edge node and the cloud node are set as 10 dBm and 24 dBm, respectively. Furthermore, we utilize the transmission pass loss model of $128.1 + 37.6 \log(d[\text{km}])$. Meanwhile, the noise power spectral density is set as $N_0 = -174$ dBm/Hz.

In the ADMM-BCU algorithm, we set the non-negative penalty parameter ν as 1. The minimum successive divergence threshold ϵ^{min} is set as 10^{-4}. In addition, the maximum iteration number of ADMM-BCU algorithm is set as 200.

B. SAHFL PERFORMANCE

In this subsection, we present the convergence performance of the proposed SAHFL model. We first introduce the following baselines.

- *Random selection*: Under this circumstance, CNN is implemented with random data selection, where both 5 and 8 edge nodes randomly selective conditions are respectively considered.
- *Full selection*: Under this circumstance, CNN is implemented by selecting all of the edge nodes.
- *Normal edge update*: Edge nodes directly use the broadcast cloud model as their updated model.

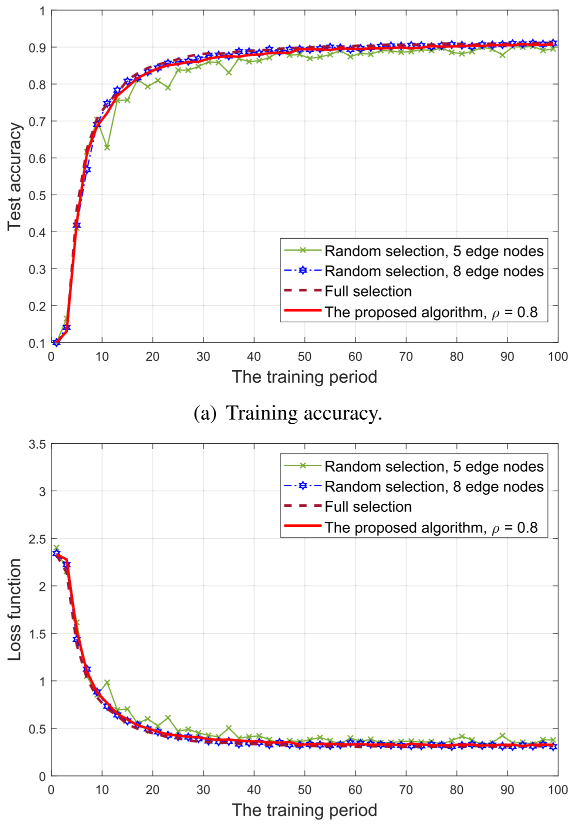
For simplicity, we assume the transmissions from the selected edge nodes to the cloud node are uniformly allocated, totally 5 MHz. Meanwhile, the transmission bandwidths from local devices to edge nodes are also set as the same, totally 15 MHz. Moreover, we set the weighted factor ρ under the proposed algorithm as 0.8. Fig. 4 shows the convergence performance of the proposed CNN based SAHFL model. From this figure, we can find that the VGG-16 network starts to converge at about 70 communication rounds for both the random selection scheme with 8 edge nodes, the Full selection scheme, and the proposed scheme. However, the random selection scheme with 5 edge nodes presents the worst convergence performance. Intuitively, it is because the more devices to be selected, the larger data information can be provided to the neural network, and thus faster convergence. Moreover, due to the non-iid datasets, each node has different contributions. Therefore, the random selection scheme may play a side effect on the whole model, leading to a decreasing model accuracy. Overall, the proposed algorithm shows a near to the full selection scheme convergence and accuracy, which can achieve better performance than the baselines that would be discussed later.

Fig. 5 presents the performance influence from the edge update model. From this figure, we find that either the training accuracy or the training loss under the elastic edge update model is better than that of the normal edge update

$$\begin{aligned}
&L\left(\tilde{\boldsymbol{\alpha}}_k, \boldsymbol{B_{k,n}}, \boldsymbol{B_{c,k}}, \boldsymbol{\beta_k}, \boldsymbol{\varkappa_k}, \boldsymbol{\varphi_k}, \boldsymbol{\tau_k}\right) \\
&\quad = Y + \frac{1}{2\nu}\sum_{k=1}^{K}\left[\alpha_k\left(1-\tilde{\alpha}_k\right)+\nu\lambda_k\right]^2 + \frac{1}{2\nu}\sum_{k=1}^{K}\left(\alpha_k-\tilde{\alpha}_k+\nu\tilde{\lambda}_k\right)^2 \\
&\qquad +\beta_k\left(\sum_{k=1}^{K}\sum_{n=1}^{\mathcal{N}_k} B_{k,n}-B_e\right)+\varkappa_k\left(\sum_{k=1}^{K}\alpha_k B_{c,k}-B_c\right) \\
&\qquad +\varphi_k\left[\frac{Z}{B_{k,n}\log_2\left(1+\frac{P_{k,n}g_{k,n}}{B_{k,n}N_0}\right)}+\frac{C_{k,n}N_{k,n}}{f_{k,n}}-X\right] \\
&\qquad +\tau_k\left(-\rho\sum_{k=1}^{K}\alpha_k\sigma_k+(1-\rho)\alpha_k\left[X+\frac{Z}{B_{c,k}\log_2\left(1+\frac{P_{c,k}g_{c,k}}{B_{c,k}N_0}\right)}\right]-Y\right),
\end{aligned} \tag{37}$$

$$\varphi_k\frac{Z}{\left(B^*_{k,n}\log_2\left(1+\frac{P_{k,n}g_{k,n}}{B^*_{k,n}N_0}\right)\right)^2}\left[\log_2\left(1+\frac{P_{k,n}g_{k,n}}{B^*_{k,n}N_0}\right)-\frac{P_{k,n}g_{k,n}}{\left(P_{k,n}g_{k,n}+B^*_{k,n}N_0\right)\ln 2}\right]=\beta_k,\ \forall k\in\mathcal{K}, n\in\mathcal{N}_k \tag{38}$$

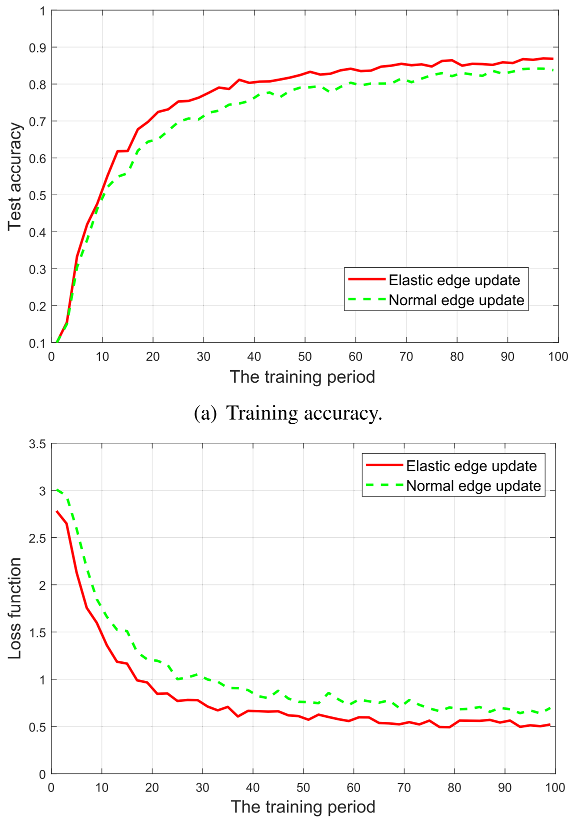
$$\tau_k\frac{(1-\rho)Z}{\left(B^*_{c,k}\log_2\left(1+\frac{P_{c,k}g_{c,k}}{B^*_{c,k}N_0}\right)\right)^2}\left[\log_2\left(1+\frac{P_{c,k}g_{c,k}}{B^*_{c,k}N_0}\right)-\frac{P_{c,k}g_{c,k}}{\left(P_{c,k}g_{c,k}+B^*_{c,k}N_0\right)\ln 2}\right]=\varkappa_k,\ \forall k\in\mathcal{K}, n\in\mathcal{N}_k \tag{39}$$



(a) Training accuracy.

(b) Training loss.

FIGURE 4. **Convergence performance of the proposed CNN model under different algorithms.**



(a) Training accuracy.

(b) Training loss.

FIGURE 5. **Performance influence from the edge update model.**

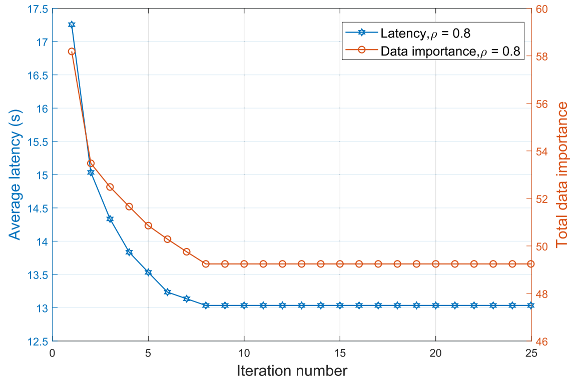


FIGURE 6. **Convergence performance of the proposed ADMM-BCU algorithm.**

model. The fluctuation of these curves are mainly due to the non-iid data form. Therefore, we can conclude that the elastic edge update model is significant to keep the personalities of the edge nodes.

C. THE SCHEDULING PERFORMANCE

In this subsection, we mainly verify the scheduling performance of the proposed algorithm. In Fig. 6, we shows that the proposed ADMM-BCU algorithm has a fast convergence and a low computational complexity.

Fig. 7 illustrates that a tradeoff exists between data importance and the transmission latency. The value of ρ starts from 0.4 to 0.8 under the step of 0.05. This figure shows that a large value of ρ leads to higher data importance and longer transmission latency, and vise versa. Thus, the operators can select a suitable value of ρ according to their specific requirements.

In Fig. 8, we present the performance among the number of selected edge nodes, data importance, and latency under various weight factors. Fig. 8(a) shows the number of selected edge nodes and total data importance in different weight factors under various algorithms. From this subfigure, the number of selected edge nodes increases with the weight factor ρ. When the value of weight factor ρ is small, i.e., the associated edge nodes are small, the proposed algorithm has a lower data importance than the random selection scheme. With the increment of associated edge nodes, the circumstance changes, which has been explained in Fig. 4. However, the full selection scheme always has the highest value of data importance at the cost of higher latency, which is shown in Fig. 8(b). Fig. 8(b) shows the full selection scheme suffers the highest latency, and the proposed algorithm has the lowest latency after scheduling. Intuitively, the transmission latency is much lower than the total latency, which means the data training time is huge. Moreover, the transmission

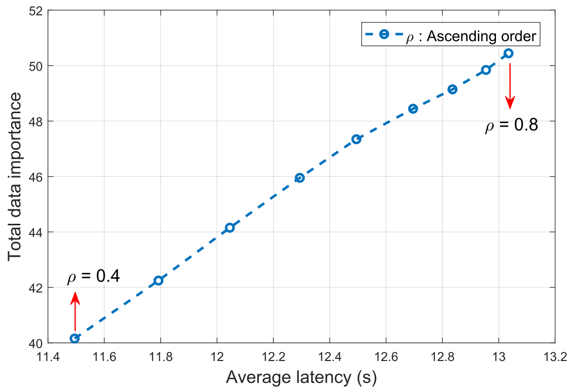
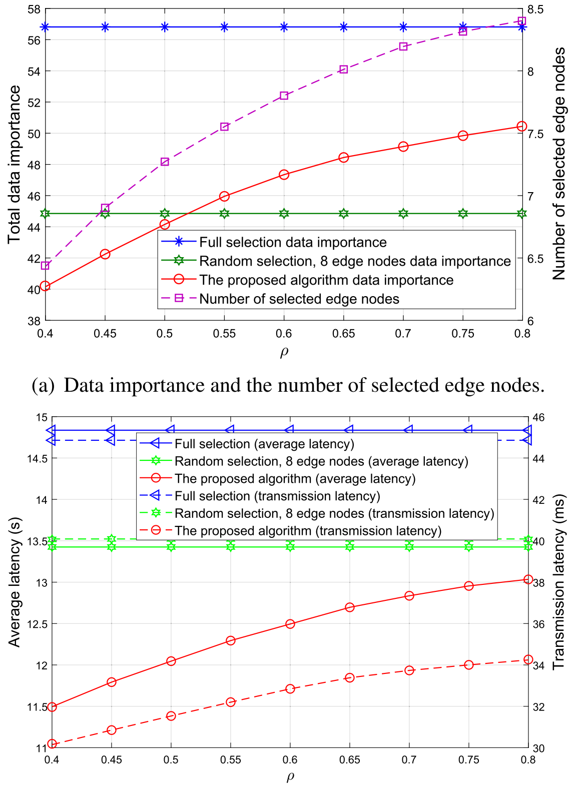


FIGURE 7. Tradeoff between data importance and delay.



(a) Data importance and the number of selected edge nodes.

(b) Latency.

FIGURE 8. Performance among the number of selected edge nodes, data importance, and latency under different mechanisms.

latency may not meet the requirements of ultra low latency mobile edge network devices. Under this circumstance, we can enlarge the wireless bandwidth by some resource management technologies.

VI. CONCLUSION

This work proposes a novel SAHFL framework that consists of local, edge, and cloud nodes to provide communication-efficient services for mobile edge networks. Specifically, homogeneous devices are allowed to associate with one edge node. Therefore, we adopt the synchronous aggregation model for edge nodes. On the contrary, for the heterogeneous edge aggregation models, we introduce a semi-asynchronous aggregation model for the cloud node, where parts of the fastest training edge models can be uploaded at each iteration. Moreover, we investigate an edge-cloud update method to keep the personalities of the edge nodes. We propose a joint edge node association and resource allocation strategy, which illustrates a tradeoff between training accuracy and transmission latency. A distributed ADMM-BCU algorithm has been adopted to solve the proposed optimal MINLP problem. Numerical results show that our proposed scheme can accelerate the training process and improve the performance for mobile edge networks.

APPENDIX A PROOF OF THEOREM 1

According to (2), (8), and (15), the global aggregation model of the cloud server can be rearranged as

$$\begin{aligned}\boldsymbol{w}_c^{t+1} &= \boldsymbol{w}_c^t + \sum_{k\in\mathcal{S}^{t+1}} \frac{|D_k|}{\sum_{k=1}^{K}|D_k|}(\boldsymbol{w}_k^{t+1} - \boldsymbol{w}_c^t)\\ &= \boldsymbol{w}_c^t + \sum_{k\in\mathcal{S}^{t+1}} \frac{|D_k|}{\sum_{k=1}^{K}|D_k|}\Big(\sum_{n=1}^{\mathcal{N}_k}\frac{|D_{k,n}|\boldsymbol{w}_{k,n}^{t+1}}{\sum_{n=1}^{\mathcal{N}_k}|D_{k,n}|} - \boldsymbol{w}_c^t\Big)\\ &= \boldsymbol{w}_c^t + \sum_{k\in\mathcal{S}^{t+1}} \frac{|D_k|}{\sum_{k=1}^{K}|D_k|}\\ &\quad\Big(\sum_{n=1}^{\mathcal{N}_k}\frac{|D_{k,n}|(\boldsymbol{w}_k^t - \eta\nabla F_{k,n}(\boldsymbol{w}_{k,n}^t))}{\sum_{n=1}^{\mathcal{N}_k}|D_{k,n}|} - \boldsymbol{w}_c^t\Big). \qquad (44)\end{aligned}$$

Since the edge model undergoes an edge elastic update process when it broadcasts to devices, we have the following edge model update according to (18), (19), (20), as

$$\boldsymbol{w}_k^t \leftarrow \frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}}\frac{\left\|\boldsymbol{w}_k^{t,l} - \boldsymbol{w}_c^{t,l}\right\|}{\left\|\boldsymbol{w}_c^{t,l}\right\|}(\boldsymbol{w}_c^t - \boldsymbol{w}_k^t) + \boldsymbol{w}_k^t. \qquad (45)$$

According to Assumption 1 and Assumption 2, the twice-continuously differentiable $F(\boldsymbol{w})$ has the inequality of

$$\delta\boldsymbol{I} \preceq \nabla^2 F(\boldsymbol{w}) \preceq L\boldsymbol{I}. \qquad (46)$$

Considering the second-order Taylor expansion, $F(\boldsymbol{w}_c^{t+1})$ can be written as

$$\begin{aligned}F(\boldsymbol{w}_c^{t+1}) &= F(\boldsymbol{w}_c^t) + \nabla F(\boldsymbol{w}_c^t)^T(\boldsymbol{w}_c^{t+1} - \boldsymbol{w}_c^t)\\ &\quad+\frac{1}{2}(\boldsymbol{w}_c^{t+1} - \boldsymbol{w}_c^t)^T\nabla^2 F(\boldsymbol{w}_c^t)(\boldsymbol{w}_c^{t+1} - \boldsymbol{w}_c^t)\\ &\overset{(a)}{\leq} F(\boldsymbol{w}_c^t) + \nabla F(\boldsymbol{w}_c^t)^T(\boldsymbol{w}_c^{t+1} - \boldsymbol{w}_c^t)\\ &\quad+\frac{L}{2}\|\boldsymbol{w}_c^{t+1} - \boldsymbol{w}_c^t\|^2, \qquad (47)\end{aligned}$$

where (a) stems from the fact that $\nabla^2 F(\boldsymbol{w}) \preceq L\boldsymbol{I}$.

By setting $0 \leq \eta \leq \frac{1}{L}$, we have (48), as shown at the top of the next page. Here, step (b) stems from the equation (45). Step (c) obtains from the fact that $0 \leq \frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}}\frac{\left\|\boldsymbol{w}_k^{t,l} - \boldsymbol{w}_c^{t,l}\right\|}{\left\|\boldsymbol{w}_c^{t,l}\right\|} \leq 1$ [23].

$$
\begin{aligned}
F(\boldsymbol{w}_c^{t+1}) - F(\boldsymbol{w}_c^{t}) &\leq (\nabla F(\boldsymbol{w}_c^t))^T(\boldsymbol{w}_c^{t+1} - \boldsymbol{w}_c^t) + \frac{L}{2}\|\boldsymbol{w}_c^{t+1} - \boldsymbol{w}_c^t\|^2 \\
&= (\nabla F(\boldsymbol{w}_c^t))^T\left(\boldsymbol{w}_c^t + \sum_{k\in\mathcal{S}^{t+1}} \frac{|D_k|}{\sum_{k=1}^K |D_k|}(\sum_{n=1}^{\mathcal{N}_k} \frac{|D_{k,n}|(\boldsymbol{w}_k^t - \eta\nabla F_{k,n}(\boldsymbol{w}_{k,n}^t))}{\sum_{n=1}^{\mathcal{N}_k}|D_{k,n}|} - \boldsymbol{w}_c^t) - \boldsymbol{w}_c^t\right) \\
&\quad + \frac{L}{2}\|\boldsymbol{w}_c^t + \sum_{k\in\mathcal{S}^{t+1}} \frac{|D_k|}{\sum_{k=1}^K |D_k|}\sum_{n=1}^{\mathcal{N}_k} \frac{|D_{k,n}|(\boldsymbol{w}_k^t - \eta\nabla F_{k,n}(\boldsymbol{w}_{k,n}^t))}{\sum_{n=1}^{\mathcal{N}_k}|D_{k,n}|} - \boldsymbol{w}_c^t) - \boldsymbol{w}_c^t\|^2 \\
&\overset{(b)}{=} (\nabla F(\boldsymbol{w}_c^t))^T\Bigg(\sum_{k\in\mathcal{S}^{t+1}} \frac{|D_k|}{\sum_{k=1}^K |D_k|}\Bigg(\sum_{n=1}^{\mathcal{N}_k} \frac{|D_{k,n}|}{\sum_{n=1}^{\mathcal{N}_k}|D_{k,n}|} \\
&\quad \Bigg(\frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}} \frac{\left\|\boldsymbol{w}_k^{t,l} - \boldsymbol{w}_c^{t,l}\right\|}{\left\|\boldsymbol{w}_c^{t,l}\right\|}(\boldsymbol{w}_c^t - \boldsymbol{w}_k^t) + \boldsymbol{w}_k^t - \eta\nabla F_{k,n}(\boldsymbol{w}_{k,n}^t)\Bigg) - \boldsymbol{w}_c^t\Bigg)\Bigg) \\
&\quad + \frac{L}{2}\|\sum_{k\in\mathcal{S}^{t+1}} \frac{|D_k|}{\sum_{k=1}^K |D_k|}\Bigg(\sum_{n=1}^{\mathcal{N}_k} \frac{|D_{k,n}|}{\sum_{n=1}^{\mathcal{N}_k}|D_{k,n}|}\Bigg(\frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}} \frac{\left\|\boldsymbol{w}_k^{t,l} - \boldsymbol{w}_c^{t,l}\right\|}{\left\|\boldsymbol{w}_c^{t,l}\right\|}(\boldsymbol{w}_c^t - \boldsymbol{w}_k^t) \\
&\quad + \boldsymbol{w}_k^t - \eta\nabla F_{k,n}(\boldsymbol{w}_{k,n}^t)\Bigg) - \boldsymbol{w}_c^t\Bigg)\|^2 \\
&\overset{(c)}{\leq} (\nabla F(\boldsymbol{w}^t))^T\left(-\sum_{k\in\mathcal{S}^{t+1}} \frac{|D_k|}{\sum_{k=1}^K |D_k|}\sum_{n=1}^{\mathcal{N}_k} \frac{|D_{k,n}|}{\sum_{n=1}^{\mathcal{N}_k}|D_{k,n}|}\nabla F_{k,n}(\boldsymbol{w}_{k,n}^t)\right) \\
&\quad + \frac{L}{2}\| - \sum_{k\in\mathcal{S}^{t+1}} \frac{|D_k|}{\sum_{k=1}^K |D_k|}\sum_{n=1}^{\mathcal{N}_k} \frac{|D_{k,n}|}{\sum_{n=1}^{\mathcal{N}_k}|D_{k,n}|}\nabla F_{k,n}(\boldsymbol{w}_{k,n}^t)\|^2 \\
&\quad \leq -\eta\|\nabla F(\boldsymbol{w}_c^t)\|^2 + \frac{L\eta^2}{2}\|\nabla F(\boldsymbol{w}_c^t)\|^2 \leq -\frac{\eta}{2}\|\nabla F(\boldsymbol{w}_c^t)\|^2
\end{aligned}
\tag{48}
$$

By applying Assumption 2, it follows that

$$
\begin{aligned}
&\mathbb{E}[F(\boldsymbol{w}_c^{t+1}) - F(\boldsymbol{w}^*)] \\
&\leq \mathbb{E}[F(\boldsymbol{w}_c^t) - F(\boldsymbol{w}^*)] - \frac{\eta}{2}\|\nabla F(\boldsymbol{w}_c^t)\|^2 \\
&\leq \mathbb{E}[F(\boldsymbol{w}_c^t) - F(\boldsymbol{w}^*)] - \delta\eta\mathbb{E}[F(\boldsymbol{w}_c^t) - F(\boldsymbol{w}^*)] \\
&= (1 - \delta\eta)\mathbb{E}[F(\boldsymbol{w}_c^t) - F(\boldsymbol{w}^*)].
\end{aligned}
\tag{49}
$$

This ends the proof.

REFERENCES

[1] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the Internet of Things (IoT) forensics: Challenges, approaches, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1191–1221, 2nd Quart., 2020.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, vol. 54, Apr. 2017, pp. 1273–1282.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, 3rd Quart., 2021.

[4] Z. Yang, M. Chen, K.-K. Wong, H. Vincent Poor, and S. Cui, "Federated learning for 6G: Applications, challenges, and opportunities," 2021, *arXiv:2101.01338*.

[5] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "," FedHome: Cloud-edge based personalized federated learning for in-home health monitoring," *Trans. Mobile Comput.*, vol. 21, no. 8, pp. 2818–2832, Aug. 2022.

[6] W. Y. B. Lim, J. Huang, Z. Xiong, J. Kang, D. Niyato, X.-S. Hua, C. Leung, and C. Miao, "Towards federated learning in UAV-enabled Internet of Vehicles: A multi-dimensional contract-matching approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5140–5154, Aug. 2021.

[7] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[8] S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto, "Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-IID private data," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 191–205, Jan. 2023.

[9] T.-C. Chiu, Y.-Y. Shih, A.-C. Pang, C.-S. Wang, W. Weng, and C.-T. Chou, "Semisupervised distributed learning with non-IID data for AIoT service platform," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9266–9277, Oct. 2020.

[10] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[11] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Jan. 2016.

[12] Y. He, J. Ren, G. Yu, and J. Yuan, "Importance-aware data selection and resource allocation in federated edge learning system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13593–13605, Nov. 2020.

[13] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[14] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, ''HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning,'' *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, Oct. 2020.

[15] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, ''Federated learning for mobile keyboard prediction,'' 2018, *arXiv:1811.03604*.

[16] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, ''Energy efficient federated learning over wireless communication networks,'' *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1938–1949, Mar. 2021.

[17] S. Liu, G. Yu, X. Chen, and M. Bennis, ''Joint user association and resource allocation for wireless hierarchical federated learning with IID and non-IID data,'' *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7852–7866, Oct. 2022.

[18] Q. Chen, X. Xu, Z. You, H. Jiang, J. Zhang, and F.-Y. Wang, ''Communication-efficient federated edge learning for NR-U-based IIoT networks,'' *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12450–12459, Jul. 2022.

[19] C. Pan, Z. Wang, H. Liao, Z. Zhou, X. Wang, M. Tariq, and S. Al-Otaibi, ''Asynchronous federated deep reinforcement learning-based URLLC-aware computation offloading in space-assisted vehicular networks,'' *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 25, 2022, doi: 10.1109/TITS.2022.3150756.

[20] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, ''Asynchronous online federated learning for edge devices with non-IID data,'' in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 15–24.

[21] Y. Chen, X. Sun, and Y. Jin, ''Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020.

[22] Z. Wang, Z. Zhang, Y. Tian, Q. Yang, H. Shan, W. Wang, and T. Q. S. Quek, ''Asynchronous federated learning over wireless communication networks,'' *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6961–6978, Sep. 2022.

[23] Y. Gao, L. Liu, X. Zheng, C. Zhang, and H. Ma, ''Federated sensing: Edge-cloud elastic collaborative learning for intelligent sensing,'' *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11100–11111, Jul. 2021.

[24] M. Chen, H. V. Poor, W. Saad, and S. Cui, ''Convergence time optimization for federated learning over wireless networks,'' *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2459–2471, Apr. 2021.

[25] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, ''Federated learning with non-IID data,'' 2018, *arXiv:1806.00582*.

[26] S. Geisser and W. M. Johnson, *Modes of Parametric Statistical Inference*. Hoboken, NJ, USA: Wiley, 2006.

[27] R. Guo, Y. Cai, M. Zhao, Q. Shi, B. Champagne, and L. Hanzo, ''Joint design of beam selection and precoding matrices for mmWave MU-MIMO systems relying on lens antenna arrays,'' *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 2, pp. 313–325, May 2018.

[28] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*.

QIMEI CHEN (Member, IEEE) received the Ph.D. degree from the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. She was a Visiting Student at the University of California, Davis, CA, USA, from 2015 to 2016. She is currently an Associate Researcher with the School of Electronic Information, Wuhan University, Wuhan, China. Her general research interests include the advantage technologies in 5G wireless communication, unlicensed spectrum, green communication, and intelligent edge.

ZEHUA YOU received the B.Eng. degree in communication engineering from the University of Electronic Science and Technology of China, in 2020. He is currently pursuing the M.Eng. degree with the School of Electronic Information, Wuhan University. His current research interests include edge intelligence, federated learning, and wireless communication.

JING WU received the B.Eng. degree in communication engineering and the Ph.D. degree in communication and information systems from Wuhan University, Wuhan, China, in 2002 and 2007, respectively. From 2004 to 2005, she was a Postdoctoral Researcher at LIMOS, Clermont-Ferrand, France. She is currently an Associate Professor with Wuhan University. Her research interests include wireless communication networks, network simulation, and intelligence data processing.

YUNPENG LIU is currently pursuing the B.S. degree in electronic and information engineering with Wuhan University. His current research interest includes federated learning.

HAO JIANG received the B.Eng. degree in communication engineering and the M.Eng. and Ph.D. degrees in communication and information systems from Wuhan University, China, in 1999, 2001, and 2004, respectively. He was a Postdoctoral Researcher at LIMOS, Clermont-Ferrand, France, from 2004 to 2005. He was a Visiting Professor at the University of Calgary, Canada, and ISIMA, B. Pascal University, France. He is currently a Professor with Wuhan University. He has authored over 60 papers in different journals and conferences. His research interests include mobile ad hoc networks and mobile big data.