Semi-Asynchronous Model Design for Federated Learning in Mobile Edge Networks

Jinfeng Zhang, Wei Liu, Yejun He, *Senior Member, IEEE*, Zhou He, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—Federated learning (FL) is a distributed machine learning (ML). Distributed clients train locally and exclusively need to upload the model parameters to learn the global model collaboratively under the coordination of the aggregation server. Although the privacy of the clients is protected, which requires multiple rounds of data upload between the clients and the server to ensure the accuracy of the global model. Inevitably, this results in latency and energy consumption issues due to limited communication resources. Therefore, mobile edge computing (MEC) has been proposed to solve communication delays and energy consumption in federated learning. In this paper, we first analyze how to select the gradient values that help the global model converge quickly and establish theoretical analysis about the relationship between the convergence rate and the gradient direction. To efficiently reduce the energy consumption of clients during training, on the premise of ensuring the local training accuracy and the convergence rate of the global model, we adopt the deep deterministic policy gradient (DDPG) algorithm, which adaptively allocates resources according to different clients' requests to minimize the energy consumption. To improve flexibility and scalability, we propose a new the semi-asynchronous federated update model, which allows clients to aggregate asynchronously on the server, and accelerates the convergence rate of the global model. Empirical results show that the proposed Algorithm 1 not only accelerates the convergence speed of the global model, but also reduces the size of parameters that need to be uploaded. Besides, the proposed Algorithm 2 reduces the time difference caused by user heterogeneity. Eventually, the semi-asynchronous update model is better than the synchronous update model in communication time.**

***Index Terms*—Federated learning, mobile edge networks, deep deterministic policy gradient, semi-asynchronous update model, energy efficiency.**

Manuscript received 30 November 2022; revised 26 February 2023 and 8 May 2023; accepted 11 July 2023. Date of publication 25 July 2023; date of current version 19 December 2023. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62071306 and in part by Shenzhen Science and Technology Program under Grants JCYJ20200109113601723, JSGG20210802154203011, JSGG20210420091805014, and GJHZ20180418190529516. The review of this article was coordinated by Prof. Zhao Ming. *(Corresponding author: Yejun He.)*

Jinfeng Zhang, Wei Liu, and Yejun He are with the State Key Laboratory of Radio Frequency Heterogeneous Integration, Guangdong Engineering Research Center of Base Station Antennas, Shenzhen Key Laboratory of Antennas and Propagation, College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: zhangjf@szu.edu.cn; 942415342@qq.com; heyejun@126.com).

Zhou He is with the Department of Mechanical Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: zhe12@umd.edu).

Mohsen Guizani is with the Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi 51133, UAE (e-mail: mguizani@ieee.org).

Digital Object Identifier 10.1109/TVT.2023.3298787

I. Introduction

Currently, there are tens of thousands of connected Internet of Things (IoT) devices around the world. These devices typically have powerful computing and communication capabilities. As such, they can be deployed in densely populated areas for data collection (e.g., medical data [1] and air quality monitoring [2]). Based on these collected data, we can make predictions for unknown and intelligent applications with classification, which can facilitate taking appropriate actions according to the application [3].

In the traditional cloud-centric approach, datasets are uploaded to a centralized server for processing. On one hand, the inevitable communication delay and energy consumption are amplified in long-distance transmissions. On the other hand, client datasets may be subjected to malicious attacks during transmissions, revealing the client's privacy. Mobile Edge Computing (MEC) is proposed as an efficient method to solve the unavoidable latency and energy consumption problems due to powerful computing and storage capabilities [4]. In order to ensure that distributed training can still maintain efficient learning in the case of heterogeneous datasets, a decentralized machine learning (ML) approach called Federated Learning (FL) is introduced in [5]. In a FL algorithm, the datasets are normally distributed over numerous clients. These unique characteristics determine that the MEC adapts to the efficient deployment of FL.

In a FL, mobile devices exploit datasets to cooperatively train a ML model. The model parameters (i.e., the model weights) are uploaded to the server for aggregation. Generally, the process of uploading parameters needs to be repeated many times in order to achieve the desired accuracy [6]. FL converges smoothly on some heterogeneous settings (e.g., non-independent and identically distributed (non-i.i.d.) datasets [7]). The method to ensure the convergence of FL has been studied [8]-[9]. However, a large number of uncertain factors (e.g., heterogeneous clients and energy consumption) need to be considered in the real FL deployment in MEC.

Important challenges and future research directions for FL are discussed in [10]. These challenges are outlined as follows: *1*) statistical challenges. *2*) communication costs (e.g., high dimensionality of the updates data leads to communication consumption). *3*) resource allocation (e.g., resource competition between heterogeneous devices). These challenges determine whether FL can have a stable training and converges quickly.

Some factors such as learning accuracy, delay, and energy consumption are also very important when FL and MEC scenarios are jointly considered. Due to the heterogeneity of a large number of users in the mobile edge computing, it is difficult to guarantee the learning accuracy in FL. In the MEC scenario, we pay more attention to the optimization of the system energy consumption and delay. When FL and MEC are considered together, optimizing the energy consumption and the delay is a challenge under considering the influence of parameters (e.g., number of clients selected K and the number of local iterations selected E) in FL.

Traditional FL applies synchronous upload strategy. The aggregation server performs weighted aggregation after all datasets are uploaded. However, the synchronous upload strategy is impractical in MEC scenarios. A large number of datasets is transmitted to the aggregation server, which leads to transmission congestions. Based on this, the semi-asynchronous update model is designed in this paper, which allows the aggregation server to aggregate individually without uploading datasets from all users. The contributions of this work are summarized as follows.

- *Importance-Based updating:* First and on the basis of the observation of many previous experiments [11], most of the updating gradient values in FL use sparse matrix. Second, many of the gradient values are unconducive for the convergence of the global model. In line with the characteristics, we design a selection algorithm based on the gradient direction, and establish theoretical guarantees. The selected gradients for uploading accelerates the convergence rate of the global model.
- *Optimization cost FL design:* On the premise of the training accuracy of the local clients and the convergence rate of the global model, the energy consumption and delay during the local training are formulated as an optimization problem. To solve this mixed integer nonlinear programming (MINLP) and NP-Hard problem, the DDPG algorithm is proposed. In addition, we analyze the parameters (e.g., the number of clients selected K and the number of local iterations E) in the object problem influence about the optimal solution.
- *Semi-Asynchronous transmission design:* In order to minimize the FL communication cost of heterogeneous clients, we design a semi-asynchronous transmission update model as shown in Fig. 1, which reduces the waiting time caused by synchronization. Additionally, we formulate the transmission cost optimization problem, and theoretically analyse the bandwidth allocation and transmission time that are biconvex functions. Based on this theoretical analysis, we propose a low-complexity search algorithm to obtain the optimal solution for minimizing the energy consumption and delay.

The rest of the paper is organized as follows. We first present the related work in Section II. Then, we present the FL model optimization design in Section III. In Section IV, we analyze the local training system model and provide the cost optimization based on the DDPG algorithm. In Section V, we present a semi-asynchronous update model to solve the synchronization delay problem. Experimental results are given in Section VI and the conclusion is presented in Section VII.

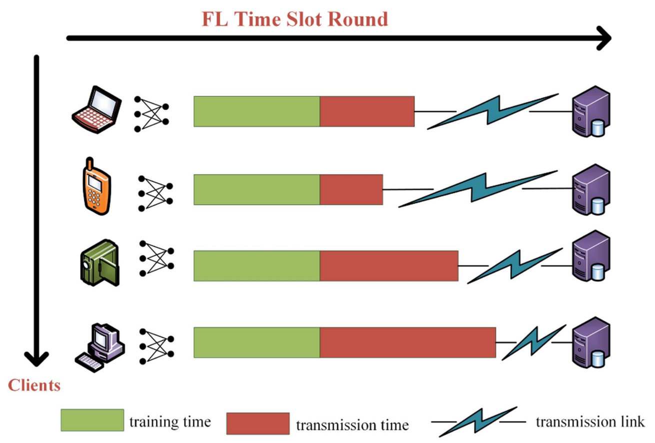

Fig. 1. Heterogeneous federated learning training round in a mobile network, where the clients complete the training immediately and the server aggregates asynchronously.

II. Related Work

Inspired by the protection of data privacy, the concept of FL was introduced in [12]. The FL allows clients to train on mobile devices, and aggregates the global model by uploading the trained parameters to avoid privacy leakage.

Recently, optimizing the FL communication delay was concerned by selecting gradient values of uploading. The focus of this strategy was not to upload the entire updates while to upload part of the updates according to a certain strategy in each communication round. Eliminating the sparse value in the updating values was applied in [13]. The latter proposed to select the updating gradient values based on the training loss value of different rounds, and established a weight table to optimize the selected gradient values. However, the simulation results showed that the algorithm was only applicable to i.i.d datasets yet the performance used non-i.i.d (i.e., heterogeneity of clients) datasets was unestablished. The optimization of correlation between gradients was studied in [14], which selected gradient values for uploading with fixed threshold correlation values. Unfortunately, the limiting section of clients that discontent the relative threshold was not applicable to real scenarios, which reduced the quality of experience for clients and diversity of datasets.

For deploying FL in MEC [15], [16], [17], we mainly focus on the optimization of the delay (e.g., training time and transmission time) and energy consumption (e.g., training energy and transmission energy). The optimization of the training time and the energy consumption was discussed in [18], [19], [20], which considered minimizing the FL energy and time consumptions through resources (e.g., client's CPU frequency and communication bandwidth) allocation under the premise of the global model convergence. Although distinctive opinions were studied in the literature as stated above, they did not consider the impact of client training accuracy when MEC and FL were combined. The challenges in optimizing FL and MEC that analytically connects the total energy and delay with the accuracy of the local client

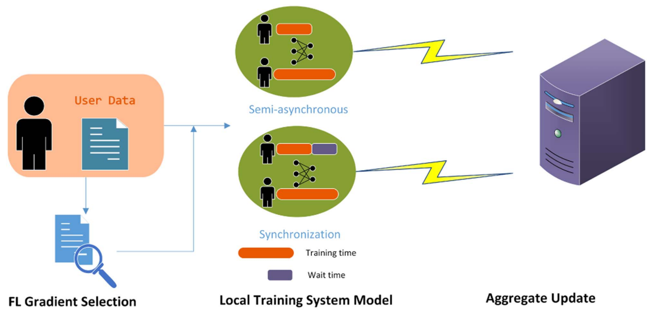

Fig. 2. Federated semi-asynchronous model design learning algorithm block diagram in a mobile edge network.

training and the convergence rate of the global model should be concerned.

The key point of the FL algorithm is uploading the gradient values to the aggregation server for averaging. The optimization of transmission time was studied in [20], [21], although different methods (e.g., convex optimization) were used to optimize the transmission time. However, they were all based on the strategy of synchronous updates. The waiting time (e.g., client disconnect and resource shortage) caused by the synchronous update strategy was unavoidable.

To address the above problem, we analyze how the characteristics (e.g., the number of clients selected K and the number of local iterations E) of FL affects the energy consumption, the local training time and the global model convergence rate. We theoretically analyze how to select useful gradient information (i.e., reduce the amount of datasets that need to be uploaded) to accelerate the convergence rate of the global model. Besides, a semi-asynchronous updating model was used to minimize the delay and energy consumption during transmission. The block diagram of the proposed algorithm is shown in Fig. 2.

III. FL Model Optimization Design

In this section, we concisely introduce FL. According to the observation results, a direction based update algorithm is proposed on the basis of the traditional FL algorithm. The complete steps are given in Algorithm 1.

A. Federated Learning

Generally, FL contains two main components: the dataset owners (e.g., participants) and the global model (e.g., FL aggregation server), respectively. Let $\mathcal{N} = \{1, ..., U\}$ indicates the set of dataset clients, and $\mathbb{D}_k$ represents a whole dataset of training data samples of client k. The process of FL can be roughly divided into the following processes

- *Step I (Initialization):* Aggregation server broadcasts the global model parameters W_G^0 to the clients participating before training.
- *Step II (Local model training):* After getting the parameters, the clients carry out local training (e.g., stochastic gradient descent) to obtain trained parameters w_k^t. Through a local iterative optimization, the optimal local parameters

Algorithm 1: FL Gradient Selection.

```
Input global model $W_G$ and the global gradient values $\vartheta_G$
  loop $k = 1, 2, 3, \ldots K$ for each epoch
    Using (5) to update local gradient values and record the loss values
    loop $\tau = 1, 2, \ldots \max$ for local training iteration
      if loss values decrease
        Execute (7) softly update gradient
      end if
      if $\tau = \max$
        Using (8) and (9) select the gradient values that meets the requirements
      end if
    end loop
  end loop
  Iteration terminates and output gradients.
```

is given by

$$w_k^{t^*} = \underset{w_k^t}{argmin}\, f(w_k^t), \tag{1}$$

where $w_k^{t^*}$ and w_k^t represent the local model parameters and optimal local parameters, respectively. Subsequently, parameters $w_k^{t^*}$ are uploaded to the server.

- *Step III (Global model aggregation):* On the server side, the synchronous update strategy is applied to obtain new global parameters W_G^t.

$$W_G^t = \frac{1}{\sum_{k\in\mathcal{N}} \mathbb{D}_k} \sum_{k=1}^{\mathcal{N}} \mathbb{D}_k w_k^t. \tag{2}$$

The local clients communicate with the server in many rounds to obtain the desired accuracy of the global model. Currently, the widespread and classical algorithm for aggregating the uploaded parameters is the FedAvg algorithm proposed in [6], which calculates the weighted average method to update the global model.

B. Key Insights

In FedAvg, the update of the global model depends on the weighted average of the gradient values uploaded by the clients participating in the training. Normally, partially uploaded gradient values incapably accelerate the convergence of the global model. When the global server executes as shown in (3), meaningless gradient parameters interfere the convergence rate of the model. The convergence of the FL has been proved in [6]. The important priori conditions in the proof are that the variance of the stochastic gradients in each client is bounded and the expected squared norm of stochastic gradients is uniformly bounded.

From [6], the variance of stochastic gradients can be expressed as

$$\mathbb{E}\|\nabla f_k(w_k^t, \xi_k^t) - \nabla f_k(w_k^t)\|^2 \le \delta_k^2. \tag{3}$$

The expected squared norm of stochastic gradients is given by

$$\mathbb{E}\|\nabla f_k(w_k^t, \xi_k^t)\|^2 \leq \mathcal{G}^2. \tag{4}$$

where $\mathbb{E}$ represents a mathematical expectation. $\| * \|$ represents square norm. δ_k^2 and $\mathcal{G}^2$ represent the bound of variance and squared norm, respectively. $\nabla f_k(w_k^t, \xi_k^t)$ represents gradient values under datasets ξ_k^t.

The above two formulae show that the modulus of the gradient and gradient divergence can be measured numerically in terms of the convergence of the analytical model.

C. Algorithm Based on Gradient Direction Analysis

Based on the local SGD and FedAvg [22], we prove the feasibility of the screening algorithm. After receiving the global model parameters, the local model parameters are calculated by

$$w_k^t = w_k^{t-1} - \eta \nabla f_k(w_k^t, \xi_k^t). \tag{5}$$

where η represents the learning rate. For the convenience of expression, we use character $\vartheta_k := -\eta\nabla f_k(w_k^t, \xi_k^t)$ to represent the latter part. From the above formula, the variation of w_k^{t+1} is determined by the ϑ_k. Based on this observation, the design of the algorithm is to pick the gradient values that accelerate the convergence of the global model. The complete proof is given in Appendix A.

D. Algorithm Design

Inspired by the analysis in [6], the average gradient direction of K clients can be used as the parameters of the global model. To obtain a fixed precision ϵ, the round of communication is given by

$$R = \mathcal{O}\left[\frac{1}{\epsilon}\left(\left(1+\frac{1}{K}\right)E\mathcal{G}^2 + \frac{\Gamma + \mathcal{G}^2}{E}\right)\right], \tag{6}$$

where Γ typically represents the differences between clients non-i.i.d datasets. $\mathcal{O}$ represents the equivalent order of magnitude and E represents the number of iterations.

We use a soft update to track the parameters of the global model after each local training to expedite the convergence of the global model.

$$\nabla f_{new,k}^t(w_k^t, \xi_k^t) = \mu\nabla f_k(w_k^t, \xi_k^t) + (1-\mu)\nabla F(W_G^{t-1}), \tag{7}$$

where μ represents a soft update parameter value. $\nabla F(W_G^{t-1})$ represents a gradient value of the global model at time $t-1$.

In the reverse update phase, we pick out the gradient values consistent with the gradient direction of the global model. The relevance of the local update ϑ_k with respect to the global update ϑ_G are calculated by

$$\vartheta_k = -\eta\nabla f_k(w_k^t, \xi_k^t) \tag{8}$$

$$\vartheta_G = -\eta\nabla F(W_G^{t-1}), \tag{9}$$

where ϑ_k indicates the size and direction of the update of the current user gradient values, and ϑ_G represents the vector direction of the global model parameters. Generally, local gradient parameters and global gradient parameters are characterized as two-dimensional parameters.

$$\mathscr{D}_{(x,y)}(\vartheta_k, \vartheta_G) = \begin{cases} 1, & sgn(\vartheta_{(x,y)}^k) = sgn(\vartheta_{(x,y)}^G) \\ 0, & sgn(\vartheta_{(x,y)}^k) \neq sgn(\vartheta_{(x,y)}^G), \end{cases} \tag{10}$$

where $\mathscr{D}$ represents the different gradient directions table between the client's model and the global model. $sgn()$ represents a symbolic function.

During the local training, the gradient table is established by comparing the positive and negative values of the local and global updated gradient values. The positive and negative gradient values reflect the directions of the weight update, and the gradient values with different directions are set to be zero. The gradient table values represent the relevance of the current position during the update process. The gradient values consistent with the global model direction are given by

$$w_k^t = w_k^{t-1} + \mathscr{D}_i\vartheta_k. \tag{11}$$

IV. Local Training System Model and Problem Formulation

In the previous section, we established a weight table to remove invalid gradient values. In FL, distributed users upload gradient values to the server for aggregation. When we jointly consider MEC-FL, removing the invalid gradient value can reduce the transmission energy consumption. In this section, we analyze the training time and energy consumption of the local distributed training in detail. Compared with previous work, we consider the impact of the local training accuracy on the energy consumption and delay. Finally, we propose the FL energy and time weighted sum optimization problem.

A. Computation Model

Despite that some previous work verified the effectiveness of FL, the optimization of the energy consumption and time were still required. The total cost of FL in the training phase, displayed in Fig. 1, involves the training time and the energy consumption.

Let j_k express the computation capacity of client k, which is measured by the number of CPU cycles per second. The computation time at client k needed for the data processing is calculated by

$$T_k^{com} = \frac{C_k D_k}{j_k}, \quad \forall k \in K, \tag{12}$$

where C_k (cycles/sample) is the number of CPU cycles required for computing one sample dataset at client k. D_k represents the dataset volume. According to [23], the energy consumption for computing the total number of $C_k D_k$ CPU cycles at client k is calculated by

$$E_k^{com} = \kappa C_k D_k j_k^2, \quad \forall k \in K. \tag{13}$$

where κ represents the battery capacity constant.

Under the above formulae, the total consumption of the local client k during training one round can be expressed as

$$\mathcal{C}_k^{com} = \sigma T_k^{com} + (1-\sigma)E_k^{com}, \quad \forall k \in K. \tag{14}$$

B. Local Training Accuracy and Problem Formulation

1) Local Training Accuracy: When the FL is deployed in the MEC, the training accuracy of the global model and the accuracy of the local model affect the training time and energy consumption. Previous work [18], [20] have considered the impact of the global model accuracy on the transmission time and energy consumption. However, the local training accuracy was not considered.

From Fig. 1, we can observe that the client's training time affects the subsequent transmission. When the training time is too long, the client is regarded as disconnected, which affects the aggregation of the global model.

In order to consider the influence of the local client training accuracy (i.e., ϱ) on FL, we utilize the convergence result in [24]

$$\mathcal{T}(\varrho, E, K) = \frac{1}{\varrho}\left(\frac{1}{2} + \frac{1}{2}\sqrt{1+\varrho(1+E+E^2 K)}\right), \quad (15)$$

where $\mathcal{T}$ represents the total number of the SGD. From the above formula, we can conclude that the training rounds required to achieve the desired accuracy (i.e., ϱ) are related to the number of iterations (i.e., E) and the number of clients selected (i.e., K).

2) Problem Formulation: Our target is to minimize the training time and the energy consumption while ensuring convergence. We transform the problem into the following

$$\mathbf{P1}: \min_{j_k, E, K, \varrho, \mathcal{R}} \sum_{r=1}^{\mathcal{R}} \sum_{k=1}^{K} \mathcal{C}_k^{com} \mathcal{T}(\varrho, E, K) \quad (16)$$

$$\text{s.t.} \quad 0 \le j_k \le j_{\max}, \quad \forall k \in K, \quad (17a)$$

$$0 \le \varrho \le 1, \quad \forall k \in K, \quad (17b)$$

$$\mathcal{R} \ge 1, E \ge 1, \quad \mathcal{R}, E \in \mathbb{Z}^+, \quad (17c)$$

$$1 \le K \le \mathcal{N}, \quad K \in \mathbb{Z}^+. \quad (17d)$$

We note that the optimal solution of problem P1 is affected by E and K, and these parameters affect the accuracy (i.e., ϵ) of the global model. To ensure the accuracy of the global model, we utilize the convergence result [6] to rewrite the problem as

$$\mathbf{P2}: \min_{j_k, E, K, \varrho, \mathcal{R}} \sum_{r=1}^{\mathcal{R}} \sum_{k=1}^{K} \mathcal{C}_k^{com} \mathcal{T}(\varrho, E, K). \quad (18)$$

$$\text{s.t.}\ 0 \le j_k \le j_{\max}, \quad \forall k \in K. \quad (19a)$$

$$0 \le \varrho \le 1, \quad 0 \le \epsilon \le 1, \quad \forall k \in K. \quad (19b)$$

$$E \ge 1, \quad 1 \le K \le \mathcal{N}, \quad K, E \in \mathbb{Z}^+. \quad (19c)$$

$$\mathcal{R} \le \frac{1}{\epsilon}\left(\left(1+\frac{1}{K}\right) E\mathcal{G}^2 + \frac{\mathcal{G}^2+\Gamma}{E}\right), \quad \mathcal{R} \in \mathbb{Z}^+. \quad (19d)$$

Obviously, P2 is an integer nonlinear programming problem, and the optimal solution is related to parameters (e.g., E, K, $\mathcal{R}$), which is difficult to infer an exact expression. In the following section, we verify the relationship between P2 and the hyperparameters.

C. Theoretical Verification

In this section, we analyze the impact of different superparameters on the accuracy of the global model.

- *Impact of* Γ*:* Inspired by [6], Γ is used to quantify the degree of non-i.i.d (i.e., clients heterogeneity), which is defined as

$$\Gamma = F(W_G)^* - \sum\nolimits_{k=1}^{\mathcal{N}} pick_k f(w_k^t)^*, \quad (20)$$

  where $F(W_G)^*$ and $f(w_k^t)^*$ indicate the minimum loss values of the global model and clients model, respectively. $pick_k$ expresses the probability of the client selection. If the dataset is non-i.i.d, Γ is non-zero, with its value reflecting the heterogeneity of the data distribution. Through the simulation results, we verify that it is generally a positive number and depends on the sampling plan.
- *Impact of* K*:* From the constraint of P2, we can conclude that the K value may affect the convergence rate, which is included in $\mathcal{O}(\frac{E\mathcal{G}^2}{K})$. Different values of K in the FL control the number of clients participating in the training. Intuitively, a larger K accelerates the convergence rate. However, the latter increases the amount of heterogeneous datasets, and reduces the convergence rate. The reason for this result is that the global model needs to trade-off the impact caused by the heterogeneity of the dataset, which leads to the stagnation of updates. The subsequent simulation results also confirm our analysis.
- *Impact of* E*:* First, we discuss the influence of E on the communication rounds, and derive the exact expression.

Theorem 1: The optimal solution E^* of problem P2 satisfies

$$E^* = \frac{\sqrt{\mathcal{G}^2+\Gamma}}{\mathcal{G}}. \quad (21)$$

This verifies our proof in Algorithm 1, which partial gradients altered does not influence the convergence. We give the proof in Appendix B.

D. Solving Problem P2

To solve P2, which is a non-convex problem with strong coupling between parameters, we propose to apply deep reinforcement learning algorithm. The reasons are as follows: 1) we incapably obtain the prior knowledge of clients, which causes heuristic algorithms impossibly. 2) it is possible to deploy deep neural networks on the client side. 3) for NP-hard problems, conventional heuristic algorithms cannot obtain a solution in a polynomial time complexity. Furthermore, the complexity increases as the number of clients increases. In this section, we first introduce the DDPG algorithm, and apply it to solve the problem of minimizing the energy consumption and delay.

1) Deep Deterministic Policy Gradient: DDPG is a deep deterministic strategy gradient algorithm, which is proposed to solve the problem of continuous motion control. DDPG optimizes agent policies through a series of agent environment interactions. Generally, DDPG modeled as MDP obtains the state space $\mathcal{S}$ and the action space $\mathcal{A}$. The agent aims to maximize the performance objective function and iteratively optimizes strategy (or policy) π. Through multiple rounds of interactions, the target of the agent is to maximize the expected reward. The expected reward is expressed as

$$\mathcal{R}_t = \sum_{t=1}^{T} \gamma r\left(s_t, a_t\right), \quad 0 < \gamma < 1, \tag{22}$$

where $r(s_t, a_t)$ indicates the reward at t slot state and action. γ represents the attenuation factor. The DDPG contains two types of networks, namely, policy network and critic network. The former outputs the action according to the results of the interaction with the environment, and the latter critics the action according to the Q values. For the policy network, with the deterministic policy π, the Q values are updated. Then, we have

$$Q^{\pi}\left(s_t, a_t\right) = \mathbb{E}\left[r\left(s_t, a_t\right) + \gamma Q^{\pi}\left(s_{t+1}, a_{t+1}\right)\right]. \tag{23}$$

The critic network guides the policy network by minimizing the experience loss function, according to [25], which can be written as

$$\mathcal{L}\left(\omega\right) = \mathbb{E}\left(y_t - Q^{\omega}\left(s_t, a_t|\omega\right)\right)^2, \tag{24}$$

$$y_t = r\left(s_t, a_t\right) + \gamma Q^{w'}\left(s_{t+1}, \pi\left(s_{t+1}|\zeta'\right)\right), \tag{25}$$

where w' and ζ' parameters represent the target critic network and the policy network, respectively, which are used to fix network parameters.

2) Algorithm Design Based on DDPG: We map the client's initial state to the environment, which is used to interact with the agent to obtain the state and reward of each time slot. The MEC server as an agent to interact with the environment obtains the state values (e.g., datasets and local client training accuracy ϱ), which reflects the heterogeneity among clients. Based on the previous analysis, we set the state values as

$$\mathbb{S}_{i\in K} = \left(\mathcal{C}_i, \mathcal{D}_i, \varrho_i, \Gamma_i\right), \tag{26}$$

where $\mathcal{C}_i$ is the number of CPU cycles required for computing one sample data. $\mathcal{D}_i$ represents the number of tasks that a client needs to upload. ϱ_i indicates the client's local training accuracy. Intuitively, different parameter's values map to different client's states.

Immediately, we input the state's values into the policy network as the input values. To maximize the $\mathcal{R}_t$ in (20), the strategy π is iteratively optimized, and the action values of the policy network is conceived as

$$\mathbb{A} \subseteq_{k=1}^{K} \left\{a_1, a_2, \ldots, a_k | a_k = \pi(s_k|\zeta)\right\}, \tag{27}$$

where a_k means that the CPU frequency with client k during the training, which can be used as a factor to adjust the training time and the energy consumption.

In order to find the optimal solution of the P2 problem, we set the reward in the DDPG algorithm as P2 problem. The reward function $\mathcal{R}_t$ is given by

$$\mathcal{R}_t = -\min_{j_k, E, K, \varrho, \mathcal{R}} \sum_{r=1}^{\mathcal{R}} \sum_{k=1}^{K} \mathcal{C}_k^{com} \mathcal{T}(\varrho, E, K), \tag{28}$$

where the reward $\mathcal{R}_t$ (optimal energy) is obtained by the appropriate action and different client's states. The solution of P2 problem corresponds to the action of the network output. The complete steps are illustrated in Algorithm 2.

Algorithm 2: Soft Update Based on DDPG Algorithm.

```
Step 0: Initialization.
    Randomly initialize critic network $Q(s, a|\omega)$ and
    policy network $\pi(s|\zeta)$
    Initialize target network $\acute{\omega}$ and $\acute{\zeta}$
    Initialize replay buffer $\mathcal{B}$.
Step 1: Training.
  loop $episode = 1, 2, 3, \ldots$ max for each epoch
    Select action $a_t = \pi(s_t|\zeta) + \phi$ according to policy and
    exploration noise;
    Execute action $a_t$ and observation $s_t$ to obtain;
    reward $r_t$ and new state $s_{t+1}$;
    Store information $(s_t, a_t, r_t, s_{t+1})$ in $\mathcal{B}$;
    Sample random mini-batch from $\mathcal{B}$;
    Execute (23) to obtain expected reward;
    According to (22) minimize loss;
  end loop
Step 2: Updating.
    Update policy network using the sampled policy
    gradient;
        $\nabla L(\pi(\zeta)) = \frac{1}{N}\sum_i \nabla Q(s_i, \pi(\zeta))\nabla\pi(s_i|\zeta).$
    Soft update target network parameters;
        $\omega' = \mu\omega + (1-\mu)\omega'$
        $\zeta' = \mu\zeta + (1-\mu)\zeta'$
    Iteration terminates and output reward.
```

V. Semi-Asynchronous Update Model Design

In the previous section, we leveraged the DDPG algorithm to optimize the training time of clients. The reason for this is that the heterogeneity of users leads to a large difference in the training time, which leads to the users dropping. Optimizing the client training time can reduce the problem that users can not participate in aggregation due to resource constraints.

Most of the previous work has considered synchronous update strategy, which undoubtedly leads to waste of the energy consumption. In this section, we first formulate the resource allocation to optimize the transmission time and energy consumption. Besides, a semi-asynchronous update model is established, and it is proven that it has a better convergence effect than the synchronous model.

A. Transmission Model and Problem Formulation

We assume that the local FL task communication adopts the orthogonal frequency division multiple access (OFDMA) technology. According to the designed semi-asynchronous model, we can upload parameters directly after training. The uplink rate of client k is given by [26]

$$\Re_k = \rho_k B \log(1 + \frac{p_k \hbar_k^2}{N_0}), \ \forall k \in K, \tag{29}$$

where $\rho_k \in [0, 1]$ is the bandwidth allocation ratio with client k. B represents the bandwidth resources. Furthermore, p_k and $\hbar_k$ indicate the client transmit power and the instant channel gain, respectively. N_0 represents a white gaussian noise.

The transmission delay between client k and a base station on the uplink is given by

$$T_k = \frac{\mathcal{L}_k}{\Re_k}, \ \forall k \in K, \tag{30}$$

where $\mathcal{L}_k$ is the uploaded gradient value. According to the above formulae, the total consumption of the local client k during the transmission can be expressed as

$$\mathcal{C}_k = p_k T_k = \frac{T_k N_0}{\hbar_k^2}\left(2^{\frac{\mathcal{L}_k}{\rho_k B T_k}} - 1\right), \ \forall k \in K. \tag{31}$$

Our goal is to minimize the energy consumption and delay during transmission. The server allocates the bandwidth before each round of transmission, and fixes it throughout the round. Therefore, the minimization of the energy consumption problem can be expressed as

$$\mathbf{P3}: \min_{\rho_k, T_k} \sum_{k=1}^{K} \mathcal{C}_k \tag{32}$$

$$\text{s.t.} \ \sum_{k=1}^{K} \rho_k = 1, \ \ \forall k \in K, \tag{33a}$$

$$0 \le \rho_k \le 1, \ \ \forall k \in K, \tag{33b}$$

$$0 < T_k \le T_{\max}, \ \ \forall k \in K. \tag{33c}$$

To solve the above problem, the base station minimizes the transmission energy consumption and delay by determining the bandwidth allocation. Here, we loosen the time requirement, because clients do not need to wait for other clients but only need to meet that the transmission time is less than the maximum endurance time.

B. Solving the Optimization Problem P3

In this subsection, we describe some properties of the optimization problem P3. Furthermore, we propose a roulette strategy with reducing the search complexity to solve P3. Obtaining the optimal ρ^* and T^* is displayed in Algorithm 3.

1) Characterizing Problem P3:

Theorem 2: Problem P3 is strictly biconvex.

Proof: For any $0 < T \le T_{\max}$, We have

$$\frac{\partial^2 \mathcal{C}}{\partial^2 T} = 2^{\frac{1}{\rho B T}} \frac{\ln 2 N_0}{T^2 B \hbar^2 \rho}\left[(\mathcal{L} - 1)\frac{2 \ln 2 N_0 \mathcal{L}}{T^3 \hbar^2 B^2 \rho^2}\right] > 0.$$

Similarly, for any $0 < \rho < 1$, we have

$$\frac{\partial^2 \mathcal{C}}{\partial^2 \rho} = 2^{\frac{\mathcal{L}}{\rho B T}} \frac{\ln 2 N_0 \mathcal{L}}{B \hbar^2}\left[\frac{2\rho B T + \ln 2\mathcal{L}}{\rho^4 B T}\right] > 0.$$

Algorithm 3: Roulette Strategy.

```
Step 0: Initialization.
    Random Initialize M feasible solution sets $X_0$
Step 1: iteration.
  loop $t = 1, 2, 3, \ldots \max$ for each time slot
    loop $i = 1, 2, 3, \ldots M$ for local training iteration
      Using (24) to calculate the energy consumption of
      the corresponding solution set;
      Execute (25), (26) mapping energy consumption
      values of different solution sets to different
      probabilities;
      Implement (27) generates better performing solution
      sets;
      Execute $\varepsilon$ greedy strategy;
      $x_i = \begin{cases} x_i, & \text{probability } (1-\varepsilon). \\ x_i = x_i + N_0, & \text{otherwise.} \end{cases}$
    end loop
  end loop
  Iteration terminates and output $(T^*, \rho^*)$.
```

We conclude that P3 is strictly biconvex.

Functions with biconvex properties can make use of search algorithms such as alternating search algorithms to achieve guaranteed optimal solutions. However, the feasible solution of the P3 problem is a real number domain, which undoubtedly increases the complexity of the search. Based on this, we design a roulette strategy to reduce the search complexity.

2) Algorithm Design: The idea of the roulette strategy is that the more valuable individuals are retained with a higher probability.

- *Enhance the individual:* Each individual corresponds to a set of feasible solutions, which obtains the corresponding cost. We map the individual to the corresponding probability.

$$ratio_i = \frac{\mathcal{C}_i}{\sum_{i=1}^{M} \mathcal{C}_i}. \tag{34}$$

$$cs_i = \sum_{i=1}^{M} ratio_i. \tag{35}$$

where $\mathcal{C}_i$ represents a group of feasible solutions. We replicate more individuals, and produce more derivation of new individuals as given by

$$new(x) \subseteq_{i=1}^{M} \begin{cases} old(x_i), & random(i) < cs(i), \\ i = i + 1, & otherwise. \end{cases} \tag{36}$$

- *Update the individual:* To avoid getting stuck in the local optima, we use ε greedy strategy to explore different solution sets, which solve the exploration-exploitation dilemma.

3) Semi-Asynchronous Update Model: To solve the inevitable waiting problem caused by the synchronous update, as displayed in Fig. 3, we discuss a single client execution FL.

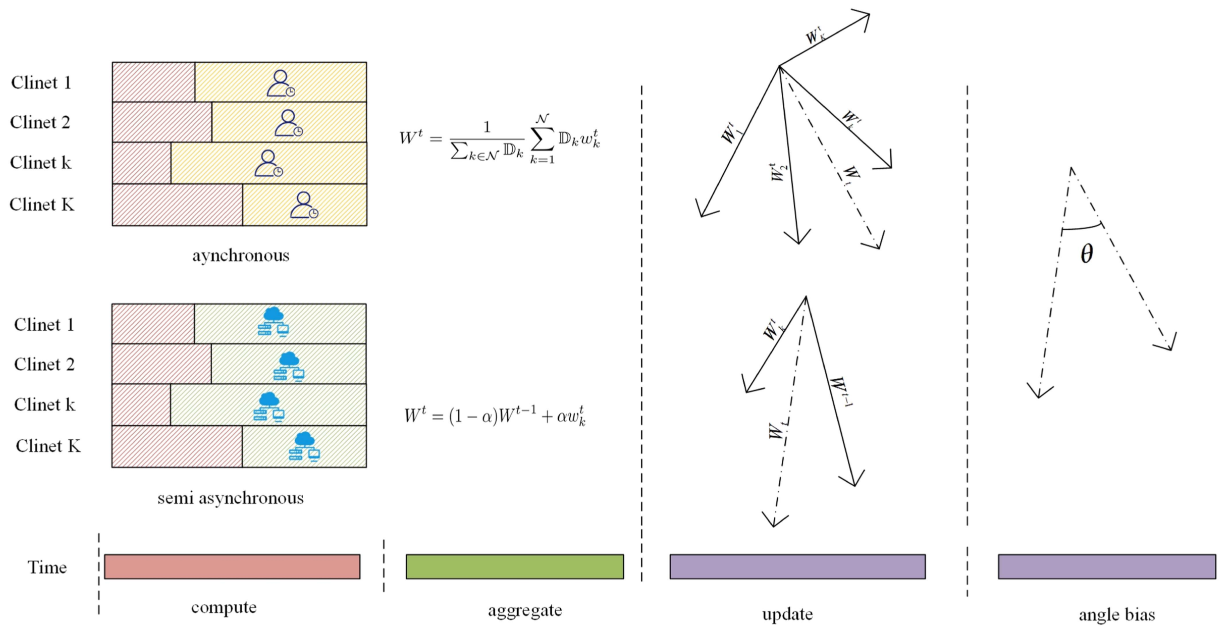

Fig. 3. Typical iteration round of the update procedure difference between synchronous and semi-asynchronous: i) synchronization needs to wait for all users to upload after training. In semi-asynchronous mode, users upload directly after training, ii) the former update requires the gradients of K users, however the latter only needs the gradients uploaded at the current moment.

In the t aggregate phase, the server accepts the trained model w_k^t from any client, and updates the global model by weighted averaging

$$W^t = (1-\alpha)W^{t-1} + \alpha w_k^t, \tag{37}$$

where $\alpha \in (0,1)$ is the mixing hyperparameter. Based on the semi-asynchronous aggregation formula, the difference between synchronous and semi-asynchronous reflects in the quantity of clients participating in the average weight, which affects the direction of the global parameters.

Corollary 1: The trade-off α on the convergence rate is given by

a) $0 < \alpha < \frac{1}{2}$:

The deflection angle θ gradually increases, and the weight of the global model parameters in the previous round is larger.

b) $\frac{1}{2} < \alpha < 1$:

The deflection angle θ gradually decreases, and the weight of the local model parameters of updating is larger.

c) $\alpha = \frac{1}{2}$:

The deflection angle is the largest. The global model parameters of the previous round and the uploaded client parameters have equivalent impact on the global model parameters of the next round.

We give the proof in Appendix C. Intuitively, the effect of reducing the quantity of clients participating in the weight update is neutralized by the adjustment of α.

VI. Simulation Results and Analysis

In this section, we assess the algorithms designed for each problem. We first introduce the simulation settings, and present the experimental results and analyses.

We evaluate our results on the real datasets (e.g., MNIST). For the non-i.i.d established datasets, following the identical settings in [5], which limits each client only to have part of the datasets.

A. Simulation Algorithm 1

1) Simulation Settings: We used the MNIST datasets for experiments. We divided the 60000 data samples into $\mathcal{N} = 100$ edge devices with the non-i.i.d dataset. Each device contains 600 unbalanced samples and the corresponding labels. For all simulation, we first randomly initialize the network parameters W_0, and set up a local training mini-batch $b = 30$. In each local training round, randomly selects K clients to execute E times local training.

2) Sampling Scheme: Intuitively, different clients selection schemes reflect the heterogeneity of datasets distribution. For the sampling scheme, we follow a similar setup in [6], which uses random sampling and uniform sampling, respectively.

- *Scheme I:* Assuming that $\mathcal{N}$ contains a subset of K randomly selected with replacement according to the sampling probabilities $pick_1, \ldots, pick_K$.
- *Scheme II:* Assuming that $\mathcal{N}$ contains a subset of K uniformly selected from $\mathcal{N}$ without replacement. Assume the data is balanced in the sense that $pick_1 = \cdots = pick_K = \frac{1}{\mathcal{N}}$.

3) Benchmark Settings: As to incarnate the performance of the optimization algorithm, we set up three baselines for comparison displayed in Fig. 4.

- *Benchmark 1:* Standard federal aggregation algorithm in [5], which is widely referenced in the literature. The main idea is that uploads all the gradient values, and

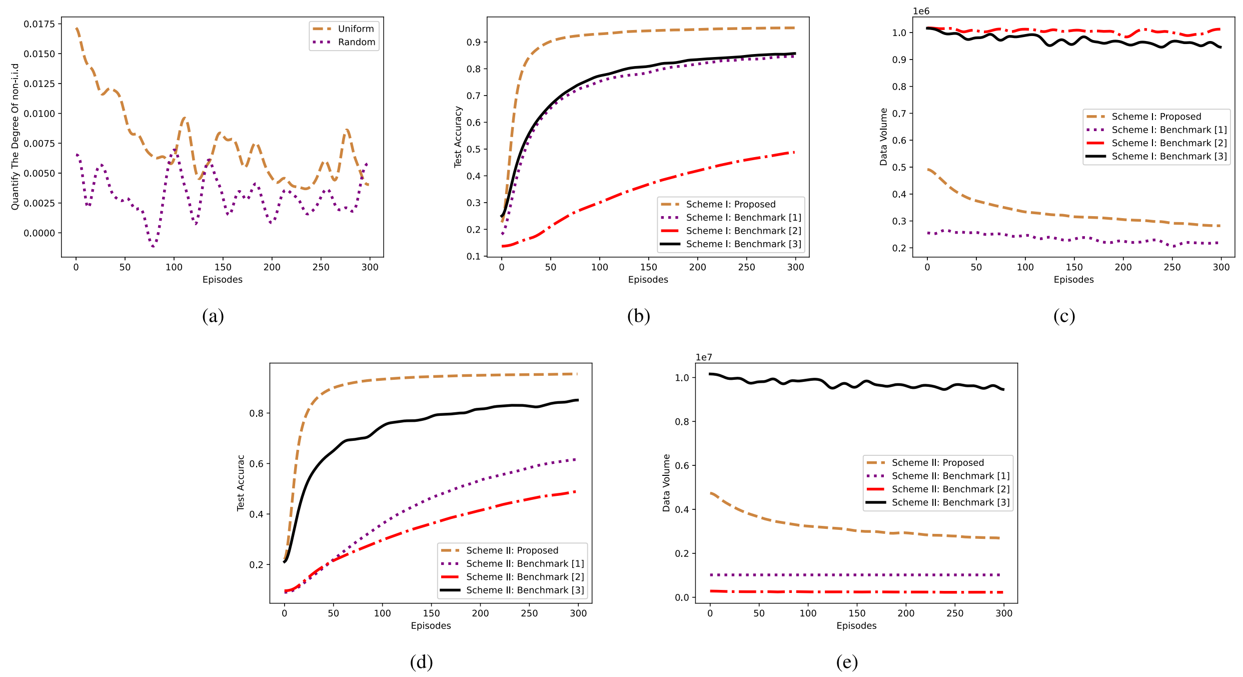

Fig. 4. Accuracy and data volume for different update strategies and sampling scheme with MNIST datesets. (a) Under the same setting, uniform sampling leads to greater heterogeneity of datasets. (b), (d) Our proposed algorithm has a faster convergence rate under both schemes. (c), (e) The proposed algorithm can use less upload data to achieve a faster convergence rate.

calculates the average values as parameter of the global model.

- *Benchmark 2:* Existing according to random weights update strategy in [13], which selects the uploaded gradient values by establishing the weight table. First, partial gradient values are randomly selected for training, and records the values change of the corresponding position. During the update phase, only the gradient value of the heavy coordinates is uploaded.
- *Benchmark 3:* Existing according to the correlation reflected in [14], which allows clients with large correlations to participate in the aggregation while clients with small correlations do not participate in the aggregation.

Fig. 4(a), (b) demonstrates the results of the proposed algorithm. It is clear that it has a faster convergence rate than other Benchmark algorithms under both sampling schemes. We proposed algorithm pick out gradient values that is conducive to accelerate the convergence of the global model.

The superiority of the proposed algorithm does not only reflect in the acceleration of convergence but also reduces the amount of datasets that need to be uploaded, which is shown in Figs. 4(c), (e). The reason for this phenomenon is that uncorrelated gradient values are eliminated to reduce the amount of uploaded datasets.

The gradient value selection method proposed in this paper eliminates sparse values and some gradient values that are different from the updating direction of the global model, which not only accelerates the convergence speed of the global model but also reduces the energy consumption and delay of datasets uploading.

B. *Simulation Algorithm 2*

1) Simulation Settings: For DDPG experiments, we initialize the parameters of the policy network and learning rate with 0.001 and 0.01, respectively. In each episode, clients participating in the training calculate the cost and each result is averaged over 50 episodes. In order to better fix the network parameters, the parameters of the two networks are updated in each 100 episodes.

2) Benchmark Settings: Fig. 4 compares the performance of the proposed algorithm with the following two benchmarks about the total learning time and energy consumption.

- *Random:* The clients randomly select the working frequency of the training, and calculate the current energy consumption and time.
- *Greedy:* Only chooses the parameters of the minimum energy consumption, without considering the constraints of the local training accuracy and the global model accuracy.

Figs. 5(a), (b) validates our analysis that the hyperparameter of K has little effects on the accuracy and empirical loss value of the global model. Fig. 5(c) demonstrates that the proposed algorithm optimizes the client's training time through resource allocation, which avoids the disconnection problem caused by the synchronized model.

Figs. 5(d), (e) demonstrate that our proposed algorithm achieves the resource optimization better than the two benchmarks. The proposed algorithm allocates resources according to the heterogeneity of clients, and minimizes the energy consumption and delay.

By combining the hyperparameters in FL with the optimization of energy consumption and delay in MEC, the DDPG algorithm designed in this paper optimizes the system energy

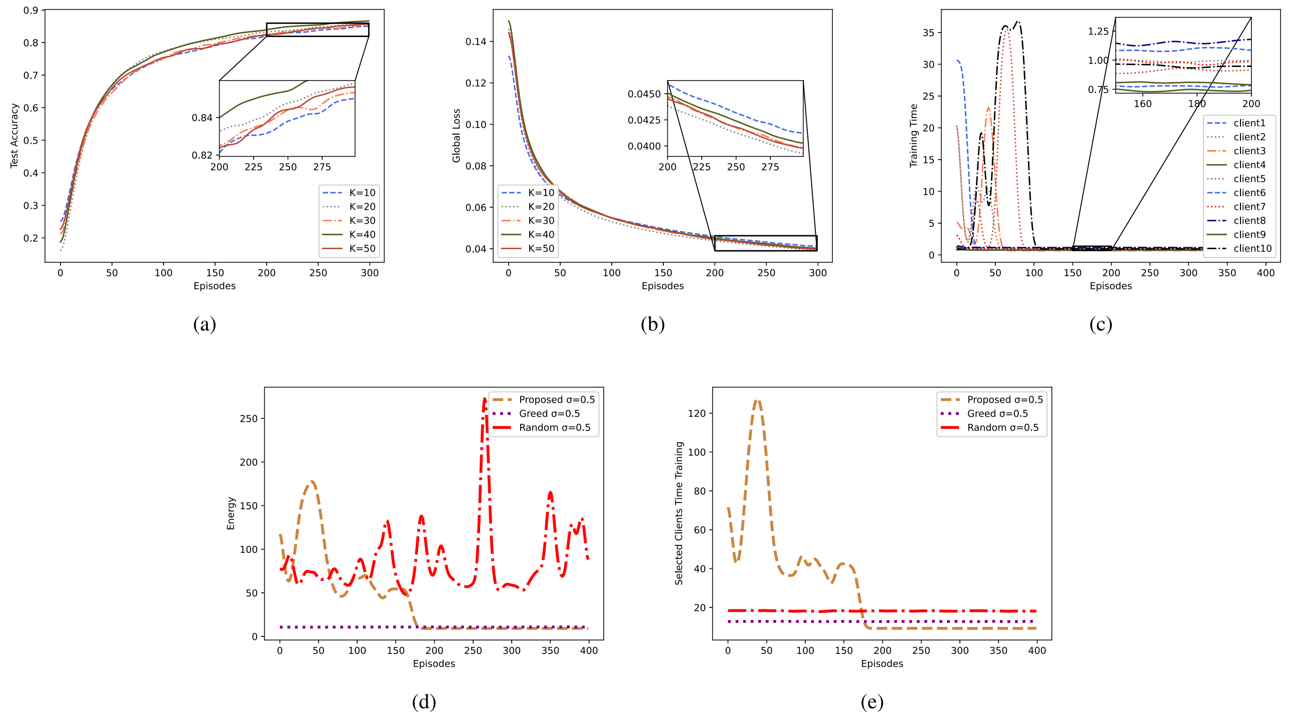

Fig. 5. Optimizing local training time and energy consumption based on DDPG. (a), (b) The effect of K value on the test accuracy and experience loss value is small. (c) Our proposed algorithm optimizes the training time of clients through resource allocation. (d), (e) The proposed algorithm optimization of time and energy consumption is better than other strategies.

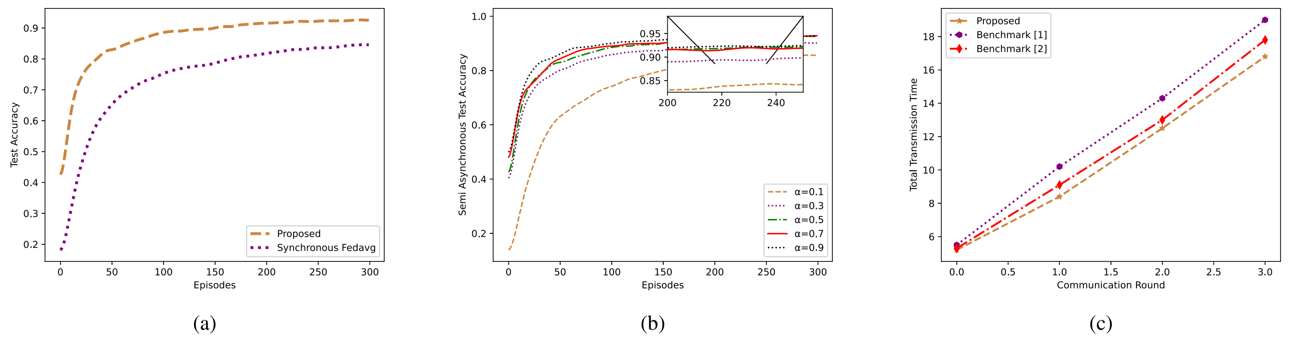

Fig. 6. Total transmission time for different scheduling strategies and semi-asynchronous update model training performance. (a) Our proposed method converges faster and more accurately than synchronous update. (b) Influence of parameter α on the convergence of semi-asynchronous models. (c) The proposed algorithm optimization of transmission time is better than other strategies.

consumption and delay without reducing the accuracy of the global model. Different sampling schemes and statistical challenges are considered.

C. Simulation Semi-Asynchronous Update Model

1) Simulation Settings: For experiments, consider an OFDMA system where the bandwidth B $= 1$ MHz, channel gains $\hbar_k$ are modeled as independent Rayleigh fading with average path loss set as 10^{-4}. The noise variance is $N_0 = 10^{-8}$ W/Hz, the number of CPU cycles for computing one sample data and task size, $\{C_k\}$ $\{D_k\}$, following the uniform distribution in the range of [10, 30] and [5, 8] MB, respectively.

2) Benchmark Settings: To evaluate our proposed semi-asynchronous update strategy, we compare it with two existing synchronous update schemes.

- *Benchmark 1:* Existing scheduling strategy in [21], which client occupies part of the bandwidth in time slot t, and the aggregation server adopts synchronous updates. This means that it needs to wait for all clients to complete the upload before updating.
- *Benchmark 2:* Establishing a scheduling strategy in [20], which client occupies all the communication resources at time t, and uploads the parameters in turn. When all client uploads parameters, the aggregation server performs the update operation.

The performance gap between the semi-asynchronous update model and the synchronous update model are displayed in Fig. 6(a). The reason for this observation is that, as heterogeneous datasets increase, the parameters uploaded by the clients only contain part of the information, and oversimplified averaging operation unable to help the global model to converge quickly.

Fig. 6(b) shows the effect of the parameter α on the convergence rate of semi-asynchronous models. Intuitively, larger α leads to a better convergence rate. The reason for this observation is that, as α increases, client's parameters occupy larger proportion during the soft update. The asynchronous upload mechanism ensures that the global model only needs to trade off a portion of the heterogeneous datasets, which means that there is no stagnation of the update due to a large amount of heterogeneous datasets.

Fig. 6(c) depicts the transmission time in different scheduling strategies, which also reflects the difference between a synchronous update strategy and an asynchronous update strategy. The semi-asynchronous update model proposed in this paper solves the synchronization waiting time issue caused by users heterogeneity. By controlling the regulator α, the convergence speed of the global model parameters can be changed, and the delay can be minimized without reducing the accuracy of the global model.

VII. Conclusion

In this paper, we studied the energy consumption and delay issues for FL in mobile edge networks. We proposed a novel method of selecting gradients based on gradient directions to solve the problem of massive uploaded datasets, and analyze the rationality of the selection algorithm. To improve the quality of the user experience, we considered the client's training accuracy in optimizing the local training energy consumption and delay, and analyzed the impact of different parameters (e.g., K, E and Γ). We also proposed a semi-asynchronous update strategy to avoid the waiting time caused by the synchronization, and analyzed the influence of different soft update parameters (e.g., α) about the global model update direction. Experimental results validated our theoretical analysis and revealed the effectiveness of the proposed algorithm.

Appendix A
Proof of Pick Algorithm

To prove by SGD and FedAvg, we first analyze the gradient direction requirements of the SGD in the case of convergence. Our object is to find a suitable ϱ^* to satisfy $f(\varrho_1) > f(\varrho_2) > \ldots > f(\varrho_k)$. We rewrite the target function as

$$\varrho^* = \underset{\varrho_k}{argmin}\, f(\varrho_k). \tag{38}$$

Assuming that the solution is found through the SGD at time t, the Taylor expansion of the difference from the optimal solution can be expressed as

$$f(\omega_t, \beta_t) - f(\omega^*, \beta^*) = \frac{\partial f(\omega^*, \beta^*)}{\partial \omega^*}(\omega - \omega^*) + \frac{\partial f(\omega^*, \beta^*)}{\partial \beta^*}(\beta - \beta^*). \tag{39}$$

The problem is transformed into finding the minimum value of the empirical loss function. Now, when the value on the right side of the equation is zero, the parameter at t is the optimal solution. We transform the form of vector inner product into the expression of a gradient update.

$$\begin{bmatrix}\omega^* \\ \beta^*\end{bmatrix} = \begin{bmatrix}\omega_t \\ \beta_t\end{bmatrix} - \lambda \begin{bmatrix}\frac{\partial f(\omega^*,\beta^*)}{\partial \omega^*} \\ \frac{\partial f(\omega^*,\beta^*)}{\partial \beta^*}\end{bmatrix}. \tag{40}$$

From the above formula, we get the expression of the gradient update as follows.

$$w_{t+1} = w_t - \lambda \nabla f_k(w_k^t). \tag{41}$$

$$\vartheta_k = -\lambda \nabla f_k(w_k^t). \tag{42}$$

To obtain the optimal parameters, the direction of the gradient affects the convergence rate. On this basis, we prove that the global model still converges when the partial gradient value is altered. $\overline{\vartheta}_k(w_k^t)$ represents the changed values.

$$\mathbb{E}\|\overline{\vartheta}_k(w_k^t) - \vartheta_k(w_k^t)\|^2 \leq \mathbb{E}\|\overline{\vartheta}_k(w_k^t) + \vartheta_k(w_k^t)\|^2 \leq 4\lambda^2\mathcal{G}^2. \tag{43}$$

The above formula shows that the direction of modifying only part of the gradient still satisfied Lipschitz.

Appendix B
Proof of Theorem 1

We first establish the functional relationship between communication rounds and iterations.

$$\mathcal{R}(E) = \frac{1}{\epsilon}\left(\left(1 + \frac{1}{K}\right)E\mathcal{G}^2 + \frac{\mathcal{G}^2 + \Gamma}{E}\right). \tag{44}$$

For arbitrary E, we have

$$\frac{\partial \mathcal{R}}{\partial E} = \frac{1}{\epsilon}\left(\left(1 + \frac{1}{K}\right)\mathcal{G}^2 + \frac{\mathcal{G}^2 + \Gamma}{E}\right), \tag{45}$$

$$\frac{\partial^2 \mathcal{R}}{\partial^2 E} = \frac{2\left(\mathcal{G}^2 + \Gamma\right)}{E^3} > 0. \tag{46}$$

Similarly, for any $1 < K < \mathcal{N}$, we obtain the same conclusion

$$\frac{\partial^2 \mathcal{R}}{\partial^2 K} = \frac{2E\mathcal{G}^2}{\epsilon K^3} > 0. \tag{47}$$

Since the domain of E and K is convex, we conclude that $\mathcal{R}(E)$ is convex.

If let $\frac{\partial \mathcal{R}}{\partial E} = 0$, we have

$$\frac{1}{\epsilon}\left(\left(1 + \frac{1}{K}\right)\mathcal{G}^2 + \frac{\mathcal{G}^2 + \Gamma}{E}\right) = 0, \tag{48}$$

$$E^* = \sqrt{\frac{K\left(\Gamma + \mathcal{G}^2\right)}{(K+1)\,\mathcal{G}^2}}. \tag{49}$$

In the reality scenario, thousands of clients participates in FL, K $\gg 1$, we have

$$E^* = \frac{\sqrt{\mathcal{G}^2 + \Gamma}}{\mathcal{G}}. \tag{50}$$

Therefore, (19) can be obtained.

Appendix C
Proof of Corollary 1

We first establish a functional relationship between weights α and angles θ.

$$\theta(\alpha) = \arccos\left[\frac{(1-\alpha)W^{t-1}\alpha w_k^t}{|(1-\alpha)W^{t-1}\alpha w_k^t|}\right]. \tag{51}$$

For the convenience of the expression, we let

$$A = |(1-\alpha)W^{t-1}\alpha w_k^t|,$$
$$B = (1-\alpha)W^{t-1}\alpha w_k^t.$$

For any $0 < \alpha < 1$, we have

$$\frac{\partial^2\theta(\alpha)}{\partial^2\alpha} = -\left[\left(1-(\frac{B}{A})^2\right)^{-\frac{3}{2}}\frac{(1-2\alpha)BW^{t-1}w_k^t}{A^2} + \frac{2\,W^{t-1}w_k^t}{A}\left(1-\left(\frac{(1-\alpha)W^{t-1}w_k^t}{A}\right)^2\right)^{-\frac{1}{2}}\right] < 0. \tag{52}$$

We conclude that $\theta(\alpha)$ is a concave function.

$$\frac{\partial\theta(\alpha)}{\partial\alpha} = \frac{1}{\sqrt{1-\left(\frac{B}{A}\right)^2}}\frac{(2\alpha-1)W^{t-1}w_K^t}{A}. \tag{53}$$

Following, we let $\frac{\partial\theta(\alpha)}{\partial\alpha} = 0$, we have

Case 1: if $\alpha = \frac{1}{2}, \theta(\alpha)$ takes the maximum value, which means the deflection angle is the maximum.

Case 2: if $0 < \alpha < \frac{1}{2}$, $\frac{\partial\theta(\alpha)}{\partial\alpha} < 0$, which is a decreasing function. The deflection angle decreases as α increases.

Case 3: if $\frac{1}{2} < \alpha < 1$, $\frac{\partial\theta(\alpha)}{\partial\alpha} > 0$, which is an increasing function. The deflection angle decreases as α decreases.

Therefore, α can trade off the value of the deflection angle, which completes this proof.

References

[1] R. Pryss, M. Reichert, J. Herrmann, B. Langguth,, and W. Schlee, “Mobile crowd sensing in clinical and psychological trials–A case study,” in *Proc. IEEE 28th Int. Symp. Comput.-Based Med. Syst.*, 2015, pp. 23–24.

[2] R. K. Ganti, F. Ye,, and H. Lei, “Mobile crowdsensing: Current state and future challenges,” *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[3] F. Samie, L. Bauer,, and J. Henkel, “From cloud down to things: An overview of machine learning in Internet of Things,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4921–4934, Jun. 2019.

[4] Y. Mao, C. You, J. Zhang, K. Huang,, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[5] B. McMahan et al., “Communication-efficient learning of deep networks from decentralized data,” in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[6] X. Li et al., “On the convergence of FedAvg on non-IID data,” in *Proc. ICLR*, 2020, *arXiv:1907.02189*.

[7] H. Yang, H. He, W. Zhang, and X. Cao, “FedSteg: A. federated transfer learning framework for secure image steganalysis,” *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1084–1094, Apr.–Jun. 2021.

[8] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. Vincent Poor, “Fast-convergent federated learning,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, Jan. 2021.

[9] H. Wu and P. Wang, “Fast-convergent federated learning with adaptive weighting,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 4, pp. 1078–1088, Dec. 2021.

[10] W. Y. B. Lim et al., “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, thirdquarter 2020.

[11] A. Jain et al., “GEMS: GPU-Enabled memory-aware model-parallelism system for distributed DNN training,” in *Proc. SC20: Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2020, pp. 1–15.

[12] W. Liu, L. Chen, Y. Chen, and W. Zhang, “Accelerating federated learning via momentum gradient descent,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 8, pp. 1754–1766, Aug. 2020.

[13] Z. Tao and Q. Li, “eSGD : Communication efficient distributed deep learning on the edge,” in *Proc. USENIX Workshop Hot Topics Edge Comput.*, 2018, pp. 1–6.

[14] L. Wang, W. Wang, and B. LI, “CMFL: Mitigating communication overhead for federated learning,” in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 954–964.

[15] P. Li, X. Huang, M. Pan, and R. Yu, “FedGreen: Federated learning with fine-grained gradient compression for green mobile edge computing,” in *Proc. IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.

[16] X. Huang, P. Li, R. Yu, Y. Wu, K. Xie, and S. Xie, “FedParking: A federated learning based parking space estimation with parked vehicle assisted edge computing,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9355–9368, Sep. 2021.

[17] D. Ye, X. Huang, Y. Wu, and R. Yu, “Incentivizing semisupervised vehicular federated learning: A multidimensional contract approach with bounded rationality,” *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18573–18588, Oct. 2022.

[18] X. Mo and J. Xu, “Energy-efficient federated edge learning with joint communication and computation design,” *J. Commun. Inf. Netw.*, vol. 6, no. 2, pp. 110–124, Jun. 2021.

[19] S. Wang et al., “Adaptive federated learning in resource constrained edge computing systems,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[20] 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. 1935–1949, Mar. 2021.

[21] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, “Cost-effective federated learning in mobile edge networks,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3606–3621, Dec. 2021.

[22] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, “Federated learning over wireless networks: Optimization model design and analysis,” in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2019, pp. 1387–1395.

[23] Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic computation offloading for mobile-edge computing with energy harvesting devices,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[24] D. Basu, D. Data, C. Karakus, and S. N. Diggavi, “Qsparse-Local-SGD: Distributed SGD with quantization, sparsification, and local computations,” *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 217–226, May 2020.

[25] H. Tan, “Reinforcement learning with deep deterministic policy gradient,” in *Proc. Int. Conf. Artif. Intell., Big Data Algorithms*, 2021, pp. 82–85.

[26] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, “User association for load balancing in heterogeneous cellular networks,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.

Jinfeng Zhang received Ph.D. degree in signal and information processing from the Dalian University of Technology, Dalian, China, in 2017. She joined the College of Information Engineering at Shenzhen University in 2004, where she is now an Associate Professor with College of Electronics and Information Engineering of Shenzhen University, Shenzhen, China. During 2017–2018, she was a Visiting Scholar with the University of Delaware, Newark, DE, USA. She was the recipient of Shenzhen high-level professional talents reserve level talent in 2019. Her main research interests include wireless communications and signal processing.

Wei Liu received the M.S. degree in electronics information from the College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China, in 2023. His research interests include wireless communications, mobile edge computing, and machine learning.

Yejun He (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2005. From 2005 to 2006, he was a Research Associate with the Department of Electronic and Information Engineering, the Hong Kong Polytechnic University, Hong Kong. From 2006 to 2007, he was a Research Associate with the Department of Electronic Engineering, Faculty of Engineering, the Chinese University of Hong Kong, Hong Kong. In 2012, he was a Visiting Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. From 2013 to 2015, he was an Advanced Visiting Scholar (Visiting Professor) with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Since 2006, he has been with College of Electronics and Information Engineering of Shenzhen University, Shenzhen, China, where he is currently a Full Professor with the College of Electronics and Information Engineering, the Director of Guangdong Engineering Research Center of Base Station Antennas and Propagation, the Director of the Shenzhen Key Laboratory of Antennas and Propagation, Shenzhen, China, and the Chair of IEEE Antennas and Propagation Society-Shenzhen Chapter. He was the recipient of the Shenzhen Overseas High-Caliber Personnel Level B (“Peacock Plan Award” B) and Shenzhen High-Level Professional Talent (Local Leading Talent). He was selected as Pengcheng Scholar Distinguished Professor, Shenzhen, and Minjiang Scholar Chair Professor of Fujian Province, in 2020 and 2022, respectively. He received the Shenzhen Science and Technology Progress Award in 2017, and has earned the Guangdong Provincial Science and Technology Progress Award for two times, in 2018 and 2023, respectively. He also obtained the IEEE APS outstanding Chapter Award in 2022. From 2023 to 2024, he is an Advanced Research Scholar (Visiting Professor) with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He has authored or coauthored more than 280 research papers, seven books, and holds about 20 patents. His research interests include wireless communications, antennas and radio frequency.

He was a Reviewer of various journals, such as IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, IEEE WIRELESS COMMUNICATIONS, IEEE COMMUNICATIONS LETTERS, *International Journal of Communication Systems* and so on. He was also a Technical Program Committee Member or a Session Chair of various conferences, including the IEEE Global Telecommunications Conference, IEEE International Conference on Communications, IEEE Wireless Communication Networking Conference, and the IEEE Vehicular Technology Conference. He was a TPC Chair of IEEE ComComAp 2021, General Chair of IEEE ComComAp 2019. He is the Principal Investigator for more than 40 current or finished research projects, including the National Natural Science Foundation of China, Science and Technology Program of Guangdong Province, and the Science and Technology Program of Shenzhen City.

He has been a Fellow of IET since 2016, a Senior Member of IEEE since 2009, a Senior Member of the China Institute of Communications since 2007 and a Senior Member of the China Institute of Electronics since 2011. He is currently an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, IEEE ANTENNAS AND WIRELESS PROPAGATION LETTERS, *IEEE Antennas and Propagation Magazine*, *International Journal of Communication Systems*, *China Communications* and *ZTE Communications*.

Zhou He is currently working toward the Ph.D. degree in mechanical engineering with the University of Maryland, College Park, MD, USA. His research interests include wireless communications, antennas, and reliability of electronic products.

Mohsen Guizani (Fellow, IEEE) received the B.S., (with distinction), M.S., and Ph.D. degrees in electrical and computer engineering from Syracuse University, Syracuse, NY, USA, in 1985, 1987, and 1990, respectively. He is currently a Professor of machine learning and the Associate Provost with the Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE. He was with different institutions in the USA. He is the author of ten books and more than publications. His research interests include applied machine learning and artificial intelligence, Internet of Things (IoT), intelligent autonomous systems, smart city, and cybersecurity. He was elevated to the IEEE Fellow in 2009 and was listed as a Clarivate Analytics Highly Cited Researcher in Computer Science in 2019, 2020, and 2021. He was the recipient of several research awards, including the 2015 IEEE Communications Society Best Survey Paper Award, Best ComSoc Journal Paper Award in 2021 and five Best Paper Awards from ICC and Globecom Conferences, 2017 IEEE Communications Society Wireless Technical Committee Recognition Award, 2018 AdHoc Technical Committee Recognition Award, and 2019 IEEE Communications and Information Security Technical Recognition Award. He was the Editor-in-Chief of IEEE Network and is currently on the Editorial Boards of many IEEE Transactions and Magazines. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He was the IEEE Computer Society Distinguished Speaker and the IEEE ComSoc Distinguished Lecturer.