Performance Optimization for Variable Bitwidth Federated Learning in Wireless Networks

Sihua Wang, *Student Member, IEEE*, Mingzhe Chen, *Member, IEEE*,
Christopher G. Brinton, *Senior Member, IEEE*, Changchuan Yin, *Senior Member, IEEE*,
Walid Saad, *Fellow, IEEE*, and Shuguang Cui, *Fellow*, *IEEE*

***Abstract*—This paper considers improving wireless communication and computation efficiency in federated learning (FL) via model quantization. In the proposed bitwidth FL scheme, edge devices train and transmit quantized versions of their local FL model parameters to a coordinating server, which, in turn, aggregates them into a quantized global model and synchronizes the devices. The goal is to jointly determine the bitwidths employed for local FL model quantization and the set of devices participating in FL training at each iteration. We pose this as an optimization problem that aims to minimize the training loss of quantized FL under a per-iteration device sampling budget and delay requirement. However, the formulated problem is difficult to solve without (i) a concrete understanding of how quantization impacts global ML performance and (ii) the ability of the server to construct estimates of this process efficiently. To address the first challenge, we analytically characterize how limited wireless resources and induced quantization errors affect the performance of the proposed FL method. Our results quantify how the improvement of FL training loss between two consecutive iterations depends on the device selection and quantization scheme as well as on several parameters inherent to the model being learned. Then, to address the second challenge, we show that the FL training process can be described as a Markov decision process (MDP) and propose a model-based reinforcement learning (RL) method to optimize action selection over iterations. Compared to model-free RL, this model-based RL approach leverages the derived mathematical characterization of the FL training process to discover an effective device selection and quantization scheme without imposing additional device communication overhead. Simulation results show that the proposed FL algorithm can reduce the convergence time by 29% and 63% compared to a model free RL method and the standard FL method, respectively.**

I. Introduction

Federated learning (FL) is an emerging edge computing technology that enables a collection of devices to collaboratively train a shared machine learning model without sharing their collected data [2]–[6]. During the FL training process, model parameters are trained locally on the device side and transmitted to a central center (e.g., at a base station (BS) coordinating the process across cellular devices) for global model aggregations. This procedure is repeated across several rounds until achieving an acceptable accuracy of the trained model [7]–[11].

The local training and device-server communication processes can each have a significant impact on the performance of FL. These considerations are particularly important in resource-constrained edge settings in which devices exhibit heterogeneity in their communication and computation resources (e.g., a low-cost sensor vs. a high powered drone collecting measurements) [12], [13]. To minimize the resulting delays due to local training and parameter transmission, one promising method that has been recently proposed is the consideration of machine learning quantization at each device [14]–[17]. In such schemes, the training and communication processes operate directly on quantized versions of the learning models, reducing the burden on device resources. However, efficient deployment of quantized FL over wireless networks poses several research challenges, related to the integration of quantization bitwidth considerations with the resulting FL training performance, which we study here.

A. Related Works

Recent works such as [18]–[29] have studied several important problems related to the implementation of quantized FL over wireless networks. The authors in [18] designed a universal vector quantization scheme for FL model transmission to minimize the quantization error. In [19], a heterogeneous quantization framework was proposed for the FL model uploading process to speed up the convergence rate. A robust FL scheme was developed in [20] to minimize the quantization errors and transmission outage probabilities under constraints on the training latency and device transmission powers. The authors in [21] proposed a hierarchical gradient quantization scheme for the FL framework to reduce the communication overhead while achieving similar learning performance. In [22], the authors investigated a communication-efficient FL approach based on gradient quantization to alleviate the required communication bits and training rounds. [23] further explored the impact of quantized communications on the performance of decentralized learning framework. The authors in [24] considered the extreme case of one-bit quantized local gradients for training the global FL model to reduce communication overhead. In [25], the energy efficiency of a quantized FL

S. Wang and C. Yin are with the Beijing Laboratory of Advanced Information Network, and the Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing 100876, China. Emails: sihuawang@bupt.edu.cn; ccyin@ieee.org.

M. Chen is with the Department of Electrical and Computer Engineering and Institute for Data Science and Computing, University of Miami, Coral Gables, FL, 33146 USA (Email: mingzhe.chen@miami.edu).

C. G. Brinton is with with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, Email: cgb@purdue.edu.

W. Saad is with the Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, 24060, USA, Email: walids@vt.edu.

S. Cui is currently with the School of Science and Engineering (SSE), the Future Network of Intelligence Institute (FNii), and the Guangdong Provincial Key Laboratory of Future Networks of Intelligence, the Chinese University of Hong Kong, and Shenzhen Research Institute of Big Data, Shenzhen, China, 518172; he is also affiliated with Peng Cheng Laboratory, Shenzhen, China, 518066, Email: shuguangcui@cuhk.edu.cn.

A preliminary version of this work [1] appears in the Proceedings of the 2022 IEEE Global Communications Conference (GLOBECOM).

This work was supported by the National Science Foundation under Grant CNS-2114267

scheme deployed over wireless networks is studied and the trade off between energy efficiency and accuracy is assessed. The authors in [26] proposed an adaptive quantized gradient method to optimize the number of communication bits employed during the FL iterations so as to reduce communication energy. In [27], an optimal vector quantizer was derived for minimizing the compression error of the local FL model update. In [28], the authors proposed a quantized FL algorithm for a device-to-device based wireless system to reduce the data transmission volume of FL models between devices. The authors in [29] developed a methodology for jointly optimizing the loss function, cost for transmitting quantized FL models, and available wireless resources to reduce communication cost and training time.

These prior works on quantized FL have each assumed that certain key parameters of the model being learned – such as smoothness and gradient diversity constants – are known in advance of the training process. Under these assumptions, traditional optimization methods can be used to capture the relationship between quantization error and FL performance so as to find the optimal FL training policy. In practice, these model parameters cannot be obtained by the central server until the FL training process has completed, and thus, the solution derived by these traditional optimization methods may not be appropriate. To address this challenge, one promising approach is to employ reinforcement learning (RL) approaches [30] for enabling the server to estimate these parameters over time through interaction with the devices during the training process, allowing discovery of a more effective FL policy.

Recently, a number of works [31]–[38] used RL algorithms to configure system parameters for FL performance optimization. In [31], the authors proposed a deep multi-agent RL to accelerate FL convergence while reducing the energy used for training. The authors in [32] designed a device selection scheme based on RL to minimize energy consumption and training delay, i.e., by searching for the most efficient set of devices to participate in each training iteration. A deep RL-based framework in [33] was proposed to maximize the long-term FL performance under energy and bandwidth constraints. In [34], the authors studied the use of a deep Q-network (DQN) to minimize wireless communication interruptions experienced by the FL framework due to device mobility. [35] used deep RL to jointly optimize training time and energy consumption via adjusting the CPU-cycle frequency of devices. The authors in [36] designed a DQN-based quantization allocation mechanism to improve the performance of FL. In [37], the authors employed a multiagent deep RL based quantization method to reduce the energy used for communication in FL framework. The authors in [38] analyzed the relationship between the global convergence and computational complexity in quantized a federated RL framework.

These prior works have thus employed RL methods to capture the relationship between FL performance and the training policy, in turn leading to improvements in different aspects of model training. However, with these methods, the coordinating server must collect numerous observations of different FL training policies by interacting with the devices over the environment, resulting in considerable delay for finding the optimal policy and encumbering FL convergence speed. To overcome this, we are motivated to develop *model-based* RL methods based on mathematical models of the quantized FL training process. Specifically, the coordinating server will estimate the associated FL model parameters based on information captured during the training process, minimizing the time and overhead required to discover the optimal FL policy.

B. Outline of Methodology and Contributions

The main contribution of this paper is a novel methodology to optimize quantized FL algorithms over wireless networks by a model based RL method that can estimate the FL training parameters and mathematically model the FL training process without continual interacting with the devices. To our best knowledge, *this is the first work that provides a systematic analysis of the integration of quantization bitwidth optimization into the FL framework.* Our key contributions include:

- We propose a novel quantized FL framework in which distributed wireless devices train and transmit their locally trained FL models to a coordinating server based on variable bitwidths. The server selects an appropriate set of devices to execute the FL algorithm with variable quantized bitwidths in each iteration. To this end, we formulate the joint device selection and FL model quantization problem as an optimization problem whose goal is to minimize training loss while accounting for communication and computation heterogeneity as well as non-i.i.d. data distribution across the devices. We quantify these heterogeneity factors in terms of service delay and communication bandwidth requirements.
- To solve this problem, we first analytically characterize the expected training convergence rate of our quantized FL framework with non-i.i.d. data distribution. Our analysis shows how the expected improvement of FL training loss between two adjacent iterations depends on the device selection scheme, the quantization scheme, and inherent properties of the model being trained under the non-i.i.d. setting. To find the tightest bound, we introduce a linear regression method for estimating these model properties according to observable training information at the server. Given these estimates, we show that the FL training process can be mathematically described as a Markov decision process (MDP) with consecutive global model losses constituting state transitions.
- To learn the optimal solution of the formulated MDP, we construct a model-based RL method that infers the action (i.e., device selection and quantization scheme) which maximizes the expected reward (i.e., minimize global model loss) in each training iteration. Compared to traditional model-free RL approaches, our proposed method enables the server to optimize the FL training process through minimal interaction with each device. The removal of continual device-server communication requirements is particularly useful in the bandwidth-limited wireless edge settings we consider.

Numerical evaluation results on real-world machine learning task datasets show that our proposed quantized FL methodology can reduce convergence time by up to 29% while

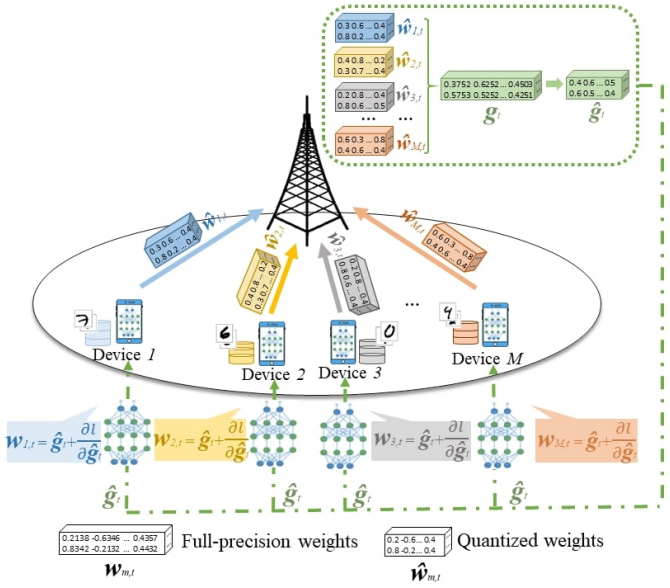

Fig. 1. Depiction of our proposed low bitwidth federated learning methodology deployed over multiple devices and one base station in a wireless network.

reducing the training iterations needed for convergence by up to 20% compared with existing FL baselines. Additionally, these results show how the number of devices and number of quantization bits jointly affect the performance of FL over wireless networks.

II. System Model and Problem Formulation

Consider a wireless network that consists of a set $\mathcal{M}$ of M devices connected upstream to a coordinating server, which we will assume is a server without loss of generality. These devices are aiming to execute an FL algorithm for training a machine learning model, as shown in Fig. 1. Each device m has N_m training data samples, and each training data sample n consists of an input feature vector $\boldsymbol{x}_{m,n} \in \mathbb{R}^{N_I \times 1}$ and (in the case of supervised learning) a corresponding label vector $\boldsymbol{y}_{m,n} \in \mathbb{R}^{N_O \times 1}$. The objective of the server and the devices is to minimize the global loss function over all data samples, which is given by

$$F(\boldsymbol{g}) = \min_{\boldsymbol{g}} \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N_m} f\left(\boldsymbol{g}, \boldsymbol{x}_{m,n}, \boldsymbol{y}_{m,n}\right), \tag{1}$$

where $\boldsymbol{g} \in \mathbb{R}^{Y \times 1}$ is a vector that captures the global FL model of dimension Y trained across the devices, with $N = \sum_{m=1}^{M} N_m$ being the total number of training data samples of all devices. $f\left(\boldsymbol{g}, \boldsymbol{x}_{m,n}, \boldsymbol{y}_{m,n}\right)$ is a loss function (e.g., squared error) that measures the accuracy of the generated global FL model $\boldsymbol{g}$ in building a relationship between the input vector $\boldsymbol{x}_{m,n}$ and the output vector $\boldsymbol{y}_{m,n}$.

A. Training Process of Low Bitwidth Federated Learning

In FL, devices and the server iteratively exchange their model parameters to find the optimal global model $\boldsymbol{g}$ that minimizes the global loss function in (1). However, due to limited computational and wireless resources, devices may not be able to train and transmit such large sized model parameters (e.g., as in the case of deep learning). To reduce the computation and transmission delays, bitwidth federated learning was proposed in [39]. Compared to the widely studied case of federated averaging [40], the FL model parameters in bitwidth FL are quantized. The overall training process of bitwidth FL is given as follows:

1) The server quantizes the initialized global learning model and broadcasts it to each device.
2) Each device calculates the training loss using the quantized global learning model and its collected data samples.
3) Based on the calculated training loss, the quantized learning model in each device is updated.
4) Each device quantizes its updated learning model.
5) The server selects a subset of devices for local FL model transmission.
6) The server aggregates the collected local FL models into a global FL model that will be transmit to devices.

Steps 2-6 are repeated until the optimal vector $\boldsymbol{g}$ is found.

From the training process, we see that, in bitwidth FL, each device uses a quantized FL model to calculate the training loss and gradient vectors during the training process. Therefore, the quantization scheme in bitwidth FL will affect the resource requirements of FL model training and transmission. This is significantly different from quantization-based FL algorithms [39] that must recover the quantized FL model during the training process, thus introducing additional computational complexity and reducing training efficiency. Next, we will introduce the training process mathematically.

1) Calculation of Training Loss of Each Device: We first introduce the calculation of each device's training loss for step 2. Without loss of generality, we will assume that a neural network is being trained; the quantization method can be used in other machine learning approaches (such as support vector machines (SVM) [41]) as well.

The weights of each device's local FL model are quantized into α_t bits. Through this, the full-precision neural network is transformed into a quantized neural network (QNN). When $\alpha_t = 1$, each QNN weight has two possible values, namely -1/0 or +1. Therefore, a neural network that consists of the weights with two possible values is called a binary neural network (BNN) [42]. Given the input vector $\boldsymbol{h}_{m,t}^{k}$ and the weight vector $\hat{\boldsymbol{g}}_t^k$ of the neurons in layer k that is represented by α_t bits, the output of each layer k at iteration t is given by [43]

$$\boldsymbol{h}_{m,t}^{k+1} = \begin{cases} \sigma\left(\boldsymbol{h}_{m,t}^{k} \odot \hat{\boldsymbol{g}}_t^k\right), & \text{if } \alpha_t = 1, \\ \sigma\left(\sum_{i=0}^{\alpha_t-1} \sum_{j=0}^{\alpha_t-1} 2^{i+j} (h_{m,t}^{k,i} \odot \hat{g}_t^{k,j})\right), & \text{if } \alpha_t > 1, \end{cases} \tag{2}$$

where $\sigma(\cdot)$ is the activation function and $\odot$ represents the inner product for vectors with bitwise operations. Given the outputs of all neuron layers $\boldsymbol{h}_{m,t} = [\boldsymbol{h}_{m,t}^1, \ldots, \boldsymbol{h}_{m,t}^K]$, the cross-entropy loss function can be expressed based on the neurons in an output layer $\boldsymbol{h}_{m,t}^K$ as

$$\begin{aligned} &f\left(\hat{\boldsymbol{g}}_t, \boldsymbol{x}_{m,n}, \boldsymbol{y}_{m,n}\right) \\ &= -\boldsymbol{y}_{m,n} \log(\boldsymbol{h}_{m,t}^K) + (1 - \boldsymbol{y}_{m,n}) \log(1 - \boldsymbol{h}_{m,t}^K), \end{aligned} \tag{3}$$

where $\hat{\boldsymbol{g}}_t = [\hat{\boldsymbol{g}}_t^1, ..., \hat{\boldsymbol{g}}_t^k, ..., \hat{\boldsymbol{g}}_t^K]$ is the quantized global FL model.

2) FL Model Update: A backward propagation (BP) algorithm based on stochastic gradient descent is used to update the parameters in QNN. The update function is expressed as

$$\boldsymbol{w}_{m,t+1} = \hat{\boldsymbol{g}}_t - \lambda \sum_{n \in \mathcal{N}_{m,t}} \frac{\partial f\left(\boldsymbol{g}, \boldsymbol{x}_{m,n}, \boldsymbol{y}_{m,n}\right)}{\partial \boldsymbol{g}}, \tag{4}$$

where λ is the learning rate, $\mathcal{N}_{m,t}$ is the subset of training data samples (i.e., minibatch) selected from device m's training dataset $\mathcal{N}_m$ at iteration t, $\boldsymbol{w}_{m,t+1}$ is the updated local FL model of device m at iteration $t+1$, and

$$\frac{\partial f}{\partial \boldsymbol{g}} = \frac{\partial f_{m,t}}{\partial \hat{\boldsymbol{g}}_t} \times \frac{\partial \hat{\boldsymbol{g}}_t}{\partial \boldsymbol{g}_t} = \frac{\partial f_{m,t}}{\partial \hat{\boldsymbol{g}}_t} \times \mathrm{Htanh}(\boldsymbol{g}_t), \tag{5}$$

where $\boldsymbol{g}_t$ represents the full-precision weights. $\mathrm{Htanh}(x) = \max(-1, \min(1, x))$ is used to approximate the derivative of the quantization function that is not differentiable. From (4) and (5), we can see that the weights are updated with full-precision values since the changes of the learning model update at each step are small.

3) FL Model Quantization at Device: As each local FL model is updated, these full-precision weights must be completely quantized into α_t bits, which is given by [44]

$$\hat{w}_{m,t}^{k,j}(\alpha_t) = Q(w_{m,t}^{k,j}, \alpha_t) = \begin{cases} \mathrm{sign}(w_{m,t}^{k,j}), & \text{if } \alpha_t = 1, \\ \frac{R\left((2^{\alpha_t}-1) w_{m,t}^{k,j}\right)}{2^{\alpha_t}-1}, & \text{if } 1 < \alpha_t < V, \\ w_{m,t}^{k,j}, & \text{if } \alpha_t = V, \end{cases} \tag{6}$$

where V is the bitwidth of the full-precision and $\mathrm{sign}(x) = 1$ if $x \geqslant 0$ and $\mathrm{sign}(x) = -1$, otherwise. $R(\cdot)$ is a rounding function with $R(x) = \lfloor x \rfloor$ if $x \leqslant \frac{\lfloor x \rfloor + \lceil x \rceil}{2}$, and $R(x) = \lceil x \rceil$, otherwise. From (6), we see that when $\alpha_t = 1$ (i.e., the binary case), if $w_{m,t}^{k,j} > 0$, we have $\hat{w}_{m,t}^{k,j} = 1$ with $w_{m,t}^{k,j}$ and $\hat{w}_{m,t}^{k,j}$ being j-th element in $w_{m,t}^{k,j}$ and $\hat{w}_{m,t}^{k,j}$ otherwise $\hat{w}_{m,t}^{k,j} = -1$. For $1 < \alpha_t < V$, $\boldsymbol{w}_{m,t}^{k}$ is quantized with increasing precision between -1 and 1. Finally, when $\alpha_t = V$, there is no quantization.

4) FL Model Transmission and Aggregation: Due to limited wireless bandwidth, the server may need to select a subset of devices to upload their local FL models for aggregation into the global model. Given the quantized local FL model $\hat{\boldsymbol{w}}_{m,t}$ of each device m at each iteration t, the update of the global FL model at iteration t is given by

$$\boldsymbol{g}_t(\boldsymbol{u}_t, \alpha_t) = \sum_{m=1}^{M} \frac{u_{m,t} N_{m,t}}{\sum_{m=1}^{M} u_{m,t} N_{m,t}} \hat{\boldsymbol{w}}_{m,t}(\alpha_t), \tag{7}$$

where $\frac{u_{m,t} N_{m,t}}{\sum_{m=1}^{M} u_{m,t} N_{m,t}}$ is a scaling update weight of $\hat{\boldsymbol{w}}_{m,t}$, with $N_{m,t}$ being the number of data samples used to train $\hat{\boldsymbol{w}}_{m,t}$ at device m. $\boldsymbol{g}_t(\boldsymbol{u}_t)$ is the global FL model at iteration t, and $\boldsymbol{u}_t = [u_{1,t}, \ldots, u_{M,t}]$ is the device selection vector, with $u_{m,t} = 1$ indicating that device m will upload its quantized local FL model $\hat{\boldsymbol{w}}_{m,t}$ to the server at iteration t, and $u_{m,t} = 0$ otherwise.

5) FL Model Quantization at the server: As the global FL model is aggregated based on the collected local FL models, the server must quantize it in low bitwidth that can be directly used to calculate the training loss at each device. This is given by [45]

$$\hat{\boldsymbol{g}}_t^k = Q(\boldsymbol{g}_t^k, \alpha_t) = \begin{cases} \mathrm{sign}(\boldsymbol{g}_t^k), & \text{if } \alpha_t = 1, \\ \frac{R\left((2^{\alpha_t}-1) \boldsymbol{g}_t^k\right)}{2^{\alpha_t}-1}, & \text{if } 1 < \alpha_t < V, \\ \boldsymbol{g}_t^k, & \text{if } \alpha_t = V. \end{cases} \tag{8}$$

B. Training Delay of Low Bitwidth Federated Learning

We next study the training delay of bitwidth FL. From the training steps, we can see that the delay consists of four components: (a) time used to calculate the training loss, (b) FL model update delay, (c) FL model quantization delay, and (d) FL model transmission delay. However, the FL model update delay is unrelated to the number of quantization bits α_t, since the models are updated with full-precision values. Thus, component (b) is constant with respect to our methodology and can be ignored. Then, the training delay is specified as follows:

1) Time Used to Calculate the Training Loss: The time used to calculate the training loss depends on the number of multiplication operations in (2) and (3). From (2), we can see that the computational complexity of each multiplication operation is related to the number of bits α_t used to represent each element in FL model vector. Specifically, given α_t, the time used to calculate the training loss is given by

$$l_{m,t}^{\mathrm{C}}(\alpha_t) = \rho \frac{\alpha_t^2 N^{\mathrm{C}}}{\vartheta f}, \tag{9}$$

where ρ is the time consumption coefficient depending on the chip of each device and N^{C} is the number of multiplication operations in the neural network. f and ϑ represent the frequency of the central processing unit (CPU) and the number of bits that can be processed by the CPU in one clock cycle, respectively.

2) FL Model Quantization Delay: Since the updated local FL model is in full-precision, each device must quantize its updated local FL model using (6) to reduce transmission delay. Given α_t, the quantization delay can be represented as [46]

$$l_{m,t}^{\mathrm{Q}}(\alpha_t) = \begin{cases} 0, & \text{if } \alpha_t = 1 \text{ or } \alpha_t = V, \\ \frac{D}{\vartheta f}, & \text{if } 1 < \alpha_t < V, \end{cases} \tag{10}$$

where D is the number of neurons in the neural network. In (10), when $\alpha_t = 1$ or $\alpha_t = V$, the quantization delay will be 0. When $\alpha_t = 1$, the value of quantized weight $\hat{\boldsymbol{w}}_{m,t}$ can be directly decided by the sign bit. When $\alpha_t = V$, no quantization takes place since we are dealing with full precision weights, i.e., $\hat{\boldsymbol{w}}_{m,t} = \boldsymbol{w}_{m,t}$. When $1 < \alpha_t < V$, the quantization delay incurred will increase based on the number of neurons in the neural network. For each neuron, the server will arithmetically perform the rounding, multiplication, and division operations according to (8).

3) FL Model Transmission Delay: To generate the global FL model that is aggregated by each quantized local FL model, each device must transmit $\hat{\boldsymbol{w}}_{m,t}$ to the server. To this end, we adopt an orthogonal frequency division multiple access (OFDMA) transmission scheme for quantized local FL model transmission. In particular, the server can allocate a set $\mathcal{U}$ of U uplink orthogonal resource blocks (RBs) to the devices for quantized weight transmission. Let W be the bandwidth of

each RB and P be the transmit power of each device. The uplink channel capacity between device m and the server over each RB i is $c_{m,t}\left(u_{m,t}\right) = u_{m,t} W \log_2\left(1+\frac{P h_{m,t}}{\sigma_N^2}\right)$ where $u_{m,t} \in \{0,1\}$ is the user association index, $h_{m,t}$ is the channel gain between device m and the server, and σ_N^2 represents the variance of additive white Gaussian noise. Then, the uplink transmission delay between device m and the server is $l_{m,t}^{\mathrm{T}}\left(u_{m,t}, \alpha_t\right) = \frac{D\alpha_t}{c_{m,t}(u_{m,t})}$ where $D\alpha_t$ is the data size of the quantized FL parameters $\hat{\boldsymbol{w}}_{m,t}$.

Since the server has enough computational resources and sufficient transmit power, we do not consider the delay used for global FL model quantization and transmission. Thus, the time that the devices and the server require to jointly complete the update of their respective local and global FL models at iteration t is given by

$$l_t(\boldsymbol{u}_t,\alpha_t) = \max_{m\in\mathcal{M}} u_{m,t}\left(l_{m,t}^{\mathrm{C}}(\alpha_t) + l_{m,t}^{\mathrm{Q}}(\alpha_t) + l_{m,t}^{\mathrm{T}}(u_{m,t},\alpha_t)\right). \tag{11}$$

Here, $u_{m,t} = 0$ implies that device m will not send its quantized local FL model to the server, and thus not cause any delay.

C. Problem Formulation

The goal is to minimize the FL training loss while meeting a delay requirement on FL completion per iteration. This minimization problem involves jointly optimizing the device selection scheme and the quantization scheme, which is formulated as follows:

$$\min_{\boldsymbol{U},\boldsymbol{\alpha}} \quad F(\boldsymbol{g}(\boldsymbol{u}_T, \boldsymbol{\alpha})), \tag{12}$$

$$\text{s.t.} \quad u_{m,t} \in \{0,1\}, \alpha \in [0, V] \text{ and } \alpha \in \mathbb{N}^+, \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{12a}$$

$$\sum_{m=1}^{M} u_{m,t} \leqslant U, \ \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{12b}$$

$$l_t\left(\boldsymbol{u}_t, \boldsymbol{\alpha}\right) \leqslant \Gamma, \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{12c}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_t, \ldots, \boldsymbol{u}_T]$ is a device selection matrix over all iterations with $\boldsymbol{u}_t = [u_{1,t}, \ldots, u_{M,t}]$ being a user association vector at iteration t, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_t, \ldots, \alpha_T]$ is a quantization precision vector of all devices for all iterations, and $\mathcal{T} = \{1, ..., T\}$ is the training period. Γ is the delay constraint for completing FL training per iteration, and T is a large constant to ensure the convergence of FL. In other words, the number of iterations that FL needs to converge will be less than T. (12a) indicates that each device can quantize its local FL model and can only occupy at most one RB for FL model transmission. (12b) ensures that the server can only select at most U devices for FL model transmission per iteration. (12c) is a constraint on the FL training delay per iteration.

The problem in (12) is challenging to solve by conventional optimization algorithms due to the following reasons. First, as the central controller, the server must select a subset of devices to collect their quantized local FL models for aggregating the global FL model. However, each local FL model that is generated by each device depends on the characteristics of the local dataset. Without such information related to the datasets, the server cannot determine the optimal device selection and quantization scheme for minimizing the FL training loss. Second, as the stochastic gradient decent method is adopted to generate each local FL model, the relationship between the training loss and device selection as well as quantization scheme cannot captured by the server via conventional optimization algorithms. This is because the stochastic gradient decent method enables each device to randomly select a subset of data samples in its local dataset for local FL model training, and hence, the server cannot directly optimize the training loss of each device. To tackle these challenges, we propose a model based RL algorithm that enables the server to capture the relationship between the FL training loss and the chosen device selection and quantization scheme. Based on this relationship, the server can proactively determine $\boldsymbol{u}_t$ and α_t so as to minimize the FL training loss.

III. Optimization Methodology

In this section, a model based RL approach for optimizing the device selection scheme $\boldsymbol{U}$ and the quantization scheme $\boldsymbol{\alpha}$ in (12) is proposed. Compared to traditional model free RL approaches that continuously interacts with edge devices to learn the device selection and quantization schemes, model based RL approaches enable the server to mathematically model the FL training process thus finding the optimal device selection and quantization scheme based on the learned state transition probability matrix. Next, we first introduce the components of the proposed model based RL method. Here, a linear regression method is used to learn the dynamic environment model in RL approach. Then, we explain the process of using the proposed model based RL method to find the global optimal $\boldsymbol{U}$ and $\boldsymbol{\alpha}$. Finally, the convergence and complexity of the proposed RL method is analyzed.

A. Components of Model Based RL Method

The proposed model based RL method consists of six components: a) agent, b) action, c) states, d) state transition probability, e) reward, and f) policy, which are specified as follows:

- **Agent**: The agent that performs the proposed model based RL algorithm is the server. In particular, at each iteration, the server must select a suitable subset of devices to transmit their local FL models and determine the number of bits used to represent each element in FL model matrix.
- **Action**: An action of the server is $\boldsymbol{a}_t = [\boldsymbol{u}_t, \alpha_t] \in \mathcal{A}$ that consists of the device selection scheme $\boldsymbol{u}_t$ and the quantization scheme α_t of all device at iteration t with $\mathcal{A}$ being the discrete sets of available actions.
- **States**: The state is $s_t = F(\boldsymbol{g}_t) \in \mathcal{S}$ that measures the performance of global FL model at iteration t with $F(\boldsymbol{g}_t)$ being the FL training loss and $\mathcal{S}$ being the sets of available states.
- **State Transition Probability**: The state transition probability $P\left(s_{t+1}|s_t, \boldsymbol{a}_t\right)$ denotes the probability of transiting from state s_t to state s_t' when action $\boldsymbol{a}_t$ is taken, which is given by

$$P\left(s_{t+1}|s_t, \boldsymbol{a}_t\right) = \Pr\{s_{t+1} = s_t'|s_t, \boldsymbol{a}_t\}. \tag{13}$$

Here, we need to note that in model free RL algorithms, the server does not know the values of a state transition

probability matrix. However, in our work, we analyze the convergence of FL and estimate the FL training parameters in the FL convergence analytical results so as to calculate the state transition probabilities. Using the state transition probability matrix can reduce the interactions between the server and edge devices thus improving the convergence speed of RL.

- **Reward**: Based on the current state s_t and the selected action $\boldsymbol{a}_t$, the reward function of the server is given by

$$r\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) = -F(\boldsymbol{g}(\boldsymbol{u}_t, \alpha_t)), \tag{14}$$

where $F(\boldsymbol{g}(\boldsymbol{u}_t, \alpha_t))$ is the training loss at iteration t. Note that, $r\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right)$ increases as $F(\boldsymbol{g}(\boldsymbol{u}_t, \alpha_t))$ decreases, which implies that maximizing the reward of the server can minimize the FL training loss.

- **Policy**: The policy is the probability of the agent choosing each action at a given state. The model based RL algorithm uses a deep neural network parameterized by $\boldsymbol{\theta}$ to map the input state to the output action. Then, the policy can be expressed as $\boldsymbol{\pi_\theta}\left(s_t, \boldsymbol{a}_t\right) = P(\boldsymbol{a}_t|s_t)$.

B. Calculation of State Transition Probability

In this section, we introduce the process of calculating the state transition probability that is used to reduce the interactions between the server and edge devices thus improving the convergence speed of RL. To this end, we must analyze the relationship between s_{t+1} and $(s_t, \boldsymbol{a}_t)$. First, we make the following assumptions, as done in [47]:

- **Assumption 1**: The loss function $F(x)$ is $L-$smooth with the Lipschitz constant $L > 0$, such that

$$||\nabla F(x) - \nabla F(y)|| \leq L||x - y||. \tag{15}$$

- **Assumption 2**: The loss function $F(x)$ is strongly convex with positive parameter μ, such that

$$F(\boldsymbol{g}_{t+1}) \geq F(\boldsymbol{g}_t) + (\boldsymbol{g}_{t+1} - \boldsymbol{g}_t)^T \nabla F(\boldsymbol{g}_t) + \frac{\mu}{2}||\boldsymbol{g}_{t+1} - \boldsymbol{g}_t||. \tag{16}$$

- **Assumption 3**: The loss function $F(x)$ is twice-continuously differentiable. Based on (15) and (16), we have

$$\mu \boldsymbol{I} \preceq \nabla^2 F(\boldsymbol{g}_t, \boldsymbol{x}_{mn}, \boldsymbol{y}_{mn}) \preceq L\boldsymbol{I}. \tag{17}$$

- We also assume that

$$\left\|\nabla f(\boldsymbol{g}_t, \boldsymbol{x}_{mn}, \boldsymbol{y}_{mn})\right\|^2 \leq \zeta_1 + \zeta_2 \left\|\nabla F(\boldsymbol{g}_t)\right\|^2, \tag{18}$$

where $F(\boldsymbol{g}_t) = \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N_m} f(\boldsymbol{g}_t, \boldsymbol{x}_{m,n}, \boldsymbol{y}_{m,n})$.

These assumptions can be satisfied by several widely used loss functions such as mean squared error, logistic regression, and cross entropy [48]. These popular loss functions can be used to capture the performance of implementing practical FL algorithms for identification, prediction, and classification. Based on these assumptions, next, we first derive the upper bound of the improvement of the FL training loss at one FL training step under the non-i.i.d. setting. Then, we further analyze the relationship between the FL training loss improvement and the selected action (i.e., the relationship between s_{t+1} and s_t when $\boldsymbol{a}_t$ is given). Based on the analytical result, we can calculate the state transition probability $P\left(s_{t+1}|s_t, \boldsymbol{a}_t\right)$. To obtain the upper bound of the FL training loss improvement at one FL training step under the non-i.i.d. setting, we first define the degree of the non-i.i.d. data distribution.

Definition 1: The degree of non-i.i.d. in the global data distribution can be characterized by [49]:

$$\epsilon = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \frac{u_{m,t} \epsilon_m N_{m,t}}{\sum_{m=1}^{M} N_{m,t}} \tag{19}$$

where $\epsilon_m = \nabla F(\boldsymbol{g}_t) - \nabla \widetilde{F}_m(\boldsymbol{g}_t)$ is the difference between the data distribution of device m and the global data distribution. We also assume that $\left\|\nabla f(\boldsymbol{g}_t, \boldsymbol{x}_{mn}, \boldsymbol{y}_{mn}) + \epsilon_m\right\|^2 \leq \zeta_1 + \zeta_2 \left\|\nabla F(\boldsymbol{g}_t)\right\|^2 + B\epsilon^2$ for some positive B with $F(\boldsymbol{g}_t) = \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N_m} f(\boldsymbol{g}_t, \boldsymbol{x}_{m,n}, \boldsymbol{y}_{m,n})$.

Using Definition 1, we derive the upper bound of the FL training loss improvement at one FL training step under the non-i.i.d. setting.

Lemma 1. The FL training loss improvement over one iteration (i.e., the gap between $\mathbb{E}\left(F(\boldsymbol{g}_{t+1})\right)$ and $\mathbb{E}\left(F(\boldsymbol{g}_t)\right)$) with a non-i.i.d. data distribution can be upper bounded as

$$\begin{aligned}\mathbb{E}\left(F(\boldsymbol{g}_{t+1})\right) - \mathbb{E}\left(F(\boldsymbol{g}_t)\right) \leq\ & \mathbb{E}\left(\left(\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t\right)\left(\nabla F(\boldsymbol{g}_t) - \epsilon\right)\right) \\ & + \frac{L}{2}\mathbb{E}\left(||\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t||^2\right) + \frac{L}{2}\mathbb{E}\left(||\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_{t+1}||^2\right),\end{aligned} \tag{20}$$

where $\boldsymbol{g}_t$ and $\hat{\boldsymbol{g}}_t$ are short for $\boldsymbol{g}_t(\boldsymbol{u}_t, \alpha_t)$ and $\hat{\boldsymbol{g}}_t(\boldsymbol{u}_t, \alpha_t)$, respectively. $\mathbb{E}(\cdot)$ is the expectation with respect to the Rayleigh fading channel gain $h_{m,t}$ and quantization error.

Proof: See Appendix A. ∎

From Lemma 1, we can see that, the upper bound of the FL training loss improvement at one iteration depends on $\hat{\boldsymbol{g}}_{t+1}(\boldsymbol{u}_{t+1}, \alpha_{t+1}) - \boldsymbol{g}_t(\boldsymbol{u}_t, \alpha_t)$ that is determined by the device selection vector $\boldsymbol{u}_t$ and quantization scheme α_t. To investigate how an action $\boldsymbol{a}_t = [\boldsymbol{u}_t, \alpha_t]$ affects the state transition in the considered bitwidth FL algorithm with non-i.i.d. data distribution, we derive the following theorem:

Theorem 1. Given the user selection vector $\boldsymbol{u}_t$ and quantization scheme α_t, the upper bound of $\mathbb{E}\left(F(\boldsymbol{g}_{t+1})\right) - \mathbb{E}\left(F(\boldsymbol{g}_t)\right)$ in non-i.i.d. data distribution can be given by

$$\begin{aligned}&\mathbb{E}\left(F(\boldsymbol{g}_{t+1})\right) - \mathbb{E}\left(F\left(\boldsymbol{g}_t\right)\right) \\ \leq\ & \frac{1}{2L}\left(-1 + \frac{4\left(N - A\right)^2\left(\mathbb{E}\left\|\Delta\left(\alpha_t\right)\right\| + 1\right)\zeta_2}{N^2}\right)\left\|\nabla F(\boldsymbol{g}_t)\right\|^2 \\ & + \frac{\mathbb{E}\left\|\Delta\left(\alpha_t\right)\right\| + 1}{2L}\left(\frac{4\left(N - A\right)^2\left(\zeta_1 + B\epsilon^2\right)}{N^2} + L^2\mathbb{E}\left\|\Delta\left(\alpha_t\right)\right\|\right) \\ & + \mathbb{E}\left(\Delta\left(\alpha_t\right)^2\right),\end{aligned} \tag{21}$$

where $A = \sum_{m=1}^{M} u_{m,t} N_{m,t}$ represents the sum of all selected devices' data samples that are used to train their local models, $\Delta(\alpha_t) = \hat{\boldsymbol{g}}_t(\alpha_t) - \boldsymbol{g}_t$ is the quantization error of the global FL model that depends on the quantization scheme $\boldsymbol{\alpha}$, $\mathbb{E}\left\|\Delta(\alpha_t)\right\| = M2^{-\alpha_t}$ is the unbiased quantization function defined in (6).

Proof: See Appendix B. ∎

From Theorem 1, we can see that, the relationship between $\mathbb{E}\left(F(\boldsymbol{g}_{t+1})\right)$ and $\mathbb{E}\left(F(\boldsymbol{g}_t)\right)$ (i.e., s_{t+1} and s_t) depends on the selected action $\boldsymbol{a}_t$ as well as the constants $1/L$, ζ_1, ζ_2, and $B\epsilon^2$. However, we do not know the values of $1/L$, ζ_1, ζ_2, and $B\epsilon^2$ since they are predefined in assumptions (15)–(19). To find the tightest bound in (21), we must find the values of $1/L$, ζ_1, ζ_2, and $B\epsilon^2$ so as to build the relationship between s_{t+1} and s_t and calculate the state transition probability $P\left(s_{t+1}|s_t,\boldsymbol{a}_t\right)$. To this end, a linear regression method [50] is used to determine the values of L, ζ_1, ζ_2, and $B\epsilon^2$ since the relationship between $\mathbb{E}\left(F(\boldsymbol{g}_{t+1})\right)-\mathbb{E}\left(F(\boldsymbol{g}_t)\right)$ and these constants are linear. The regression loss function defined as

$$\begin{aligned}&\mathcal{J}(L,\zeta_1,\zeta_2,B\epsilon^2)\\ =&\frac{1}{I}\sum_{i=1}^{I}\left(\left(\mathbb{E}\left(F(\boldsymbol{g}_{t+1})^{(i)}\right)-\mathbb{E}\left(F(\boldsymbol{g}_t)^{(i)}\right)\right)\right.\\ &\left.-K\left(L,\zeta_1,\zeta_2,B\epsilon^2|F(\boldsymbol{g}_t)^{(i)},\boldsymbol{a}_t^{(i)},F(\boldsymbol{g}_{t+1})^{(i)}\right)\right)^2,\end{aligned}\tag{22}$$

where I is the number of real interactions between the server and edge devices used to estimate $1/L$, ζ_1, ζ_2, and $B\epsilon^2$. $K\Big(L,\zeta_1,\zeta_2,B\epsilon^2|F(\boldsymbol{g}_t)^{(i)},\boldsymbol{a}_t^{(i)},F(\boldsymbol{g}_{t+1})^{(i)}\Big)$ is the upper bound of the FL training loss at one FL training step obtained in (31). $\boldsymbol{b}^{(i)}=(F(\boldsymbol{g}_t)^{(i)},\boldsymbol{a}_t^{(i)},F(\boldsymbol{g}_{t+1})^{(i)})$ is the set of recorded pairs consisted of FL training loss and the selected action observed by the server and devices. $\boldsymbol{b}^{(i)}$ will be used to estimate the values of $1/L$, ζ_1, ζ_2, and $B\epsilon^2$. Specifically, given $\mathcal{B}=\left\{\boldsymbol{b}^{(0)},\ldots,\boldsymbol{b}^{(i)},\ldots,\boldsymbol{b}^{(I)}\right\}$, L, ζ_1, ζ_2, and $B\epsilon^2$ are updated using a standard gradient descent method

$$\begin{aligned}&L=L-\iota_L\frac{\partial\mathcal{J}(L,\zeta_1,\zeta_2,B')}{\partial L},\quad \zeta_1=\zeta_1-\iota_{\zeta_1}\frac{\partial\mathcal{J}(L,\zeta_1,\zeta_2,B')}{\partial\zeta_1},\\ &\zeta_2=\zeta_2-\iota_{\zeta_2}\frac{\partial\mathcal{J}(L,\zeta_1,\zeta_2,B')}{\partial\zeta_2},\quad B'=B'-\iota_{B'}\frac{\partial\mathcal{J}(L,\zeta_1,\zeta_2,B')}{\partial B'},\end{aligned}\tag{23}$$

where $B'=B\epsilon^2$. ι_L, ι_{ζ_1}, ι_{ζ_2}, and $\iota_{B'}$ are learning rates for parameters L, ζ_1, ζ_2, and B'.

Given the values of L, ζ_1, ζ_2, and $B\epsilon^2$, the gap between $\mathbb{E}(F(\boldsymbol{g}_{t+1}))$ and $\mathbb{E}(F(\boldsymbol{g}_t))$ can be estimated according to our upper bound. Based on the definition of the state, the state transition probability $P(s_t+1|s_t,\boldsymbol{a}_t)$ is given by

$$P\left(s_{t+1}|s_t,a_t\right)=\begin{cases}1,\text{if } s_{t+1}=s_t+K',\\ 0,\text{ otherwise.}\end{cases}\tag{24}$$

where $K'=K\left(L,\zeta_1,\zeta_2,B\epsilon^2|F(\boldsymbol{g}_t)^{(i)},\boldsymbol{a}_t^{(i)},F(\boldsymbol{g}_{t+1})^{(i)}\right)$.

C. Optimization of Device Selection and Quantization Scheme

Having the state transition probability $P\left(s_{t+1}|s_t,\boldsymbol{a}_t\right)$, next, we introduce the optimization of $\boldsymbol{\pi_\theta}$ so as to find the optimal device selection scheme $\boldsymbol{u}_t$ and quantization scheme α_t. Optimizing $\boldsymbol{\pi_\theta}$ for minimizing the FL training loss corresponds to minimizing

$$\mathcal{L}(\boldsymbol{\theta})=\sum_{(s_t,\boldsymbol{a}_t)\in\boldsymbol{\tau}}P(s_0)\prod_{t=1}^{T}\boldsymbol{\pi}_\theta(s_{t-1},\boldsymbol{a}_t)P(s_t|s_{t-1},\boldsymbol{a}_t)\sum_{t=1}^{T}r\left(\boldsymbol{s}_t,\boldsymbol{a}_t\right),\tag{25}$$

where $\boldsymbol{\tau}=\{s_0,\boldsymbol{a}_0,\ldots,s_T,\boldsymbol{a}_T\}$ is the trajectory replay buffer.

Given (25), the optimization of policy network $\boldsymbol{\theta}$ is

$$\max_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}).\tag{26}$$

We update $\boldsymbol{\pi_\theta}$ using a standard gradient descent method

$$\boldsymbol{\theta}=\boldsymbol{\theta}+\iota\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}),\tag{27}$$

where α is the learning rate and the policy gradient is

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})=&\sum_{(s_t,\boldsymbol{a}_t)\in\boldsymbol{\tau}}P(s_0)\prod_{t=1}^{T}\boldsymbol{\pi}_\theta(s_{t-1},\boldsymbol{a}_t)P(s_t|s_{t-1},\boldsymbol{a}_t)\sum_{t=1}^{T}r\left(\boldsymbol{s}_t,\boldsymbol{a}_t\right)\\ =&\frac{1}{T}\sum_{t=1}^{T}r\left(\boldsymbol{s}_t,\boldsymbol{a}_t\right)\nabla\log\boldsymbol{\pi}_{\boldsymbol{\theta}}(s_t,\boldsymbol{a}_t).\end{aligned}\tag{28}$$

D. Proposed Method for FL with Nonconvex Loss Functions

In Sections III. A, B, and C, we proposed a novel model based RL to optimize the device selection and quantization scheme so as to minimize FL training loss. Here, we extend the designed RL for FL with non-convex loss functions. First, we derive the convergence of FL with non-convex loss functions. In particular, we first replace convex Assumptions 2 and 3 with the following conditions:

Condition 1 [51]: The gradient of the non-convex loss function $F(x)$ is bounded by a nonnegative constant B, i.e., $||\nabla F(x)||\leqslant C$.

Condition 2 [52]: Function $F(x)$ is μ-nonconvex such that all eigenvalues of ∇^2F lie in $[-\mu,L]$, for some $\mu\in(0,L]$.

Condition 3 [52]: The Hessian of the loss function $F(x)$ is γ-Lipschitz continuous, such that

$$||\nabla^2F(x)-\nabla^2F(y)||\leqslant\gamma||x-y||.\tag{29}$$

Together with Assumption 1, and Conditions 1 and 3, loss function $F(x)$ is L_γ-smooth for $L_\gamma=4L+\gamma\iota C$ with $\iota\in[0,1/L]$ [Lemma 4.2, [52]]. Based on (29), Lemma 1 can be rewritten as

$$\begin{aligned}\mathbb{E}\left(F\left(\boldsymbol{g}_{t+1}\right)\right)-\mathbb{E}\left(F\left(\boldsymbol{g}_t\right)\right)\leq&\mathbb{E}\left(\left(\hat{\boldsymbol{g}}_{t+1}-\boldsymbol{g}_t\right)\left(\nabla F(\boldsymbol{g}_t)-\epsilon\right)\right)\\ &+\frac{L_\gamma}{2}\mathbb{E}\left(\left\|\hat{\boldsymbol{g}}_{t+1}-\boldsymbol{g}_t\right\|^2\right)+\frac{L_\gamma}{2}\mathbb{E}\left(||\hat{\boldsymbol{g}}_{t+1}-\boldsymbol{g}_{t+1}||^2\right).\end{aligned}\tag{30}$$

Then, the convergence of our FL methodology with nonconvex loss functions is shown in the following theorem.

Theorem 2. Given the user selection vector $\boldsymbol{u}_t$ and quantization scheme α_t, an upper bound $\mathbb{E}\left(F(\boldsymbol{g}_{t+1})\right)-\mathbb{E}\left(F(\boldsymbol{g}_t)\right)$ can be obtained as

$$\begin{aligned}&\mathbb{E}\left(F(\boldsymbol{g}_{t+1})\right)-\mathbb{E}\left(F(\boldsymbol{g}_t)\right)\\ \leq&\frac{1}{2L_\gamma}\left(-1+\frac{4(N-A)^2\mathbb{E}\left(\|\Delta(\alpha_t)\|+1\right)\zeta_2}{N^2}\right)||\nabla F(\boldsymbol{g}_t)||^2\\ &+\frac{\mathbb{E}\left\|\Delta(\alpha_t)\right\|+1}{2L_\gamma}\left(\frac{4(N-A)^2(\zeta_1+B\epsilon^2)}{N^2}+{L_\gamma}^2\mathbb{E}\left\|\Delta(\alpha_t)\right\|\right)\\ &+\frac{M^2\Upsilon^4}{2L_\gamma}+\mathbb{E}\left(\Delta(\alpha_t)^2\right),\end{aligned}\tag{31}$$

where $A=\sum_{m=1}^{M}u_{m,t}N_{m,t}$ represents the sum of all selected devices' data samples, $\Delta(\alpha_t)=\hat{\boldsymbol{g}}_t(\alpha_t)-\boldsymbol{g}_t$ is the quantization

Algorithm 1 Model-based RL for device selection and quantization optimization

Input: The environment state $\mathcal{S}$, the action space $\mathcal{A}$.
Output: The device selection and quantization scheme.
1: Initialize policy $\boldsymbol{\pi_\theta}$, transition replay buffer $\mathcal{B}$, trajectory replay buffer $\boldsymbol{\tau}$.
2: **for** iteration $i = 1 : I$ **do**
3: Randomly selects a subset of devices to generate the global FL model that are quantized into α_t bits.
4: Records $F(\boldsymbol{g}_t)$, $F(\boldsymbol{g}_{t+1})$, α_t, and device selection scheme $\boldsymbol{u}_t$ in $\mathcal{B}$.
5: **end for**
6: Estimate $1/L$, ζ_1, ζ_2, and $B\epsilon^2$ to construct $P\left(s_{t+1}|s_t, \boldsymbol{a}_t\right)$ using (23) based on the real transition in $\mathcal{B}$.
7: **for** iteration $i = 1 : H$ **do**
8: Sample initial state from $\mathcal{S}$, then use policy $\boldsymbol{\pi_\theta}$ and learned $P\left(s_{t+1}|s_t, \boldsymbol{a}_t\right)$ to perform T trajectories and update $\boldsymbol{\tau}$.
9: Sample from $\boldsymbol{\tau}$, and update the current policy evaluation by solving Equation (28).
10: **end for**

error of the global FL model, and $\mathbb{E}\left\|\Delta(\alpha_t)\right\| = M2^{-\alpha_t}$ is the unbiased quantization function.

Proof: The detailed proof can be found in [53]. ■

Given Theorem 2, the server is able to obtain the state transition probability $P(s_{t+1}|s_t, a_t)$ via estimating L_γ, Υ, ζ_1, ζ_2, $B\epsilon^2$. Given $P(s_{t+1}|s_t, a_t)$, we can use the proposed model based RL to find the optimal device selection and quantization scheme.

E. Implementation and Complexity

Next, we first analyze the training process of the model based RL algorithm. To train the proposed model based RL, the server needs to collect the time consumption coefficient ρ, the frequency of the CPU f, the number of bits that can be processed by the CPU in one clock cycle ϑ, the channel gain $h_{m,t}$, and the transmit power of each device P. These parameters are constant and can be obtained from devices. Meanwhile, the server already knows FL model meta-parameters such as the number of multiplication operations N^{C} and the number of neurons D, when it initializes the FL model. Additionally, during the initial I FL training iterations, the server must first randomly select a subset of devices to participate in FL, obtaining their training loss values and the selected actions so as to estimate the values of $1/L$, ζ_1, ζ_2, and $B\epsilon^2$, and calculate $P\left(s_{t+1}|s_t, \boldsymbol{a}_t\right)$, as shown in (22)-(28). Then, using $P\left(s_{t+1}|s_t, \boldsymbol{a}_t\right)$, a model based RL algorithm is used to find the optimal α_t and $\boldsymbol{u}_t$ without any interactions between the server and devices. The entire process of training the proposed model based RL algorithm is shown in Algorithm 1.

The computational complexity of the proposed algorithm lies in the calculation of the state transition probability $P\left(s_{t+1}|s_t, \boldsymbol{a}_t\right)$ by a standard gradient descent method as well as optimizing α_t and $u_{m,t}$ by the proposed model based RL algorithm, which is detailed as follows:

a) In terms of computational complexity of calculating $P\left(s_{t+1}|s_t, \boldsymbol{a}_t\right)$, the server needs to find the values of $1/L$, ζ_1, ζ_2, and $B\epsilon^2$ using a linear regression method. The regression loss function in (22) is strongly convex and smooth. Hence, the fixed step-size gradient descent method at least updates $\mathcal{O}\Big(\frac{||L^0-L^*||_2^2}{\varepsilon\iota_L}+\frac{||{\zeta_1}^0-{\zeta_1}^*||_2^2}{\varepsilon\iota_{\zeta_1}}+\frac{||{\zeta_2}^0-{\zeta_2}^*||_2^2}{\varepsilon\iota_{\zeta_2}}+\frac{||B'^0-B'^*||_2^2}{\varepsilon\iota_{B'}}\Big)$ iterations for reaching the optimal L^*, ${\zeta_1}^*$, ${\zeta_2}^*$, B'^* from initialized L^0, ${\zeta_1}^0$, ${\zeta_2}^0$, B'^0 with ε error [54].

b) The computational complexity of the proposed RL algorithm depends on the number of the parameters in the policy network $\boldsymbol{\theta}$ which depends on the size of action space $\mathcal{A}$ and the size of state space $\mathcal{S}$. In particular, the possible combinations of each $\boldsymbol{a}_t$ in action space $\mathcal{A}$ is to choose i devices ($i \leq U$) from all M devices to participate in quantized FL. Thus, the size of the possible device selections is $\sum_{i=1}^{U} C_M^i = \sum_{i=1}^{U} \frac{M!}{i!(M-i)!}$ and the number of quantization actions is Y. The state space $\mathcal{S}$ consists of the continuous values of loss function $F(\boldsymbol{g}_t)$. To ensure a finite state space, the continuous loss function values is divided into $|\mathcal{S}|$ levels. Then, the size of state space $\mathcal{S}$ is $|\mathcal{S}|$. Therefore, the computational complexity of the proposed RL algorithm is $\mathcal{O}\left\{Y|\mathcal{S}|\sum_{i=1}^{U}\frac{M!}{i!(M-i)!}\prod_{k=2}^{K-1}H_k\right\}$, where H_k is the number of the neurons in layer k of the policy network $\boldsymbol{\theta}$.

TABLE I
SIMULATION PARAMETERS

Parameters	Values	Parameters	Values	Parameters	Values
M	15	U	6	N_m	200
W	15 kHz	P	0.5 W	σ_N^2	-174 dBm
K	8	I	20	Γ	1 s
T	1000	f	3.3 GHz	B	64
D	217728	ρ	2.8×10^6	ι	0.02
ι_L	0.02	ι_{ζ_1}	0.02	ι_{ζ_2}	0.02

IV. NUMERICAL EVALUATION

For our simulations, we consider a circular network area having a radius $r = 1500$ m with one server at its center serving $M = 15$ uniformly distributed devices. The other parameters used in simulations are listed in Table I, unless otherwise stated. For comparison purposes, we use three baselines:

- a) The binary FL scheme from [55] that enables the server to randomly select a subset of devices to cooperatively train the FL model at each iteration. Each parameter in the trained FL model is quantized into one bit.
- b) An FL algorithm that enables the server to randomly select a subset of devices to cooperatively train the FL model in full-precision (i.e., without quantization), which can be seen as a standard FL [11].
- c) An FL algorithm that optimizes the device selection and quantization schemes using a model free RL method [32]. For c), a policy gradient-based RL update is employed to learn the state transition probabilities.

A. Datasets and ML Models

We consider two popular ML tasks: handwritten digit identification on the MNIST dataset [56], and image classification on the CIFAR-10 dataset [57]. The quantized FL algorithm that is used for handwritten digit identification consists of three full-connection layers. The total number of model parameters in the used fully-connected neural network (FNN) is 217728 ($= 28\times28\times256+256\times64+64\times10$). To verify the feasibility

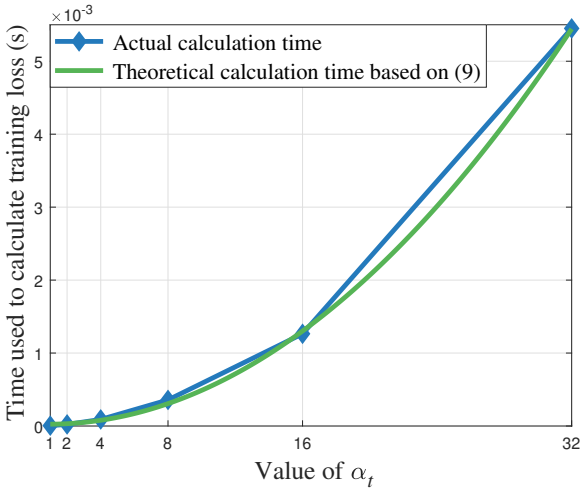

Fig. 2. Calculation time of low bitwidth federated learning vs. the quantization precision.

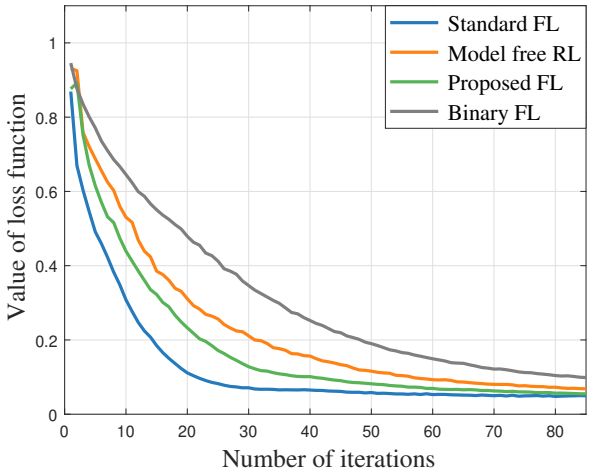

Fig. 3. Training loss vs. the number of iterations.

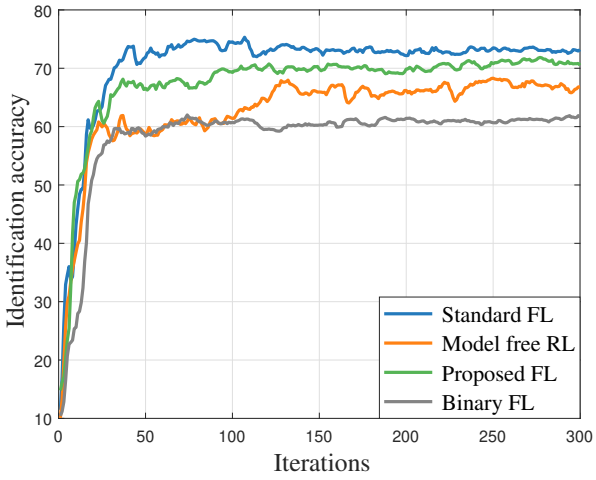

Fig. 4. Identification accuracy vs. number of iterations.

of the proposed calculation time model in (9), we first simulate the actual calculation time using the clock module in GEneral Matrix Multiply (GEMM) [58], as shown in Fig. 2. Fig. 2 shows that the actual calculation time is almost the same as the theoretical calculation time in (9).

The quantized FL algorithm that is used for image classification consists of three convolutional layers and two full-connection layers. In the used convolutional neural network (CNN), the size of the convolutional kernel is 5×5 and the total number of model parameters in CNN is 116704 $(= 5 \times 5 \times (3 \times 32 + 32 \times 32 + 32 \times 64) + 576 \times 64 \times 64 + 10)$.

For both datasets, we will consider two cases of data distributions across clients: (i) non-i.i.d., where each client is allocated samples from only 3 of 10 labels; and (ii) i.i.d., where each client is allocated samples from all labels. All FL algorithms are considered to be converged when the value of the FL loss variance calculated over 20 consecutive iterations is less than 0.001.

B. Convergence Performance Analysis

Fig. 3 shows how the FL training loss changes as the number of iterations varies for MNIST. From Fig. 3, we can see that, the proposed model based RL algorithm can reduce the number of iterations needed to converge by 14% and 24% compared to the model free RL method and the binary FL method, respectively. This is due to the fact that the proposed method enables the server to estimate the FL training parameters in the first few iterations so as to model the FL training process mathematically thus reducing the number of iterations required to converge.

Fig. 4 gives the accuracy plot corresponding to Fig. 3. From this figure, we can see once again that, the proposed algorithm obtains a noticeable improvement in convergence speed compared with model free RL in the non-i.i.d. case. This implies that our proposed algorithm models the FL training process effectively in the non-i.i.d. case via estimating the key meta-parameters that lead to speeding up the convergence. Fig. 4 also shows our algorithm comes within 2% of the accuracy obtained by standard FL at convergence due to quantization errors. In contrast, by optimizing the device selection and quantization precision, our methodology reduces the necessary bitwidth by 68% and the number of devices participating in each round by 30%.

Fig. 5 shows how the FL training loss changes as the number of iterations varies on the CIFAR-10 dataset. We can see that the value of the loss function decreases as the number of iterations increases, and as the quantization bitwidth increases, the value of the loss function decreases. This is because as the quantization bitwidth increases, the introduced error resulting from the quantization decreases, which enables the trained FL model achieves a better performance in terms of training loss.

Fig. 6 shows how the identification accuracy of all considered algorithms changes as the number of iterations varies on the CIFAR-10 dataset in the non-i.i.d. case. We see that our proposed methodology can achieve up to 22% improvement in terms of the number of iterations required to converge compared to the model free RL method. Similar to the previous figures, this demonstrates the advantage of the server estimating the associated FL model parameters based on information captured during the training process, thus optimizing the FL training process through minimal interaction with each device. Fig. 6 also shows that the binary FL algorithm (i.e.,

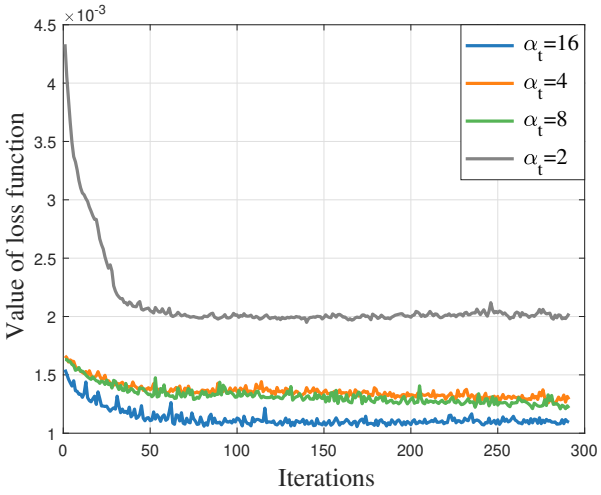

Fig. 5. Training loss vs. number of iterations.

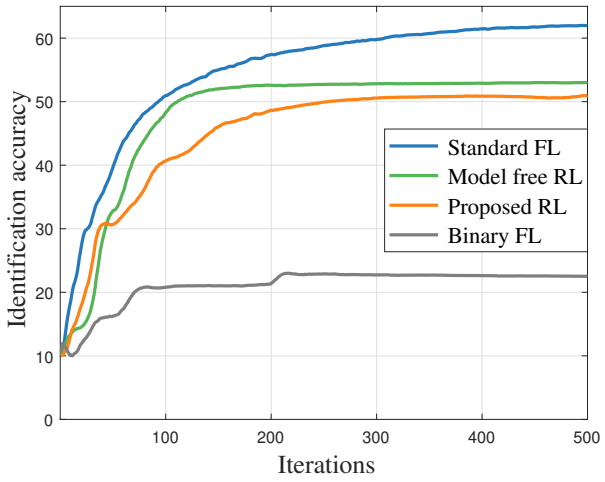

Fig. 6. Identification accuracy vs. number of iterations.

when the weights of CNN are binary) can only achieve 21% identification accuracy. This is due to the fact that the binary FL neither has a pre-training process [59] nor uses full-precision scale factors [60] to recover the full-precision model, again emphasizing the benefit of optimizing the quantization precision.

C. Training Accuracy and Latency Comparison

Fig. 7 shows one example of implementing the proposed FL algorithm for 40 handwritten digit identification. From this figure, we see that, as the delay requirement Γ for completing each FL training iteration increases, the average quantization bits α and the identification accuracy increase. This is because as Γ increases, the time that can be used for training and transmitting local FL parameters in the selected devices increases thus resulting in an improvement of α and identification accuracy. From Fig. 7, we can see that, for 40 handwritten digit identification, the proposed algorithm correctly identifies 35 handwritten digits. In contrast, the model free RL identifies 34 handwritten digits and the binary FL correctly identifies 33 handwritten digits. This is because the proposed FL algorithm can mathematically model the FL training process by obtaining the transition probability so as to find out the optimal device selection and quantization scheme for achieving a higher identification accuracy.

Fig. 8 shows how the identification accuracy changes as the number of devices varies on the MNIST dataset, in the non-i.i.d. case. From Fig. 8, we see that the proposed model based RL method can improve the identification accuracy by up to 6% compared with binary FL when the number of devices is small, while it is closer to 3% when the number of devices grows larger. Our method is able to obtain a stronger robustness against the number of devices than model-free RL and binary FL through analyzing the relationship among the FL training loss at different iterations, thus finding a better FL training policy especially when less information is available. Fig. 8 also shows that the performance of model-free RL almost converges to our method as the number of devices increases (in this case, once it reaches 9), This is because a smaller number of devices results in a smaller number of data samples available throughout the system for training the RL policy and FL model. One of the advantages of our model-based methodology is that it enables the server to model the FL training process using limited device-to-server interactions, which translates to lower data requirements. As the number of devices continues to increase, the considered network will have sufficient data samples for policy learning even with a model-free RL approach. Thus, the identification accuracy of the proposed method and that of the model-free RL become the same for larger M.

Fig. 9 gives the training loss corresponding to the testing accuracy plot in Fig. 8. The result is consistent with what we observe in Fig. 8: as the number of devices increases, all considered algorithms have more data available for training, which gives a lower training loss. On the other hand, when there are fewer devices ($M < 9$), the training loss increases more quickly.

Fig. 10 shows how the identification accuracy of the proposed FL framework changes as the delay requirement varies. This figure is simulated using CIFAR-10 dataset. From Fig. 10, we see that, as the delay requirement increases, the identification accuracy of all considered learning algorithms increases. This is because that as the delay requirement increases, all considered learning algorithms enables the selected devices to fully utilize the training and transmitting time to perform FL framework, which results in an increase of average quantization bits and achievable accuracy. Fig. 10 also shows that as the average quantization bits α decreases, the number of iterations required to reach a fixed achievable accuracy increases slightly. This is due to the fact that as α decreases, the quantization error increases, which decreases the accuracy for modeling FL tranining process. However, with a decrease of α, the time used to perform FL training at each iteration decreases and thus, the total time used to reach a fixed achievable accuracy decreases rapidly, which implies that the time used for training the proposed quantized FL framework decreases.

In Fig. 11, we show how the identification accuracy of the proposed FL algorithm changes as the number of iterations varies. This figure is simulated using CIFAR-10 dataset. From Fig. 11, we see that, as the number of iterations increases, the identification accuracy of the proposed algorithms with different α first increases and, then remains unchanged. This

Delay requirement	Schemes	Average quantization bits										
Γ=0.1 s	Proposed FL	$\alpha=1.8$	8	5	3	9	8	6	0	7	4	7
	Model free RL	$\alpha=1.5$	8	5	3	9	6	6	2	1	4	7
	Binary FL	$\alpha=1$	8	5	6	9	8	6	2	1	4	9
Γ=0.3 s	Proposed FL	$\alpha=4.2$	8	5	3	7	8	6	2	1	4	1
	Model free RL	$\alpha=3.8$	8	5	6	9	8	6	0	1	4	1
	Binary FL	$\alpha=1$	8	5	6	9	8	6	2	1	9	7
Γ=0.5 s	Proposed FL	$\alpha=8.4$	8	5	3	9	8	6	0	1	4	7
	Model free RL	$\alpha=8.1$	8	5	3	9	8	9	2	1	4	7
	Binary FL	$\alpha=1$	8	5	3	9	6	6	2	1	4	7
Γ=1.0 s	Proposed FL	$\alpha=16.4$	8	5	3	9	8	6	2	1	4	7
	Model free RL	$\alpha=16.1$	8	5	3	9	8	6	2	1	4	1
	Binary FL	$\alpha=1$	8	5	3	9	9	6	0	1	4	7

Fig. 7. An example of implementing quantized FL for handwritten digit identification.

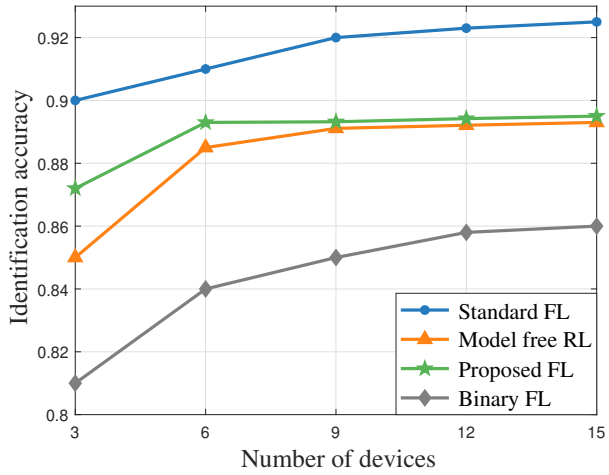

Fig. 8. Identification accuracy vs. number of devices.

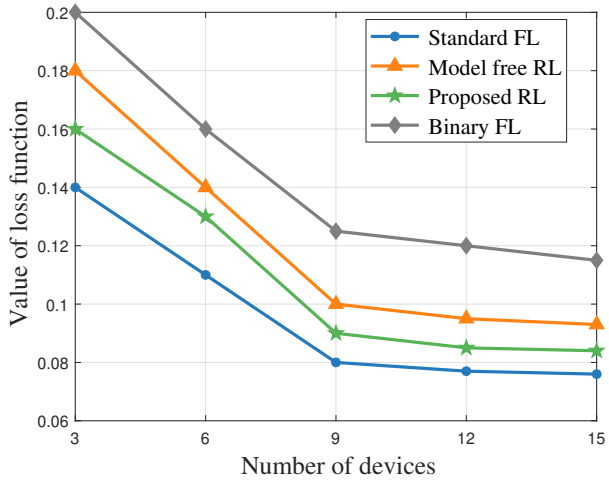

Fig. 9. Training loss vs. number of devices.

is because FL algorithms converge. From Fig. 11, we can also see that as α increases, the identification accuracy of the proposed algorithm increases. This is due to the fact that as α increases, the quantization error decreases. Fig. 11 also shows that as α decreases, the instability of the proposed algorithm increases. This is because the quantization error in the weights of the convolution kernel can significantly affect the result of convolution operation, thus a decrease of α will result in a degeneration of identification accuracy in CNN.

Fig. 12 shows how the identification accuracy changes as the convergence time varies. This figure is simulated using CIFAR-10 dataset. In this figure, we can see that, the proposed FL reduces the convergence time by up to 29% and 63% compared to model free RL method and the standard FL method, respectively. The 29% gain stems from the fact that the proposed model based RL approach can mathematically derive the transition probability across different actions and states thus speeding up the FL training process and achieving a better identification accuracy. The 63% gain stems from the fact that the proposed model based RL approach enables the devices to quantize its local FL model with optimal bitwidth.

Fig. 13 shows an example of the optimized device selection and quantization scheme. This simulation employs the MNIST dataset, with i.i.d. data distribution across devices. In this figure, we can see that, as the distance between the server and each device m increases, the probability of quantizing the FL model on device m with a high-precision bitwidth decreases. This is consistent with the fact that as the distance between the server and device m increases, the time used to transmit the FL model increases under a fixed transmit power. Hence, when the data distribution across devices is i.i.d., the server will tend to have further away devices quantize their local models in lower precision so as to reduce the time used for model exchanging and training to satisfy the delay constraint Γ. Similarly, we can also see that, as the distance decreases, the probability for choosing device m to participate in FL training increases. Closer devices tend to transmit higher precision FL models (when the data distributions are i.i.d.), which enables the server and the devices to obtain a FL model with a better performance of identification accuracy.

V. Conclusion

In this article, we developed a novel quantized FL framework in which distributed wireless devices train and transmit their locally trained FL models to a coordinating server based on variable bitwidths. We formulated an optimization problem that jointly considers the device selection and quantization scheme to minimize FL training loss while accounting for communication and computation heterogeneity across the devices. To solve this problem, we first analytically derived the

Delay requirement	Schemes	Average quantization bits	Identification accuracy 30%	40%	45%	50%	55%	60%
Γ=0.1 s	Proposed FL	α=1.8	350	*	*	*	*	*
	Model free RL	α=1.5	380	*	*	*	*	*
Γ=0.3 s	Proposed FL	α=4.5	27	76	160	280	*	*
	Model free RL	α=3.8	27	80	180	320	*	*
Γ=0.5 s	Proposed FL	α=8.4	26	45	80	220	*	*
	Model free RL	α=8.1	28	45	90	250	*	*
Γ=1.0 s	Proposed FL	α=16.4	25	43	60	100	300	750
	Model free RL	α=16.1	28	45	68	125	380	760

Fig. 10. An example of implementing quantized FL for CIFAR-10 identification.

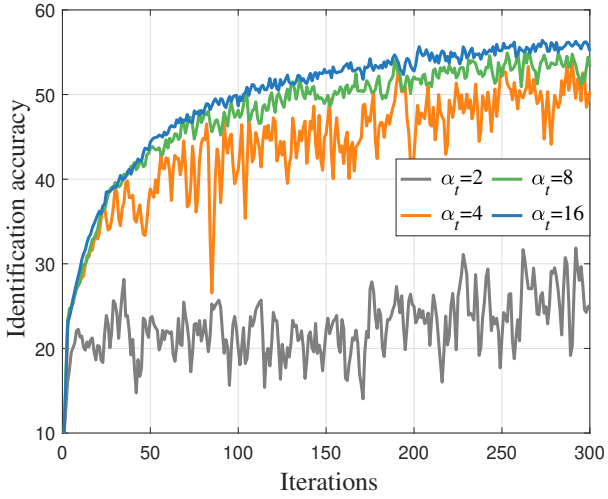

Fig. 11. Identification accuracy vs. number of iterations.

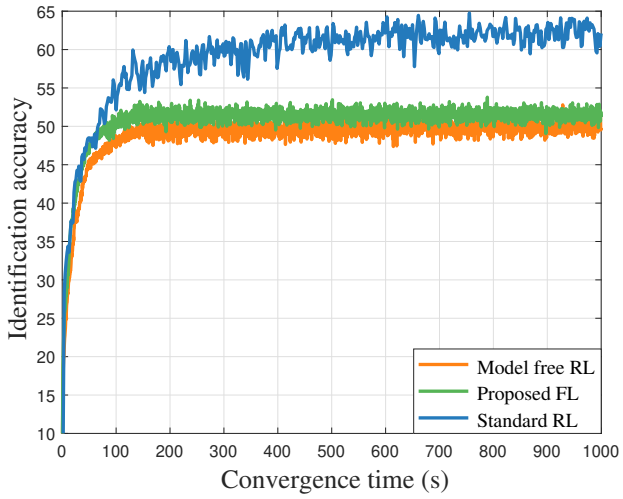

Fig. 12. Identification accuracy vs. convergence time.

expected training convergence rate of our quantized FL framework. Our analysis showed how the expected improvement of FL training loss between two adjacent iterations depends on the device selection scheme, the quantization scheme, and inherent properties of the model being trained. To find the tightest bound, we introduced a linear regression method for estimating these model properties according to observable training information at the server. Given these estimates, the improvement of FL performance at adjacent iterations was described as an MDP. We then proposed a model-based RL

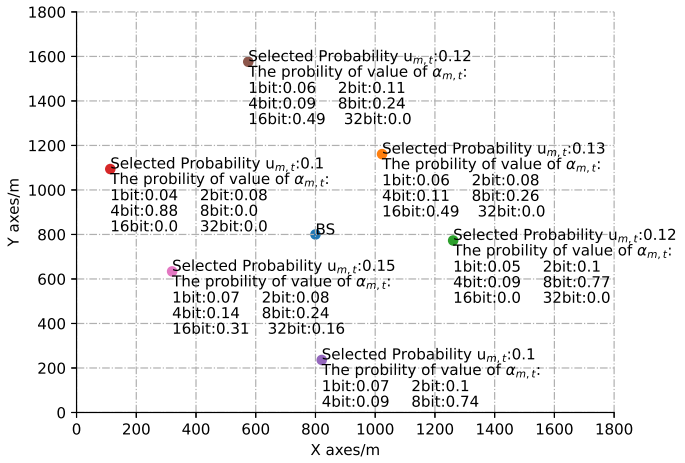

Fig. 13. Example of the optimized quantization and device selection scheme in the proposed FL framework.

method to learn the relationship between FL performance and the choice of device selection and quantization scheme so as to converge on the policy minimizing FL loss. Numerical evaluation on real-world machine learning tasks demonstrated that the proposed methodology yields significant gains in classification accuracy and convergence speed compared to conventional approaches.

VI. Appendix

A. Proof of Lemma 1

To prove Lemma 1, we first rewrite $F\left(\boldsymbol{g}_{t+1}\left(\boldsymbol{u}_{t+1}\right),\alpha_{t+1}\right)$ using the second-order Taylor expansion and the L-smoothness of property in Assumption 1, which can be expressed as

$$\begin{aligned} &F\left(\boldsymbol{g}_{t+1}\right)\\ \leq &F\left(\hat{\boldsymbol{g}}_{t+1}\right)+\left(\boldsymbol{g}_{t+1}-\hat{\boldsymbol{g}}_{t+1}\right)\nabla\widetilde{F}\left(\hat{\boldsymbol{g}}_{t+1}\right)+\frac{L}{2}\left\|\boldsymbol{g}_{t+1}-\hat{\boldsymbol{g}}_{t+1}\right\|^{2}, \end{aligned} \tag{32}$$

where $\boldsymbol{g}_t$ and $\hat{\boldsymbol{g}}_t$ are short for $\boldsymbol{g}_t\left(\boldsymbol{u}_t,\alpha_t\right)$ and $\hat{\boldsymbol{g}}_t\left(\boldsymbol{u}_t,\alpha_t\right)$, respectively.

Taking expectations of both sides of (32), we have

$$\begin{aligned} \mathbb{E}\left(F\left(\boldsymbol{g}_{t+1}\right)\right) \leq &\mathbb{E}\left(F\left(\hat{\boldsymbol{g}}_{t+1}\right)+\left(\boldsymbol{g}_{t+1}-\hat{\boldsymbol{g}}_{t+1}\right)\nabla\widetilde{F}\left(\hat{\boldsymbol{g}}_{t+1}\right)\right.\\ &\left.+\frac{L}{2}\left\|\boldsymbol{g}_{t+1}-\hat{\boldsymbol{g}}_{t+1}\right\|^{2}\right)\\ \overset{(a)}{=}&\mathbb{E}\left(F\left(\hat{\boldsymbol{g}}_{t+1}\right)\right)+\frac{L}{2}\mathbb{E}\left(\left\|\boldsymbol{g}_{t+1}-\hat{\boldsymbol{g}}_{t+1}\right\|^{2}\right), \end{aligned} \tag{33}$$

where (a) stems from the fact that the unbiased quantization function $Q\left(\boldsymbol{g}_{t+1}, \boldsymbol{\alpha}\right)$ satisfies $\mathbb{E}[\hat{\boldsymbol{g}}_{t+1}] = \boldsymbol{g}_{t+1}$. Similarly, we rewrite $F\left(\hat{\boldsymbol{g}}_{t+1}\right)$ as

$$F\left(\hat{\boldsymbol{g}}_{t+1}\right) \leq F\left(\boldsymbol{g}_t\right) + \left(\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t\right) \nabla \widetilde{F}\left(\boldsymbol{g}_t\right) + \frac{L}{2}\left\|\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t\right\|^2. \tag{34}$$

Taking the expectation over (34) and combining it with (33), we have

$$\begin{aligned}\mathbb{E}\left(F\left(\boldsymbol{g}_{t+1}\right)\right) - \mathbb{E}\left(F\left(\boldsymbol{g}_t\right)\right) \leq{}& \mathbb{E}\left(\left(\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t\right) \nabla \widetilde{F}\left(\boldsymbol{g}_t\right)\right) \\ &+ \frac{L}{2}\mathbb{E}\left(\left\|\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t\right\|^2\right) + \frac{L}{2}\mathbb{E}\left(\left\|\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_{t+1}\right\|^2\right).\end{aligned} \tag{35}$$

Then, we substitute (19) into (35) and have

$$\begin{aligned}\mathbb{E}\left(F(\boldsymbol{g}_{t+1})\right) - \mathbb{E}\left(F(\boldsymbol{g}_t)\right) \leq{}& \mathbb{E}\left(\left(\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t\right)\left(\nabla F(\boldsymbol{g}_t) - \epsilon\right)\right) \\ &+ \frac{L}{2}\mathbb{E}\left(\left\|\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t\right\|^2\right) + \frac{L}{2}\mathbb{E}\left(\left\|\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_{t+1}\right\|^2\right).\end{aligned} \tag{36}$$

This completes the proof.

B. Proof of Theorem 1

To prove Theorem 1, we first investigate the gap between the expectation of the quantized model $\hat{\boldsymbol{g}}_{t+1}$ and the expectation of the full-precision model $\boldsymbol{g}_t$ in Lemma 1, which is given by

$$\begin{aligned}\mathbb{E}\left(\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t\right) &= \mathbb{E}\left(\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_{t+1} + \boldsymbol{g}_{t+1} - \boldsymbol{g}_t\right) \\ &= \mathbb{E}\left(\Delta\left(\alpha_t\right) - \lambda\left(\nabla F\left(\boldsymbol{g}_t\right) - \boldsymbol{e}_t\right)\right),\end{aligned} \tag{37}$$

where $\Delta\left(\alpha_t\right) = \left\|\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_{t+1}\right\|$ is the quantization error at iteration $t+1$ and $\boldsymbol{e}_t$ is a gradient deviation caused by the quantization errors of the local FL models that are transmitted by selected devices and the devices that do not transmit their local FL models to the server at iteration t. In particular, $\boldsymbol{e}_t$ can be expressed as

$$\boldsymbol{e}_t = \nabla F\left(\boldsymbol{g}_t\right) - \frac{\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_{m,t}} u_{m,t}\left(\nabla f\left(\hat{\boldsymbol{g}}_t, \boldsymbol{x}_{mn}, \boldsymbol{y}_{mn}\right) + \epsilon_m\right)}{N\sum_{m=1}^{M} N_{m,t}u_{m,t}}. \tag{38}$$

Substituting (37) into Lemma 1, we have

$$\begin{aligned}&\mathbb{E}\left(F\left(\boldsymbol{g}_{t+1}\right)\right) - \mathbb{E}\left(F\left(\boldsymbol{g}_t\right)\right) \\ \leq{}& \mathbb{E}\left(\left(\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t\right)\left(\nabla F\left(\boldsymbol{g}_t\right) - \epsilon\right)\right) + \frac{L}{2}\mathbb{E}\left(\left\|\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_t\right\|^2\right) \\ &+ \frac{L}{2}\mathbb{E}\left(\left\|\hat{\boldsymbol{g}}_{t+1} - \boldsymbol{g}_{t+1}\right\|^2\right) \\ ={}& \mathbb{E}\left(\left(\Delta\left(\alpha_t\right) - \lambda\left(\nabla F\left(\boldsymbol{g}_t\right) - \boldsymbol{e}_t\right)\right)\left(\nabla F\left(\boldsymbol{g}_t\right) - \epsilon\right)\right) \\ &+ \frac{L}{2}\mathbb{E}\left(\left\|\Delta\left(\alpha_t\right) - \lambda\left(\nabla F\left(\boldsymbol{g}_t\right) - \boldsymbol{e}_t\right)\right\|^2\right) + \frac{L}{2}\mathbb{E}\left(\left\|\Delta\left(\alpha_t\right)\right\|^2\right) \\ ={}& -\frac{1}{2L}\left\|\nabla F\left(\boldsymbol{g}_t\right)\right\|^2 + \frac{1}{2L}\mathbb{E}\left(\left\|\boldsymbol{e}_t + L\Delta\left(\alpha_t\right)\right\|^2\right) + \frac{L}{2}\mathbb{E}\left(\Delta\left(\alpha_t\right)^2\right) \\ \overset{(a)}{=}{}& -\frac{1}{2L}\left\|\nabla F\left(\boldsymbol{g}_t\right)\right\|^2 + \frac{1}{2L}\mathbb{E}\left(\left\|\boldsymbol{e}_t\right\|^2\right) + \mathbb{E}\left(\Delta\left(\alpha_t\right)^2\right),\end{aligned} \tag{39}$$

where (a) stems from the fact that $\mathbb{E}\left(\hat{\boldsymbol{g}}_{t+1}\right) = \boldsymbol{g}_{t+1}$. Next, we derive $\mathbb{E}\left(\left\|\boldsymbol{e}_t\right\|^2\right)$, which can be given as follows

$$\begin{aligned}&\mathbb{E}\left(\left\|\boldsymbol{e}_t\right\|^2\right) \\ ={}& \mathbb{E}\left(\left\|\nabla F\left(\boldsymbol{g}_t\right) - \frac{\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_{m,t}} u_{m,t}\nabla f\left(\hat{\boldsymbol{g}}_t\right)}{N\sum_{m=1}^{M} N_{m,t}u_{m,t}}\right\|^2\right) \\ ={}& \mathbb{E}\left(\left\|\nabla F\left(\boldsymbol{g}_t\right) - \frac{\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_{m,t}} u_{m,t}\left(\nabla f\left(\boldsymbol{g}_t + \Delta\left(\alpha_t\right)\right) + \epsilon_m\right)}{\sum_{m=1}^{M} N_{m,t}u_{m,t}}\right\|^2\right) \\ \overset{(a)}{\leq}{}& \mathbb{E}\left(\left\|\nabla F\left(\boldsymbol{g}_t\right) - \frac{\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_{m,t}} u_m\nabla\left[f\left(\boldsymbol{g}_t\right) + \Delta\left(\alpha_t\right)\nabla f\left(\boldsymbol{g}_t\right) + \epsilon_m + \frac{1}{2}||\boldsymbol{o}||^2\right]}{\sum_{m=1}^{M} N_{m,t}u_{m,t}}\right\|^2\right) \\ \overset{(b)}{\leq}{}& \mathbb{E}\left(\left\|\nabla F\left(\boldsymbol{g}_t\right) - \frac{\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_{m,t}} u_{m,t}\left(\nabla f\left(\boldsymbol{g}_t\right) + \epsilon_m + \Delta\left(\alpha_t\right)L\right)}{\sum_{m=1}^{M} N_{m,t}u_{m,t}}\right\|^2\right)\end{aligned} \tag{40}$$

where $f\left(\boldsymbol{g}_t\right)$ is short for $f\left(\boldsymbol{g}_t, \boldsymbol{x}_{mn}, \boldsymbol{y}_{mn}\right)$, (a) stems from the second-order Taylor expansion of $f\left(\boldsymbol{g}_t + \Delta\left(\alpha_t\right), \boldsymbol{x}_{mn}, \boldsymbol{y}_{mn}\right)$, (b) stems from the twice-continuously differentiable of $f\left(\boldsymbol{g}_t, \boldsymbol{x}_{mn}, \boldsymbol{y}_{mn}\right)$ with $\mu\boldsymbol{I} \preceq \nabla^2 f\left(\boldsymbol{g}_t, \boldsymbol{x}_{mn}, \boldsymbol{y}_{mn}\right) \preceq L\boldsymbol{I}$. The inequality equation in (40) is achieved by the triangle-inequality (i.e., $\left\|\nabla f(\boldsymbol{g}_t, \boldsymbol{x}_{mn}, \boldsymbol{y}_{mn})\right\| \leq \sqrt{\zeta_1 + \zeta_2\left\|\nabla F(\boldsymbol{g}_t)\right\|^2}$). Substituting the triangle-inequality and $\sum_{m=1}^{M} N_{m,t}u_{m,t} = A$ into (40), we have

$$\begin{aligned}&\mathbb{E}\left(||\boldsymbol{e}_t||^2\right) \\ ={}& \mathbb{E}\left(\left\|\frac{(N-A)\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_{m,t}}\nabla f(\boldsymbol{g}_t) + \epsilon_m}{NA} - \frac{1}{A}\sum_{m=1}^{M} u_{m,t}L\Delta(\alpha_t)\right. \right. \\ &\left.\left. + \frac{1}{N}\sum_{m=1}^{M} N_m(1-u_{m,t})\sum_{n\in\mathcal{N}_{m,t}}\nabla f(\boldsymbol{g}_t) + \epsilon_m\right\|\right)^2 \\ \leq{}& \mathbb{E}\left(\frac{1}{NA}(N-A)\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_{m,t}}\left\|\nabla f(\boldsymbol{g}_t) + \epsilon_m\right\|\right. \\ &+ \frac{1}{A}\sum_{m=1}^{M} u_{m,t}L\left\|\Delta(\alpha_t)\right\| \\ &\left. + \frac{1}{N}\sum_{m=1}^{M} N_m(1-u_{m,t})\sum_{n\in\mathcal{N}_{m,t}}\left\|\nabla f(\boldsymbol{g}_t) + \epsilon_m\right\|\right)^2\end{aligned}$$

$$
\begin{aligned}
&\overset{(a)}{\leq} \mathbb{E}\left(\frac{(N-A)\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_{m,t}}\|\nabla f(\boldsymbol{g}_t)+\epsilon_m\|}{N}+L\|\Delta(\alpha_t)\|\right.\\
&\qquad\left.+\frac{(N-A)\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_{m,t}}\|\nabla f(\boldsymbol{g}_t)+\epsilon_m\|}{N}\right)^2\\
&=\mathbb{E}\left(\frac{2(N-A)\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_{m,t}}\|\nabla f(\boldsymbol{g}_t)+\epsilon_m\|}{N}+L\|\Delta(\alpha_t)\|\right)^2\\
&\overset{(b)}{\leq}(\mathbb{E}\|\Delta(\alpha_t)\|+1)\left(\frac{4(N-A)^2\left(\zeta_1+\zeta_2\|\nabla F(\boldsymbol{g}_t)\|^2+B\epsilon^2\right)}{N^2}\right.\\
&\qquad\left.+L^2\mathbb{E}\|\Delta(\alpha_t)\|\right)
\end{aligned}
\tag{41}
$$

where (a) stems from the fact that $U\leq A$ and (b) stems from the inequality $2\boldsymbol{a}\boldsymbol{b}\leq\|\Delta(\alpha_t)\|\boldsymbol{a}^2+\|\Delta(\alpha_t)\|^{-1}\boldsymbol{b}^2$ for any two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ with scalar $\|\Delta(\alpha_t)\|\geq 0$. Substituting (41) into (39), we have

$$
\begin{aligned}
&\mathbb{E}\left(F(\boldsymbol{g}_{t+1})\right)-\mathbb{E}\left(F\left(\boldsymbol{g}_t\right)\right)\\
\leq&-\frac{1}{2L}||\nabla F\left(\boldsymbol{g}_t\right)||^2+\frac{1}{2L}\mathbb{E}\left(||\boldsymbol{e}_t||^2\right)+\mathbb{E}\left(\Delta\left(\alpha_t\right)^2\right)\\
\leq&-\frac{1}{2L}||\nabla F(\boldsymbol{g}_t)||^2+\mathbb{E}\left(\Delta\left(\alpha_t\right)^2\right)\\
&+\frac{1}{2L}\mathbb{E}\left(\|\Delta(\alpha_t)\|+1\right)\left(\frac{4(N-A)^2\left(\zeta_1+\zeta_2\|\nabla F(\boldsymbol{g}_t)\|^2+B\epsilon^2\right)}{N^2}\right.\\
&\left.+L^2\mathbb{E}\|\Delta\left(\alpha_t\right)\|\right)\\
=&\frac{1}{2L}\left(-1+\frac{4(N-A)^2\left(\mathbb{E}\|\Delta\left(\alpha_t\right)\|+1\right)\zeta_2}{N^2}\right)||\nabla F(\boldsymbol{g}_t)||^2\\
&+\frac{\mathbb{E}\|\Delta\left(\alpha_t\right)\|+1}{2L}\left(\frac{4(N-A)^2\left(\zeta_1+B\epsilon^2\right)}{N^2}+L^2\mathbb{E}\|\Delta\left(\alpha_t\right)\|\right)\\
&+\mathbb{E}\left(\Delta\left(\alpha_t\right)^2\right).
\end{aligned}
\tag{42}
$$

This completes the proof.

REFERENCES

[1] N. Yang, S. Wang, M. Chen, C. G. Brinton, C. Yin, W. Saad, and S. Cui, "Model-based reinforcement learning for quantized federated learning performance optimization," in *Proc. of IEEE Global Communications*, Rio de Janeiro, Brazil, Dec. 2022.

[2] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3579–3605, Dec. 2021.

[3] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar, "Over-the-air federated learning from heterogeneous data," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3796–3811, June 2021.

[4] C. Ma, J. Li, M. Ding, K. Wei, W. Chen, and H. V. Poor, "Federated learning with unreliable clients: Performance analysis and mechanism design," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17308–17319, Dec. 2021.

[5] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, "Distributed multi-agent meta learning for trajectory design in wireless drone networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3177–3192, June 2021.

[6] S. Hu, X. Chen, W. Ni, E. Hossain, and X. Wang, "Distributed machine learning for wireless communication networks: Techniques, architectures, and applications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1458–1493, 3rd Quart., 2021.

[7] 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 Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, April 2021.

[8] W. Y. B. Lim, Z. Xiong, J. Kang, D. Niyato, C. Leung, C. Miao, and X. Shen, "When information freshness meets service latency in federated learning: A task-aware incentive scheme for smart industries," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 457–466, Jan. 2022.

[9] X. Yuan, W. Ni, M. Ding, K. Wei, J. Li, and H. V. Poor, "Amplitude-varying perturbation for balancing privacy and utility in federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1884–1897, Mar. 2023.

[10] Z. Zhao, C. Feng, W. Hong, J. Jiang, C. Jia, T. Q. S. Quek, and M. Peng, "Federated learning with non-IID data in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1927–1942, March 2022.

[11] J. Zhang, N. Li, and M. Dedeoglu, "Federated learning over wireless networks: A band-limited coordinated descent approach," in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, Vancouver, Canada, May 2021.

[12] C. G. Brinton S. Hosseinalipour and, V. Aggarwal, H. Dai, and M. Chiang, "From federated to fog learning: Distributed machine learning over heterogeneous wireless networks," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 41–47, Jan. 2020.

[13] J. Kang, Z. Xiong, N. Dusit, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10700–10714, Sept. 2019.

[14] O. A. Hanna, Y. H. Ezzeldin, C. Fragouli, and S. Diggavi, "Quantization of distributed data for learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 987–1001, Sept. 2021.

[15] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 5–36, Jan. 2022.

[16] D. Basu, D. Data, C. Karakus, and S. N. Diggavi, "Qsparse-local-SGD: Distributed SGD with quantization, sparsification, and local computations," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 217–226, May 2020.

[17] M. El Chamie, J. Liu, and T. Başar, "Design and analysis of distributed averaging with quantized communication," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3870–3884, Dec. 2016.

[18] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "UVeQFed: Universal vector quantization for federated learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 500–514, Dec. 2021.

[19] S. Chen, C. Shen, L. Zhang, and Y. Tang, "Dynamic aggregation for heterogeneous quantization in federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6804–6819, May 2021.

[20] Y. Wang, Y. Xu, Q. Shi, and T. H. Chang, "Quantized federated learning under transmission delay and outage constraints," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 323–341, Jan. 2022.

[21] Y. Du, S. Yang, and K. Huang, "High-dimensional stochastic gradient quantization for communication-efficient edge learning," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2128–2142, March 2020.

[22] J. Sun, T. Chen, G. B. Giannakis, Q. Yang, and Z. Yang, "Lazily aggregated quantized gradient innovation for communication-efficient federated learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2031–2044, Oct. 2022.

[23] X. Cao and T. Başar, "Decentralized multi-agent stochastic optimization with pairwise constraints and quantized communications," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3296–3311, May 2020.

[24] S. Lee, C. Park, S. Hong, Y. C. Eldar, and N. Lee, "Soft-sign stochastic gradient descent algorithm for wireless federated learning," in *Proc. IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Lucca, Italy, Sept. 2021.

[25] M. Kim, W. Saad, M. Mozaffari, and M. Debbah, "On the tradeoff between energy, precision, and accuracy in federated quantized neural networks," in *Proc. of the IEEE International Conference on Communications (ICC)*, Seoul, South Korea, May 2022.

[26] A. Mahmoudi, J. M. B. D. S. Junior, H. S. Ghadikolaei, and C. Fischione, "A-laq: Adaptive lazily aggregated quantized gradient," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Rio de Janeiro, Brazil, Dec. 2022.

[27] Y. Oh, Y. S. Jeon, M. Chen, and W. Saad, "Quantized distributed federated learning for industrial internet of things," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Rio de Janeiro, Brazil, Dec. 2022.

[28] T. Ma, H. Wang, and C. Li, "Quantized distributed federated learning for industrial internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3027–3036, Feb. 2023.

[29] Y. J. Liu, S. Qin, G. Feng, D. Niyato, Y. Sun, and J. Zhou, "Adaptive quantization based on ensemble distillation to support fl enabled edge intelligence," in *Proc. IEEE Global Communications Conference*, Rio de Janeiro, Brazil, Dec. 2022.

[30] X. Yuan, S. Hu, W. Ni, R. P. Liu, and X. Wang, "Joint user, channel, modulation-coding selection, and RIS configuration for jamming resistance in multiuser OFDMA systems," *IEEE Transactions on Communications*, vol. 71, no. 3, pp. 1631–1645, Mar. 2023.

[31] W. Zhang, D. Yang, W. Wu, H. Peng, N. Zhang, H. Zhang, and X. Shen, "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3688–3703, Oct. 2021.

[32] P. Zhang, C. Wang, C. Jiang, and Z. Han, "Deep reinforcement learning assisted federated learning algorithm for data management of IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8475–8484, March 2021.

[33] Q. V. Do, Q. -V. Pham, and W. -J. Hwang, "Deep reinforcement learning for energy-efficient federated learning in UAV-enabled wireless powered networks," *IEEE Communications Letters*, vol. 26, no. 1, pp. 99–103, Oct. 2022.

[34] H. T. Nguyen, N. Cong Luong, J. Zhao, C. Yuen, and D. Niyato, "Resource allocation in mobility-aware federated learning networks: A deep reinforcement learning approach," in *Proc. of IEEE World Forum on Internet of Things (WF-IoT)*, LA, USA, June 2020.

[35] Y. Zhan, P. Li, and S. Guo, "Experience-driven computational resource allocation of federated learning by deep reinforcement learning," in *Proc. of IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, LA, USA, May 2020.

[36] M. Han, X. Sun, S. Zheng, X. Wang, and H. Tan, "Resource rationing for federated learning with reinforcement learning," in *Proc. Computing, Communications and IoT Applications (ComComAp)*, Shenzhen, China, Nov. 2021.

[37] Y. Song, H. H. Chang, and L. Liu, "Federated dynamic spectrum access through multi-agent deep reinforcement learning," in *Proc. IEEE Global Communications Conference*, Rio de Janeiro, Brazil, Dec. 2022.

[38] Y. Jiang, M. Zhang, F. Zheng, Y. Chen, M. Bennis, and X. You, "A federated reinforcement learning method with quantization for cooperative edge caching in fog radio access networks," Available Online*: https://arxiv.org/abs/2206.11556*, Jun. 2022.

[39] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, to appear, 2021.

[40] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, March 2021.

[41] M. Sharma, S. Soman, and Jayadeva, "Minimal complexity machines under weight quantization," *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1189–1198, March 2021.

[42] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, "Binary neural networks: A survey," *Pattern Recognition*, vol. 105, pp. 1–14, July 2020.

[43] S. I. Young, W. Zhe, D. Taubman, and B. Girod, "Transform quantization for cnn compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5700–5714, Sept. 2022.

[44] Y. Cai, T. Tang, L. Xia, M. Cheng, Z. Zhu, Y. Wang, and H. Yang, "Training low bitwidth convolutional neural network on RRAM," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jeju, Korea (South), Jan. 2018.

[45] Y. Ji and L. Chen, "Fedqnn: A computation-communication-efficient federated learning framework for iot with low-bitwidth neural network quantization," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2494–2507, Feb. 2023.

[46] S. Wang, M. Chen, X. Liu, C. Yin, S. Cui, and H. Vincent Poor, "A machine learning approach for task and resource allocation in mobile-edge computing-based networks," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1358–1372, Feb. 2021.

[47] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[48] M. P. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *Siam Journal on Scientific Computing*, vol. 34, no. 3, pp. A1380–A1405, Jan. 2012.

[49] Y. Tian, Z. Zhang, Z. Yang, and R. Jin, "Hierarchical federated learning with adaptive clustering on non-iid data," in *Proc. IEEE Global Communications Conference*, Rio de Janeiro, Brazil, May 2022.

[50] N. Megiddo and A. Tamir, "Finding least-distances lines," *Journal on Algebraic and Discrete Methods*, vol. 4, no. 2, pp. 207–211, June 1983.

[51] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," in *Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, Dec. 2020.

[52] S. Wang, S. Hosseinalipour, M. Gorlatova, C. G. Brinton, and M. Chiang, "UAV-assisted online machine learning over multi-tiered networks: A hierarchical nested personalized federated learning approach," Available Online*: https://arxiv.org/abs/2106.15734*, Oct. 2023.

[53] S. Wang, M. Chen, C. G. Brinton, C. Yin, W. Saad, and S. Cui, "Performance optimization for variable bitwidth federated learning in wireless networks," Available Online*: https://arxiv.org/abs/2209.10200*, Sep. 2022.

[54] T. Ren, F. Cui, A. Atsidakou, S. Sanghavi, and N. Ho, "Towards statistical and computational complexities of polyak step size gradient descent," Available Online*: https://arxiv.org/abs/2110.07810v1*, Oct. 2020.

[55] Y. Yang, Z. Zhang, and Q. Yang, "Communication-efficient federated learning with binary neural networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3836–3850, Dec. 2021.

[56] Y. LeCun, "The MNIST database of handwritten digits," Available Online*: http://yann.lecun.com/exdb/mnist/*, Sep. 2020.

[57] A. Krizhevsky, "Learning multiple layers of features from tiny images," Available Online*: https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf*, April 2009.

[58] Y. Umuroglu and M. Jahre, "Streamlined deployment for quantized neural networks," Available Online*: https://arxiv.org/abs/1709.04060v2*, May 2018.

[59] A. K. Mishra and D. Marr, "Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy," in *International Conference on Learning Representations (ICLR)*, Vancouver, Canada, April 2018.

[60] Z. Li, B. Ni, W. Zhang, X. Yang, and W. Gao, "Performance guaranteed network acceleration via high-order residual quantization," in *Proc. of IEEE International Conference on Computer Vision (ICVV)*, Venice, Italy, Dec. 2017.