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.

Index Terms—Bitwidth federated learning (FL), FL training loss optimization, model-based reinforcement learning (RL).

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], [3], [4], [5], [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], [8], [9], [10], [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], [15], [16], [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], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], and [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 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], [32], [33], [34], [35], [36], [37], [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. Reference [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 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 $x_{m,n} \in \mathbb{R}^{N_{x} \times 1}$ and (in the case of supervised learning) a corresponding label vector $y_{m,n} \in \mathbb{R}^{N_{y} \times 1}$. The objective of the server and the devices is to minimize the global loss function over all data samples, which is given by

$$F(g) = \min_{g} \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N_m} f(g, x_{m,n}, y_{m,n}),$$

where $g \in \mathbb{R}^{Y \times 1}$ 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. $f(g, x_{m,n}, y_{m,n})$ is a loss function (e.g., squared error) that measures the accuracy of the generated global FL model $g$ in building a relationship between the input vector $x_{m,n}$ and the output vector $y_{m,n}$.

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 $g$ 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 $g$ 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 \( \alpha_t \) 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 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 \( h_{m,t}^k \) and the weight vector \( g_t^k \) of the neurons in layer \( k \) that is represented by \( \alpha_t \) bits, the output of each layer \( k \) at iteration \( t \) is given by [43]

\[
h_{m,t}^{k+1} = \begin{cases} \sigma(h_{m,t}^k \odot g_t^k), & \text{if } \alpha_t = 1, \\ \sigma\left(\sum_{i=0}^{\alpha_t-1}\sum_{j=0}^{\alpha_t-1}2^{i+j}(h_{m,t}^k \odot g_t^j)\right), & \text{if } \alpha_t > 1, \end{cases}
\]

(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 \( h_{m,t} = [h_{m,t}^1, \ldots, h_{m,t}^K] \), the cross-entropy loss function can be expressed based on the neurons in an output layer \( h_{m,t}^K \) as

\[
f(g_t, x_{m,n}, y_{m,n}) = -y_{m,n}\log(h_{m,t}^K) + (1 - y_{m,n})\log(1 - h_{m,t}^K),
\]

(3)

where \( g_t = [g_t^1, \ldots, g_t^k, \ldots, g_t^K] \) 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

\[
w_{m,t+1} = g_t - \lambda \sum_{m \in \mathcal{N}_{m,t}} \frac{\partial f(g_t, x_{m,n}, y_{m,n})}{\partial g_t},
\]

(4)

where \( \lambda \) 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 \( t \), \( w_{m,t+1} \) is the updated local FL model of device \( m \) at iteration \( t+1 \), and

\[
\frac{\partial f}{\partial g_t} = \frac{\partial f_{m,t}}{\partial g_t} \times \frac{\partial g_t}{\partial w_{k,j,t}} = \frac{\partial f_{m,t}}{\partial g_t} \times \text{Htanh}(g_t),
\]

(5)

where \( g_t \) represents the full-precision weights. \( \text{Htanh}(x) = \max(-1, \min(1, x)) \) 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 \( \alpha_t \) bits, which is given by [44]

\[
\hat{w}_{m,t}^{k,i}(\alpha_t) = Q(w_{m,t}^{k,i}, \alpha_t) = \begin{cases} \text{sign}(w_{m,t}^{k,i}), & \text{if } \alpha_t = 1, \\ \frac{R(2^{\alpha_t-1}-1)w_{m,t}^{k,i}}{2^{\alpha_t-1}w_{m,t}^{k,i}}, & \text{if } 1 < \alpha_t < V, \\ \frac{R(2^{\alpha_t-1})w_{m,t}^{k,i}}{2^{\alpha_t-1}w_{m,t}^{k,i}}, & \text{if } \alpha_t = V, \end{cases}
\]

(6)

where \( V \) is the bitwidth of the full-precision and \( \text{sign}(x) = 1 \) if \( x \geq 0 \) and \( \text{sign}(x) = -1 \) otherwise. \( R(\cdot) \) is a rounding function with \( R(x) = [x] \) if \( x \in [\frac{1}{2}, \frac{1}{2}] \), and \( R(x) = [x] \) otherwise. From (6), we see that when \( \alpha_t = 1 \) (i.e., the binary case), if \( w_{m,t}^{k,i} > 0 \), we have \( \hat{w}_{m,t}^{k,i} = 1 \) with \( \hat{w}_{m,t}^{k,i} \) and \( \hat{w}_{m,t}^{k,i} \) being \( j \)-th element in \( w_{m,t}^{k,i} \) and \( w_{m,t}^{k,i} \), otherwise \( \hat{w}_{m,t}^{k,i} = -1 \). For \( 1 < \alpha_t < V \), \( w_{m,t}^{k,i} \) 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 \( \hat{w}_{m,t} \) of each device \( m \) at each iteration \( t \), the update of the global FL model at iteration \( t \) is given by

\[
g_t(u_t, \alpha_t) = \sum_{m=1}^{M} \frac{u_{m,t}N_{m,t}}{\sum_{m=1}^{M} u_{m,t}N_{m,t}} \hat{w}_{m,t}(\alpha_t),
\]

(7)

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

\[
g_t = Q(g_t, \alpha_t) = \begin{cases} \text{sign}(g_t), & \text{if } \alpha_t = 1, \\ \frac{R(2^{\alpha_t-1}-1)g_t}{2^{\alpha_t-1}g_t}, & \text{if } 1 < \alpha_t < V, \\ \frac{R(2^{\alpha_t-1})g_t}{2^{\alpha_t-1}g_t}, & \text{if } \alpha_t = V. \end{cases}
\]

(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 \( \alpha_t \), 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 \( \alpha_t \) used to represent each element in FL model vector. Specifically, given \( \alpha_t \), the time used to calculate the training loss is given by

\[
j_{m,t}(\alpha_t) = \rho \frac{\alpha_t^2 N^C}{\partial f},
\]

(9)
where $\rho$ 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 $\vartheta$ 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 $\alpha_t$, the quantization delay can be represented as [46]

$$t^Q_{m,t}(\alpha_t) = \begin{cases} 0, & \text{if } \alpha_t = 1 \\
\frac{D}{\vartheta_f}, & \text{if } 1 < \alpha_t < V, \end{cases}$$

(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 $\hat{w}_{m,t}$ 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., $\hat{w}_{m,t} = w_{m,t}$. 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 $\hat{w}_{m,t}$ 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 $c_{m,i}(u_{m,t}) = u_{m,t} W \log_2 \left(1 + \frac{P h_{m,i}}{\sigma^2_N}\right)$ where $u_{m,t} \in \{0, 1\}$ is the user association index, $h_{m,i}$ is the channel gain between device $m$ and the server, and $\sigma^2_N$ represents the variance of additive white Gaussian noise. Then, the uplink transmission delay between device $m$ and the server is $t^T_{m,t}(u_{m,t}, \alpha_t) = \frac{D\alpha_t}{c_{m,t}(u_{m,t})}$ where $D\alpha_t$ is the data size of the quantized FL parameters $\hat{w}_{m,t}$.

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

$$l_t(u_t, \alpha_t) = \max_{m \in M} u_{m,t} \left( t^C_{m,t}(\alpha_t) + t^Q_{m,t}(\alpha_t) + t^T_{m,t}(u_{m,t}, \alpha_t) \right).$$

(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:

$$\min_{U, \alpha} F(g(u_T, \alpha)), \quad \text{subject to} \quad u_{m,t} \in \{0, 1\}, \alpha \in [0, V] \text{ and } \alpha \in \mathbb{N}^+, \forall m \in M, \forall t \in T.$$  

(12a)

$$\sum_{m=1}^{M} u_{m,t} \leq U, \forall m \in M, \forall t \in T,$$

(12b)

$$l_t(u_t, \alpha) \leq \gamma, \forall m \in M, \forall t \in T.$$  

(12c)

where $U = [u_1, \ldots, u_t, \ldots, u_T]$ is a device selection matrix over all iterations with $u_t = [u_{1,t}, \ldots, u_{M,t}]$ being a user association vector at iteration $t$, $\alpha = [\alpha_1, \ldots, \alpha_t, \ldots, \alpha_T]$ is a quantization precision vector of all devices for all iterations, and $T = \{1, \ldots, T\}$ is the training period. $\Gamma$ 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 be 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 $u_t$ and $\alpha_t$ 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 $U$ and the quantization scheme $\alpha$ 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 $U$ and $\alpha$. 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 $a_t = [u_t, \alpha_t] \in A$ that consists of the device selection scheme $u_t$ and the quantization scheme $\alpha_t$ of all device at iteration $t$ with $A$ being the discrete sets of available actions.

- **States**: The state is $s_t = F(g_t) \in S$ that measures the performance of global FL model at iteration $t$ with $F(g_t)$ being the FL training loss and $S$ being the sets of available states.

- **State Transition Probability**: The state transition probability $P(s_{t+1}|s_t, a_t)$ denotes the probability of transiting from state $s_t$ to state $s_t'$ when action $a_t$ is taken, which is given by

  \[ P(s_{t+1}|s_t, a_t) = \Pr\{s_{t+1} = s_t'|s_t, a_t\}. \tag{13} \]

  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 $a_t$, the reward function of the server is given by

  \[ r(s_t, a_t) = -F(g(u_t, \alpha_t)), \tag{14} \]

  where $F(g(u_t, \alpha_t))$ is the training loss at iteration $t$. Note that, $r(s_t, a_t)$ increases as $F(g(u_t, \alpha_t))$ 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 $\theta$ to map the input state to the output action. Then, the policy can be expressed as $\pi_\theta(s_t, a_t) = P(a_t|s_t)$.

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 $(s_t, a_t)$. 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 $\mu$, such that

  \[ F(g_{t+1}) \geq F(g_t) + (g_{t+1} - g_t)^T \nabla F(g_t) + \frac{\mu}{2} ||g_{t+1} - g_t||. \tag{16} \]

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

  \[ \mu I \leq \nabla^2 F(g_t,x_{mn},y_{mn}) \leq LI. \tag{17} \]

  We also assume that

  \[ \|\nabla f(g_t,x_{mn},y_{mn})\|^2 \leq \zeta_1 + \zeta_2 \|\nabla F(g_t)\|^2, \tag{18} \]

  where $F(g_t) = \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N_m} f(g_t,x_{m,n},y_{m,n})$.

  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 $a_t$ is given). Based on the analytical result, we can calculate the state transition probability $P(s_{t+1}|s_t, a_t)$. 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]:

\[ \epsilon = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \sum_{t=1}^{T} \epsilon_{m,t} N_{m,t} \tag{19} \]

where $\epsilon_{m,t} = \nabla F(g_{t}) - \nabla \overline{F}_{m}(g_{t})$ is the difference between the data distribution of device $m$ and the global data distribution. We also assume that $||\nabla f(g_t,x_{mn},y_{mn}) + \epsilon_{m}||^2 \leq \zeta_1 + \zeta_2 ||\nabla F(g_t)||^2 + B\epsilon^2$ for some positive $B$ with $F(g_t) = \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N_m} f(g_t,x_{m,n},y_{m,n})$.

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 $\mathbb{E}(F(g_{t+1}))$ and $\mathbb{E}(F(g_t))$) with a non-i.i.d. data distribution can be upper bounded as

$$
\mathbb{E}(F(g_{t+1})) - \mathbb{E}(F(g_t)) \\
\leq \mathbb{E}((\hat{g}_{t+1} - g_t) \nabla F(g_t) - c) \\
+ \frac{L}{2} \mathbb{E}((\hat{g}_{t+1} - g_t)^2) + \frac{L}{2} \mathbb{E}((\hat{g}_{t+1} - g_{t+1})^2), \quad (20)
$$

where $g_t$ and $\hat{g}_t$ are short for $g_t(u_t, \alpha_t)$ and $\hat{g}_t(u_t, \alpha_t)$, respectively. $c$ 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 $\hat{g}_{t+1}(u_{t+1}, \alpha_{t+1}) - g_t(u_t, \alpha_t)$ that is determined by the device selection vector $u_t$ and quantization scheme $\alpha_t$. To investigate how an action $\alpha_t = [u_t, \alpha_t]$ 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 $u_t$ and quantization scheme $\alpha_t$, the upper bound of $\mathbb{E}(F(g_{t+1})) - \mathbb{E}(F(g_t))$ in non-i.i.d. data distribution can be given by

$$
\mathbb{E}(F(g_{t+1})) - \mathbb{E}(F(g_t)) \\
\leq \frac{1}{2L} \left(-1 + \frac{4(N - 2)^2}{\mathbb{E} \|\Delta(\alpha_t)/\| + 1} \right) \mathbb{E} \|\nabla F(g_t)\|^2 \\
+ \frac{\mathbb{E} \|\nabla F(g_t)\|^2 + 4(N - 2)^2}{2L} \left(\frac{4(N - 2)^2}{\mathbb{E} \|\nabla F(g_t)\|^2} + L^2 \mathbb{E} \|\Delta(\alpha_t)\|^2 \right) \\
+ \mathbb{E} \left(\|\Delta(\alpha_t)\|^2\right), \quad (21)
$$

where $A = \sum_{m=1}^{N} u_{m,t} \times 1$ represents the sum of all selected devices’ data samples that are used to train their local models, $\Delta(\alpha_t) = \hat{g}_t(\alpha_t) - g_t$ is the quantization error of the global FL model that depends on the quantization scheme $\alpha_t$, $\mathbb{E} \|\Delta(\alpha_t)\|^2 = M^2 - \alpha_t$ is the unbiased quantization function defined in (6).

Proof: See Appendix B.

From Theorem 1, we can see that, the relationship between $\mathbb{E}(F(g_{t+1}))$ and $\mathbb{E}(F(g_t))$ (i.e., $s_{t+1}$ and $s_t$) depends on the selected action $\alpha_t$ as well as the constants $1/L$, $\zeta_1$, $\zeta_2$, and $Bc^2$. However, we do not know the values of $1/L$, $\zeta_1$, $\zeta_2$, and $Bc^2$ since they are predefined in assumptions (15)–(19). To find the tightest bound in (21), we must find the values of $1/L$, $\zeta_1$, $\zeta_2$, and $Bc^2$ so as to build the relationship between $s_{t+1}$ and $s_t$ and calculate the state transition probability $P(s_{t+1}|s_t, \alpha_t)$. To this end, a linear regression method [50] is used to determine the values of $1/L$, $\zeta_1$, $\zeta_2$, and $Bc^2$ since the relationship between $\mathbb{E}(F(g_{t+1}))$ and $\mathbb{E}(F(g_t))$ and these constants are linear. The regression loss function defined as

$$
\mathcal{J}(L, \zeta_1, \zeta_2, Bc^2) \\
= \frac{1}{T} \sum_{t=1}^{T} \left(\mathbb{E}(F(g_{t+1}^{(i)})) - \mathbb{E}(F(g_t^{(i)}))\right) \\
- K \left(L, \zeta_1, \zeta_2, Bc^2 | F(g_t^{(i)}, a_t^{(i)}, F(g_{t+1}^{(i)}))\right)^2, \quad (22)
$$

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

$$
L = L - \frac{\partial \mathcal{J}(L, \zeta_1, \zeta_2, B')}{\partial L}, \\
\zeta_1 = \zeta_1 - \frac{\partial \mathcal{J}(L, \zeta_1, \zeta_2, B')}{\partial \zeta_1}, \\
\zeta_2 = \zeta_2 - \frac{\partial \mathcal{J}(L, \zeta_1, \zeta_2, B')}{\partial \zeta_2}, \\
B' = B' - \frac{\partial \mathcal{J}(L, \zeta_1, \zeta_2, B')}{\partial B'}, \quad (23)
$$

where $B' = Bc^2$, $L$, $\zeta_1$, $\zeta_2$, and $Bc^2$ are learning rates for parameters $L$, $\zeta_1$, $\zeta_2$, and $Bc^2$.

C. Optimization of Device Selection and Quantization Scheme

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

$$
\mathcal{L}(\theta) = \sum_{(s_t, a_t) \tau} \prod_{t=1}^{T} P(s_t) \prod_{t=1}^{T} \pi_\theta(s_{t-1}, a_t) \\
\times P(s_t|s_{t-1}, a_t) \sum_{t=1}^{T} \tau'(s_t, a_t), \quad (25)
$$

where $\tau = \{s_0, a_0, \ldots, s_T, a_T\}$ is the trajectory replay buffer. Given (25), the optimization of policy network $\theta$ is

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

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

$$
\theta = \theta + \alpha \nabla_\theta \mathcal{L}(\theta), \quad (27)
$$

where $\alpha$ is the learning rate and the policy gradient is

$$
\nabla_\theta \mathcal{L}(\theta) = \sum_{(s_t, a_t) \tau} \prod_{t=1}^{T} P(s_t) \prod_{t=1}^{T} \pi_\theta(s_{t-1}, a_t) \\
\times P(s_t|s_{t-1}, a_t) \sum_{t=1}^{T} \tau'(s_t, a_t)
$$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
\[
= \frac{1}{T} \sum_{t=1}^{T} r(s_t, a_t) \nabla \log \pi_\theta(s_t, a_t).
\]  

(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., \( \| \nabla F(x) \| \leq C \).

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

**Condition 3** [52]: The Hessian of the loss function \( F(x) \) is \( \gamma \)-Lipschitz continuous, such that
\[
\| \nabla^2 F(x) - \nabla^2 F(y) \| \leq \gamma \| x - y \|.
\]

(29)

Together with Assumption 1, and Conditions 1 and 3, loss function \( F(x) \) is \( L_\gamma \)-smooth for \( L_\gamma = 4L + \gamma \epsilon C \) with \( \epsilon \in [0, 1/L] \). [Lemma 4.2, 51].

Based on (29), Lemma 1 can be rewritten as
\[
\mathbb{E} (F(g_{t+1})) - \mathbb{E} (F(g_t)) \leq \mathbb{E} (\left( \hat{g}_{t+1} - g_t \right) (\nabla F(g_t) - \epsilon)) + \frac{L_\gamma}{2} \mathbb{E} \left( \| \hat{g}_{t+1} - g_t \|^2 \right) + \frac{L_\gamma}{2} \mathbb{E} \left( \| g_{t+1} - g_{t+1} \|^2 \right).
\]

(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 \( u_t \) and quantization scheme \( \alpha_t \), an upper bound \( \mathbb{E} (F(g_{t+1})) - \mathbb{E} (F(g_t)) \) can be obtained as
\[
\mathbb{E} (F(g_{t+1})) - \mathbb{E} (F(g_t)) \leq \frac{1}{2L_\gamma} \left( -1 + \frac{4(N-A)^2}{N^2} \mathbb{E} \left( \| \Delta(\alpha_t) \| \right) + 1 \right) \mathbb{E} \left( \| \nabla F(g_t) \|^2 \right) + \frac{\mathbb{E} \left( \| \Delta(\alpha_t) \| \right) + 1}{2L_\gamma} \left( \frac{4(N-A)^2}{N^2} \zeta_1 + B\epsilon^2 \right) + \frac{L_\gamma^2}{2} \mathbb{E} \left( \| \Delta(\alpha_t) \|^2 \right) + \frac{M^2 \mathbb{E}^2}{2L_\gamma} + \mathbb{E} \left( \Delta(\alpha_t)^2 \right),
\]

(31)

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

**Theorem 2**: Given the user selection vector \( u_t \) and quantization scheme \( \alpha_t \), an upper bound \( \mathbb{E} (F(g_{t+1})) - \mathbb{E} (F(g_t)) \) can be obtained as
\[
\mathbb{E} (F(g_{t+1})) - \mathbb{E} (F(g_t)) \leq \frac{1}{2L_\gamma} \left( -1 + \frac{4(N-A)^2}{N^2} \mathbb{E} \left( \| \Delta(\alpha_t) \| \right) + 1 \right) \mathbb{E} \left( \| \nabla F(g_t) \|^2 \right) + \frac{\mathbb{E} \left( \| \Delta(\alpha_t) \| \right) + 1}{2L_\gamma} \left( \frac{4(N-A)^2}{N^2} \zeta_1 + B\epsilon^2 \right) + \frac{L_\gamma^2}{2} \mathbb{E} \left( \| \Delta(\alpha_t) \|^2 \right) + \frac{M^2 \mathbb{E}^2}{2L_\gamma} + \mathbb{E} \left( \Delta(\alpha_t)^2 \right),
\]

(31)

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

**Theorem 2**: Given the user selection vector \( u_t \) and quantization scheme \( \alpha_t \), an upper bound \( \mathbb{E} (F(g_{t+1})) - \mathbb{E} (F(g_t)) \) can be obtained as
\[
\mathbb{E} (F(g_{t+1})) - \mathbb{E} (F(g_t)) \leq \frac{1}{2L_\gamma} \left( -1 + \frac{4(N-A)^2}{N^2} \mathbb{E} \left( \| \Delta(\alpha_t) \| \right) + 1 \right) \mathbb{E} \left( \| \nabla F(g_t) \|^2 \right) + \frac{\mathbb{E} \left( \| \Delta(\alpha_t) \| \right) + 1}{2L_\gamma} \left( \frac{4(N-A)^2}{N^2} \zeta_1 + B\epsilon^2 \right) + \frac{L_\gamma^2}{2} \mathbb{E} \left( \| \Delta(\alpha_t) \|^2 \right) + \frac{M^2 \mathbb{E}^2}{2L_\gamma} + \mathbb{E} \left( \Delta(\alpha_t)^2 \right),
\]

(31)

where \( A = \sum_{m=1}^{M} u_{m,t} N_{m,t} \) represents the sum of all selected devices’ data samples, \( \Delta(\alpha_t) = \hat{g}_t (\alpha_t) - g_t \) is the quantization error of the global FL model, and \( \mathbb{E} \| \Delta(\alpha_t) \| \| M^2 - \alpha_t \| \) 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 \( L_\gamma, \zeta_1, \zeta_2, B\epsilon^2 \). 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 \( \rho \), the frequency of the CPU \( f \), the number of bits that can be processed by the CPU in one clock cycle \( \theta \), 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 \( 1/L, \zeta_1, \zeta_2, \) and \( B\epsilon^2 \), and calculate \( P(s_{t+1}|s_t, a_t) \), as shown in (22)-(28). Then, using \( P(s_{t+1}|s_t, a_t) \), a model based RL algorithm is used to find the optimal \( \alpha_t \) and \( u_t \) without any interactions between the server and devices. The entire process of training the proposed model based RL algorithm is shown in Algorithm 1.

**Algorithm 1** Model-Based RL for Device Selection and Quantization Optimization

**Input**: The environment state \( S \), the action space \( A \).

**Output**: The device selection and quantization scheme.

1. Initialize policy \( \pi_\theta \), transition replay buffer \( B \), trajectory replay buffer \( \tau \).
2. for iteration \( i = 1 : I \) do
3. Randomly selects a subset of devices to generate the global FL model that are quantized into \( \alpha_t \) bits.
4. Records \( F(g_t), F(g_{t+1}), \alpha_t \), and device selection scheme \( u_t \) in \( B \).
5. end for
6. Estimate \( 1/L, \zeta_1, \zeta_2, \) and \( B\epsilon^2 \) to construct \( P(s_{t+1}|s_t, a_t) \) using (23) based on the real transition in \( B \).
7. for iteration \( i = 1 : H \) do
8. Sample initial state from \( S \), then use policy \( \pi_\theta \) and learned \( P(s_{t+1}|s_t, a_t) \) to perform \( T \) trajectories and update \( \tau \).
9. Sample from \( \tau \), and update the current policy evaluation by solving Equation (28).
10. end for

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

a) In terms of computational complexity of calculating \( P(s_{t+1}|s_t, a_t) \), the server needs to find the values of \( 1/L, \zeta_1, \zeta_2, \) 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 \( T \) with a radius \( r \geq 1 \) to reach the optimal \( L^* \), \( \zeta \), \( \epsilon \), and \( B^* \) from initialized \( L^0 \), \( \zeta_1^0 \), \( \zeta_2^0 \), \( B^0 \) with \( \epsilon \) error [54].

b) The computational complexity of the proposed RL algorithm depends on the number of the parameters in the policy network \( \theta \) which depends on the size of action space \( A \) and the size of state space \( S \). In particular, the possible combinations of each \( a_i \) in action space \( A \) 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 \( \sum_{i=1}^{U} C_M^i \) and the number of quantization actions is \( Y \). The state space \( S \) consists of the continuous values of loss function \( F(g_s) \). To ensure a finite state space, the continuous loss function values is divided into \(|S|\) levels. Then, the size of state space \( S \) is \(|S|\). Therefore, the computational complexity of the proposed RL algorithm is \( O \left\{ Y|S| \sum_{i=1}^{U} C_M^i \frac{M!}{(M-i)!} H_k \right\} \), where \( H_k \) is the number of the neurons in layer \( k \) of the policy network \( \theta \).

IV. NUMERICAL EVALUATION

For our simulations, we consider a circular network area having a radius \( r = 1500 \text{ m} \) 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 x 28 x 256 + 256 x 64 + 64 x 10). To verify the feasibility of the proposed calculation time model in (9), we first simulate the actual calculation time using the clock module in 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 \( 5 \times 5 \) and the total number of model parameters in CNN is 116704 (= 5 x 5 x (3 x 32 x 32 + 32 x 32 + 32 x 64) + 576 x 64 x 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

---

**TABLE I**

| Parameters | Values | Parameters | Values | Parameters | Values |
|------------|--------|------------|--------|------------|--------|
| \( M \)    | 15     | \( U \)    | 6      | \( N_m \) | 200    |
| \( W \)    | 15 kHz | \( P \)    | 0.5 W  | \( \sigma_k \) | -174 dBm |
| \( K \)    | 8      | \( I \)    | 20     | \( I \)    | 1 s    |
| \( T \)    | 1000   | \( J \)    | 3.3 GHz| \( B \)    | 64     |
| \( D \)    | 217728 | \( \rho \)  | 2.8 x 10^5 | \( \epsilon \) | 0.02  |
| \( \zeta_k \) | 0.02   | \( \zeta_1 \) | 0.02   | \( \zeta_2 \) | 0.02   |

---

*Fig. 2. Calculation time of low bit-width federated learning vs. the quantization precision.*
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., 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 \( \Gamma \) for completing each FL training iteration increases, the average quantization bits \( \alpha \) and the identification accuracy increase. This is because as \( \Gamma \) 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 \( \alpha \) 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 $\alpha$ decreases, the number of iterations required to reach a fixed achievable accuracy increases slightly. This is due to the fact that as $\alpha$ decreases, the quantization error increases, which decreases the accuracy for modeling FL training process. However, with a decrease of $\alpha$, 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 algorithm increases. This is due to the fact that as $\alpha$ increases, the quantization error decreases. Fig. 11 also shows that as $\alpha$ 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 $\alpha$ 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 fewer 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 $\Gamma$. 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 device 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 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 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 $F(\hat{g}_{t+1}(u_{t+1}), \alpha_{t+1})$ using the second-order Taylor expansion and the $L$-smoothness of property in Assumption 1, which can be expressed as

$$F(\hat{g}_{t+1}) \leq F(\hat{g}_{t+1}) + (\hat{g}_{t+1} - \hat{g}_{t+1}) \nabla F(\hat{g}_{t+1}) + \frac{L}{2} \|\hat{g}_{t+1} - \hat{g}_{t+1}\|^2,$$  \hspace{1cm} (32)

where $\hat{g}_{t}$ and $\hat{g}_{t}$ are short for $g_{t}(u_{t}, \alpha_{t})$ and $\hat{g}_{t}(u_{t}, \alpha_{t})$, respectively.

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

$$E\left(F(\hat{g}_{t+1})\right) \leq E\left(F(\hat{g}_{t+1}) + (\hat{g}_{t+1} - \hat{g}_{t+1}) \nabla F(\hat{g}_{t+1}) + \frac{L}{2} \|\hat{g}_{t+1} - \hat{g}_{t+1}\|^2\right),$$  \hspace{1cm} (33)

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

$$F(\hat{g}_{t+1}) \leq F(g_{t}) + (\hat{g}_{t+1} - g_{t}) \nabla F(g_{t}) + \frac{L}{2} \|g_{t+1} - g_{t}\|^2.$$  \hspace{1cm} (34)

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

$$E\left(F(g_{t+1})\right) - E\left(F(g_{t})\right) \leq E\left((g_{t+1} - g_{t}) \nabla F(g_{t}) + \frac{L}{2} \|g_{t+1} - g_{t}\|^2\right) + \frac{L}{2} E\left(||\hat{g}_{t+1} - g_{t}||^2\right).$$  \hspace{1cm} (35)

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

$$E\left(F(\hat{g}_{t+1})\right) - E\left(F(g_{t})\right) \leq E\left((\hat{g}_{t+1} - g_{t}) \nabla F(g_{t}) - \epsilon\right) + \frac{L}{2} E\left(||\hat{g}_{t+1} - g_{t}||^2\right) + \frac{L}{2} E\left(||\hat{g}_{t+1} - g_{t+1}||^2\right).$$  \hspace{1cm} (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 $\hat{g}_{t+1}$ and the expectation of the full-precision model $g_t$ in Lemma 1, which is given by

$$E(\hat{g}_{t+1} - g_{t}) = E(\hat{g}_{t+1} - g_{t} + g_{t+1} - g_{t})$$
$$= E(\Delta(\alpha_{t}) - \lambda (\nabla F(g_{t}) - e_{t})), \hspace{1cm} (37)$$

where $\Delta(\alpha_{t}) = ||\hat{g}_{t+1} - g_{t+1}||$ is the quantization error at iteration $t+1$ and $e_{t}$ 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...
Substituting (37) into Lemma 1, we have

\[ e_t = \nabla F (g_t) - \frac{\sum_{m=1}^{M} \sum_{n \in N_{m,t}} u_{m,t} (\nabla f (g_t, x_{mn}, y_{mn}) + \epsilon_m)}{N \sum_{m=1}^{M} N_{m,t} u_{m,t}} \]  

(38)

Substituting (37) into Lemma 1, we have

\[ \mathbb{E} (F (g_{t+1})) - \mathbb{E} (F (g_t)) \]

\[ \leq \mathbb{E} (\| \hat{g}_{t+1} - g_t \|_2^2) + \frac{L}{2} \mathbb{E} (\| \hat{g}_{t+1} - g_t \|_2^2) \]

\[ + \frac{L}{2} \mathbb{E} (\| g_{t+1} - g_t \|_2^2) \]

\[ \leq \mathbb{E} \left( \left( \frac{(N - A) \sum_{m=1}^{M} \sum_{n \in N_{m,t}} \nabla f (g_t) + \epsilon_m}{NA} \right)^2 \right) \]

\[ - \frac{1}{A} \sum_{m=1}^{M} u_{m,t} L \Delta (\alpha_t) \]

\[ + \frac{1}{N} \sum_{m=1}^{M} N_m (1 - u_{m,t}) \sum_{n \in N_{m,t}} \nabla f (g_t) + \epsilon_m \right)^2 \]

\[ \leq \mathbb{E} \left( \frac{1}{N} (N - A) \sum_{m=1}^{M} \sum_{n \in N_{m,t}} \nabla f (g_t) + \epsilon_m \right)^2 \]

\[ + \frac{1}{A} \sum_{m=1}^{M} u_{m,t} L \Delta (\alpha_t) \]

\[ + \frac{1}{N} \sum_{m=1}^{M} N_m (1 - u_{m,t}) \sum_{n \in N_{m,t}} \nabla f (g_t) + \epsilon_m \right)^2 \]

(39)

where (a) stems from the fact that \( \mathbb{E} (\hat{g}_{t+1}) = g_{t+1} \).

Next, we derive \( \mathbb{E} (\| e_t \|_2^2) \), which can be given as in (40), shown at the bottom of the page, where \( f (g_t) \) is short for \( f (g_t, x_{mn}, y_{mn}) \), (a) stems from the second-order Taylor expansion of \( f (g_t + \Delta (\alpha_t), x_{mn}, y_{mn}) \), (b) stems from the twice-continuously differentiable of \( f (g_t, x_{mn}, y_{mn}) \) with \( \mu I \preceq \nabla^2 f (g_t, x_{mn}, y_{mn}) \preceq LI \). The inequality equation in (40) is achieved by the triangle-inequality (i.e., \( \| \nabla f (g_t, x_{mn}, y_{mn}) \| \leq \sqrt{\zeta_1 + \zeta_2 \| \nabla F (g_t) \|^2} \)). Substituting the triangle-inequality and \( \sum_{m=1}^{M} N_{m,t} u_{m,t} = A \) into (40), we have

\[ \mathbb{E} (\| e_t \|_2^2) = \frac{\sum_{m=1}^{M} \sum_{n \in N_{m,t}} u_{m,t} \nabla f (g_t)}{N \sum_{m=1}^{M} N_{m,t} u_{m,t}} \]

\[ \leq \mathbb{E} \left( \left( \frac{\sum_{m=1}^{M} \sum_{n \in N_{m,t}} u_{m,t} \nabla f (g_t) + \epsilon_m + \frac{1}{2} \| \sigma \|_2^2}{\sum_{m=1}^{M} N_{m,t} u_{m,t}} \right)^2 \right) \]

\[ \leq \mathbb{E} \left( \left( \frac{\sum_{m=1}^{M} \sum_{n \in N_{m,t}} u_{m,t} \nabla f (g_t) + \epsilon_m + \Delta (\alpha_t) L}{\sum_{m=1}^{M} N_{m,t} u_{m,t}} \right)^2 \right) \]  

(40)
\[
(a) \quad \mathbb{E} \left( \left( N - A \right) \sum_{m=1}^{M} \sum_{n \in N_{m,t}} \frac{\| \nabla f(g_t) + \epsilon_m \|}{N} + L \| \Delta(\alpha_t) \| \right) \\
= \mathbb{E} \left( \left( N - A \right) \sum_{m=1}^{M} \sum_{n \in N_{m,t}} \frac{\| \nabla f(g_t) + \epsilon_m \|}{N} + L \| \Delta(\alpha_t) \| \right) \right)^2
\]

\[
(b) \quad \leq \left( \mathbb{E} \| \Delta(\alpha_t) \| + 1 \right) \left( 4 \left( N - A \right)^2 \left( \zeta_1 + \zeta_2 \| \nabla F(g_t) \|^2 + Bc^2 \right) \frac{\| \epsilon_t \|^2}{N^2} + L^2 \mathbb{E} \| \Delta(\alpha_t) \| \right)
\]

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

\[
\mathbb{E} \left( F(g_{t+1}) \right) - \mathbb{E} \left( F(g_t) \right) \\
\leq -\frac{1}{2L} \left( -1 + 4 \left( N - A \right)^2 \left( \mathbb{E} \| \Delta(\alpha_t) \| + 1 \right) \zeta_2 \| \nabla F(g_t) \|^2 \right) + \frac{4 \left( N - A \right)^2 \left( \zeta_1 + \zeta_2 \| \nabla F(g_t) \|^2 \right)}{N^2} + L^2 \mathbb{E} \| \Delta(\alpha_t) \|
\]

This completes the proof.

**REFERENCES**

[1] N. Yang et al., “Model-based reinforcement learning for quantized federated learning performance optimization,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Rio de Janeiro, Brazil, Dec. 2022, pp. 5063–5068.

[2] M. Chen et al., “Distributed learning in wireless networks: Recent progress and future challenges,” *IEEE J. Sel. Areas Commun.*, 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 Trans. Signal Process.*, vol. 69, pp. 3796–3811, 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 Things J.*, 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 J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3177–3192, Oct. 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 Commun. Surveys Tuts.*, 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 Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, 3rd Quart., 2021.

[8] W. Y. B. Lim et al., “When information freshness meets service latency in federated learning: A task-aware incentive scheme for smart industries,” *IEEE Trans. Ind. Informat.*, 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 Trans. Inf. Forensics Secur.*, vol. 18, pp. 1884–1897, 2023.

[10] Z. Zhao et al., “Federated learning with non-IID data in wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1927–1942, Mar. 2022.

[11] J. Zhang, N. Li, and M. Dedegolu, “Federated learning over wireless networks: A band-limited coordinated descent approach,” in *Proc. IEEE Comput. Commun. (INFOCOM)*, Vancouver, BC, Canada, May 2021, pp. 1–10.

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

[13] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, “Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[14] O. A. Hanna, Y. H. Ezzeldin, C. Fragioulis, and S. Diggavi, “Quantization of distributed data for learning,” *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 3, pp. 987–1001, Sep. 2021.

[15] K. B. Letiaf, Y. Shi, J. Lu, and J. Lu, “Edge artificial intelligence for 6G: Vision, enabling technologies, and applications,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 1–8, Jan. 2022.

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

[17] M. El Chamie, J. Liu, and T. Basar, “Design and analysis of distributed averaging with quantized communication,” *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3870–3884, Dec. 2016.

[18] Y. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, “UvEQofd: Universal vector quantization for federated learning,” *IEEE Trans. Signal Process.*, vol. 69, pp. 500–514, 2021.

[19] S. Chen, C. Shen, L. Zhang, and Y. Tang, “Dynamic aggregation for heterogeneous quantization in federated learning,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6804–6819, Oct. 2021.

[20] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, “Quantized federated learning under transmission delay and outage constraints,” *IEEE J. Sel. Areas Commun.*, 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 Trans. Signal Process.*, vol. 68, pp. 2128–2142, 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 Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2031–2044, Apr. 2022.

[23] X. Cao and T. Basar, “Decentralized multi-agent stochastic optimization with pairwise constraints and quantized communications,” *IEEE Trans. Signal Process.*, vol. 68, pp. 3296–3311, 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 22nd Int. Workshop Signal Process. Adv. Wireless Commun.* (SPAWC), Lucca, Italy, Sep. 2021, pp. 241–245.

[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. IEEE Int. Conf. Commun. (ICC)*, Seoul, South Korea, May 2022, pp. 2194–2199.
Y. Tian, Z. Zhang, Z. Yang, and R. Jin, “Hierarchical federated learning with adaptive clustering on non-IID data,” in *IEEE Global Commun. Conf. (GLOBECOM)*, Rio de Janeiro, Brazil, Dec. 2022, pp. 627–632.

N. Megiddo and A. Tamir, “Finding least-distances lines,” *SIAM J. Algebraic Discrete Methods*, vol. 4, no. 2, pp. 207–211, Jun. 1983.

Z. Li, B. Ni, W. Zhang, X. Yang, and W. Gao, “Performance guaranteed network acceleration via high-order residual quantization,” in *IEEE Trans. Commun.* vol. 69, no. 2, pp. 591–598, Feb. 2021.

A. Fallah, A. Mokhtari, and A. Ozdaglar, “Personalized federated learning: A meta-learning approach,” in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, Dec. 2020, pp. 1–11.

S. Wang, S. Hosseinalipour, M. Gorlatova, C. G. Brinton, and M. Chiang, “VA- assisted online machine learning over multi-tiered networks: A hierarchical nested personalized federated learning approach,” 2021, arXiv:2106.15754.

S. Wang, M. Chen, C. G. Brinton, C. Yin, W. Saad, and S. Cui, “Performance optimization for variable bitwidth federated learning in wireless networks,” 2022, arXiv:2209.10200.

T. Ren, F. Cui, A. Atsidakou, S. Sanghavi, and N. Ho, “Towards statistical and computational complexities of Polyak step size gradient descent,” 2021, arXiv:2110.07810.

Y. Yang, Z. Zhang, and Q. Yang, “Communication-efficient federated learning with binary neural networks,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3836–3850, Dec. 2021.

Y. LeCun (Sep. 2020). The MNIST Database of Handwritten Digits. [Online]. Available: http://yann.lecun.com/exdb/mnist/

A. Krizhevsky (Apr. 2017). CIFAR-10: A dataset for benchmarking machine learning algorithms. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

Y. Umuroglu and M. Jahre, “Streamlined deployment for quantized neural networks,” 2017, arXiv:1709.04060.

A. K. Mishra and D. Marr, “Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy,” in *Proc. ICLR*, 2021, pp. 1–17.

Z. Li, B. Ni, W. Zhang, X. Yang, and W. Gao, “Performance guaranteed network acceleration via high-order residual quantization,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2603–2611.

Sihua Wang: Received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2021.

Mingzhe Chen: (Member, IEEE) is currently an Assistant Professor with the Department of Electrical and Computer Engineering and the Institute of Data Science and Computing, University of Miami. His research interests include federated learning, reinforcement learning, virtual reality, unmanned aerial vehicles, and the Internet of Things. He has received from the IEEE Communication Society four journal article awards, including the Fred W. Ellersick Prize Award in 2022, the IEEE Marconi Prize Paper Award in Wireless Communications in 2023, the Young Author Best Paper Award in 2021 and 2023, and three conference best paper awards at IEEE ICC in 2020, IEEE GLOBECOM in 2020, and IEEE WCNC in 2021. He also serves as an Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and IEEE TRANSACTIONS ON MACHINE LEARNING IN COMMUNICATIONS AND NETWORKING.
Christopher G. Brinton (Senior Member, IEEE) received the Ph.D. (Hons.) and M.S. degrees in electrical engineering from Princeton University in 2016 and 2013, respectively. He is currently the Elmore Chaired Assistant Professor in electrical and computer engineering with Purdue University. Prior to joining Purdue University, he was the Associate Director of the EDGE Laboratory and a Lecturer in electrical engineering with Princeton University. His research interests are at the intersection of networked systems and machine learning, specifically in distributed machine learning, fog/edge network intelligence, and data-driven network optimization. He was a recipient of the NSF CAREER Award, ONR Young Investigator Program (YIP) Award, DARPA Young Faculty Award (YFA), and Intel Rising Star Faculty Award. He also serves as an Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, in the ML and AI for wireless area.

Changchuan Yin (Senior Member, IEEE) received the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications, Beijing, China, in 1998. In 2004, he was a Visiting Scholar with the Faculty of Science, The University of Sydney, Sydney, NSW, Australia. From 2007 to 2008, he held a visiting position with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA. He is currently a Professor with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His research interests include wireless networks and statistical signal processing. He was a co-recipient of the IEEE Guglielmo Marconi Prize Paper Award in 2023 and the IEEE International Conference on Wireless Communications and Signal Processing Best Paper Award in 2009. He has served as the symposium co-chair and a TPC member for many IEEE conferences.

Walid Saad (Fellow, IEEE) received the Ph.D. degree from the University of Oslo, Norway, in 2010. He is currently a Professor with the Department of Electrical and Computer Engineering, Virginia Tech, where he leads the Network science, Wireless, and Security (NEWS) Laboratory. He is also the Next-G Wireless Faculty Lead of Virginia Tech’s Innovation Campus. His research interests include wireless networks (5G/6G/beyond), machine learning, game theory, security, UAVs, semantic communications, cyber-physical systems, and network science. He was a recipient of the NSF CAREER Award in 2013, the AFOSR Summer Faculty Fellowship in 2014, and the Young Investigator Award from the Office of Naval Research (ONR) in 2015. He was the coauthor of 11 conference best paper awards at IEEE WiOpt in 2009, ICIMP in 2010, IEEE WCNC in 2012, IEEE PIMRC in 2015, IEEE SmartGridComm in 2015, EuCNC in 2017, IEEE GLOBECOM (2018 and 2020), IFIP NTMS in 2019, and IEEE ICC (2020 and 2022). He was a recipient of the 2015 and 2022 Fred W. Ellersick Prize from the IEEE Communications Society and the IEEE Communications Society Marconi Prize Award in 2023. He was also the coauthor of the papers that received the IEEE Communications Society Young Author Best Paper Award in 2019, 2021, and 2023. Other recognitions include the 2017 IEEE ComSoc Best Young Professional in Academia Award, the 2018 IEEE ComSoc Radio Communications Committee Early Achievement Award, and the 2019 IEEE ComSoc Communication Theory Technical Committee Early Achievement Award. From 2015 to 2017, he was named the Stephen O. Lane Junior Faculty Fellow with Virginia Tech. In 2017, he was named the College of Engineering Faculty Fellow. He received the Dean’s Award for Research Excellence from Virginia Tech in 2019. He was also an IEEE Distinguished Lecturer in 2019 and 2020. He has been annually listed in the Clarivate Web of Science Highly Cited Researcher List since 2019. He also serves as an Area Editor for the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING and the IEEE TRANSACTIONS ON COMMUNICATIONS. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON MACHINE LEARNING IN COMMUNICATIONS AND NETWORKING.

Shuguang Cui (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Stanford University, CA, USA, in 2005. Afterwards, he has been working as an Assistant Professor, an Associate Professor, a Full Professor, and a Chair Professor in electrical and computer engineering with the University of Arizona, Texas A&M University, UC Davis, and CUHK, Shenzhen, respectively. He was the Executive Dean of the School of Science and Engineering, CUHK, Shenzhen, and the Executive Vice Director of the Shenzhen Research Institute of Big Data. His current research interests focus on data driven large-scale system control and resource management, large data set analysis, the IoT systemic design, energy harvesting-based communication system design, and cognitive network optimization. He was selected as the Thomson Reuters Highly Cited Researcher and listed in the Worlds’ Most Influential Scientific Minds by ScienceWatch in 2014. He was a recipient of the IEEE Signal Processing Society 2012 Best Paper Award. He has served as the general co-chair and the TPC co-chairs for many IEEE conferences. He has also been serving as an Area Editor for IEEE Signal Processing Magazine and an Associate Editor for IEEE TRANSACTIONS ON BIG DATA, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Series on Green Communications and Networking, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He has been an elected member of IEEE Signal Processing Society SPCom Technical Committee (2009-2014) and the elected Chair of IEEE ComSoc Wireless Technical Committee (2017-2018). He is a member of the Steering Committee of IEEE TRANSACTIONS ON BIG DATA and the Chair of the Steering Committee of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is also the Vice Chair of the IEEE VT Fellow Evaluation Committee and a member of the IEEE ComSoc Award Committee. He was an IEEE ComSoc Distinguished Lecturer in 2014 and IEEE VT Society Distinguished Lecturer in 2019. In 2020, he won the IEEE ICC Best Paper Award, ICIP Best Paper Finalist, and the IEEE Globecom Best Paper Award. In 2021, he won the IEEE WCNC Best Paper Award. In 2023, he won the IEEE Marconi Best Paper Award, got elected as a fellow of Canadian Academy of Engineering and starts to serve as the Editor-in-Chief for IEEE TRANSACTIONS ON MOBILE COMPUTING.