This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3264463

Deep Reinforcement Learning Assisted Client Selection in Non-orthogonal Multiple Access based Federated Learning

Rana Albelaihi, *Graduate Student Member, IEEE,* Akhil Alasandagutti, Liangkun Yu, *Graduate Student Member, IEEE,* Jingjing Yao, *Member, IEEE,* and Xiang Sun, *Member, IEEE*

***Abstract*—To reap the benefit of big data generated by the massive number of Internet of Things (IoT) devices while preserving data privacy, federated learning (FL) has been proposed to enable IoT devices to train machine learning models locally. That is, instead of sharing the local data sets, different clients in terms of IoT devices only need to upload their local models to a centralized FL server. Client selection in FL is critical to maximize the number of qualified clients, who can successfully upload their local models to the FL server before the predefined deadline. Normally, client selection is coupled with wireless resource management owing to the fact that different clients need to share the same spectrum to upload their local models. The existing solutions of joint optimizing client selection and resource management are designed based on frequency division multiple access (FDMA) or time division multiple access (TDMA), which do not consider the dynamics of the clients and lead to low bandwidth utilization. In this paper, we propose the Non-Orthogonal Multiple Access (NOMA) based resource allocation for client selection in FL to dynamically and jointly optimize client selection for each global iteration as well as the transmission power of each selected client in each time slot within a global iteration. We design the Deep Reinforcement lEarning based client selection in non-orthogonAl Multiple access based Federated Learning (DREAM-FL) algorithm to solve the problem. Extensive simulations are conducted to demonstrate that DREAM-FL can select more qualified clients and has higher model accuracy than FDMA and TDMA-based solutions.**

***Index Terms*—Federated Learning, client selection, deep reinforcement learning, NOMA**

I. Introduction

Analyzing the high volume of data generated by the Internet of Things (IoT) devices are valuable [1]. However, due to some personal and sensitive information hiding behind IoT data, users are reluctant to upload their IoT data to a centralized facility for analysis. Hence, it is necessary to design a distributed data analysis framework to locally analyze IoT data to preserve data privacy and reduce network traffic. Federated learning (FL) is this type of framework to train machine learning models over local data sets that are distributed among IoT devices [2]–[5]. In an IoT-based FL system, a machine learning model is trained via many global iterations. In each global iteration, a centralized FL server selects a number of clients in terms of IoT devices and broadcasts an initialized global model to the selected clients via the existing wireless infrastructure, such as base stations (BSs), as shown in Fig. 1. After receiving the global model, each client trains the model using its local data set to generate its local model and uploads the local model to the FL server via the BS. Once the FL receives the local models from all the selected clients, it aggregates the local models to update the global model. The global model keeps training in each global iteration until it is converged [6].

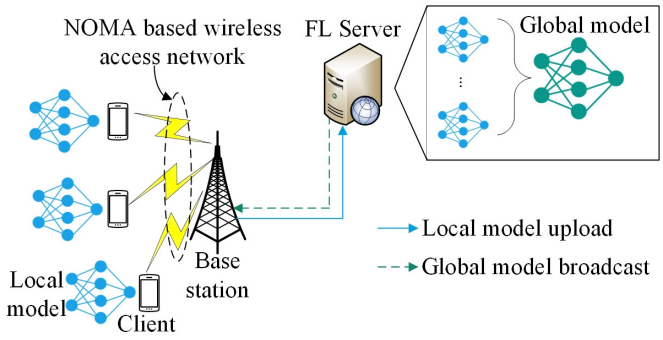


Fig. 1: NOMA-based wireless federated learning.

The client selected in heterogeneous FL is critical to determine the global model training latency. That is, in heterogeneous FL, clients have different computing and communications capabilities to train and upload their local models.Hence, the FL server has to wait for those straggler clients, who need a long time to train and upload their models, in each global iteration. The straggler clients can significantly prolong the overall training time in FL [7]. One of the most popular client selection methods to avoid the straggler problem is to set up a deadline for each global iteration. The FL server only selects the clients, who can upload their local models before the deadline, thus avoiding the straggler issue [8]. On the other hand, client selection is always coupled with wireless resource allocation since the selected clients would share the same wireless spectrum to upload their models [9], [10]. For example, assigning more bandwidth resources to a client may enable this client to be a qualified client to successfully upload their local models to the FL server before the predefined deadline. The existing joint resource allocation and client selection in FL is based on frequency division multiple access (FDMA) or time division multiple

R. Albelaihi, A.Alasandagutti, L. Yu, and X. Sun are with the University of New Mexico, Albuquerque, NM 87131, USA. E-mail: {ralbelaihi,aalasand1,liangkun,sunxiang}@unm.edu.

J. Yao is with the Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA. E-mail: jingjing.yao@ttu.edu.

This work was supported by the National Science Foundation under Award CNS-2148178.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3264463

access (TDMA) spectrum sharing techniques. In FDMA-based FL, the amount of bandwidth will be reserved for each selected client in each global iteration to meet the deadline requirement. In TDMA-based FL, a number of time slots are reserved for each selected client, who would utilize the whole spectrum to upload its local model during the allocated time slots. The existing joint resource allocation and client selection solutions in FL are mainly to reserve wireless resources in terms of bandwidth and time slots for the selected clients. Wireless resource reservation would lead to low bandwidth utilization due to client dynamics [1]. Taking an FDMA-based FL system as an example, assume that Client-A is selected to participate in training the global model during the current global iteration and the system reserves, for example, one wireless channel (e.g., 1 MHz bandwidth) to Client-A based on its current channel condition to ensure it can upload its local model before the deadline of the current global iteration. As shown in Fig. 2, reserving this wireless channel for Client-A means that other clients cannot reuse the wireless channel to upload their local models even if Client-A has already uploaded its local model or has not started to upload its local model, thus leading to low bandwidth utilization and potentially reducing the number of selected clients. Fig. 2 also provides an example in TDMA-based FL, where two time slots are reserved for Client-A. However, Client-A only needs one time slot to finish its local model uploading because of, for example, better channel condition between Client-A and the BS. Hence, bandwidth utilization and the number of selected clients are reduced.

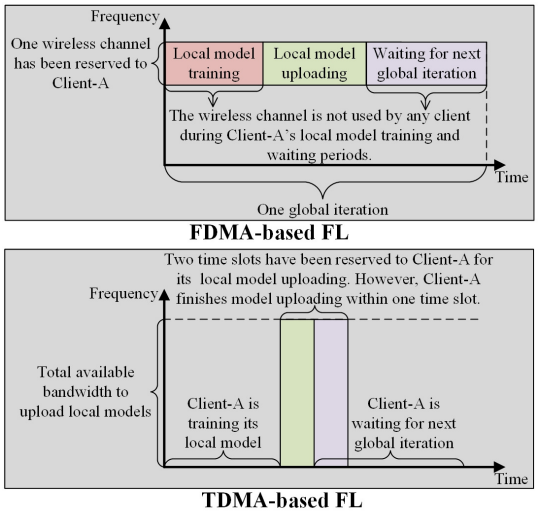


Fig. 2: Illustration of drawbacks for FDMA and TDMA-based FL.

To fully utilize bandwidth resources, we propose the Non-Orthogonal Multiple Access (NOMA) based resource allocation for client selection in FL. In NOMA-based FL, all the selected clients can upload their local models over the same time slot and frequency band, but with different transmission power, which has to be optimized to meet the deadline. The BS receives the superimposed signals from the clients and applies successive interference cancellation (SIC) to decode the local models from the superimposed signals. Note that the major reason to apply NOMA rather than TDMA/FDMA is that NOMA has been demonstrated to have higher bandwidth utilization, higher network capacity, and more simultaneous connections than TDMA/FDMA [9]–[11]. Also, to adapt to client dynamics in NOMA-based FL, we propose to dynamically optimize the transmission power of the selected clients in each time slot based on their channel conditions and residual traffic loads (i.e., the number of residual bits for uploading a client's local model). Here, dynamically optimizing the transmission power of the clients basically means that wireless resources will not be reserved for the clients at all the time slots in a global iteration. Dynamically determining the transmission power of the selected clients can potentially increase the bandwidth utilization and the number of selected clients, but may lead to new challenges. 1) It is difficult to estimate the communications latency of a client in uploading its local model, which is critical to client selection. This is because only the clients, which are able to upload their local models before the deadline, can be selected. 2) The resource allocation and client selection problems are coupled together but are optimized at different levels of time granularity. That is, client selection should be determined at the beginning of each global iteration, but the resource allocation is optimized for each time slot (e.g., every 100 ms) in a global iteration.

To resolve the challenges, we propose the Deep Reinforcement lEarning based client selection in non-orthogonAl Multiple access based Federated Learning (DREAM-FL). In DREAM-FL, the policies that are used to determine client selection and dynamic power allocation will be trained via simulation experiences. In addition, to handle different levels of time granularity for client selection and resource allocation, we design two Advantage Actor Critic (A2C) models [12], [13] to solve the resource allocation and client selection problems, respectively. The reward of client selection depends on the policy of resource allocation. The major contributions of the paper are summarized as follows.

1) We propose to jointly optimize client selection and resource management in the context of NOMA-based FL.
2) We formulate the problem to optimize client selection and transmission power allocation such that the number of selected clients is maximized, while ensuring all selected clients upload their local models before the deadline.
3) We design the DREAM-FL algorithm, which is based on the advantage actor-critic (A2C) deep reinforcement learning method, to dynamically and efficiently select clients and allocate transmission power.
4) The performance of DREAM-FL is evaluated via extensive simulations.

The rest of the paper is organized as follows. Section II summarizes the related works. Section III introduces the system models to estimate the latency of a client in computing its local model and uploading the local model to the FL server.

[1]Here, client dynamics comprises two aspects, i.e., 1) dynamic channel conditions, where the clients may move around, thus leading to varying channel conditions and uploading data rates to the BS, and 2) dynamic traffic loads, where different clients may start and finish their local model uploading processes in different time slots.

Also, the problem of optimizing client selection and resource management in NOMA-based FL is formulated. Section IV describes the detail of DREAM-FL, and simulation results are discussed in Section V. Section VI concludes the paper.

II. Related Work

Due to the heterogeneous and dynamic features of clients, client selection is critical to maximize the number of qualified clients in each global iteration, which can substantially reduce the overall training time and improve the model accuracy [8], [14]. In wireless FL, since clients share the same spectrum to upload their local models, client selection is always coupled with wireless resource allocation. Different spectrum sharing solutions have different resource allocation strategies, thus leading to various client selection solutions.

In the **FDMA-based FL system**, the amount of bandwidth reserved for each selected client would be calculated to ensure a selected client can upload its local model before the deadline while guaranteeing the sum of the bandwidth assigned to all the selected clients to be less than or equal to the bandwidth available at the BS. Shi *et al.* [15] estimated the total number of global iterations required to obtain the converged global model, which is a function of the number of selected clients. That is, selecting more clients in each global iteration leads to fewer global iterations to have the converged global model, and vice versa. Based on that, they designed a joint bandwidth allocation and client selection method to minimize the overall training time, which equals the sum of the delay for all the global iterations. Xu *et al.* [16] proved that selecting fewer clients in early global iterations but more clients in later global iterations can potentially improve the global model accuracy. Based on this finding, they selected more clients as the number of global iterations increases. Chen *et al.* [17] considered the transmission errors in uploading clients' local models. They designed a jointly optimizing client selection, resource block allocation, and transmission power management to minimize the training loss in each global iteration.

In the **TDMA-based FL system**, different numbers of time slots would be allocated to the selected clients in a global iteration to ensure they successfully upload their local models before the deadline. Each of these selected clients can utilize the whole spectrum to upload its local model during the allocated time slots. Hence, a selected client has extra waiting time if the client tries to upload its local model, but another selected client is occupying the spectrum to upload its local model. Albelaihi *et al.* [18] applied a queuing model to estimate the waiting time of a selected client, which is a function of the number of selected clients and their channel gain values. Based on that, they designed an algorithm to solve an optimization problem, which maximizes the number of qualified clients. Yu *et al.* [19] formulated the waiting time of a selected client as a recursive function, and designed a heuristic algorithm to optimize the trade-off between minimizing the energy consumption of the selected clients and maximizing the number of selected clients. Albelaihi *et al.* [20] considered the green FL scenario, where the clients are powered by portable batteries that can harvest green energy from the environment to prolong the battery life [21]. They designed a client selection method to optimize the trade-off between maximizing the number of qualified clients and minimizing the energy pulled from the batteries for the selected clients.

Although many solutions have been proposed to jointly optimize client selection and transmission power allocation in both FDMA and TDMA-based FL systems, none of them considers client dynamics defined in Footnote 1. As a result, they reserve the amount of wireless resources (i.e., bandwidth in FDMA and time slots in TDMA) to these selected clients at the beginning of a global iteration to ensure all the selected clients can finish their local model uploading before the deadline. However, wireless resource reservation leads to low resource utilization since other clients cannot utilize the resources, which have been reserved for a selected client, even if the selected client has not started to upload its local model or has already uploaded its local model.

NOMA is a promising wireless access technology that achieves higher bandwidth utilization than TDMA and FDMA [22], [23]. In a **NOMA-based FL system**, many selected clients could utilize the same spectrum to simultaneously upload their local models to the BS. The transmission power of the selected clients should be optimized to ensure the local models can be uploaded before the deadline. Without optimizing client selection and transmission power, Sun *et al.* [24] conducted extensive simulations, where 10 clients are randomly selected in each iteration and the transmission power of these clients are the same. The results demonstrated that as compared to the TDMA-based FL system, the NOMA-based FL system can significantly reduce communication latency. Ma *et al.* [25] proposed a transmission power allocation solution in NOMA-based FL systems to maximize the weighted sum data rate of all the selected clients. Assuming all the clients in the system will participate in the model training in each global iteration, Bouzinis *et al.* [26] designed a transmission power allocation method to minimize the latency of a global iteration, while ensuring that the total energy consumption of a client for both computation and communications cannot exceed the maximum available energy of the client in each global iteration. Both works do not consider the deadline for each global iteration and client dynamics, which may lead to the long latency of a global iteration and inefficient resource allocation. In addition, the existing NOMA-based FL solutions only optimize the transmission power allocation by assuming the selected clients are given. To the best of our knowledge, we are the first to jointly optimize client selection and transmission power allocation in NOMA-based FL systems by considering the client dynamics.

III. System Models and Problem Formulation

As shown in Fig. 1, IoT devices in terms of clients communicate with the BS via NOMA-based wireless networks. Let $\mathcal{I}$ be the set of clients in a BS's coverage area. Denote x_i as the binary variable to indicate whether client i is selected to participate in the FL training process (i.e., $x_i = 1$) or not (i.e., $x_i = 0$). Other key notations that are used in the system models and problem formulation are listed in Table I.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3264463

TABLE I: Summary of key notations

Notation	Definition
$\boldsymbol{\mathcal{I}}$	Set of all the clients
$\boldsymbol{\mathcal{J}}$	Set of the selected clients
i	Index of clients in $\boldsymbol{\mathcal{I}}$
j	Index of selected clients in $\boldsymbol{\mathcal{J}}$
$\boldsymbol{\mathcal{D}}_i$	Set of samples for client i to train its local model
t_i^{comp}	Computational latency of client i
t_j^{upload}	Uploading latency of selected client j
τ	Deadline of each global iteration
C_i	Number of CPU cycles for an SGD epoch
f_i	Computational capacity
h_j^t	Channel coefficient of selected client j at time slot t
p_j^t	Transmission power of selected client j at time slot t
r_j^t	Uploading data rate of selected client j at time slot t
N_0	Noise power
B	Available bandwidth
s	Size of a local model
Δt	Length of a time slot

A. Foundation of Federated Learning

The goal of the FL is to derive the vector of parameters, denoted as $\boldsymbol{\omega}$, for a global model in order to minimize the global loss function $\boldsymbol{\mathcal{F}}(\boldsymbol{\omega})$, i.e.,

$$\arg\min_{\boldsymbol{\omega}} \boldsymbol{\mathcal{F}}(\boldsymbol{\omega}) = \arg\min_{\boldsymbol{\omega}} \sum_{i\in\boldsymbol{\mathcal{I}}} \frac{|\boldsymbol{\mathcal{D}}_i|}{|\boldsymbol{\mathcal{D}}|} f_i(\boldsymbol{\omega})\, x_i, \tag{1}$$

where $|\boldsymbol{\mathcal{D}}|$ is the size of the overall training data set, $|\boldsymbol{\mathcal{D}}_i|$ is the size of the training data set at client i (where $\boldsymbol{\mathcal{D}} = \bigcup_{i\in\boldsymbol{\mathcal{I}}} (\boldsymbol{\mathcal{D}}_i \cdot x_i)$), and $f_i(\boldsymbol{\omega})$ is the local loss function of client i over $\boldsymbol{\mathcal{D}}_i$, i.e., $f_i(\boldsymbol{\omega}) = \frac{1}{|\boldsymbol{\mathcal{D}}_i|} \sum_{n\in\boldsymbol{\mathcal{D}}_i} f(\boldsymbol{\omega}, \boldsymbol{a}_{i,n}, b_{i,n})$. Here, $(\boldsymbol{a}_{i,n}, b_{i,n})$ is the input-output pair for the n^{th} data sample in client i's data set, and $f(\boldsymbol{\omega}, \boldsymbol{a}_{i,n}, b_{i,n})$ captures the error of the local model over $(\boldsymbol{a}_{i,n}, b_{i,n})$. FL is to solve Problem (1) in a distributed manner. Each global iteration k comprises four steps.

1) The BS broadcasts the current global model, denoted as $\boldsymbol{\omega}^{(k)}$, to all the selected clients.
2) Each selected client i (i.e., $x_i = 1$) performs local computation on the received model to train its local model over local data set $\boldsymbol{\mathcal{D}}_i$ based on the gradient descent method, i.e., $\boldsymbol{\omega}_i^{(k+1)} = \boldsymbol{\omega}_i^{(k)} - \delta \nabla f_i\left(\boldsymbol{\omega}_i^{(k)}\right)$, where δ indicates the step size or learning rate.
3) After obtaining the local model $\boldsymbol{\omega}_i^{(k+1)}$, client i uploads its local model to the BS.
4) The BS aggregates the local models from the selected clients to update the global model based on, for example, FedAvg [3], i.e., $\boldsymbol{\omega}^{(k+1)} = \frac{\sum_{i\in\boldsymbol{\mathcal{I}}} \boldsymbol{\omega}_i^{(k+1)} x_i}{\sum_{i\in\boldsymbol{\mathcal{I}}} x_i}$.

The FL keeps updating the global model in each iteration until the global model is converged.

B. Computational latency

Assume that stochastic gradient descent (SGD) is used by each client to train its local model, where one SGD epoch comprises one forward propagation and one backward propagation by feeding one local sample into the local model. The latency of client i in training its local model during a global iteration is mainly determined by its computational capacity f_i in terms of CPU cycles per second and the number of SGD epochs/local samples $|\boldsymbol{\mathcal{D}}_i|$ [27]. That is,

$$t_i^{comp} = \frac{C_i |\boldsymbol{\mathcal{D}}_i|}{f_i}, \tag{2}$$

where t_i^{comp} is the computation latency of client i, C_i is the average number of CPU cycles for one epoch of SGD, and $\frac{C_i}{f_i}$ indicates the latency of one epoch of SGD.

C. Communication latency in NOMA-based FL

The latency of a selected client in uploading its local model depends on the uploading date rate of the selected client. Since NOMA is applied, the uploading date rate of a selected client is determined by the interference from other selected clients, who are currently uploading their local models. Specifically, let $\boldsymbol{\mathcal{J}}$ be the set of selected clients under the BS's coverage area, i.e., $\boldsymbol{\mathcal{J}} = \{i \in \boldsymbol{\mathcal{I}} \mid x_i = 1\}$. Denote j as the index of these selected clients. Note that the indices i and j represent different physical meanings, where i is the index of the clients in the BS's coverage area, and j is the index of the clients that are selected to participate in the FL model training during the current global iteration. Even though i and j may point to the same client, they could have different values because the selected clients are re-ordered based on their received power at the BS. The reason for re-ordering is because, in NOMA, a client with higher received power will be decoded by the BS before a client with lower received power. For each selected client j (where $j \in \boldsymbol{\mathcal{J}}$), it would start to upload its local model once it finishes local model training at t_j^{comp}, where t_j^{comp} is estimated based on Eq. (2). Denote y_j^t as a binary variable to indicate if selected client j has already finished its model uploading or not at the beginning of time slot t, i.e.,

$$y_j^t = \begin{cases} 1, & \text{if } \sum_{t'=t_j^{comp}}^{t-1} l_j^{t'} < s, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where s is the size of the local model and $l_j^{t'}$ indicates the amount of data in its local model that has been transmitted from selected client j to the BS in time slot t', i.e.,

$$l_j^{t'} = \Delta t \times r_j^{t'}. \tag{4}$$

Here, Δt is the duration of a time slot, $r_j^{t'}$ is the uploading data rate of selected client j in time slot t'. Plugging Eq. (4) into Eq. (3), we have

$$y_j^t = \begin{cases} 1, & \text{if } \sum_{t'=t_j^{comp}}^{t-1} r_j^{t'} < \frac{s}{\Delta t}. \\ 0, & \text{otherwise}. \end{cases} \tag{5}$$

As we mentioned before, to calculate $r_j^{t'}$ in NOMA-based FL, we sort the selected clients based on their received power at the BS in time slot t'. That is, if a client has a lower received power at the BS, the client has a lower index, i.e.,

$$(j - j') \left(\left| h_j^{t'} \right|^2 p_j^{t'} - \left| h_{j'}^{t'} \right|^2 p_{j'}^{t'} \right) \geq 0, \tag{6}$$

where $p_{j'}^{t'}$ and $h_{j'}^{t'}$ are the transmission power and channel response of clients j' in time slot t', respectively, and j and j' are both the indices of the clients in $\boldsymbol{\mathcal{J}}$. Then, $r_j^{t'}$ can be calculated by

$$r_j^{t'} = Blog_2\left(1+\frac{p_j^{t'}\left|h_j^{t'}\right|^2}{N_0+\sum\limits_{j'=1}^{j-1} y_{j'}^{t'}p_{j'}^{t'}\left|h_{j'}^{t'}\right|^2}\right), \tag{7}$$

where B is the total amount of bandwidth at the BS, $p_j^{t'}$ is the transmission power of selected client j, $\left|h_j^{t'}\right|^2$ is the channel gain from selected client j to the BS, N_0 is the noisy power level, and $\sum\limits_{j'=1}^{j-1} y_{j'}^{t'}p_{j'}^{t'}\left|h_{j'}^{t'}\right|^2$ is the interference from other selected clients, who have the lower indices than client j.

Accordingly, the communication latency of selected client j in uploading its local model is

$$t_j^{upload} = \Delta t \sum_{t=t_j^{comp}}^{\tau} y_j^t, \tag{8}$$

where τ is the deadline of each global iteration.

D. Problem formulation

Based on the defined latency models, we formulate the problem of jointly optimizing client selection and transmission power of the selected clients in NOMA-based FL as follows.

$$\boldsymbol{P0}: \max_{\boldsymbol{x},\boldsymbol{p^t}} \sum_{i\in\boldsymbol{\mathcal{I}}} x_i, \tag{9}$$

$$\text{s.t. } x_i \in \{0,1\},\ \forall i \in \boldsymbol{\mathcal{I}}, \tag{10}$$

$$\boldsymbol{\mathcal{J}} = \{i \in \boldsymbol{\mathcal{I}} \mid x_i = 1\}, \tag{11}$$

$$0 \le p_j^t \le p_j^{max},\ \forall j \in \boldsymbol{\mathcal{J}}, \forall t, \tag{12}$$

$$(j-j')\left(\left|h_j^t\right|^2 p_j^t - \left|h_{j'}^t\right|^2 p_{j'}^t\right) \ge 0,\ \forall j,j' \in \boldsymbol{\mathcal{J}}, \forall t \tag{13}$$

$$y_j^t = \begin{cases} 1, & \text{if } \sum\limits_{t'=t_j^{comp}}^{t-1} r_j^{t'} < \frac{s}{\Delta t}, \\ 0, & \text{otherwise,} \end{cases} \quad \forall j \in \boldsymbol{\mathcal{J}}, \tag{14}$$

$$\Delta t \sum_{t=t_j^{comp}}^{\tau} r_j^t y_j^t \ge s,\ \forall j \in \boldsymbol{\mathcal{J}}. \tag{15}$$

The objective, i.e., Eq. (9), is to maximize the number of selected clients in a global iteration. Constraint (10) indicates x_i to be a binary variable. Constraint (11) defines the set of selected clients. Constraint (12) implies that the transmission power of a client is no larger than its maximum transmission power. Constraint (13) indicates the index order of the selected clients in $\boldsymbol{\mathcal{J}}$. Constraint (14) defines y_j^t to be a binary variable, which is used to indicate if selected client j has already finished its model uploading or not at the beginning of time slot t. Constraint (15) indicates that selected client j has to upload its local model before the deadline. Note that $\boldsymbol{P0}$ is non-trivial to be solved because 1) the system has to predict channel conditions h_j^t of all the clients before the deadline in order to verify if Constraint (15) can be satisfied. However, accurately predicting the channel conditions of the clients is difficult even if it is possible; 2) $\boldsymbol{P0}$ is an NP-hard problem[2] with a high-dimensional variable space.

IV. Deep Reinforcement Learning based Client Selection in Non-orthogonal Multiple access based Federated Learning

We apply Markov Decision Process (MDP) to reformulate the $\boldsymbol{P0}$. As we mentioned before, client selection and power allocation are optimized in different levels of time granularity, and so we will build two MDPs for client selection $(\boldsymbol{\mathcal{S}}_{cs}, \boldsymbol{\mathcal{A}}_{cs}, \boldsymbol{\mathcal{F}}_{cs}, \boldsymbol{\mathcal{R}}_{cs})$ and power allocation $(\boldsymbol{\mathcal{S}}_{pa}, \boldsymbol{\mathcal{A}}_{pa}, \boldsymbol{\mathcal{F}}_{pa}, \boldsymbol{\mathcal{R}}_{pa})$, respectively. In the client selection MDP, the FL server iteratively evaluates if a client should be selected or not in each round. The transmission power allocation MDP then optimizes the transmission power of the clients, which have been selected by client selection during this round, in each time slot.

A. MDP for client selection

1) State: In round k, the FL server evaluates if client k should be selected (i.e., $x_k = 1$) or not (i.e., $x_k = 0$), where $k \in \boldsymbol{\mathcal{I}}$. Thus, the states of round k for client selection is defined as $\boldsymbol{s}_{cs}^k = \left[\mathbf{H}_{cs}, \mathbf{T}, \boldsymbol{x}^k\right]$, where $\boldsymbol{s}_{cs}^k \in \boldsymbol{\mathcal{S}}_{cs}$. Here,

- $\mathbf{H}_{cs} = \left\{|h_1|^2, |h_2|^2, ..., \left|h_{|\boldsymbol{\mathcal{I}}|}\right|^2\right\}$ is the set of the channel gain of all the clients in $\boldsymbol{\mathcal{I}}$.
- $\mathbf{T} = \left\{t_1^{comp}, t_2^{comp}, ..., t_{|\boldsymbol{\mathcal{I}}|}^{comp}\right\}$ is the set of the computational latency of all the clients in $\boldsymbol{\mathcal{I}}$.
- $\boldsymbol{x}^k = \left\{x_1^k, x_2^k, ..., x_{|\boldsymbol{\mathcal{I}}|}^k\right\}$ is the set of the selection status for all the clients in round k.

2) Action: The action of the FL system for client selection in round k, denoted as a_{cs}^k (where $a_{cs}^k \in \boldsymbol{\mathcal{A}}_{cs}$), is $a_{cs}^k = x_k$.

3) Reward: The reward function of client selection in round k, denoted as ρ_{cs}^k (where $\rho_{cs}^k \in \boldsymbol{\mathcal{R}}_{cs}$), is the sum of the rewards for all the clients, i.e., $\rho_{cs}^k = \sum\limits_{i=1}^{|\boldsymbol{\mathcal{I}}|} \rho_{cs}^k(i)$, where $\rho_{cs}^k(i)$ is the reward of client i in round k, which is defined as

$$\rho_{cs}^k(i) = \begin{cases} 1, & \text{if } x_i^k = 1 \,\&\, \text{Constraint (15) is met,} \\ 0, & \text{else if } x_i^k = 0, \\ -10, & \text{otherwise.} \end{cases} \tag{16}$$

Basically, Eq. (16) implies if client i is selected and can finally upload its local model before the deadline (i.e., Eq. (15) is met), the reward $\rho_{cs}^k(i) = 1$; if client i is not selected, then its reward is 0; if client i is selected but cannot upload its local model before the deadline, then it is a bad client selection, and so the reward $\rho_{cs}^k(i) = -10$. Note that the reward of client selection is also affected by transmission power allocation.

[2]Any mixed-integer nonlinear programming (MINLP) problem, in general, is NP-hard [28] because it can be reducible into component redundancy allocation problem, which has been proved to be NP-hard [29]. Based on this conclusion, we can derive $\boldsymbol{P0}$ is NP-hard if it is MINLP. Apparently, $\boldsymbol{P0}$ is MINLP since 1) $\boldsymbol{x}$ is the set of binary variables and $\boldsymbol{p^t}$ is the set of continuous variables, and 2) Constraints (14) and (15) are nonlinear with respect to $\boldsymbol{x}$ and $\boldsymbol{p^t}$.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3264463

B. MDP for transmission power allocation

1) State: Each global iteration is further divided into a number of time slots with equal length, and NOMA is applied to enable the selected clients to share the frequency spectrum in uploading their local models. In time slot t, the FL server would iteratively pick a client and assign the corresponding transmission power. Thus, the state of the FL system for transmission power allocation is defined as $\boldsymbol{s}_{pa}^t(j) = \left[\left(|h_j|^2\right)^t, m_j^t\right]$, where $\left(|h_j|^2\right)^t$ is the channel gain of selected client j at time slot t and m_j^t is the remaining bits for selected client j's local model at time slot t, i.e.,

$$m_j^t = \begin{cases} s, & \text{if } t-1 < t_j^{comp}, \\ \left[s - \sum_{t'=t_j^{comp}}^{t-1} \Delta t \times r_j^{t'}\right]^+, & \text{otherwise.} \end{cases} \tag{17}$$

Here, if selected client j is still training and not starting to upload its local model (i.e., if $t-1 < t_j^{comp}$), the number of remaining bits for selected client j always equal to the size of the model s. If selected client j starts uploading its local model, the number of remaining bits equals the size of the model s minus the number of bits that have been transmitted.

2) Action: The action set of the FL system for transmission power allocation at time slot t, denoted as $\boldsymbol{a}_{pa}^t$ (where $\boldsymbol{a}_{pa}^t \in \boldsymbol{\mathcal{A}}_{pa}$), is defined as the set of transmission power for all the selected clients, i.e., $\boldsymbol{a}_{pa}^t = \boldsymbol{p}^t = \left\{p_1^t, p_2^t, ..., p_{|\boldsymbol{\mathcal{I}}|}^t\right\}$. Note that, as we mentioned before, the transmission power of each selected client p_j^t is iteratively selected. Also, by considering the real implementation, we divide the transmission power into 100 levels starting from 0 to p^{max}, and a selected client can only pick its transmission power p_j^t from the predefined levels.

3) Reward: The reward function of transmission power allocation at time slot t, denoted as ρ_{pa}^t (where $\rho_{pa}^t \in \boldsymbol{\mathcal{R}}_{pa}$), is the sum of the rewards for all the selected clients, i.e., $\rho_{pa}^t = \sum_{j=1}^{|\boldsymbol{\mathcal{I}}|} \rho_{pa}^t(j)$, where $\rho_{pa}^t(j)$ is the reward of selected client j at time slot t, which is defined as follows.

$$\rho_{pa}^t(j) = \begin{cases} 1, & \text{if } \Delta t \sum_{t'=t_j^{comp}}^{t} r_j^{t'} y_j^{t'} \geq s \ \&\ t \leq \tau, \\ \frac{\Delta t r_j^t}{s}, & \text{else if } \Delta t \sum_{t'=t_j^{comp}}^{t} r_j^{t'} y_j^{t'} < s \ \&\ t < \tau, \\ -1, & \text{otherwise.} \end{cases} \tag{18}$$

Here, Eq. (18) means that if selected client j has already uploaded its local model before the deadline, then its reward at time slot t is 1. If selected client j has not uploaded its local model and the current time slot is not the deadline, then its reward at time slot t is the currently transmitted bits, i.e., $\Delta t \times r_j^t / s$. If selected client j cannot upload its local model before the deadline, then its reward is -1.

C. Deep Reinforcement lEarning based client selection in non-orthogonAl Multiple access based Federated Learning (DREAM-FL)

We design the DREAM-FL algorithm, which comprises two Advantage Actor Critic (A2C) [30] to solve the client selection and transmission power allocation problems, respectively. A2C is a DRL method that combines policy-based and value-based reinforcement learning. In A2C, there are actor and critic neural networks. The actor network provides the stochastic policy to choose the corresponding action(s) such that the expected cumulative reward is maximized. Denote $J^{cs}(\theta^{cs})$ and $J^{pa}(\theta^{pa})$ as the cumulative reward for client selection and transmission power allocation, respectively. We have

$$\begin{cases} J_{cs}^{actor}(\theta_{cs}) = \mathbb{E}\left[\sum_{k=1}^{|\boldsymbol{\mathcal{I}}|} (\gamma_{cs})^k \rho_{cs}^k\right], \\ J_{pa}^{actor}(\theta_{pa}) = \mathbb{E}\left[\sum_{t=1}^{\tau} (\gamma_{pa})^t \rho_{pa}^t\right], \end{cases} \tag{19}$$

where θ_{cs} and θ_{pa} are the parameters of the actor networks for client selection and transmission power allocation, respectively, and γ_{cs} and γ_{ap} are the related discount factors, where $\gamma_{cs}, \gamma_{ap} \in [0,1]$. Hence, the gradients of θ^{cs} and θ^{pa} are

$$\begin{cases} \nabla_{\theta_{cs}} J_{cs}^{actor}(\theta_{cs}) = \mathbb{E}[\nabla_{\theta_{cs}} \log \pi_{\theta_{cs}}(a_{cs}^k | \boldsymbol{s}_{cs}^k) A(\boldsymbol{s}_{cs}^k, a_{cs}^k)], \\ \nabla_{\theta_{pa}} J_{pa}^{actor}(\theta_{pa}) = \mathbb{E}[\nabla_{\theta_{pa}} \log \pi_{\theta_{pa}}(\boldsymbol{a}_{pa}^t | \boldsymbol{s}_{pa}^t) A(\boldsymbol{s}_{pa}^t, \boldsymbol{a}_{pa}^t)], \end{cases} \tag{20}$$

where $\pi_{\theta_{cs}}(a_{cs}^k | \boldsymbol{s}_{cs}^k)$ and $\pi_{\theta_{pa}}(\boldsymbol{a}_{pa}^t | \boldsymbol{s}_{pa}^t)$ are the stochastic policies of client selection and transmission power allocation, respectively, and $A(\boldsymbol{s}_{cs}^k, a_{cs}^k)$ and $A(\boldsymbol{s}_{pa}^t, \boldsymbol{a}_{pa}^t)$ are the Advantage functions of client selection and transmission power allocation, respectively. We apply the one-step temporal difference (TD) error to estimate the Advantage functions, i.e.,

$$\begin{cases} A(\boldsymbol{s}_{cs}^k, a_{cs}^k) = \rho_{cs}^k + \gamma_{cs} V_{\zeta_{cs}}(\boldsymbol{s}_{cs}^{k+1}) - V_{\zeta_{cs}}(\boldsymbol{s}_{cs}^k). \\ A(\boldsymbol{s}_{pa}^t, \boldsymbol{a}_{pa}^t) = \rho_{pa}^t + \gamma_{pa} V_{\zeta_{pa}}(\boldsymbol{s}_{pa}^{t+1}) - V_{\zeta_{pa}}(\boldsymbol{s}_{pa}^t). \end{cases} \tag{21}$$

Here, $V_{\zeta_{cs}}(\boldsymbol{s}_{cs}^k)$ and $V_{\zeta_{cs}}(\boldsymbol{s}_{cs}^{k+1})$ are the state-values in rounds k and $k+1$, respectively, which are estimated by the critic network for client selection, and ζ_{cs} is the parameter of the critic network. Similarly, $V_{\zeta_{pa}}(\boldsymbol{s}_{pa}^t)$ and $V_{\zeta_{pa}}(\boldsymbol{s}_{pa}^{t+1})$ are the state-values in time slots t and $t+1$, respectively, which are estimated by the critic network for transmission power allocation, and ζ_{pa} is the parameter of the critic network. Here, the two critic networks are used to evaluate the actions (i.e., a_{cs}^k and $\boldsymbol{a}_{pa}^t$) taken by the actor networks based on the Advantage values, thus improving the policies. The objectives of the critic networks are to minimize the one-step TD error for client selection and power allocation, respectively i.e.,

$$\begin{cases} J_{cs}^{critic}(\zeta_{cs}) = \left(\rho_{cs}^k + \gamma_{cs} V_{\zeta_{cs}}(\boldsymbol{s}_{cs}^{k+1}) - V_{\zeta_{cs}}(\boldsymbol{s}_{cs}^k)\right)^2. \\ J_{pa}^{critic}(\zeta_{pa}) = \left(\rho_{pa}^t + \gamma_{pa} V_{\zeta_{pa}}(\boldsymbol{s}_{pa}^{t+1}) - V_{\zeta_{pa}}(\boldsymbol{s}_{pa}^t)\right)^2. \end{cases} \tag{22}$$

Note that in each round k, the parameters of actor and critic networks for client selection, i.e., θ_{cs} and ζ_{cs}, would be updated via gradient descend, i.e.,

$$\begin{cases} \theta_{cs} := \theta_{cs} - \varsigma_{cs}^{actor} \nabla_{\theta_{cs}} J_{cs}^{actor}(\theta_{cs}), \\ \zeta_{cs} := \zeta_{cs} - \varsigma_{cs}^{critic} \nabla_{\zeta_{cs}} J_{cs}^{critic}(\zeta_{cs}), \end{cases} \tag{23}$$

where ς_{cs}^{actor} and ς_{cs}^{critic} are the learning rates of the actor and critic networks for client selection, respectively. Similarly, in each time slot t, the parameters of actor and critic networks for transmission power allocation, i.e., θ_{pa} and ζ_{pa}, would be updated via gradient descend, i.e.,

$$\begin{cases} \theta_{pa} := \theta_{pa} - \varsigma_{pa}^{actor} \nabla_{\theta_{pa}} J_{pa}^{actor}(\theta_{pa}), \\ \zeta_{pa} := \zeta_{pa} - \varsigma_{pa}^{critic} \nabla_{\zeta_{pa}} J_{pa}^{critic}(\zeta_{pa}), \end{cases} \tag{24}$$

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3264463

where ς_{pa}^{actor} and ς_{pa}^{critic} are the learning rates of the actor and critic networks for transmission power allocation, respectively.

Fig. 3 illustrates our designed A2C networks for client selection and transmission power allocation. Specifically, we incorporate the actor and critic networks into one deep neural network (DNN), i.e., $\theta_{cs} = \zeta_{cs}$ and $\theta_{pa} = \zeta_{pa}$. The client selection A2C network comprises an input layer taking the input states $\boldsymbol{s}_{cs}^k = \left[\mathbf{H}_{cs}, \mathbf{T}, \boldsymbol{x}^k\right]$, three hidden layers with 20, 80, and 120 neurons, respectively (and with sigmoid activation functions in each layer), and an output layer that generates the action of client selection in the current round (i.e., x_k) as well as the state value $V_{\zeta_{cs}}(\boldsymbol{s}_{cs}^k)$. Similarly, the transmission power allocation A2C network comprises an input layer taking the input states $\boldsymbol{s}_{pa}^t(j) = \left[\left(|h_j|^2\right)^t, m_j^t\right]$ (which are generated based on the action derived by the client selection A2C network), three hidden layers with 20, 80, and 120 neurons, respectively (and with sigmoid activation functions in each layer), and an output layer that generates the actions in terms of p_j^t and the state value $V_{\zeta_{pa}}(\boldsymbol{s}_{pa}^t)(j)$.

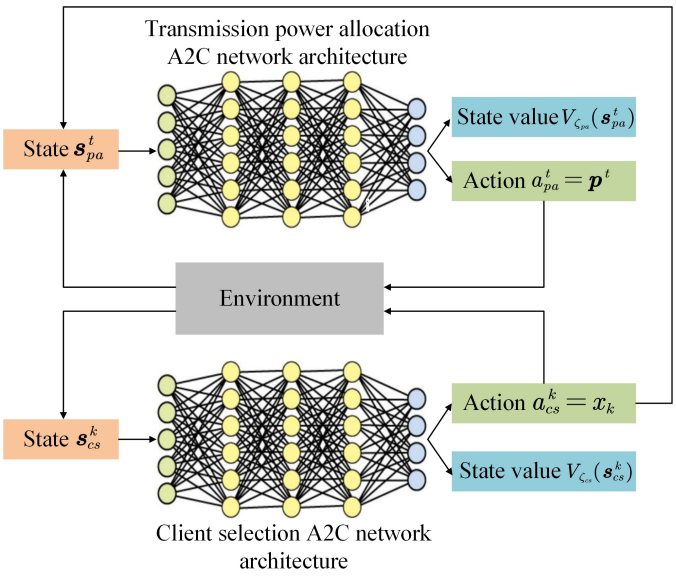


Fig. 3: The A2C networks for client selection and transmission power allocation.

Algorithm 1 summarizes the DREAM-FL algorithm, which is the process of training the A2C networks for client selection and transmission power allocation. Specifically, in each global iteration, we first initialize the actions $\boldsymbol{x} = \mathbf{0}$ and $\boldsymbol{p^t} = \mathbf{0}$ in Step 2. Steps 7-11 indicate that, in each round k, the client selection A2C network would apply the current policy to generate the action based on the current state $\boldsymbol{s}_{cs}^k$. The action implies client k should be selected (i.e., $x_k = 1$) or not (i.e., $x_k = 0$). Note that there is at most one client would be selected in each round. The state-action pair $\left\langle \boldsymbol{s}_{cs}^k, \boldsymbol{a}_{cs}^k \right\rangle$ in replay buffer $\boldsymbol{\mathcal{B}}_{cs}$. In Steps 14-18, based on the selected clients from the transmission power allocation A2C network, the transmission power allocation A2C network applies the current policy to iteratively generate the action of each selected client $\boldsymbol{a}_{pa}^t(j)$, i.e., the transmission power of selected client j, at time slot t. The state-action pair $\left\langle \boldsymbol{s}_{pa}^t(j), \boldsymbol{a}_{pa}^t(j) \right\rangle$ is stored in $\boldsymbol{\mathcal{B}}_{pa}$.

Algorithm 1: DREAM-FL algorithm

1 Initialize the discount factors γ_{cs} and γ_{pa} as well as the learning rates ς_{cs}^{actor}, ς_{cs}^{critic}, ς_{pa}^{actor}, and ς_{cs}^{actor}.
2 Initialize $\boldsymbol{x} = \mathbf{0}$ and $\boldsymbol{p^t} = \mathbf{0}$.
3 **for** *each global iteration* **do**
4 Uniformly distribute the clients in the BS's coverage area;
5 Initialize replay buffer $\boldsymbol{\mathcal{B}}_{cs}$;
6 **for** *each round* k *(where* $1 \leq k \leq |\boldsymbol{\mathcal{I}}|$*)* **do**
7 **for** *each user* i **do**
8 Obtain the current state $\boldsymbol{s}_{cs}^k = \left[\mathbf{H}_{cs}, \mathbf{T}, \boldsymbol{x}^k\right]$;
9 Input $\boldsymbol{s}_{cs}^k$ to the client selection A2C network to derive the action $a_{cs}^k = x_k$;
10 Store the state-action pair $\left\langle \boldsymbol{s}_{cs}^k, \boldsymbol{a}_{cs}^k \right\rangle$ in $\boldsymbol{\mathcal{B}}_{cs}$;
11 **end**
12 **for** *each time slot* t *(where* $1 \leq t \leq \tau$*)* **do**
13 Initialize replay buffer $\boldsymbol{\mathcal{B}}_{pa}$;
14 **for** *each selected client* j **do**
15 Obtain the current state $\boldsymbol{s}_{pa}^t(j) = \left[\left(|h_j|^2\right)^t, m_j^t\right]$;
16 Input $\boldsymbol{s}_{pa}^t(j)$ to the transmission power allocation A2C network to derive the actions $a_{pa}^t = p_j^t$;
17 Store the state-action pair $\left\langle \boldsymbol{s}_{pa}^t(j), \boldsymbol{a}_{pa}^t(j) \right\rangle$ in $\boldsymbol{\mathcal{B}}_{pa}$;
18 **end**
19 Calculate reward ρ_{pa}^t based on Eq. (18) and save the reward in $\boldsymbol{\mathcal{B}}_{pa}$;
20 **end**
21 Calculate reward ρ_{cs}^k based on Eq. (16) and save the reward in $\boldsymbol{\mathcal{B}}_{cs}$;
22 Update the transmission power allocation A2C network based on $\boldsymbol{\mathcal{B}}_{pa}$ and Eq. (24);
23 **end**
24 Update the client selection A2C network based on $\boldsymbol{\mathcal{B}}_{cs}$ and Eq. (23);
25 **end**

Once the transmission powers of all the selected clients are derived by the transmission power allocation A2C network, the corresponding reward ρ_{pa}^t can be calculated based on Eq. (18) in Step 19. In Steps 21-22, we calculate the reward ρ_{cs}^k of client selection in round k based on Eq. (16) and update the parameters in the transmission power allocation A2C network based on Eq. (24) at time slot t. The same process in Steps 7-22 would repeat for each round until $k = |\boldsymbol{\mathcal{I}}|$, meaning that all the clients have to be evaluated by the client selection A2C network. Then, in Step 24, we can update the parameters of the client selection A2C network based on Eq. (23) by utilizing the transitions in $\boldsymbol{\mathcal{B}}_{cs}$ during this global iteration.

V. Simulations

We apply Tensorflow [31] to build and train the A2C networks in the DREAM-FL method. The simulation environment

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3264463

is written in C++ and interacts with the A2C networks in Tensorflow based on ZMQ [32], which achieves an asynchronous messaging library to achieve communications between two applications (i.e., the simulated environment and the A2C networks). In the simulated environment, we assume that there are 50 clients in the BS's coverage area. The locations of these clients are uniformly distributed in the BS's coverage area, and we use the 3GPP macro cell propagation model [33], i.e., $128.1 + 37.6 \log_{10} d$, to estimate the path loss between a client and the BS, where d is the distance between a client and the BS in kilometers. Also, the computational capacity f_i of each client is generated based on a uniform distribution, i.e., $f_i \sim U(1.5, 3)$ GHz. Other simulation parameters and hyperparameters are listed in Table II.

TABLE II: Simulation parameters and hyperparamters

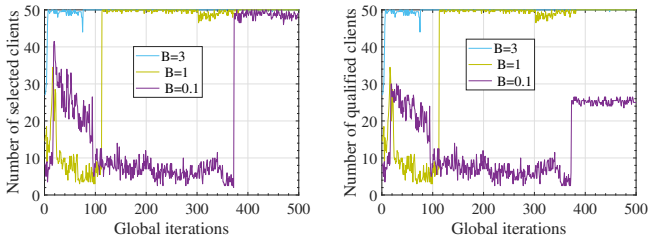
Parameter	Value
Number of CPU cycles (C_i)	3×10^4 cycles/sample
Computation capacity (f_i)	$U(1.5, 3)$ GHz
Maximum transmission power (p^{max})	1 Watt [34], [35]
Noise power (N_0)	-114 dBm
Size of local model (s)	100 Kbits
Length of a time slot (Δt)	0.1 second
Discount factor (γ_{cs}/γ_{pa})	0.99 [36]
Learning rate (ς_{cs}^{actor}/ς_{cs}^{critic}/ς_{pa}^{actor}/ς_{pa}^{critic})	0.001 [37]

A. Convergence analysis

Assume that the number of training data samples for each client is 40 (i.e., $|\boldsymbol{\mathcal{D}}_i| = 40$) and the deadline is 2 seconds (i.e., $\tau = 2$). Figs. 5a and 5b show the learning curves of the A2C networks for client selection and transmission power allocation, respectively, under different bandwidth settings. In Fig. 5, we can see that both A2C networks quickly converge around 100 global iterations if the bandwidth is sufficient (i.e., B =3 and 1 MHz) to support almost all 50 clients to upload their local models before the deadline. However, once the bandwidth is not sufficient (i.e., B =0.1 MHz), the A2C networks require more global iterations ($>$ 350 global iterations) to converge. In Fig 5a, we can find that the reward of the A2C network for client selection is finally converged close to -200 when $B = 0.1$ MHz. This can be explained by Fig. 4, where Fig. 4a indicates the number of clients selected by the A2C network for client selection during the training process, and Fig. 4b shows the number of qualified clients, i.e., the selected clients who can successfully upload their local models before the deadline. In Fig. 4, we can find that when the training curves are converged, the client selection network selects an average of 49 clients; however, only an average of 26 clients can upload their models before the deadline, and the rest of the 23 clients, who cannot upload their models before the deadline, generates -10 reward according to Eq. (16).

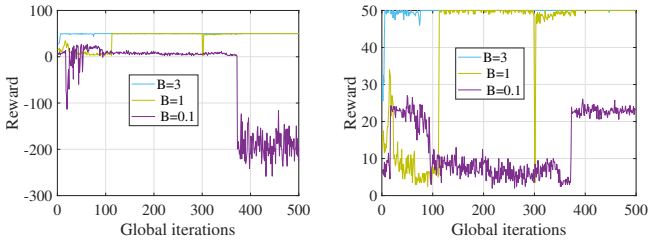
B. Performance evaluation

We further compare DREAM-FL with the other two baseline methods, i.e., Latency awarE pARticipant selectioN (LEARN) [18] and Frequency division duplex-based deadline Aware paRticipant selectioN (FARN) [15]–[17]. All three



(a) Number of selected clients. (b) Number of qualified clients.

Fig. 4: Performance of DREAM-FL over 500 global iterations, where a) measures the number of selected clients, and b) measures the number of qualified local clients.



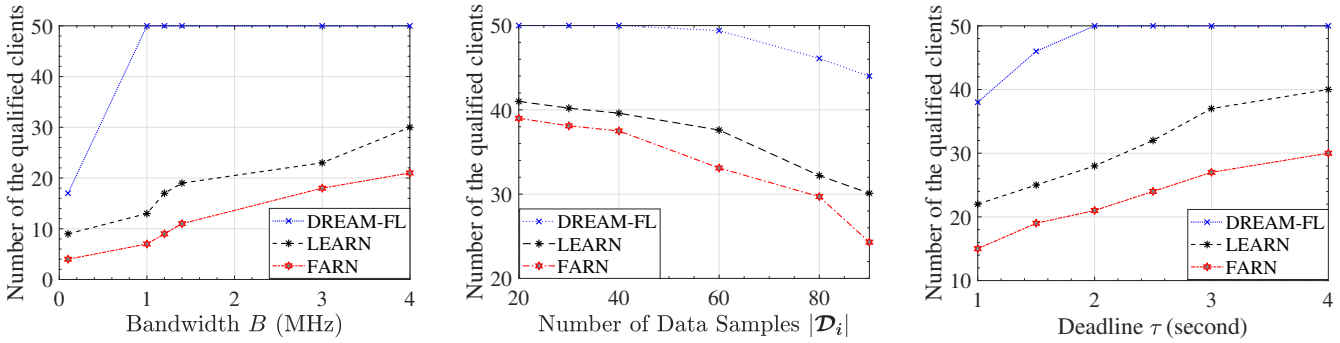
(a) A2C network for client selection. (b) A2C network for transmission power allocation.

Fig. 5: The learning curves of DREAM-FL over 500 global iterations, where a) measures the cumulative rewards for the client selection network, and b) measures the cumulative rewards for the transmission power network.

algorithms have the same goal, i.e., to maximize the number of selected clients in each global iteration, while ensuring the selected clients to upload their local models before the deadline. However, different algorithms apply different wireless access technologies. LEARN applies TDMA to enable different clients to upload their local models to the BS in different time slots. If a client uploads its local model during the current time slot, it utilizes the whole spectrum, and so other clients have to wait until the spectrum is available. So, it is critical to estimate the time to wait for the spectrum to be available for a client. FARN applies FDMA to upload local models of the selected clients to the BS. That is, sufficient bandwidth would be reserved for each selected client no matter if it hasn't started to upload its local model yet or has already uploaded its local models.

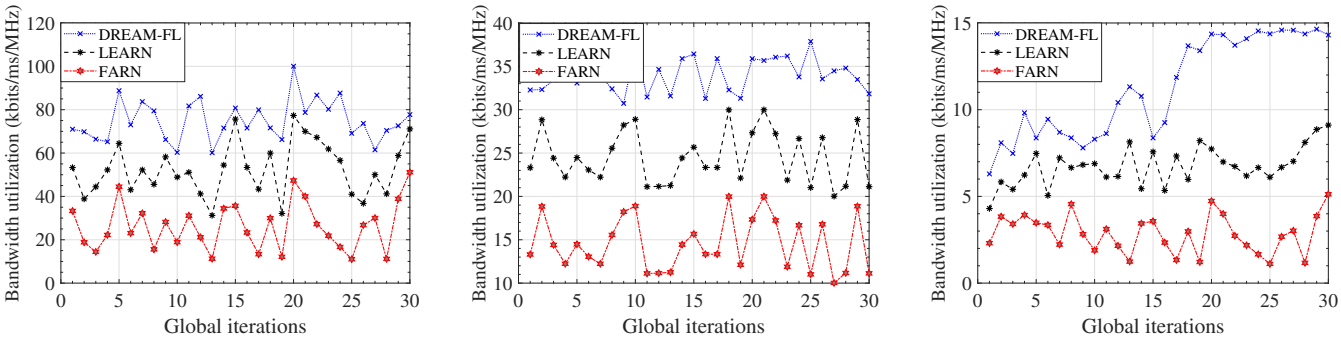
1) Number of the qualified clients versus bandwidth: Assume that $|\boldsymbol{\mathcal{D}}_i| = 40$ and $\tau = 2$ seconds. Fig. 6a shows the number of qualified clients for different algorithms by changing the amount of available bandwidth B. Varying B can affect the uploading latency of all the clients. The DREAM-FL algorithm performs better than LEARN and FARN to generate the most qualified clients because DREAM-FL can more efficiently utilize bandwidth to upload more client models before the deadline. To prove this conclusion, we calculate the average bandwidth utilization of a global iteration generated by different algorithms. Figs 7a, 7b, and 7c show the average bandwidth utilization over 30 global iterations when B = 0.1, 1, and 3 MHz, respectively. DREAM-FL

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3264463



(a) Performance of the algorithms over B. (b) Performance of the algorithms over $|\mathcal{D}_i|$. (c) Performance of the algorithms over τ.

Fig. 6: The number of qualified clients for different algorithms by varying (a) available bandwidth B, (b) the number of training samples $|\mathcal{D}_i|$, and (c) deadline τ.



(a) B=0.1 MHz. (b) B=1 MHz. (c) B=3 MHz.

Fig. 7: Bandwidth utilization of different algorithm when a) B=0.1 MHz, b) B=1 MHz, and c) B=3 MHz.

always generates higher average bandwidth utilization than LEARN and FARN. For instance, when B = 0.1 MHz, the average bandwidth utilization among 30 global iterations for DREAM-FL, LEARN, and FARN are around 80, 60, and 30 kbits/ms/MHz, respectively. Moreover, the average bandwidth utilization reduces as B increases. This is because the clients with better channel conditions are preferred to be selected than the clients with worse channel conditions to achieve higher average bandwidth utilization. As B increases, more clients with worse channel conditions are selected to reduce the average bandwidth utilization. Note that, when $B \geq 1$ MHz, as shown in Fig. 6a, all the clients are selected by DREAM-FL, meaning that the traffic load in terms of the number of selected/qualified clients no longer increases as B increases. As a result, the average bandwidth utilization for DREAM-FL significantly reduces as B increases.

2) Number of the qualified clients versus the number of the training data samples: Assume that $B = 1$ MHz and $\tau = 2$ seconds. Fig. 6b shows the number of qualified clients for different algorithms by changing the number of training samples $|\mathcal{D}_i|$. Note that having a larger $|\mathcal{D}_i|$ means that each client would have a longer time to compute its local model and a shorter time to upload its local model in a global iteration. DREAM-FL can adapt to the change by retraining the A2C networks. From the figure, we can see that DREAM-FL can always generate the most qualified clients in a global iteration. For example, when $|\mathcal{D}_i| = 40$, the number of qualified clients for DREAM-FL, LEARN, and FARN are 50, 40, and 37, respectively. The results can also be explained by the high average bandwidth utilization incurred by DREAM-FL. That is, even though clients have less time to upload their local models before the deadline as $|\mathcal{D}_i|$ increases, DREAM-FL can still consider the client dynamics to generate higher average bandwidth utilization than LEARN and FARN by optimizing the transmission power in each time slot. As a result, DREAM-FL incurs more qualified clients than LEARN and FARN.

3) Number of the qualified clients versus the number of the deadline: Assume that $B = 1$ MHz and $|\mathcal{D}_i| = 40$. Fig. 6c shows the number of qualified clients for different algorithms by changing the deadline τ. The results also demonstrate that DREAM-FL can efficiently and adaptively optimize the transmission power of the clients based on the changes of the deadline to maximize the number of qualified clients.

4) Model accuracy evaluation: To evaluate how the three algorithms affect the accuracy of a global model, we train a convolutional neural network (CNN) network over CIFAR-10, which is a benchmark dataset containing 10 images classes, each of which has 5,000 images for training and 1,000 images for testing. For each image class, we distribute 5,000 training images to $|\mathcal{I}|$ clients, where $|\mathcal{I}| = 50$, according to a Dirichlet distribution, whose probability density function is

$$f(\eta_1, \eta_2, \ldots, \eta_{|\mathcal{I}|}; \beta) = \frac{\Gamma(\beta \times |\mathcal{I}|)}{\Gamma(\beta)^{|\mathcal{I}|}} \prod_{i=1}^{|\mathcal{I}|} \eta_i^{\beta-1}, \qquad (25)$$

where η_i implies the probability of assigning an image to client i, $\Gamma()$ is the gamma function, and β is the concentration parameter to adjust the variance of the number of images (from the same class) among clients. For example, a larger β indicates a more balanced image partition among the clients, i.e., the number of images with the same class among different clients has a lower variance. A smaller β, on the other hand, implies less balanced image partition among the clients, i.e., the number of images with the same class among different classes has a higher variance. The structure of a CNN model to be trained comprises four 3x3 convolution layers (where the first layer has 32 channels, and each of the following three layers has 64 channels. Also, only the first two layers are followed with 2×2 max pooling), followed with a dropout layer with the rate of 75%, a fully connected 256 units ReLu layer, and a 10 units softmax output layer. There are total of 1,144,650 parameters in this CNN model.

Assume that the amount of bandwidth $B = 1$ MHz, the size of data samples for each client $|\boldsymbol{\mathcal{D}}_i| = 40$, and the deadline $\tau = 1$ second. Different algorithms would select different numbers of clients to participate in the model training in each global iteration, and Figs. 8a, 8b, and 8c show how the model accuracy changes over the global iterations by applying different algorithms. It is easy to observe that, when $\beta = 10$ in Fig 8c, DREAM-FL has higher model test accuracy (~ 0.7) than LEARN (< 0.6) and FARN (< 0.55) after 3,000 global iterations. Also, DREAM-FL has a faster model convergence rate than LEARN and FARN. The reason of DREAM-FL having higher model accuracy and a faster convergence rate is because DREAM-FL has more qualified clients than LEARN and FARN. As shown in Fig. 6c, when $B = 1$, $|\boldsymbol{\mathcal{D}}_i| = 40$, and $\tau = 1$, the average number of the qualified clients for DREAM-FL, LEARN, and FARN are 38, 22, and 13, respectively. The results demonstrate the rationale of maximizing the number of selected clients as the objective of the problem. In addition, as shown in Figs. 8a, 8b, and 8c, as β decreases, the model accuracy gap among DREAM-FL, LEARN, and FARN increases. For example, when $\beta = 0.1$, the model accuracy for DREAM-FL can still be ~ 0.7; however, the model accuracy for LEARN and FARM are reduced to < 0.5 and < 0.4, respectively. The results imply that increasing the number of qualified clients can significantly increase the model accuracy, especially when data samples are non-independent and identically distributed among the clients.

VI. Conclusion

This paper investigated the joint resource allocation and client selection problem in a NOMA-based FL system. To efficiently solve the problem, a deep reinforcement learning-based algorithm, i.e., DREAM-FL, has been designed. The simulation results have demonstrated that DREAM-FL outperforms the other two baseline algorithms, i.e., LEARN and FARN, by selecting the most qualified clients in each global iteration, thus accelerating the FL process.

References

[1] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the Internet of Things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.

[2] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. [Online]. Available: https://arxiv.org/abs/1610.05492.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, 2017, pp. 1273–1282.

[4] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[5] J. Yao and N. Ansari, "Enhancing federated learning in fog-aided IoT by CPU frequency and wireless power control," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3438–3445, 2021.

[6] ——, "Secure federated learning by power control for internet of drones," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1021–1031, 2021.

[7] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," *arXiv e-prints*, p. arXiv:1812.06127, Dec. 2018.

[8] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[9] G. Sreya, S. Saigadha, P. D. Mankar, G. Das, and H. S. Dhillon, "Adaptive rate noma for cellular iot networks," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 478–482, 2022.

[10] Z. Zhang and R. Q. Hu, "Uplink non-orthogonal multiple access with fractional power control," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[11] X. Sun, L. Yu, and Y. Yang, "Jointly optimizing user clustering, power management, and wireless channel allocation for noma-based internet of things," *Digital Communications and Networks*, vol. 7, no. 1, pp. 29–36, 2021.

[12] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, vol. 12. MIT Press, 1999.

[13] Z. Akhavan, M. Esmaeili, B. Badnava, M. Yousefi, X. Sun, M. Devetsikiotis, and P. Zarkesh-Ha, "Deep reinforcement learning for online latency aware workload offloading in mobile edge computing," in *2022 IEEE Global Communications Conference*, 2022, pp. 2218–2223.

[14] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1552–1564, 2021.

[15] W. Shi, S. Zhou, and Z. Niu, "Device scheduling with fast convergence for wireless federated learning," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[16] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1188–1200, 2021.

[17] 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, 2021.

[18] R. Albelaihi, X. Sun, W. D. Craft, L. Yu, and C. Wang, "Adaptive participant selection in heterogeneous federated learning," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[19] L. Yu, R. Albelaihi, X. Sun, N. Ansari, and M. Devetsikiotis, "Jointly optimizing client selection and resource management in wireless federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4385–4395, 2022.

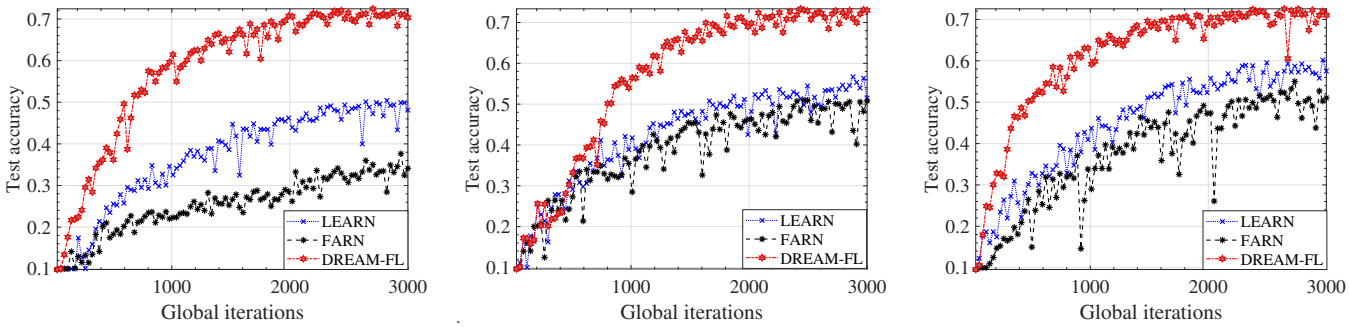
[20] R. Albelaihi, L. Yu, W. D. Craft, X. Sun, C. Wang, and R. Gazda, "Green federated learning via energy-aware client selection," in *2022 IEEE Global Communications Conference*, 2022, pp. 13–18.

[21] X. Sun and N. Ansari, "Dynamic resource caching in the iot application layer for smart cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 606–613, 2018.

[22] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A survey on non-orthogonal multiple access for 5g networks: Research challenges and future trends," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2181–2195, 2017.

[23] J. Ghosh, I.-H. Ra, S. Singh, H. Haci, K. A. Al-Utaibi, and S. M. Sait, "On the comparison of optimal noma and oma in a paradigm shift of emerging technologies," *IEEE Access*, vol. 10, pp. 11 616–11 632, 2022.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3264463



(a) $\beta = 0.1$. (b) $\beta = 1$. (c) $\beta = 10$.

Fig. 8: Test accuracy over global iterations for different algorithms, where a) β=0.1, b) β=1, and c) β=10.

[24] H. Sun, X. Ma, and R. Q. Hu, "Adaptive federated learning with gradient compression in uplink noma," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16 325–16 329, 2020.

[25] X. Ma, H. Sun, and R. Q. Hu, "Scheduling policy and power allocation for federated learning in noma based mec," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–7.

[26] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless federated learning (WFL) for 6g networks - part II: the compute-then-transmit NOMA paradigm," *CoRR*, vol. abs/2104.12005, 2021. [Online]. Available: https://arxiv.org/abs/2104.12005

[27] Z. Yang, M. Chen, W. Saad, C. S. Hong, M. Shikh-Bahaei, H. V. Poor, and S. Cui, "Delay minimization for federated learning over wireless communication networks," *arXiv preprint arXiv:2007.03462*, 2020.

[28] Y.-C. Hsieh, Y.-C. Lee, and P.-S. You, "Solving nonlinear constrained optimization problems: An immune evolutionary based two-phase approach," *Applied Mathematical Modelling*, vol. 39, no. 19, pp. 5759–5768, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0307904X14006878

[29] M.-S. Chern, "On the computational complexity of reliability redundancy allocation in a series system," *Operations research letters*, vol. 11, no. 5, pp. 309–315, 1992.

[30] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in neural information processing systems*, vol. 12, 1999.

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[32] P. Hintjens, *ZeroMQ: messaging for many applications*. " O'Reilly Media, Inc.", 2013.

[33] European Telecommunications Standards Institute (ETSI), "3gpp tr 36.931 version 9.0.0," 2011. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/136900_136999/136931/09.00.00_60/tr_136931v090000p.pdf

[34] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proceedings of the National Academy of Sciences*, vol. 118, no. 17, p. e2024789118, 2021.

[35] T. Zeng, O. Semiari, M. Chen, W. Saad, and M. Bennis, "Federated learning on the road autonomous controller design for connected and autonomous vehicles," *IEEE Transactions on Wireless Communications*, vol. 21, no. 12, pp. 10 407–10 423, 2022.

[36] J. Zheng, K. Li, N. Mhaisen, W. Ni, E. Tovar, and M. Guizani, "Exploring deep-reinforcement-learning-assisted federated learning for online resource allocation in privacy-preserving edgeiot," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 099–21 110, 2022.

[37] W. Huang, T. Li, D. Wang, S. Du, and J. Zhang, "Fairness and accuracy in federated learning," *arXiv preprint arXiv:2012.10069*, 2020.

Rana Albelaihi [S'20] received her M.S. and B.S. degrees in computer science from Tennessee State University, Tennessee, USA in 2014 and 2016, respectively. She started her doctoral studies at the SENet lab at the University of New Mexico in 2020. Her research interests include IoT, federated learning, and green communications and computing.

Akhil Alasandagutti received his B.S. in Computer Science at the University of Mississippi in 2021. He is currently pursuing a Ph.D. in Computer Science at the University of New Mexico. His research interests include predictive performance modeling in high-performance computing applications and applied machine learning.

Liangkun Yu [S'20] obtained his B.E. and M.E. degrees both in Communications Engineering from Fuzhou University in 2014 and 2017, respectively. Following this, he joined China Telecom as a wireless network engineer from 2017 to 2019. He commenced his doctoral program in the SECNet Lab at the University of New Mexico in 2019. His research covers various topics, including reinforcement learning for UAV swarm management, aerial corridors, federated learning, and drone-assisted mobile networks.

Jingjing Yao received the B.S. degree in information and communication engineering from Dalian University of Technology, Dalian, China, in 2013, the M.S. degree in information and communication engineering from the University of Science and Technology of China, Hefei, China, in 2016, and the Ph.D. degree in computer engineering from New Jersey Institute of Technology, Newark, NJ, USA, in 2021. She is an Assistant Professor of Computer Science with Texas Tech University, Lubbock, TX, USA. Her research interests include Internet of Things, applied machine learning in communication and networking, drone-assisted networking, and mobile edge computing.

Xiang Sun [S'13, M'18] is an assistant professor at the Department of Electrical and Computer Engineering at the University of New Mexico. He received his Ph.D. degree in Electrical Engineering from New Jersey Institute of Technology (NJIT) in 2018, and his M.E. and B.E. degrees both from Hebei University of Engineering in 2011 and 2008, respectively. His research interests include free space optics, wireless networks, distributed machine learning, Internet of Things, edge computing, and green communications and computing. He has received several honors and awards, including NJIT Ross Fellowship 2014-2015, 2016 IEEE International Conference on Communications (ICC) Best Paper Award, 2017 IEEE Communications Letters Exemplary Reviewers Award, 2018 NJIT Hashimoto Price, 2018 InterDigital Innovation Award on IoT Semantic Mashup, and 2019 IEICE Communications Society Best Tutorial Paper Award. He is an Associate Editor of the IEEE Open Journal of the Computer Society and Elsevier Digital Communications and Networks.