Digital Twin-Assisted Federated Learning Service Provisioning Over Mobile Edge Networks

Ruirui Zhang, Zhenzhen Xie, *Member, IEEE*, Dongxiao Yu, *Senior Member, IEEE*, Weifa Liang, *Senior Member, IEEE*, and Xiuzhen Cheng, *Fellow, IEEE*

***Abstract*—Federated Learning (FL) offers collaborative machine learning without data exposure, but challenges arise in the mobile edge network (MEC) environment due to limited resources and dynamic conditions. This paper presents a Digital Twin (DT)-assisted FL platform for MEC networks and introduces a novel multi-FL service framework to address resource dynamics and mobile users. We leverage DT models to optimize device scheduling and MEC resource allocation, aiming to maximize utility across FL services. Our work includes heuristic and constant approximation algorithms for offline multi-FL service scenarios and we also investigate an online setting of our solution with dynamic bandwidth and moving client conditions. To adapt to changing network conditions, we utilize historical bandwidth data in DTs and implement a deep reinforcement learning algorithm, Ra_DDPG, for automatic bandwidth allocation. Evaluation results demonstrate a significant 49.8% increase in system utility compared to a benchmark algorithm, showcasing the effectiveness of our approach.**

***Index Terms*—Digital twin (DT), digital twin-assisted mobile edge computing (DT-assisted MEC), multiple federated learning services, deep reinforcement learning algorithm.**

I. Introduction

FEDERATED learning is a promising collaborative learning method that focuses on training machine learning models in a manner that augments privacy safeguards. Motivated by the current trend of machine learning as a service (MLaaS), which promises to build, deploy and manage high-quality machine learning models for various real-world applications [1], there is an urgent need for enabling federated learning paradigm to process multiple service requests simultaneously [2], [3]. Considering the competition among different service requests, the needs of various data sources, and privacy concerns, mobile edge computing (MEC)-based Federated learning (FL) presents an opportunity to remove the barriers by enabling multiple stakeholders to train machine learning models collaboratively, meanwhile, providing computation and communication resources to response various service request efficiently [4], [5], [6]. In particular, multiple FL services provisioning in such an FL platform can select FL clients with heterogeneous resources to respond to different FL service requests. Furthermore, the selected FL clients can utilize their private data and the resources provided by the MEC environment to fulfill specific task requests.

We note that, since the edge resources are distributed and scattered, the MEC network is featured by highly dynamic and limited computing and communication resource constraints [7], which may adversely affect the FL training performance, due to the UEs often demand different amounts of resources under diverse network conditions. Worse, an FL platform with multi-service provisioning could exaggerate the resource competition, and the dynamic network workload also inevitably affects the performance of each training task, resulting in a lower total income for the FL platform. Moreover, in MEC networks, UEs exhibit variations in processing power and private data sources. In the context of resource-constrained multi-service provisioning, if the FL platform tends to involve every UE while having diverse computing and communication resources and varying data quality, it may inevitably result in an inefficient training process. As mentioned in previous studies [8], [9], [10], [11], [12], [13], optimization methods that allow for flexible task offloading and adaptive resource allocation are popular approaches for improving the FL training efficiency in MEC environment. However, most existing works focused on a single FL service request, and these investigations did not fully address the underlying challenges associated with imbalanced data, computing resources, and unstable network conditions arising in multiple FL service provisioning.

Hence, it poses a key problem. That is, how to allocate the limited computing and communication resources efficiently to support multiple FL service provisioning in an MEC network simultaneously? To address this problem, we take the first step to utilize digital twins (DTs) for resource allocation and client selection to realize multiple services provisioning. A DT network refers to a software replica of a real communication network along with its operating environment. Therefore, adopting DT in MEC networks ensures the FL platform access to accurate, relevant, and timely data of the MEC network status. However,

Manuscript received 29 August 2023; revised 15 November 2023; accepted 19 November 2023. Date of publication 29 November 2023; date of current version 15 January 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62122042 and in part by the Major Basic Research Program of Shandong Provincial Natural Science Foundation under Grant ZR2022ZD02. Recommended for acceptance by H. Wu. *(Corresponding authors: Zhenzhen Xie; Dongxiao Yu.)*

Ruirui Zhang, Zhenzhen Xie, Dongxiao Yu, and Xiuzhen Cheng are with the School of Computer Science and Technology, Shandong University, Qingdao, Shandong 266237, China (e-mail: sherryz@mail.sdu.edu.cn; xiezz21@sdu.edu.cn; dxyu@sdu.edu.cn; xzcheng@sdu.edu.cn).

Weifa Liang is with the Department of Computer Science, City University of Hong Kong, Hong Kong 999077, P. R. China (e-mail: weifa.liang@cityu.edu.hk).

Digital Object Identifier 10.1109/TC.2023.3337317

most of the existing digital twin-enabled MEC network has not solved the challenges posed by heterogeneous FL participants with dynamic mobility patterns and different historical behaviors.

In this paper, we study the multiple FL services provisioning problem for a DT-assisted MEC network under changing network conditions with dynamic bandwidth and constantly moving UEs. In order to facilitate the provisioning of multiple FL services within an MEC network, it is important to concurrently address FL client selection and bandwidth allocation for the selected clients. Therefore, we introduce RL technique based method into our implementation, and it adopts MDP model to model the problem environment and offers guidance on how to learn a near-optimal policy. We utilize the network DT to interact with the MEC network environment continuously and adopt Long-Short Term Memory (LSTM) network to predict future network conditions based on accumulating knowledge stored in MEC network DT. Furthermore, we also profile each participant by analyzing the data quality, and the staleness to measure the contribution value in each communication round of FL. Based on the network insights provided by DTs and FL participant behavior evaluation, we jointly consider the FL client selection and bandwidth allocation strategy during FL training process to enable multiple FL services provisioning in an MEC network. Then, we present the design and implementation of a deep reinforcement learning (DRL) solution that improves the utility of the FL platform through intelligent bandwidth allocation. The proposed DRL algorithm, based on DDPG, leverages the network insights from DT and actively gives a bandwidth allocation decision in each communication round that can adapt to the dynamic MEC network conditions.

The main contributions of this paper are summarized as follows.

- We propose a novel DT-assisted MEC network service provisioning framework for multiple FL services, where DT offers continuous synchronization between the MEC and the twin for multiple FL services.
- We formulate a multiple FL services provisioning problem with the aim to maximize the accumulative utility of all FL model training, by intelligently choosing UE participation and the BS channel assignment to the chosen UEs. We also show the NP-hardness of the problem of concern through a reduction from a classical General Assignment Problem (GAP).
- We propose a fast, scalable heuristic algorithm for the offline setting of the defined problem, which delivers a near-optimal solution. Under the constraint that each FL training request has a fixed number of training rounds and each UE's transmission range can cover the entire network, we devise a constant approximation algorithm for this special case of the problem.
- For the online scenario with dynamic bandwidth and constantly moving UEs, we adopt reinforcement learning to enhance the multiple FL services provisioning in a DT-assisted MEC network. Deep Deterministic Policy Gradient (DDPG) is adopted for bandwidth allocation based on the monitored information of DT.
- We evaluate the performance of the proposed algorithms against several baseline algorithms through extensive simulations. Numerical results demonstrate that the proposed solution has promising performance in terms of bandwidth usage and utility. The proposed online algorithm outperforms baseline algorithms by increasing the utility by around 30%.

The rest of this paper is organized as follows. Section II introduces the system model, notations and notions, and problem definitions. Section III formulates the offline version of the problem, proposes a heuristic algorithm for the problem, and devised an approximation algorithm for a special case of the problem. Section IV devises an online algorithm for the online version of the problem. Section V evaluates the performance of the proposed algorithms through simulations. Section VI surveys related works, and Section VII concludes the paper.

II. System Model

We consider a DT-assisted MEC network $G = (V \cup U, E)$ with a set V of base stations (BSs) and a set U of UEs, where E is a set of links between BSs. We assume that a link between two BSs is an optical link with high capacity, and thus the bandwidth on the link is not considered as a constraint. Each BS at location Loc_i is connected to a cloudlet v_i, and C_i^{cmp} and C_i^{sto} are the computing and storage resource capacities on v_i, respectively. For a single FL training request, we assume that the training time T is divided into equal time slots indexed by t with $1 \le t \le T$. The learned model parameters of the request are uploaded to a cloudlet by each involving UE, and multiplexing methods such as TDMA and OFDMA can be adopted for data transmissions from UEs to the BSs. The available bandwidth for FL training requests may vary over time due to the presence of other FL training requests competing for communication resources in the MEC network. Let $b_{i,t}$ denote the amount of available bandwidth between UEs and cloudlet v_i in the time slot t.

To make full use of the historical MEC network information to support multi-FL service provisioning in varying network conditions, this paper adopts DT as a promising solution to provide a virtual replica of the MEC network. Referring to the characteristics of the digital twin network, a DT-assisted MEC network can use the virtual representation of the network to analyze and emulate the physical network environment. Specifically, we assume each UE $u_{j,k}$ has a corresponding DT $DT(u_{j,k})$ to capture its mobility changes and data uploading channels, and each $v_i \in V$ has a DT $DT(v_i)$ to record its historical bandwidth usage. For clarity, we use Fig. 1 to illustrate a DT-assisted MEC environment with five BSs to support multi-service provisioning in an FL platform. Each BS has a cloudlet and all the UEs have a fixed transmission range. Different FL service requests need to be sent to different FL servers for better MEC resource utilization, for example, in Fig. 1, The FL servers of the two FL services are located in cloudlet v_5 and v_1, respectively.

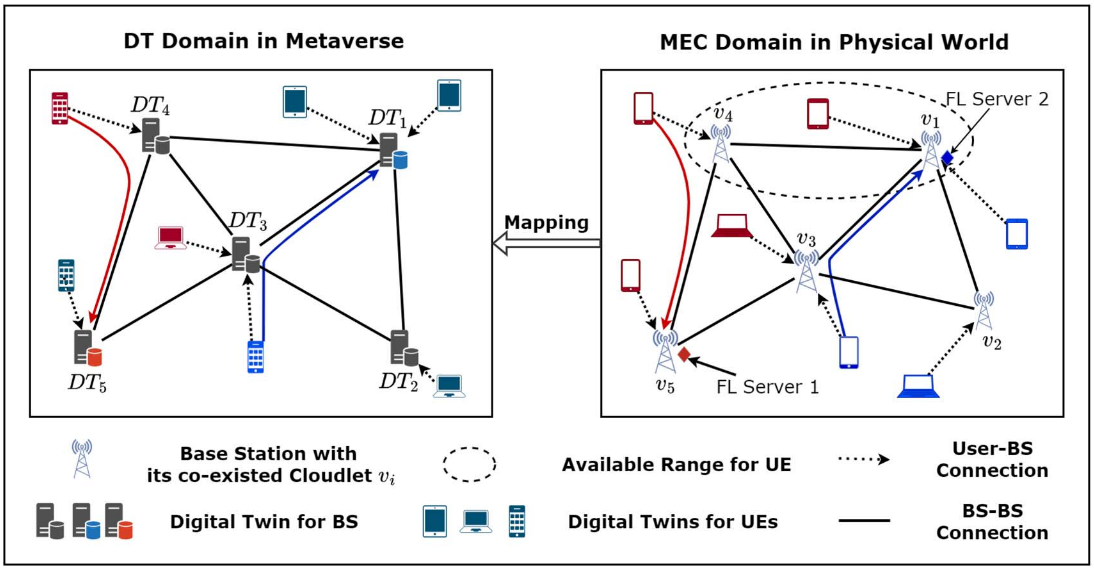


Fig. 1. An example of a digital twin-assisted MEC network.

A. Multiple FL Services Provisioning in DT-Assisted MEC Networks

In a DT-assisted MEC network, an FL platform with multiple FL services needs to broadcast K FL training requests to M UEs. The set of FL training requests is denoted by $R = \{r_1, r_2, \ldots, r_K\}$ and each request r_k is bounded with a specific learning goal on a global model w_k, which is defined as follows.

$$\min_{w_k \in \mathbb{R}_d} F_k(w_k) \tag{1}$$

where $F_k(w_k)$ is the global loss function of FL training request r_k. We assume that each r_k has an FL server s_k that locates on the cloudlet v_i.

The *contribution* of any UE to support an FL training request is defined as follows. In each FL training round, UE $u_{j,k}$ in $U_k = \{u_{1,k}, u_{2,k}, \ldots\}$ performs local training on his local data set $D_{j,k}$. The number of data samples in dataset $D_{j,k}$ is denoted as $|D_{j,k}|$. Then UEs of FL service r_k will send their local training updates to FL server s_k to update global model w_k.

B. Communication Model

We assume that each FL learning process follows a synchronous communication strategy. That is, the FL server of the FL service aggregates the local models sent by the chosen UEs until it receives all local training updates from the chosen UEs. Notably, the FL server needs to aggregate the local training updates and generates a new global model during an FL training process, it is necessary to deploy an FL server on a cloudlet with sufficient resource to meet its resource need. Let c_k^{cmp} and c_k^{sto} be the computing and storage resources requirement of FL server s_k, respectively. At time slot t, the location of UE $u_{j,k}$ is denoted as $Loc_{j,k,t}$. Let $|w_k|$ be the size of global model w_k transmitted between a UE and FL server s_k. In a single training round, denote by $\eta_{j,k}$ the upload transmission rate of UE $u_{j,k}$. For UE $u_{j,k}$'s local training update, the transmission delay between $u_{j,k}$ and v_i is $|w_k|/(b_{i,j,k,t}\eta_{j,k})$, where $b_{i,j,k,t}$ is the amount of bandwidth allocated to $u_{j,k}$. Then, the number of FL training rounds $l_{k,t}$ for global model w_k is $l_{k,t} = \min_{u_{j,k} \in U_k} \lfloor \frac{b_{i,j,k,t}\eta_{j,k}}{|w_k|} \rfloor$.

C. FL Client Selection

We here first introduce a key observation that bridges the training efficiency of a model and the number of selected clients for multiple FL services that co-exists in an FL platform. We then formulate an ILP solution for FL client selection.

The utility of each FL service is closely related to the contribution of its chosen UEs. A subset U_k of UEs with high contribution quality should be selected to boost the FL training process to achieve an accurate and fast-converging global model. To maximize the system utility, an FL client selection strategy should utilize the limited computing and communication resources in the MEC network to pick up the UEs with large contributions to the objective function.

We denote $d_{j,k,t}$ as the amount of contribution made by UE $u_{j,k}$ to the global model w_k training at time slot t, which is defined as follows.

$$d_{j,k,t} = \alpha \cdot \sqrt{|D_{j,k}|} + \eta \cdot tc_{j,k}^{local} - \varphi \cdot dis_{j,k,t}, \tag{2}$$

where α, η and φ are constant coefficients, $tc_{j,k}^{local}$ is the local training delay, and $dis_{j,k,t}$ is the Euclidean distance between location $Loc_{j,k}$ of UE $u_{j,k}$ and location Loc_i of the FL server s_k.

The first item $\alpha\sqrt{|D_{j,k}|}$ in the right-hand side of Eq. (2) weighted by parameter α denotes the data quality of UE's local data set, while $\eta \cdot tc_{j,k}^{local}$ is weighted by η to evaluate the staleness of $u_{j,k}$'s local training update. The final term $\varphi \cdot dis_{j,k,t}$ is considered as the impact of the physical distance and is weighted by φ.

Let d_k and l_k be the accumulative values of the contributions made by all selected UEs and global training rounds of model w_k after T time slots, respectively. The utility function F_k^u of model w_k in the end of its training process is defined as

$$F_k^u = \beta \cdot f_k(d_k) + (1-\beta) \cdot g_k(l_k) \tag{3}$$

where functions $f_k(d_k)$ and $g_k(l_k)$ are both continuous DR-submodular (see Definition 1 below) and $\beta \in [0,1]$ is constant. The system utility of an FL platform is $\sum_{k=1}^{K} F_k^u$.

Continuous submodular functions are defined on subsets of $\mathbb{R}^n$: $\mathcal{X} = \prod_{i=1}^{n} \mathcal{X}_i$, where each $\mathcal{X}_i$ is a compact subset of $\mathbb{R}$. Over continuous domains, a DR-submodular function [14] is a submodular function with the diminishing returns (DR) property,

Definition 1: (DR property). A function $f(\cdot)$ defined over $\mathcal{X}$ is DR-submodular (has the DR property) if $\forall \boldsymbol{a} \le \boldsymbol{b} \in \mathcal{X}$, $\forall i \in [n]$, $\forall j \in \mathbb{R}_+$ s.t. $(j\boldsymbol{\chi}_i + \boldsymbol{a})$ and $(j\boldsymbol{\chi}_i + \boldsymbol{b})$ are still in $\mathcal{X}$, it holds,

$$f(j\boldsymbol{\chi}_i + \boldsymbol{a}) - f(\boldsymbol{a}) \ge f(j\boldsymbol{\chi}_i + \boldsymbol{b}) - f(\boldsymbol{b})$$

$f(\cdot)$ is a DR-submodular function.

D. Problem Formulation

Given an MEC network $G = (V \cup U, E)$ with mobile UEs and dynamically changing computing and communication resources, we divided the training time into T time slots. For the corresponding digital twins of BSs and UEs, we let R be the set of FL training requests of an FL platform, where each request $r_i \in R$ demands a global model w_k for an independent learning goal. To maximize the system utility $\sum_{k=1}^{K} F_k^u$ of the FL platform, the *multi-FL service provisioning* problem is to jointly consider the FL client (UE) selection and the bandwidth allocation to the clients for each FL service provisioning, with the aim to maximize the accumulative utility gain of all service model training, through choosing a subset of UEs to participate in the training process of each FL model and allocating the communication resources for the chosen UEs, subject to the bandwidth resource capacity on each BS and the maximum transmission range of each UE.

Theorem 1: The multiple-FL service provisioning problem in $G = (V \cup U, E)$ is NP-hard.

Proof: We first show the NP-hardness of the multiple-FL service provisioning problem by reducing the Generalized Assignment Problem (GAP) to a special case of the problem in time slot t, where all UEs have already been assigned channels to upload their data with a fixed bandwidth cost and each FL server has already been placed in one cloudlet. Moreover, this special case has a fixed iterations requirement and thus the system utility is $\sum_{t=1}^{T} \sum_{u_{j,k}} d_{j,k,t}$. The reduction proceeds as follows.

The GAP contains $|V|$ bins and $|U|$ items, each bin $v_i \in V$ is associated with a budget $b_{i,t}$. For a bin v_i, each item $u_{j,k}$ has a profit $d_{j,k,t}$ with a weight $b_{j,k,t}$. The GAP is to pack as many items as possible to the $|V|$ bins such that the total profit is maximized, subject to the budget of each bin. It can be seen that this is a special case of the problem of concern in this paper, and a solution to the latter will deliver a solution to the GAP. It is well known that the GAP is NP-hard [15], so the *multiple-FL service provisioning* problem is NP-hard, too. □

For clarity, the symbols used in this paper are summarised in Table I.

TABLE I
THE SUMMARY OF NOTATIONS

Notion	Description
$V = v_1, v_2, \ldots, v_N$	the set of BSs and each BS has a co-existed cloudlet
N	Total number of BSs (with its co-existed cloudlet)
M	Total number of UEs in the network
K	Total number of FL training requests
$W = \{w_1, w_2, \ldots, w_k\}$	the set of FL models
r_k and R	an FL training request, a set of FL training requests
s_k	the FL server of r_k
U	the set of UEs in the network
U_k	the set of UEs participating in the training of global model w_k
Loc_i	a potential location for BS with its co-existed cloudlet v_i
$Loc_{j,k,t}$	a potential location for $u_{j,k}$ at time slot t
$b_{i,t}$	the available channel bandwidth of v_i at time slot t
C_i^{cmp} and C_i^{sto}	the computing and storage resource capacities on v_i
E	the set of wired links between BSs.
$DT(v_i)$	digital twin of v_i
$DT(u_{j,k})$	digital twin of $u_{j,k}$
$l_{k,t}$	the number of training rounds of global model w_k at time slot t
c_k^{cmp}	the cost of computing resource of placing s_k
c_k^{sto}	the cost of storage resource of placing s_k
$b_{i,j,k,t}$	the bandwidth allocated to $u_{j,k}$ when it chooses channel v_i for uploading data at time slot t
$\eta_{j,k}$	the upload transmission rate of UE $u_{j,k}$
$dis_{j,k,t}$	the Euclidean distance between UE $u_{j,k}$ and its FL server s_k

III. Algorithms for the Offline Multiple FL Services Provisioning Problem

In this section, we first formulate a Mixed Integer Nonlinear Programming (MINLP) for the offline multi-FL services provisioning problem. We then present a heuristic algorithm for the offline problem and an approximation algorithm for a special case.

A. MINLP Formulation of the Offline Multi-FL Service Provisioning Problem

Given a set R of FL training requests, the problem aims to maximize the system utility by jointly considering the FL server placement, client selection, and bandwidth allocation. The FL server s_k of request $r_k \in R$ needs to be placed on a cloudlet v_i with the amount of computing resource c_k^{cmp} and storage resource c_k^{sto}, respectively.

Let $z_{i,k}$ and $y_{i,j,k}$ be binary decision variables that show whether s_k is placed in v_i and UE $u_{j,k}$ uploads its local updates through the channel of v_i in the training process of w_k.

Recall that request r_k demands the number of training rounds l_k no less than l_k^{min}. The amount of contribution of each training round d_k should be at least d_k^{min}, and the optimization objective is to maximize the system utility as follows.

$$\text{Maximize} \quad \sum_{k=1}^{K} F_k^u \tag{4a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} c_k^{cmp} z_{i,k} \leq C_i^{cmp}, \forall i \tag{4b}$$

$$\sum_{k=1}^{K} c_k^{sto} z_{i,k} \leq C_i^{sto}, \forall i \tag{4c}$$

$$\sum_{i} z_{i,k} = 1, \forall k \tag{4d}$$

$$\sum_{j} x_{j,k} d_{j,k} \geq d_k^{min} \ , \forall k \tag{4e}$$

$$l_k \geq l_k^{min} \ , \forall k \tag{4f}$$

$$\sum_{i} y_{i,j,k} = x_{j,k}, \forall u_{j,k} \tag{4g}$$

$$\sum_{u_{j,k} \in U} |w_k| l_k y_{i,j,k} \leq b_i \eta_{j,k} \ , \forall i \tag{4h}$$

$$x_{j,k}, z_{i,k}, y_{i,j,k} \in \{0, 1\}, \forall i, j, k \tag{4i}$$

$$l_k \in \mathbb{Z}, \forall k \in \{1, 2, \ldots, K\} \tag{4j}$$

Let $Z = \{z_{j,k} \mid u_{j,k} \in U\}$, $X = \{x_{j,k} \mid u_{j,k} \in U\}$ and $Y = \{y_{i,j,k} \mid u_{j,k} \in U \ \& \ v_i \in V\}$ be FL server placement decision, UE selection, and BS channel assignment, respectively.

Constraints Eq. (4b) and Eq. (4c) guarantee that the total amounts of computing resource and storage resource demands do not exceed the computing and storage capacity on each cloudlet. Eq. (4e) and Eq. (4f) give the minimum contribution value and expected number of iterations, respectively. Eq. (4h) is the available channel bandwidth constraint.

B. Heuristic Algorithm

The offline problem is decomposed into two subproblems and we solve the two problems separately by using a heuristic algorithm For each FL training request, we first allocate the FL server to an available cloudlet. We then select a subset of UEs to participate in the training process and allocate bandwidth resources to each selected UE.

Considering Constraint (4h), the bandwidth allocation is a non-convex problem and a heuristic algorithm for this problem is detailed in `Algorithm` 1. Firstly, let U_k be the sequence of sorted UEs in U in non-decreasing order of $\Delta e_{j,k}$ (Line 1-4). A UE $u_{j,k} \in U_k$ with a larger $\Delta e_{j,k}$ indicates the UE would be chosen in an FL training request with a higher possibility. Then, in each round, we use binary search to find a L and a feasible solution to Y (Line 7-21). A pruning method is utilized to reduce the calculation time (Line 18-21). Finally, the strategy X is updated (Line 22-24).

Algorithm 1 A Heuristic algorithm for Bandwidth Allocation

Input: Initialization: M, N, K, an MEC network $G = (V, E), V = \{v_1, \ldots, v_N\}$, a set of FL training requests $R = \{r_1, \ldots, r_K\}$, a set of FL models $W = \{w_1, \ldots, w_K\}$, and an associated set of utility functions $\mathcal{F} = \{F_1, \ldots, F_K\}$, the available channel bandwidth b_i of each base station $v_i \in V$

Output: system utility value F for strategies X' and L'.

1: **for** Each UE group $U_k \in U$ **do**
2: Compute $\hat{b}_{j,k} = \frac{\sum_i b_{i,j,k}}{\sum_i 1}$ for each UE $u_{j,k}$
3: Sort U_k in ascending order by $\Delta e_{j,k} = \hat{b}_{j,k}/d_{j,k}$
4: Get the initialized U_k.
5: Initialize F as the given utility in the very beginning, X and $L = \{l_0, l_1, \ldots, l_K\}$ as the corresponding strategies;
6: **while** $\sum_{j,k} x_{j,k} < M$ **do**
7: Compute L_{min} and L_{max};
8: Initialize parameters as $L \leftarrow L_{min}$; $\hat{k} \leftarrow 1$; $\hat{l} \leftarrow l_1$; $\Delta l \leftarrow l_1^{max} - l_1^{min}$;
9: **while** $\Delta l > 1$ **do**
10: **if** there is a feasible solution under the constraints X and L: **then**
11: $L_{min} \leftarrow L$;
12: Compute utility F';
13: **if** $F' > F$ **then**
14: $F \leftarrow F'$; $X' \leftarrow X$; $L' \leftarrow L$;
15: **else**
16: Update L_{max} by setting $l_{\hat{k}}^{max} \leftarrow l_{\hat{k}}$;
17: $l_k \leftarrow \hat{l}$;
18: $\hat{k} \leftarrow \arg\max_{k \in \{1,2,\ldots,K\}} \{l_k^{max} - l_k^{min}\}$;
19: $\hat{l} \leftarrow l_{\hat{k}}$;
20: $\Delta l \leftarrow l_{\hat{k}}^{max} - l_{\hat{k}}^{min}$;
21: Update $l_{\hat{k}} \leftarrow (l_{\hat{k}}^{max} + l_{\hat{k}}^{min})/2$;
22: $i \leftarrow \arg\min_k \{\sum_j x_{j,k} \cdot d_{j,k}\}$;
23: $\hat{i}$ is index of the first zero value in X_i;
24: Update strategy X by setting $x_{\hat{i},i} \leftarrow 1$;
25: **return** Utility value F and corresponding strategy X' and L'.

C. Approximation Algorithm for a Special Case

In the following, we consider a special case of the problem under several restrictions by proposing an approximation algorithm for it. Specifically, given a set R of FL training requests, we assume that each request r_k requires a fixed number of rounds of the FL training process. Moreover, we assume that all BSs are under the transmission range of each UE, so the available bandwidth of each BS v_i is summed up as a total available bandwidth B, and let $b_{j,k}$ be the bandwidth requirement of UE $u_{j,k}$ if it is selected. We need to select a subset of UEs to participate in the global training process to maximize the accumulative utility gain of all participating UEs by maximizing the sum of contribution values of all FL training requests, under the available bandwidth resource constraint.

To this end, we formulate an Integer Linear Programming (ILP) for this special case of the problem as follows.

$$\text{Maximize} \quad \beta \cdot \sum_{u_{j,k} \in U_k} d_{j,k} \cdot x_{j,k} \tag{5a}$$

$$\text{s.t.} \quad \sum_{u_{j,k} \in U} x_{j,k} \cdot b_{j,k} \leq B, \tag{5b}$$

$$x_{j,k} \in \{0, 1\}, \forall j, k \tag{5c}$$

Algorithm 2 Approximation algorithm for Special Case

Input: Initialization: M, N, K, an MEC network $G = (V, E), V = \{v_1, \ldots, v_N\}$, a set of FL training requests $R = \{r_1, \ldots, r_K\}$, a set of FL models $W = \{w_1, \ldots, w_K\}$, the bandwidth requirement $b_{j,k}$ of UE $u_{j,k}$ and its contribution value $d_{j,k}$, and the available bandwidth B

Output: system optimization objective F for strategy X.

```
Initialize the utility value F = C and x_{j,k} ← 0, ∀u_{j,k} ∈ U;
B' ← 0;
Sort UEs in decreasing order by d_{j,k}/b_{j,k} and get a sequence U';
while B' < B do
    Pick the UE u_{j,k} with maximum d_{j,k}/b_{j,k} in U';
    if b_{j,k} + B' ≤ B then
        x_{j,k} ← 1;
        B' ← B' + b_{j,k};
        F ← F + d_{j,k};
    Remove u_{j,k} from U';
return Utility value F and corresponding strategy X = {x_{j,k}|u_{j,k} ∈ U}.
```

Theorem 2: The problem under fixed requirement restrictions is NP-hard.

Proof: We show the NP-hardness of the problem under fixed requirement restrictions by reducing the 0-1 knapsack problem to it. The reduction is as follows.

The 0-1 knapsack problem contains one knapsack with a maximum weight capacity of B. There are $|U|$ items, and each item $u_{j,k} \in U$ has a weight $b_{j,k}$ and value $d_{j,k}$. The 0-1 knapsack problem is to maximize the sum of the values of the items in the knapsack so that the sum of the weights is less than or equal to the knapsack's capacity. It can be seen a solution to the problem under fixed requirement restrictions will deliver a solution to the 0-1 knapsack problem. It is well known that the 0-1 knapsack problem is NP-hard, so the problem under fixed requirement restrictions is NP-hard, too. □

We find that this problem can be reduced to a 0-1 knapsack problem. That is, each UE $u_{j,k} \in U$ has a corresponding item with size $b_{j,k}$ and value $d_{j,k}$, and the system corresponds to the knapsack with the capacity B. We then give the approximation algorithm detailed in `Algorithm` 2.

D. Algorithm Analysis

We determine the time complexity of `Algorithm` 1 and analyze the approximation ratio of `Algorithm` 2.

Theorem 3: Given a DT-assisted MEC network $G = (V \cup U, E)$, a set R of FL training requests, a set of FL models W, and an associated set of utility functions, there is an algorithm, `Algorithm 1`, for the offline version of multiple FL training request assignment problem in G, which is $O(|U| \log |U| + |U| \cdot |R| \log |R|)$ time.

Proof: We first show that the solution delivered by `Algorithm 1` is feasible. As none of the constraints is violated, it can be verified that the solution indeed is a feasible solution.

We then analyze the time complexity of `Algorithm` 1. The computation time for the initialization is $O(|U| \log |U|)$, while binary search takes $O(|R| \log |R|)$ time. In each iteration, at least one UE is considered, therefore, there are at most $|U|$ iterations in the proposed algorithm. The time complexity of `Algorithm` 1 thus is $O(|U| \log |U| + |U| \cdot |R| \log |R|)$. □

Theorem 4: Given a DT-assisted MEC network $G = (V \cup U, E)$ and a set R of FL training requests, a set of FL models W, the bandwidth requirement $b_{j,k}$ and contribution value $d_{j,k}$ of UE $u_{j,k}$, and the available bandwidth B, there is an approximation algorithm, `Algorithm 2`, for the proposed problem, where the approximation ratio is 2.

Proof: We analyze the approximation ratio of `Algorithm 2`. Let OPT and $\mathbb{S}$ be the optimal solution and the feasible solution of `Algorithm` 2 for the problem under fixed requirement restrictions, respectively. If $\sum_{u_{j,k} \in U} b_{j,k} \leq B$ then $\mathbb{S} = OPT$. Thus, we assume $\sum_{u_{j,k} \in U} b_{j,k} > B$. Let $u_{j',k'}$ be the first one that is not selected in the training process in $\mathbb{S}$. We claim that

$$OPT < \mathbb{S} + d_{j',k'} \leq \mathbb{S} + \max_{u_{j,k} \in U} d_{j,k} \leq 2 \cdot \mathbb{S} \tag{6}$$

For the first part of the inequality 6, we note that UEs are sorted in decreasing order of $d_{j,k}/b_{j,k}$ in $\mathbb{S}$. Therefore, if we are allowed to just select a portion of one UE, then the most efficient way is to load the UEs in $\mathbb{S}$, plus a portion of $U_{j',k'}$, because replacing any portion of these UEs by other UEs decreases the utility value of the solution. This shows that the maximum total value we can get is less than $\mathbb{S} + d_{j',k'}$, which is therefore an upper bound of OPT. The second part of the inequality lies in the fact that $d_{j',k'} \leq \max_{u_{j,k} \in U} d_{j,k}$, and the last part is because $\max_{u_{j,k} \in U} d_{j,k} \leq \mathbb{S}$. So the approximation ratio of the proposed algorithm for the problem under fixed requirement restrictions, `Algorithm` 2, is 2. □

IV. Online Multiple-FL Service Provisioning With RL Support

In this section, we propose an online algorithm for the *dynamic multiple FL services provisioning* problem with the aim to maximize the system utility within T time slots, assuming that FL requests arrive one by one without the knowledge of future arrivals.

A. Algorithm Overview

The basic idea of the proposed algorithm consists of two stages: data preprocessing and bandwidth allocation. For data preprocessing, we leverage the historical trace information of BSs and UEs stored in their DTs, which later are used to predict the bandwidth of each channel and locations of UEs in the next time period. A greedy algorithm is then employed to determine the best channels for parameter transmission in multiple FL service model training, and the bandwidth allocation is carried out by a Markov Decision Process (MDP) $M = (s_t, act_t, r(s_t; act_t), s_{t+1})$, where s_t is the current state of MEC network, act_t is the action adopted at time slot t, $r(s_t; act_t)$ is the reward function and s_{t+1} is the next state. To solve the MDP, the Deep Deterministic Policy Gradient (DDPG) method is utilized to find the near-optimal bandwidth allocation solution to the MDP.

B. Data Preprocessing

We use the DTs to store historical data of UEs, BSs and their channels in the MEC environment. The historical trajectory dataset $TD(u_{j,k})$ of UE $u_{j,k}$ is also stored in $DT(u_{j,k})$, which includes the information about the mobility, the contribution value, and the timestamp of $u_{j,k}$ and represented by a tuple $\langle loc, d_{j,k,timestamp}, timestamp\rangle$. The DT $DT(v_i)$ of BS v_i contains channel available bandwidth and corresponding timestamp as $< b_i, timestamp >$.

Let $|TD(u_{j,k}, loc)|$ be the number of data samples in $TD(u_{j,k})$ when $u_{j,k}$ reached location loc and $|TD|$ be the number of data samples in $TD(u_{j,k})$. Then the predicted probability $p(u_{j,k}, loc)$ of UE $u_{j,k}$ at location loc is formulated as $p(u_{j,k}, loc) = \frac{|TD(u_{j,k}, loc)|}{|TD|}$. We adopt the Long Short-Term Memory (LSTM) method to predict the amount of available bandwidth of v_i at time slot t, and the predictive value $\hat{b}_{i,t}$ gives the information on future bandwidth changes.

Having obtained the predicted location of each UE and bandwidth on each BS, we then determine the channel of each UE to transmit the local training updates. For UE $u_{j,k}$ at time slot t, we select BS v_i within $u_{j,k}$'s transmission range while having the maximum capacity and save the index of the BS in $ca_{j,k,t}$. Therefore, the channel allocation result is defined as $CA_t = \{ca_{j,k,t} \mid u_{j,k} \in U\}$.

C. Bandwidth Allocation

We adopt a deep reinforcement learning method for bandwidth allocation. Specifically, we first formulate the bandwidth allocation as a Markov Decision Process (MDP), where an agent interacts with an environment for a given number of time periods, with the aim to maximize the system utility of a multiple FL services platform. At each time slot, the cloud server acts as the agent and observes the state of the DT-assisted MEC network environment, and then takes actions based on the state. The action then impacts the environment and transits to the next state. The agent receives a reward as a consequence of its action and the change of the environment. The MDP is formulated as follows.

State: The state of time slot t is represented by a vector $s_t = (B(t), D(t), CA_t)$, where $B(t) = \{\hat{b}_{i,t} \mid v_i \in V\}$ denotes the predicted bandwidth on each channel of BS v_i, $D(t) = \{d_{j,k,t} \mid u_{j,k} \in U\}$ denotes the predicted contribution value of all UEs and $d_{j,k,t}$ denotes the predicted contribution value of $u_{j,k}$ at time slot t, $CA_t = \{ca_{j,k,t} \mid u_{j,k} \in U\}$ denotes the channel allocation for all UEs at time slot t. The value of $d_{j,k,t}$ will be updated at time slot t only if UE $u_{j,k}$ is selected for training in the last time slot.

Action: At the beginning of each time slot t, the agent needs to select the subset of UEs U_k to participate in the training of global model w_k. Then the agent allocates the bandwidth to the selected UEs based on the channel allocation CA_t in state s_t. If UE $u_{j,k}$ is not selected in an FL training request, then the bandwidth $b^t_{j,k}$ is set to 0. In particular, the action space can be represented as $act_t = \{\mathbf{b}(t)\}$ where $\mathbf{b}(t) = \{b_{j,k,t} \mid u_{j,k} \in U\}$ denotes the bandwidth of channel $v_{ca_{j,k,t}}$ allocated to $u_{j,k}$ for uploading the local training updates.

Algorithm 3 Resource Allocation based on DDPG

Input: Initialization: Episode number M', Time horizon T, MEC network $G = (V \cup U, E), V = \{v_1, \ldots, v_N\}$, the set of FL training requests R, the set of FL global models $Model = \{w_1, \ldots, w_k\}$, predicted channel bandwidth $\hat{b}^t_i$ for each BS v_i at time slot t, predicted contribution value $d^t_{j,k}$ for each user $u_{j,k}$ at time slot t, the utility function $F^u_k(t) = \beta \cdot f_k(d_{k,t}) + (1-\beta) \cdot g_k(\sum_{t'=1}^{t} l_{k,t'})$;

Output: The trained actor and critic networks $\mu(s|\theta^\mu)$, $Q(s, act|\theta^Q)$.

Randomly initialize the critic network $Q(s, act|\theta^Q)$ and the actor network $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ, respectively;
Initialize the target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$;
Initialize replay buffer $\mathcal{R}$;
for $episode \leftarrow 1$ to M' **do**
  Initialize a random process $\mathcal{N}$ for action exploration;
  Receive the initial state observations s_1;
  for $t \leftarrow 1$ to T **do**
    Get the predicted bandwidth and UE location using the proposed method;
    Select an action according to the current policy π and exploration noise $\mathcal{N}_t$;
    $act_t \leftarrow \mu(s_t|\theta^\mu) + \mathcal{N}_t$;
    The network allocates bandwidth to each UE;
    UEs who have non-zero bandwidth join local model training and upload local training updates;
    while There exists one global model for an FL service in time slot t **do**
      Wait;
    Update the predicted contribution value of selected UEs;
    Observe reward $r_t(s_t, act_t)$ and observe new state s_{t+1};
    Store transition $(s_t, act_t, r_t, s_{t+1})$ in $\mathcal{R}$;
    Sample a random minibatch of C transitions $(s_i, act_i, r_i, s_{i+1})$ from $\mathcal{R}$;
    $y_i \leftarrow r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} \mid \theta^{\mu'}) \mid \theta^{Q'})$;
    Update critic by minimizing the Loss: $\frac{1}{C}\sum_i (y_i - Q(s_i, act_i \mid \theta^Q))^2$;
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} \approx \frac{1}{C}\sum_i \nabla_{act} Q(s, act|\theta^Q)|_{s=s_i, act=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i},$$

    $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$;
    $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$;
return Reward value.

Reward: The reward at the end of each time slot t is $r(s_t; act_t)$, which is defined as follows.

$$r(s_t; act_t) = \sum_{k=1}^{K}\left(\beta * f_k(d_{k,t}) + (1-\beta) * g_k\left(\sum_{t'=1}^{t} l_{k,t'}\right)\right), \quad \forall t = 1, \ldots, T \tag{7}$$

where $\sum_{t'=1}^{t} l_{k,t'}$ is the accumulative number of communication rounds in model w_k training process and $d_{k,t}$ is the sum of the amounts of contributions by all selected UEs.

Given the MDP, we use the Deep Deterministic Policy Gradient (DDPG) [16] to allocate channel bandwidth to each UE.

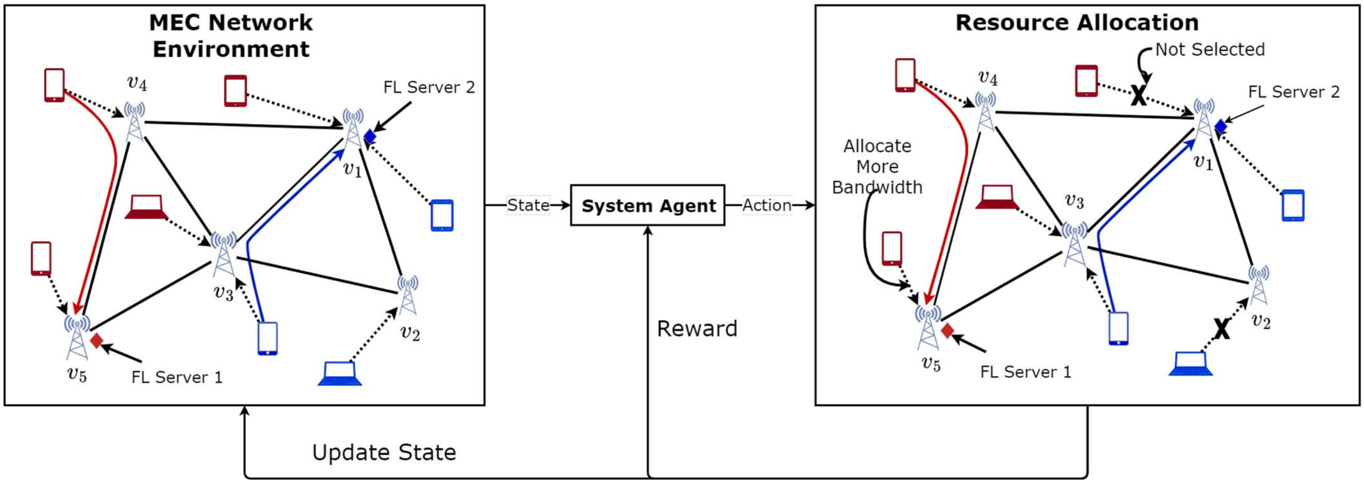


Fig. 2. An illustration of the communication resource allocation procedure.

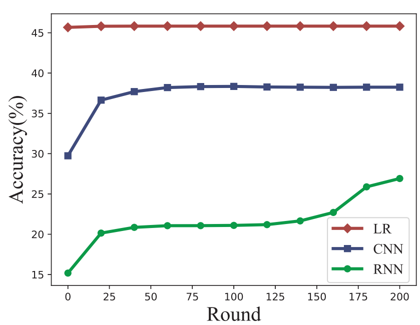


Fig. 3. Performance of different FL models.

The proposed DDPG-based channel bandwidth uses the historical information in DTs to predict future bandwidth and UEs' mobility. The proposed algorithm is given in `Algorithm` 3.

In the beginning, both the actor networks and critic networks are initialized randomly as the initial training parameters. Then, the replay memory buffer is initialized for storing the experiential samples in the training process. In each time slot, the agent selects its action toward its current observation and obtains the corresponding reward. Then, the new observation of the system state is obtained. The experience tuple $(s_t, act_t, r(s_t; act_t), s_{t+1})$ is stored to the replay buffer. Finally, the critic network and actor network are trained by sampling records from the replay buffer. An example is shown in Fig. 2.

V. SIMULATIONS

In this section, we evaluate the performance of the proposed algorithm through experimental simulations and investigate the impact of important parameters on the performance of the proposed algorithm.

A. Simulation Settings

1) Network Settings: We consider a DT-assisted MEC network with 5 base stations and 150 users with 10 FL training requests in a 20×20 square kilometers area. The distance between any two BSs is set to be at least 3 kilometers. The available wireless bandwidth at each BS is randomly drawn from [20 MHz, 50 MHz]. For each FL training request, the size of the trained model parameters is drawn between 0.02 MB and 0.5 MB randomly, which is validated based on existing networks, such as the Linear Regression network, CNN and RNN, etc.

2) Task and Dataset Description: The whole training process of FL tasks consists of $T = 100$ time slots and the length of each time slot is set as 5 minutes. The number of UEs in an FL training request is drawn from a Gaussian distribution which has a mean of 15. Each FL training request is training on datasets MNIST, FeMNIST, and Shakespeare with three different models: we perform Logistic regression on the MNIST dataset, CNN on the FeMNIST dataset, and RNN on the Shakespeare dataset. FedAvg is adopted in the simulations for local model aggregation in FL training process. The performance of those three models is shown in Fig. 3.

3) Training Settings: In each FL training request, data samples are uniformly assigned to the UEs. The local batch size is set to 10, local epochs are 5 and the learning rate is 0.01. The parameter φ in Eq. (2) for computing the value of each user's contribution value is in the range of $[0, 1]$, and the staleness penalty factor is $\beta = 0.5$.

4) Metrics: We evaluate all algorithms on three important evaluation Metrics: bandwidth usage, utility value, and wasted bandwidth.

- **Bandwidth usage** is characterized as the rate of bandwidth resource utilization that effectively supports the synchronous FL communication protocol.
- **Utility value** gauges the performance of each algorithm within the context of the multi-service FL platform, measuring its contribution to overall system utility.
- **Wasted bandwidth:** Notably, under synchronous FL communication protocol settings, the performance of an FL service request depends on the 'slowest' UE so a well-designed bandwidth allocation strategy should ensure all the training updates from selected UE can have

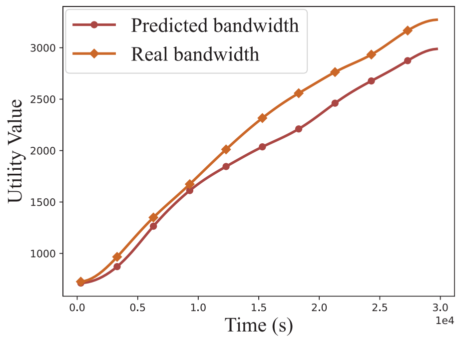


Fig. 4. Performance of algorithm Ra_DDPG by varying the value of bandwidth.

a similar arrival time. Nonetheless, it's essential to acknowledge the presence of excess bandwidth within the network that doesn't contribute to expediting the training process. This surplus bandwidth is appropriately defined as *Wasted bandwidth*.

B. Impact of Key Factors

1) Impact on Bandwidth Prediction Method: We first investigated the performance of the proposed algorithm Ra_DDPG by using LSTM as the bandwidth prediction method. Fig. 4 plots the utility value of Ra_DDPG under the predictive value and measured value of bandwidth. The difference between the two utility value curves is not significant, since the utility value under the predictive bandwidth is up to 9% difference when compared to the utility calculated under measured bandwidth. The results demonstrate the effectiveness of the proposed bandwidth prediction based on the LSTM network since the utility curve under predictive bandwidth is close to the one under measured bandwidth. This is because of the accurate bandwidth prediction employed by LSTM and DTs.

2) Impact on UE Numbers: We then evaluated the performance of Ra_DDPG in terms of bandwidth usage, utility value, and wasted bandwidth, by varying the number of users (UEs) from 50 to 200 with a fixed network size of 20×20. Fig. 5(a) illustrates the actual bandwidth usage of the FL platform in the MEC networks. It shows the actual bandwidth usage of the FL platform in the MEC networks, where the bandwidth usage decreases when the number of UEs increases. This is because the competition increased with the growth of the number of UEs. When the number of UEs reaches 200, Ra_DDPG can still maintain the bandwidth usage of 51.72%, which indicates a higher bandwidth utilization. We further evaluated bandwidth wastage in Fig. 5(c). It shows that when the number of UEs is increased, more bandwidth resource is wasted. This can be attributed to the growing uploading time gap between different UEs under the same FL task exacerbated by the fiercer competition for bandwidth resources. In particular, our proposed algorithm under 50 UEs wasted at most 30.78% during the whole training process. Meanwhile, the wasted bandwidth only increased 10% when the number of UEs is increased to 200.

We then compare the utility value of the proposed algorithm by varying the number of UEs and depict the results in Fig. 5(b). We can observe that the utility value of our proposed algorithm growing from 831 to 5246 under the condition of 200 UEs, while the utility value under 50 UEs is from 3324 to 13019 which indicates the utility is doubled. This is because when the number of UEs increases, one FL training request can have more available UEs to increase the utility gains.

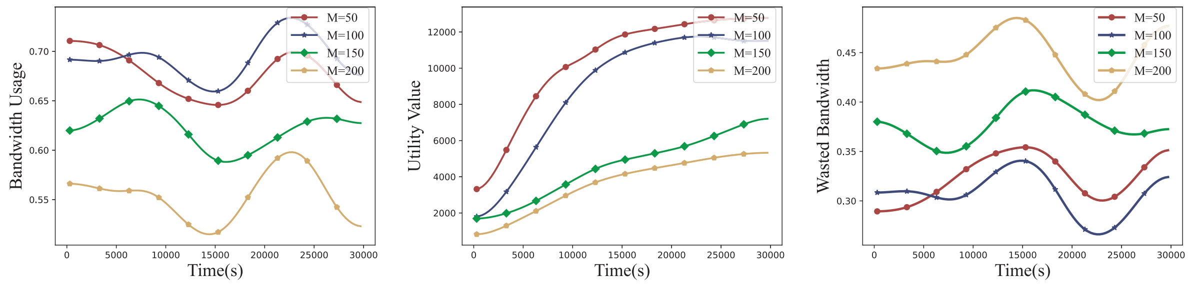
3) Impact on Network Sizes: Fig. 6 shows the impact of the network sizes with fixed 150 UEs and 5 base stations on the solution of algorithm Ra_DDPG. Fig. 6(a) depicts the bandwidth usage of the whole process by varying the size of the network from 10 to 100. The utility value and the wasted bandwidth with a different network size of 150 UEs are shown in Figs. 6(b) and 6(c), respectively. When the network size increased, we observe that the bandwidth usage decreased while the wasted bandwidth increased. At time slot 21, the proposed algorithm has the lowest bandwidth usage 59.84% with network 100, and it has the highest 66.48% with network size 10. It is seen that the utility value decreases with the increase in the network size. When the network size grows, the utility value decreased. This is because the distance between UE and BSs in a large MEC network could be longer, which leads to a larger transmission delay and lower utility value. The utility value of the proposed algorithm with network size 10 has the highest value of 3172, while the utility value of that with network size 70 is the lowest one of 1607.

C. Numerical Results

Given the channel allocation for each UE, we evaluated the proposed algorithm Ra_DDPG against the offline algorithm and greedy algorithm:

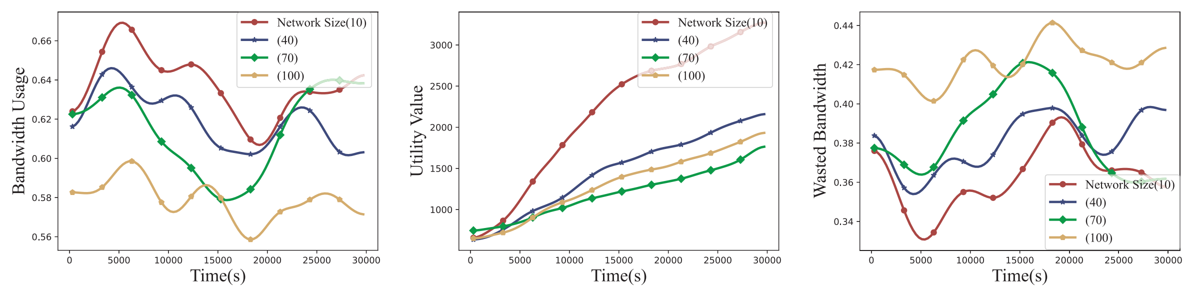
- **Offline Algorithm:** By running the offline algorithm, we can get a nearly optimal bandwidth allocation. Such comparison is conservative because this algorithm represents a lower bound of the optimal solution of the problem of concern.
- **Greedy:** Bandwidth allocation using Greedy strategy. It always allocates more bandwidth to the training requests that have the lowest performance.
- **Eq_client:** For all UEs involved in the training of global models, the bandwidth of each BS is allocated equally. Moreover, UEs always choose the nearest base station to upload the trained model parameters.
- **Eq_task:** The bandwidth of the system is equally allocated to each FL training request. For each FL training request, it adopts the offline algorithm as the bandwidth alloca tion method.
- **Par_task:** For each FL training request, Par_task allocates bandwidth proportional to the number of its UEs. For each FL training request, it adopts the offline algorithm as the bandwidth allocation method.

We compared Ra_DDPG against the proposed baseline algorithms for the problem of joint participant selection and communication resource allocation with 10 FL training requests and different numbers of UEs in a DT-assisted MEC network as shown in Fig. 7. Fig. 7(a) depicts the bandwidth usage curves of different algorithms by varying the number of UEs from



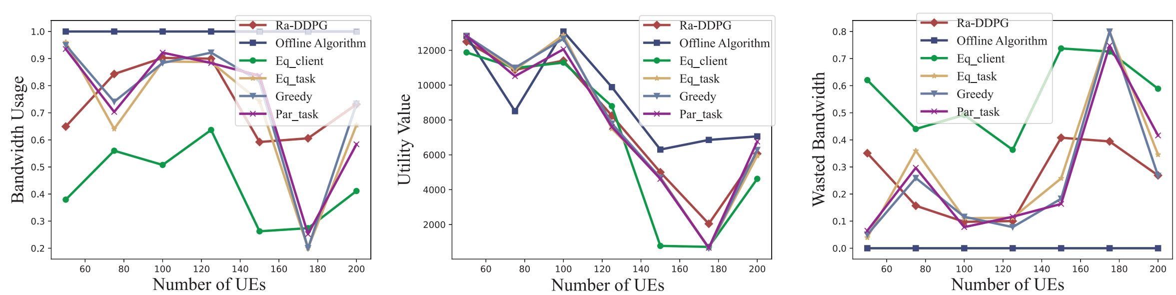
(a) (b) (c)

Fig. 5. Comparison results for the comprehensive performance of different UEs in Ra_DDPG. (a) The bandwidth usage with different UEs in the network. (b) The utility value with different UEs in the network. (c) The wasted bandwidth with different UEs in the network.



(a) (b) (c)

Fig. 6. Comparison results for the comprehensive performance of different network sizes in Ra_DDPG. (a) The bandwidth usage with different network sizes. (b) The utility value with different network sizes. (c) The wasted bandwidth with different network sizes.



(a) (b) (c)

Fig. 7. Comparison results for the comprehensive performance of different amounts of UEs in different algorithms. (a) The bandwidth usage with different amounts of UEs among six algorithms. (b) The utility value with different amounts of UEs among six algorithms. (c) The wasted bandwidth with different amounts of UEs among six algorithms.

50 to 200 in the network with a fixed size of 20. The results of wasted bandwidth are shown in Fig. 7(c). We can see that while all algorithms' bandwidth usage decreases with the increasing number of UEs, algorithm Eq_client has almost the lowest bandwidth usage 44.5% among the four algorithms. When the number of UEs grows from 50 to 175, we can see the bandwidth usage of Ra_DDPG is similar to Eq_task, Greedy, and Par_task, which is around 80% on average. However, when the number of UEs arrives at 175, the bandwidth usage of Ra_DDPG is 60%, which is 35% higher than those of the baseline algorithms. The results indicate that Ra_DDPG performs better than the other baseline algorithms and is close to the offline algorithm.

We then evaluate the utility value and depict the results in Fig. 7(b). It manifests that when the number of UEs grows, the utility value of all algorithms decreases rapidly. When the number of UEs grows from 50 to 150, the utility value decreased and the variance among 6 algorithms is less than 1000. We observed that Ra_DDPG tends to have the highest utility value most of the time among the compared algorithms, even when the number of UEs increases up to 200.

We further compared the performance of Ra_DDPG with the baseline algorithms by varying the network size from 10 to 100 with 150 UEs and 5 BSs. The results are shown in Fig. 8. Fig. 8(a) shows that, when the network size increased, the bandwidth usage in the solutions delivered by the five

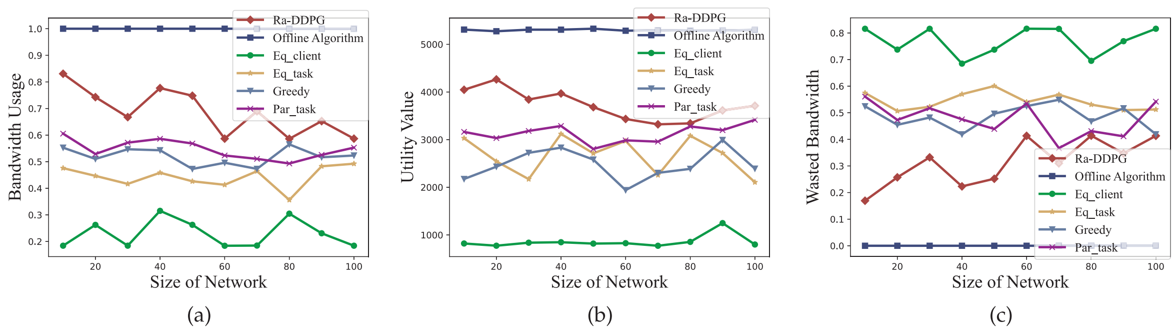


Fig. 8. Comparison results for the comprehensive performance of different network sizes with different algorithms. (a) The bandwidth usage with different network sizes among six algorithms. (b) The utility value with different network sizes among six algorithms. (c) The wasted bandwidth with different network sizes among six algorithms.

comparison algorithms slowly downturn. Eq_client always has the lowest bandwidth usage up to 31.91%. Meanwhile, the average bandwidth usage of Ra_DDPG is 68.83% which is 45.63% higher than Eq_client and 15.04% higher than the results of the other three algorithms.

The results of the utility value of different algorithms are shown in Fig. 8(b). It shows that the average utility value of algorithm Ra_DDPG is 3742.62 which is higher than the other four algorithms. The average utility value of Greedy is 2497.90 which is close to Eq_task, and the average utility value of algorithm Par_task, which is 3144.33, is slightly lower than that of Ra_DDPG. The wasted bandwidth of different algorithms is shown in Fig. 8(c). We can see that Ra_DDPG has the highest bandwidth utilization and delivered the highest utility value among the five algorithms.

VI. Related Work

Due to the growth of big data, IoT and AI applications, distributed machine learning has been extensively studied in recent years[17], [18], [19], [20], [21], [22]. Considering the data privacy, [23] first introduced FL as the primary solution for training models without exposing raw data. There are many investigations on implementing FL in MEC networks [24], [25], [26], [27]. For example, Dinh et al. [28] is among the first to extend FL over the wireless network. They considered a resource allocation optimization problem in the FL training process with heterogeneous computing and power resources. [11] proposed a joint control solution of FL in MEC networks with the aim of minimizing the long-term cumulative training cost. [29] aimed at alleviating the communication bottleneck of one-bit over-the-air aggregation for FL in edge networks and proposed a one-bit broadband digital aggregation solution.

In the dynamic landscape of federated learning, the integration of deep reinforcement learning (DRL) has attracted increasing attention due to its potential to address the unique challenges inherent in this decentralized, privacy-centric paradigm. Several research endeavors have explored the synergies between DRL and federated learning, each offering valuable insights and methodological contributions. In [30], [31], [32], the authors concentrated on enhancing the performance of federated learning by utilizing DRL to assess the learning quality of clients and select reliable participants in the federated learning process. Moreover, [33] directed their efforts towards addressing the issue of unreliable clients within hierarchical federated learning. Furthermore, DRL has found application in fortifying the resilience of federated learning systems against adversarial attacks. For example, [34] studied the poisoning attacks in FL systems and proposed a malicious model detection mechanism with the help of DRL. In [35], Weng et al. introduced an auditable and privacy-preserving deep learning framework that incorporates a value-driven incentive mechanism based on blockchain technology.

There are several recent studies on utilizing the DT technology in FL service provisioning [36], [37]. For instance, [38] studied the problem of adaptive FL and DT for Industrial IoT. Lu [39] leveraged the FL to construct DT models of IoT devices based on their running data and formulated an asynchronous model update scheme as an optimization problem, where they utilized the deep neural network model to reduce the transmission energy cost. [40] designed a mechanism with DT assisted for edge computing under asynchronous FL settings which obtained optimal resource management solution. [41] studied a blockchain-empowered FL scheme to strengthen communication security and data privacy protection in DT-edge networks and proposed an asynchronous aggregation scheme, which enhanced both communication efficiency and data security for IoT applications.

Unlike the aforementioned studies that considered only a single FL request in the FL platform, in this paper, we focus on multiple FL services provisioning in an MEC network. We consider the network dynamics and resource competitions among different FL requests, by adopting the DT of the network to provide accurate and reliable prediction of bandwidth and UE mobility.

VII. Conclusion

In this paper, we investigated multiple FL services provisioning in a DT-assisted MEC network, which can provide different FL services based on different FL training requests. We first formulated a novel multiple-FL services provisioning problem of FL client selection and bandwidth allocation for each FL training request. We then proposed a heuristic algorithm for the problem in the offline setting, and we also devised a constant approximation algorithm for its special case under fixed requirement constraints. We thirdly devised a deep reinforcement learning based online algorithm, Ra_DDPG, for the problem in the online setting, where the bandwidth resource and locations of UEs vary over time. We finally evaluated the performance of the online algorithm against other baseline algorithms through simulations. Simulation results demonstrated that the online algorithm Ra_DDPG outperforms the benchmarks. The future work built upon this research includes exploring resource allocation with the trade-off between fairness and training efficiency in multiple FL services provisioning.

REFERENCES

[1] M. Ribeiro, K. Grolinger, and M. A. Capretz, "MLaaS: Machine learning as a service," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2015, pp. 896–902.

[2] Q. Weng et al., "MLaaS in the wild: Workload analysis and scheduling in Large-Scale heterogeneous GPU clusters," in *Proc. 19th USENIX Symp. Netw. Syst. Des. Implementation (NSDI)*, Renton, WA, USA: USENIX Association, Apr. 2022, pp. 945–960. Accessed: Dec. 22, 2022. [Online]. Available: https://www.usenix.org/conference/nsdi22/presentation/weng

[3] F. Ponzina, M. Peón-Quirós, A. Burg, and D. Atienza, "E2CNNs: Ensembles of convolutional neural networks to improve robustness against memory errors in edge-computing devices," *IEEE Trans. Comput.*, vol. 70, no. 8, pp. 1199–1212, Aug. 2021.

[4] S. Xie, Y. Xue, Y. Zhu, and Z. Wang, "Cost effective MLaaS federation: A combinatorial reinforcement learning approach," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, 2022, pp. 2078–2087.

[5] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. Cervelló-Pastor, and l. Monje, "On the optimal allocation of virtual resources in cloud computing networks," *IEEE Trans. Comput.*, vol. 62, no. 6, pp. 1060–1071, Jun. 2013.

[6] C. Feng, Z. Zhao, Y. Wang, T. Q. S. Quek, and M. Peng, "On the design of federated learning in the mobile edge computing systems," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5902–5916, Sep. 2021.

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

[8] S. Liu, G. Yu, R. Yin, J. Yuan, L. Shen, and C. Liu, "Joint model pruning and device selection for communication-efficient federated edge learning," *IEEE Trans. Commun.*, vol. 70, no. 1, pp. 231–244, Jan. 2022.

[9] Z. Lin, S. Bi, and Y.-J. A. Zhang, "Optimizing AI service placement and resource allocation in mobile edge intelligence systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7257–7271, Nov. 2021.

[10] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, "Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration," *IEEE Trans. Comput.*, vol. 67, no. 9, pp. 1287–1300, Sep. 2018.

[11] Y. Jin, L. Jiao, Z. Qian, S. Zhang, and S. Lu, "Learning for learning: Predictive online control of federated learning with edge provisioning," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 1–10.

[12] F. Wang, H. Xing, and J. Xu, "Real-time resource allocation for wireless powered multiuser mobile edge computing with energy and task causality," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 7140–7155, Nov. 2020.

[13] J. Zhang et al., "Adaptive federated learning on Non-IID data with resource constraint," *IEEE Trans. Comput.*, vol. 71, no. 7, pp. 1655–1667, Jul. 2022.

[14] A. A. Bian, B. Mirzasoleiman, J. Buhmann, and A. Krause, "Guaranteed non-convex optimization: Submodular maximization over continuous domains," in *Proc. Artif. Intell. Statist.*, PMLR, 2017, pp. 111–120.

[15] L. Özbakir, A. Baykasoğlu, and P. Tapkan, "Bees algorithm for generalized assignment problem," *Appl. Math. Comput.*, vol. 215, no. 11, pp. 3782–3795, 2010.

[16] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[17] D. Wang, J. Ren, Z. Wang, Y. Wang, and Y. Zhang, "PrivAim: A dual-privacy preserving and quality-aware incentive mechanism for federated learning," *IEEE Trans. Comput.*, vol. 72, no. 7, pp. 1913–1927, Jul. 2023.

[18] D. Peteiro-Barral and B. Guijarro-Berdiñas, "A survey of methods for distributed machine learning," *Prog. Artif. Intell.*, vol. 2, no. 1, pp. 1–11, 2013.

[19] M. Raceala-Motoc, S. Limmer, I. Bjelakovic, and S. Stanczak, "Distributed machine learning in the context of function computation over wireless networks," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, 2018, pp. 291–297.

[20] Y. Gao et al., "Evaluation and optimization of distributed machine learning techniques for Internet of Things," *IEEE Trans. Comput.*, vol. 71, no. 10, pp. 2538–2552, Oct. 2022.

[21] E. Muscinelli, S. S. Shinde, and D. Tarchi, "Overview of distributed machine learning techniques for 6G networks," *Algorithms*, vol. 15, no. 6, p. 210, 2022, doi: 10.3390/A15060210.

[22] J. Wang, J. Hu, G. Min, W. Zhan, A. Y. Zomaya, and N. Georgalas, "Dependent task offloading for edge computing based on deep reinforcement learning," *IEEE Trans. Comput.*, vol. 71, no. 10, pp. 2449–2461, Oct. 2022.

[23] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, PMLR, 2017, pp. 1273–1282.

[24] W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang, "Client selection for federated learning with non-IID data in mobile edge computing," *IEEE Access*, vol. 9, pp. 24462–24474, 2021.

[25] Z. He, L. Wang, and Z. Cai, "Clustered federated learning with adaptive local differential privacy on heterogeneous IoT data," *IEEE Internet Things J.*, early access, Aug. 7, 2023.

[26] H. G. Abreha, M. Hayajneh, and M. A. Serhani, "Federated learning in edge computing: A systematic survey," *Sensors*, vol. 22, no. 2, 2022, Art. no. 450.

[27] J. Mills, J. Hu, G. Min, R. Jin, S. Zheng, and J. Wang, "Accelerating federated learning with a global biased optimiser," *IEEE Trans. Comput.*, vol. 72, no. 6, pp. 1804–1814, Jun. 2023.

[28] C. T. Dinh et al., "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2021.

[29] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2120–2135, Mar. 2021.

[30] Y. Deng et al., "AUCTION: Automated and quality-aware client selection framework for efficient federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 8, pp. 1996–2009, Aug. 2022.

[31] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, 2020, pp. 1698–1707.

[32] Y. Deng et al., "Improving federated learning with quality-aware user incentive and auto-weighted model aggregation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 4515–4529, Dec. 2022.

[33] N. M. Al-Maslamani, M. Abdallah, and B. S. Ciftler, "Reputation-aware multi-agent DRL for secure hierarchical federated learning in IoT," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1274–1284, 2023.

[34] W. Liu et al., "D2MIF: A malicious model detection mechanism for federated-learning-empowered Artificial Intelligence of Things," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2141–2151, Feb. 2023.

[35] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2438–2455, Sep./Oct. 2021.

[36] J. Pang, Y. Huang, Z. Xie, J. Li, and Z. Cai, "Collaborative city digital twin for the COVID-19 pandemic: A federated learning solution," *Tsinghua Sci. Technol.*, vol. 26, no. 5, pp. 759–771, Oct. 2021.

[37] M. A. Ferrag, B. Kantarci, L. C. Cordeiro, M. Debbah, and K.-K. R. Choo, "Poisoning attacks in federated edge learning for digital twin 6G-enabled IoTs: An anticipatory study," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Rome, Italy, 2023, pp. 1253–1258, doi: 10.1109/ICCWorkshops57953.2023.10283797.

[38] W. Sun, S. Lei, L. Wang, Z. Liu, and Y. Zhang, "Adaptive federated learning and digital twin for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5605–5614, Aug. 2021.

[39] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, “Communication-efficient federated learning for digital twin edge networks in industrial IoT,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5709–5718, Aug. 2021.

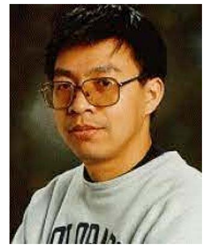
[40] Y. Xu, H. Zhou, J. Chen, T. Ma, and S. Shen, “Cybertwin assisted wireless asynchronous federated learning mechanism for edge computing,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 1–6.

[41] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, “Communication-efficient federated learning and permissioned blockchain for digital twin edge networks,” *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2276–2288, Feb. 2021.

Ruirui Zhang received the B.S. degree from Taishan College, Shandong University, in 2022. She is currently working toward the Ph.D. degree with the School of Computer Science and Technology, Shandong University. Her research interests include mobile edge computing, Internet of Things, and federated learning.

Zhenzhen Xie (Member, IEEE) received the M.S. degree and the Ph.D. degree in computer science from Jilin University, China, in 2014 and 2021, respectively. She currently holds a Postdoctoral position with the School of Computer Science and Technology, Shandong University. Her research interests include edge computing, the Internet of Things, and federated learning.

Dongxiao Yu (Senior Member, IEEE) received the B.S. degree from the School of Mathematics, Shandong University, in 2006, and the Ph.D. degree from the Department of Computer Science, University of Hong Kong, in 2014. He became an Associate Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, in 2016. He is currently a Professor with the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing, and graph algorithms.



Weifa Liang (Senior Member, IEEE) received the B.Sc. degree in computer science from Wuhan University, China, in 1984, the M.E. degree in computer science from the University of Science and Technology of China, in 1989, and the Ph.D. degree in computer science from Australian National University, in 1998. He is a Professor with the Department of Computer Science, City University of Hong Kong. Prior to that, he was a Professor with the Australian National University. His research interests include design and analysis of energy efficient routing protocols for wireless ad hoc and sensor networks, mobile edge computing (MEC), network function virtualization (NFV), digital twins and Internet of Things, software-defined networking (SDN), design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory. He currently serves as an Editor in the Editorial Board of IEEE TRANSACTIONS ON COMMUNICATIONS.

Xiuzhen Cheng (Fellow, IEEE) received the M.S. and Ph.D. degrees in computer science from the University of Minnesota Twin Cities, in 2000 and 2002, respectively. She is a Professor with the School of Computer Science and Technology, Shandong University, Qingdao, China. She worked as a Professor with the Department of Computer Science, George Washington University, Washington, DC, USA, from 2013 to 2017. She worked as a Program Director with the U.S. National Science Foundation (NSF) from April to October, in 2006 (full time), and from April 2008 to May 2010 (part time). She has published more than 170 peer-reviewed papers. Her current research interests include cyber physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She received the NSF CAREER Award in 2004. She is a member of ACM.