DT-Empowered, Social-Aware Service Provisioning in Edge Computing

Jing Li, Jianping Wang, *Fellow, IEEE*, Weifa Liang, *Fellow, IEEE*, Jie Wu, *Fellow, IEEE*, Quan Chen, *Member, IEEE*, and Zichuan Xu, *Member, IEEE*

***Abstract*—The Internet of Things (IoT) is gathering paces in the new era of Industry 4.0, and the Digital Twin (DT) technology bridges the gap between the bursting amounts of data generated by IoT devices and the user requirements for real-time data processing. DT services maintain living digital models of physical objects, and a DT network enables comprehensive service provisioning with the global knowledge of a group of DTs. On the other hand, exposing serverless computing at network edges, the recent advances in Mobile Edge Computing (MEC) introduce new inspirations to the DT landscape that ensure fine-grained resource management and low network-wide delay of DT services. However, social relationships among IoT devices and DT data privacy impact DT orchestrations. In this paper, we first design a differential privacy-based federated learning framework to build a DT network for DT services in response to user requests in an MEC, thereby enhancing the Quality of Services (QoS). Built upon the proposed framework, we then formulate two novel social-aware DT placement problems: the static social-aware S_DT placement problem, and the dynamic social-aware S_DT placement problem, respectively. We also show the NP-hardness of the defined problems. Then, we formulate an Integer Linear Program (ILP) solution to the static social-aware S_DT placement problem when the problem size is small; otherwise we develop an approximation algorithm with a provable approximation ratio for it. Third, we study the dynamic social-aware S_DT placement problem when requests arrive one by one without the knowledge of future request arrivals over the time horizon, for which we devise an online algorithm with a provable competitive ratio. Finally, we conduct simulations to evaluate the performance of the proposed algorithms. Simulation results show that the proposed algorithms outperform their counterparts, improving the performance compared with their baselines by no less than 14.9%.**

***Index Terms*—Digital twin (DT) placement, mobile edge computing, social-aware DT relationships, DT-enabled service provisioning, approximation algorithm, online algorithm, federated learning, differential privacy budget, resource allocation.**

Received 14 April 2025; revised 13 October 2025 and 26 January 2026; accepted 26 March 2026; approved by IEEE TRANSACTIONS ON NETWORKING Editor A.-C. Pang. Date of publication 20 April 2026; date of current version 28 April 2026. The work of Jing Li was supported in part by Hong Kong Research Grants Council (RGC) through the Collaborative Research Fund (CRF) under Grant C1042-23GF; and in part by the Post-Doctoral Fellowship Award from the Research Grants Council of Hong Kong Special Administrative Region, China, under Project CityU PDFS2425-1S02. The work of Jianping Wang was supported by Hong Kong Research Grants Council (RGC) through the Collaborative Research Fund (CRF) under Grant C1042-23GF. The work of Weifa Liang was supported in part by Hong Kong Research Grants Council (RGC) through the Collaborative Research Fund (CRF) under Grant C1042-23GF; and in part by the Research Grants Council of Hong Kong Special Administrative Region, China, under Grant CityU 11202723, Grant CityU 11202824, Grant 7005845, Grant 8730094, and Grant 9380137. The work of Zichuan Xu was supported in part by NSFC under Grant 62172068 and in part by Shandong Provincial Natural Science Foundation under Grant ZR2023LZH013. *(Corresponding author: Jianping Wang.)*

Jing Li, Jianping Wang, and Weifa Liang are with the Department of Computer Science, City University of Hong Kong, Hong Kong, China (e-mail: jing.li@cityu.edu.hk; jianwang@cityu.edu.hk; weifa.liang@cityu.edu.hk).

Jie Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: jiewu@temple.edu).

Quan Chen is with the School of Computer, Guangdong University of Technology, Guangzhou 510006, China (e-mail: quan.c@gdut.edu.cn).

Zichuan Xu is with the School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: z.xu@dlut.edu.cn).

Digital Object Identifier 10.1109/TON.2026.3684790

I. INTRODUCTION

PROPELLED by the increasing scale of digitization, the Internet of Things (IoT) is opening the door to a smarter world with its capacity to sense data in physical environments [23]. Meanwhile, Digital Twins (DTs) are being rolled out as virtual representations of physical objects to reflect their real-time statuses, augmenting the performance of a surging number of IoT devices [1]. DTs are data-intensive, and their continuous synchronizations with physical objects require real-time data analytics [34].

Different from traditional cloud computing with significant service delays, Mobile Edge Computing (MEC) deploys cloudlets (edge servers) near users with the nature of ubiquitousness and low network delay to ensure fast responses and timely DT maintenance [16], [21], [42]. Albeit with these benefits of MEC, it raises issues about resource scarcity and inefficient resource management [15]. Taking advantage of the container technology, serverless computing provides fine-grained resource allocation with high elasticity for MEC [39]. In this context, serverless edge computing integrates MEC and serverless computing, and has emerged as a crucial research area and endows DTs with new vigor [33].

With the accelerated penetration of DTs, the DT network paradigm is increasingly expected to deliver comprehensive DT services to users, through grouping a collection of DTs and analyzing their global information [34]. However, building a DT network confronts significant challenges, including the growing concerns on privacy and security, and frequent network communications. This highlights the importance of distributed learning architectures, among which the federated learning framework is envisioned as a key paradigm for IoT and 6G networking, where models are trained locally and the model parameters, rather than raw data, are uploaded to a server for aggregation [14]. Nevertheless, federated learning remains vulnerable to potential leakages of private data, through examining the differences of model

2998-4157 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

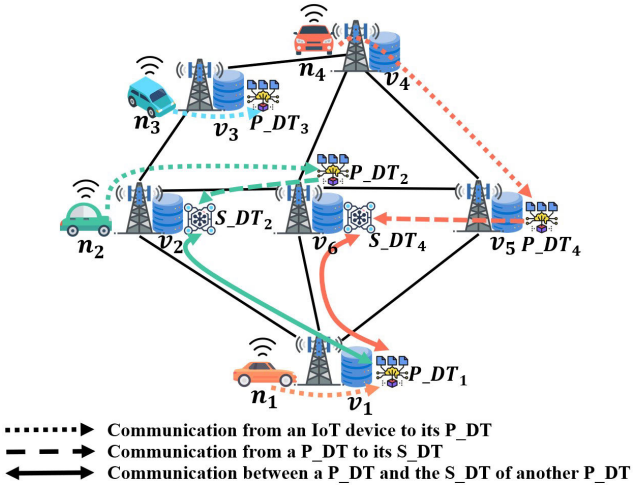


Fig. 1. An illustrative example of a DT network for a request in an MEC network where each Access Point (AP) has a co-located cloudlet. Each IoT device has a *Primary DT* (P_DT) deployed in a cloudlet, i.e., P_DT_1, P_DT_2, P_DT_3 and P_DT_4 of IoT devices n_1, n_2, n_3 and n_4 are deployed in cloudlets v_1, v_6, v_3, and v_5, respectively. IoT device n_1 issues a request and selects n_2 and n_4 for DT data, through instantiating containers to implement their *Sub-DTs (S_DTs)* with the needed features, S_DT_2 and S_DT_4, in cloudlets v_2 and v_6, respectively. Thus, the DT network of the request consists of P_DT_1, S_DT_2 and S_DT_4, where P_DT_1 transmits its trained global model to S_DT_2 and S_DT_4 for their local training. The trained models of S_DT_2 and S_DT_4 are then sent back to P_DT_1 for aggregation, by a differential privacy-based federated learning framework.

parameters that are uploaded via local devices. To this end, a privacy-preserving method - differential privacy, which prevents data leakage through adding artificial noise, has been widely adopted in practice [36].

Recent research results demonstrate that the emerging notion of Social IoT (SIoT) relationships, such as ownership and co-location, among IoT devices impact the orchestration and management of DTs [7], [8]. On behalf of their physical counterparts, DTs inherently inherit and maintain social relationships of IoT devices, which provide further insights that can inform the design and optimization of DT networks. E.g., when two IoT devices are owned by the same entity, their DTs may trust each other closely, and the added noise by the differential privacy policy can be set as a small value [9]. Also, when two devices are in the same area with frequent interactions, their DT data are important to each other.

In this paper, we study social-aware DT placements for admitting service requests in a DT-enabled MEC network, where each request is based on a DT network. We focus on the general case of data sharing among DTs to facilitate the DT network construction for providing comprehensive DT services in edge environments. As illustrated in Fig. 1, each IoT device has a *Primary DT* (P_DT) placed in a cloudlet, containing all features of the IoT device through monitoring its state continuously. An IoT device may issue a request for the DT data of other IoT devices, i.e., a request has a candidate set of IoT devices as its potential participants through inter-twin communication [35], e.g., the driving simulation of a vehicle may need the DT data of other vehicles in the same area. Each selected IoT device then extracts the needed features from its P_DT to establish a *Sub-DT (S_DT)* to participate in the execution of the request, where each S_DT is deployed in a container. E.g., a traffic monitoring request prefers the features, such as locations and driving behaviors of vehicles. Namely, the DT network of a request consists of the P_DT of the IoT device issuing the request and the S_DTs of selected IoT devices for providing DT data.

The key differences between the P_DT and one S_DT of an IoT device lie in that the P_DT contains all features of the IoT device and is implemented by a network function for a long lifetime. In contrast, a S_DT is implemented by a short-lived container that contains partial features of the IoT device, which means a S_DT requires much less resources compared with its P_DT. The model of a S_DT built on the required features is trained to meet the requirements of the request, e.g., considering a request to simulate the driving of a vehicle in an area, the driving behaviors of other vehicles are needed, which can be simulated by their S_DTs. Meanwhile, a vehicle maintenance request pays much more attentions to the features such as the states of software and hardware of vehicles.

To protect DT data privacy, each request is executed by adopting a differential privacy-based federated learning framework, i.e., the P_DT of an IoT device issuing the request first uses itself to train a global model and then transmits the trained global model to the S_DT of each selected IoT device to establish a local model. Each of such a S_DT performs local training and sends the updated model to the P_DT for aggregation with an amount of allocated privacy budget, depending on the social relationships (e.g., trust) among IoT devices [9]. Each service request has a *global privacy budget* to bound its privacy leakage [27], and the total allocated privacy budget on S_DTs of the request should not exceed the global privacy budget.

We measure the Quality of Service (QoS) for admitting a request, i.e., its utility gain, based on the importance of the DT data of each chosen IoT device, and we model such data importance based on interaction intensiveness relationships between IoT devices, which is also indicated by their social relationships.

Providing efficient social-aware DT services in an MEC network poses several challenges. First, based on a differential privacy-based federated learning framework, how to select appropriate IoT devices for providing DT data demanded by each service request, subject to a given global privacy budget on the request? Second, how to deploy S_DTs of the selected IoT devices into cloudlets to build the DT network for each admitted request in an MEC network, considering limited resource capacities on cloudlets? Thirdly, how to cope with DT network-enabled request admissions through S_DT deployments to maximize the total utility of admitted requests, by exploring the social-aware DT relationships? Finally, how to handle the dynamic arrivals of DT network-enabled requests in an online manner and deploy S_DTs of admitted requests onto cloudlets without any future knowledge?

The novelty of the study of this paper lies in exploring the social-awareness among DTs, and leveraging this relationship for DT-assisted service provisioning in MEC. To provide privacy protection on DT data, a differential privacy-based federated learning framework is developed to respond to each service request. Built upon the proposed framework, novel

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

optimization problems for placing S_DTs in an MEC network are formulated, and performance-guaranteed algorithms for the problems are devised.

The main contributions of this paper are as follows.

- We formulate two novel social-aware S_DT placement problems for DT-enabled service provisioning in MEC, considering social relationships, data privacy and resource constraints, while illustrating the NP-hardness of the formulated problems.
- We propose an Integer Linear Program (ILP) solution for the static social-aware S_DT placement problem when the problem size is small, and devise an approximation algorithm for the problem with a provable approximation ratio.
- We develop an online algorithm for the dynamic social-aware S_DT placement problem with a provable competitive ratio, by leveraging the primal-dual dynamic updating technique, at the expense of bounded resource violations.
- We evaluate the algorithm performance by simulations. The results demonstrate that the proposed approximation and online algorithms are promising, which improves the performance of the benchmarks by no less than 14.9%.

The remainder of the paper is arranged as follows. Section II surveys related works on social-aware DT service provisioning in MEC. Section III introduces the system model, DT network framework, and problem definitions. Section IV proposes an approximation algorithm for the static social-aware S_DT placement problem. Section V devises an online algorithm for the dynamic social-aware S_DT placement problem. Section VI evaluates the algorithm performance, and Section VII concludes the paper.

II. Related Work

Network services in serverless edge computing: Serverless computing endows network service provisioning with great elasticity, which has been extended to edge environments, and serverless edge computing is receiving a surge of attention recently [24], [30], [33], [34], [39], [40], [41]. Li et al. [24] considered the choice of communication styles for assigning functions in serverless edge computing, and proposed a priority-based approach to minimize the completion time of applications. Pan et al. [30] addressed a problem of container caching in MEC networks to reduce system costs. They mapped the problem to the ski-rental problem to devise online algorithms. Shang et al. [33] formulated a problem for data flow and container placement in serverless edge computing networks, and introduced an online algorithm to mitigate both operational costs and service delays. Ma et al. [28] explored dynamic tradeoffs between communication and computing resources allocated to network function services. That is, when the communication resource in a system becomes scarce, the proposed algorithm replicates service instances to multiple cloudlets by utilizing rich computing resources; otherwise the algorithm routes service requests to service hosts (cloudlets) for processing and then routes the results back to their users, by making use of abundant communication resource. Pasteris et al. [31] considered the problem of placing multiple services in a heterogeneous MEC system to maximize the total reward, by showing NP-hardness and proposing a $\left(1-\frac{1}{e}\right)/4$-approximation algorithm for the problem. Xu et al. [39] paid attention to reducing the age of big data query results. They developed an approximate solution for query admissions and an online learning approach for dynamic query admissions in serverless edge computing. However, none of the aforementioned works considered DT services in serverless edge computing.

Social-aware DT services: In recent years, DT techniques applied in edge environments have been extensively investigated [2], [3], [16], [17], [18], [19], [20], [34], [35], [42], [45]. For instance, Ai et al. [2], [3] studied the tradeoffs of resource allocations between DT-empowered service model retraining and service fidelity of service models, and proposed efficient algorithms for the placements of DTs and service model instances. Li et al. [18], [20] integrated object mobility in edge-cloud networks to dynamically deploy DTs in edge servers, while proposing online and approximation algorithms to provide users access to fresh DT data. Based on Lyapunov optimization, Lin et al. [26] presented an incentive-based solution to cope with dynamic DT service requirements in order to boost the long-term profit. Ren et al. [32] developed a Reinforcement Learning (RL)-based technique to enhance service quality, through constructing a DT network that is consistent with a real-world network. Yao et al. [42] presented an architecture for scheduling tasks and caching services in a DT edge network. They delivered a graph attention-based algorithm to improve the QoS. Zhang et al. [45] dealt with inference services in a DT-assisted MEC network, and they employed primary and secondary DTs for inference services by leveraging DT migrations. A few recent studies take social relationships into consideration for providing DT services [7], [8], [46]. Chukhno et al. [7] focused on the optimal social-aware deployment of DTs in MEC, and provided an efficient solution to minimize the communication delay (i) between objects and their DTs; and (ii) among DTs of friend IoT devices. Chukhno et al. [8] also examined the dynamic placements of social-aware DTs, and provided a heuristic algorithm to mitigate the communication delay. Zhao et al. [46] explored the social relationships among vehicles in DT-based vehicle edge computing, and proposed intelligent partial offloading strategies to optimize the system performance. However, none of the aforementioned works adopted federated learning to protect privacy in DT service provisioning.

Federated learning-enabled DT services: Federated learning allays user privacy worries and offers a promising distributed machine learning architecture for enabling DT services [1], [14], [47]. Abdulrahman et al. [1] leveraged federated learning techniques to build a shared DT model, and designed a trust-based client clustering method, incorporating the social relationship among clients. They proposed an intelligent framework to optimize communication costs and computing resources. Jiang et al. [14] exploited blockchain and cooperative federated learning to build DTs with flexibility and security. They designed an auction-based scheme to maximize social welfare in edge networks.

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

Zhou et al. [47] constructed a federated RL scheme with knowledge distillation in MEC to reduce communication cost, while enhancing the model performance, where DTs are adopted to obtain real-time data for model training. However, none of the aforementioned works considered social relationships among IoT devices to determine the allocated privacy budgets and to determine the importance of the collected DT data.

Different from the aforementioned works, in this paper we study social-aware DT placements in an MEC network for DT-assisted service provisioning, by exploring social relationships among DTs of different objects. We develop a differential privacy-based federated learning framework to construct a DT network for each request admission, and devise efficient approximation and online algorithms for the social-aware DT placement problems with guaranteed performance. It must be mentioned that this journal extension comes from a conference paper [22]. The main differences from its conference version lie in a new problem - the dynamic social-aware DT placement problem, and an online algorithm guaranteed performance for the problem is devised. Also, the detailed proofs of the theorems and lemmas for the approximation algorithm in the conference version are provided, which were sketched due to lack of space.

III. Preliminaries

A. System Model

Consider an MEC network $G = (V, E)$, consisting of a set V of Access Points (APs) and a set E of optic links interconnecting APs. Each AP has a co-located cloudlet, and we use the notation $v \in V$ to indicate a cloudlet or its co-located AP. Let M_v be the available memory resource in a cloudlet $v \in V$. The system model is illustrated in Fig. 1.

Denote by q_{n_1,n_2} the trust of IoT device n_1 to IoT device n_2 with $0 \leq q_{n_1,n_2} \leq 1$ [9]. A larger value of q_{n_1,n_2} indicates that device n_1 trusts device n_2 with high fidelity. Denote by l_{n_1,n_2} the interaction intensiveness between IoT devices n_1 and n_2 with $0 \leq l_{n_1,n_2} \leq 1$ [8], where a larger l_{n_1,n_2} indicates that IoT device n_1 interacts with n_2 more frequently, i.e., the DT data of IoT device n_2 is more important for n_1.

B. Federated Learning Framework for Constructing a DT Network for Each Request

We introduce a differential privacy-based federated learning framework for constructing a DT network for each request. Each IoT device $n \in \mathcal{N}$ has a *Primary DT (P_DT)* placed in a cloudlet $v_n \in V$, which contains all features of IoT device n through monitoring its state continuously. The P_DT is implemented by a network function for a long lifetime with storing all the historical data of its IoT device, and we can see that the dynamic deployment of P_DTs will cause a non-negligible migration cost, therefore, we adopt the static deployment of P_DTs for simplicity.

There is a set R of DT service requests. Each request $r \in R$ is issued by an IoT device n_r and executed by its P_DT, which may need the DT data of a candidate set of IoT devices $N_r \subseteq \mathcal{N}$ as potential participants through inter-twin communication [35].

Given a request r, each selected IoT device $n \in N_r$ extracts the needed features from its P_DT to establish a *Sub-DT (S_DT)*, and each S_DT is deployed in a container in the MEC network. Note that for a given P_DT, it can have multiple S_DTs with different S_DTs having different features. The demanded resource of a container is mainly the memory resource [41], according to serverless platforms [4], [5], because the amount of data loaded to the memory is related to the amount of memory assigned to the container [39]. Denote by $m_{r,n}$ the amount of memory resource needed in a container to deploy a S_DT of IoT device n for request r.

Following the federated learning framework, each request r can be implemented through building a DT network consisting of the P_DT of the IoT device n_r issuing the request r and the S_DTs of the chosen candidate IoT devices in N_r. Especially, the P_DT of IoT device n_r first trains a global model and then transmits the trained global model to the S_DTs of all selected IoT devices. The S_DTs then conduct local training and transmit the updated model to the P_DT for aggregation. Because the data volume of a S_DT of an IoT device is much less than that of its P_DT, adopting S_DTs can reduce the local training time of a request on the DT data, thereby reducing the data traffic burden on the MEC network.

C. Differential Privacy and Privacy Budget Constraint

The model of a S_DT can be trained based on its personal data, which, however, is likely to leak some of the DT information. The differential privacy policy makes a commitment to enabling secure model sharing by offering dependable assurances on the data exposure of individuals [36].

Definition 1 ((ϵ, δ)-differential privacy [11]): An algorithm $F : \mathcal{D} \longmapsto \Lambda$ is (ϵ, δ)-differential privacy if for any set $\Omega \subseteq \Lambda$ and for any two neighboring datasets $\mathbb{D}_1$ and $\mathbb{D}_2$, they is only one sample difference, i.e., $|\mathbb{D}_1| \leq |\mathbb{D}_2| + 1$ or $|\mathbb{D}_2| \leq |\mathbb{D}_1| + 1$, $\mathbb{D}_1, \mathbb{D}_2 \subseteq \mathcal{D}$ and vice versa, $Pr[F(\mathbb{D}_1) \in \Omega] \leq e^{\epsilon} \cdot Pr[F(\mathbb{D}_2) \in \Omega] + \delta$, where $\epsilon > 0$ is the privacy budget and indicates the upper bound of the degree of privacy leakage. δ indicates the probability that the privacy leakage exceeds the upper bound.

A smaller privacy budget ϵ assigned to a piece of data indicates that the data will receive greater privacy protection. In terms of the DT data privacy issue, a S_DT adopts the differential privacy strategy through adding the Gaussian noise to model parameters to avoid revealing the original data [36].

We assume that each request $r \in R$ has a global privacy budget B_r to bound its privacy leakage [27]. The required privacy protection level can be obtained based on social relationships (trust) among IoT devices [38]. Following [9], [10], the privacy budget for data communication from IoT device n_1 to IoT device n_2 is calculated as follows.

$$\epsilon_{n_1,n_2} = \frac{q_{n_1,n_2}}{q_{n_1,n_2} + \tau} \cdot \kappa, \tag{1}$$

where q_{n_1,n_2} is the value of the trust from IoT device n_1 to device n_2 with $0 \leq q_{n_1,n_2} \leq 1$, τ is a constant with $0 < \tau \leq 1$

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

to avoid the denominator to be zero, and $\kappa > 0$ is a tuning parameter to scale the allocated privacy budget. The values of τ and κ can be set by analyzing the historical traces.

The global privacy budget constraint for each request r is as follows. Suppose that a request r issued by IoT device n_r is admitted. Then, it establishes the S_DTs of each IoT device in a set $\mathbb{N}_r$ with $\mathbb{N}_r \subseteq N_r \subseteq \mathcal{N}$, while the total privacy budget allocated of request r is no more than B_r, i.e.,

$$\sum_{n \in \mathbb{N}_r} \epsilon_{n,n_r} \leq B_r. \tag{2}$$

D. Utility Gain

Recall that IoT device n_r issues a request r executed on its P_DT for the DT data from a candidate set of IoT devices $N_r \subseteq \mathcal{N}$ as potential participants. The utility gain of such a request r depends on the importance of the DT data of each chosen IoT device from the candidate set N_r.

Considering the social relationships, we define the utility gain $u_{r,n}$ of selecting an IoT device n from N_r as

$$u_{r,n} = \frac{l_{n_r,n}}{\sum_{n' \in N_r} l_{n_r,n'}}, \tag{3}$$

where $l_{n_r,n}$ is the interaction intensiveness between IoT devices n_r and n with $0 \leq l_{n_r,n} \leq 1$ [8], indicating the importance of the DT data of IoT device n for n_r.

Especially, we assume that the utility gain of a request is the sum of the utility gains of retrieving DT data from all selected IoT devices, and this metric definition is similar to throughput in literature [6], [25], [43], [44].

E. Problem Definitions

Definition 2: Given an MEC network $G = (V, E)$, a set $\mathcal{N}$ of IoT devices, a set R of requests, and each request $r \in R$ has a candidate set of IoT devices N_r for S_DT deployment with a given global privacy budget B_r, *the static social-aware S_DT placement problem* is to maximize the utility gain of the requests, through deploying S_DTs in G, subject to a given global privacy budget and memory capacities on cloudlets.

The static social-aware S_DT placement problem can be mathematically defined as follows. Let $x_{r,n,v}$ be a binary variable, where $x_{r,n,v} = 1$ presents that the S_DT of IoT device n is deployed in a cloudlet $v \in V$ for request r, and $x_{r,n,v} = 0$ otherwise. If there is no S_DT deployed for request r with $x_{r,n,v} = 0, \forall n \in N_r, \forall v \in V$, then request r is rejected.

The ILP of the static social-aware S_DT placement problem is formulated as follows.

$$\text{Maximize} \sum_{r \in R} \sum_{n \in N_r} \sum_{v \in V} u_{r,n} \cdot x_{r,n,v}, \tag{4}$$

subject to: from (1) to (3),

$$\sum_{r \in R} \sum_{n \in N_r} m_{r,n} \cdot x_{r,n,v} \leq M_v, \quad \forall v \in V \tag{5}$$

$$\sum_{n \in N_r} \sum_{v \in V} \epsilon_{n,n_r} \cdot x_{r,n,v} \leq B_r, \quad \forall r \in R \tag{6}$$

$$\sum_{v \in V} x_{r,n,v} \leq 1, \quad \forall r \in R, \forall n \in N_r \tag{7}$$

$$x_{r,n,v} \in \{0,1\}, \ \forall r \in R, \ \forall n \in N_r, \ \forall v \in V, \tag{8}$$

where Constraint (5) ensures the memory capacity on each cloudlet. Constraint (6) ensures the global privacy budget on each request by Eq. (2). Constraint (7) indicates that each S_DT is deployed in at most one cloudlet.

In the following, we consider the dynamic request admissions over a given monitoring period, i.e., the DT network-enabled requests arrive one by one without the knowledge of future arrivals, and each incoming request must be responded by either acceptance or rejection immediately. In this way, we define the dynamic social-aware S_DT placement problem as follows.

Definition 3: Given an MEC network $G = (V, E)$ and a given time horizon T, a set $\mathcal{N}$ of IoT devices with each generating sensory data continuously, and a sequence R of service requests that arrives one by one without the knowledge of future request arrivals, *the dynamic social-aware S_DT placement problem* is to maximize the accumulative utility gain for the given time horizon T, through deploying S_DTs in G for each incoming request by either accepting or rejecting it immediately, subject to the privacy budget of the request and memory capacities on cloudlets in G.

F. NP-Hardness of the Defined Problems

Theorem 1: The static social-aware S_DT placement problem for DT network-enabled service provisioning in an MEC network is NP-hard.

Proof: We show the NP-hardness of the problem through a reduction from the maximum-profit Generalized Assignment Problem (GAP) as follows.

We consider a special case of the static social-aware S_DT placement problem without the privacy budget constraint. For this maximum-profit GAP instance, there are $|V|$ bins, and each bin $v \in V$ has memory capacity M_v, i.e., the memory resource capacity on cloudlet v. There are $\sum_{r \in R} |N_r|$ items, and each item $i_{r,n}$ has a size $m_{r,n}$, i.e., the amount of memory resource needed in a container to deploy a S_DT of IoT device n for request r. Assigning item $i_{r,n}$ to a bin brings a profit $u_{r,n}$. The maximum-profit GAP is to maximize the total profit of the items assigned to bins, considering the capacities on bins. This special maximum-profit GAP is equivalent to a special static social-aware S_DT placement problem, and the static social-aware S_DT placement problem is NP-hard due to the fact that the maximum-profit GAP is NP-hard [29]. ■

Corollary 1: The dynamic social-aware S_DT placement problem for DT network-enabled service provisioning in an MEC network is NP-hard.

Proof: We show that the static social-aware S_DT placement problem is NP-hard by Theorem 1, which is a special case of the dynamic social-aware S_DT placement problem when all DT service requests are given in advance. Hence, the dynamic social-aware S_DT placement problem is NP-hard, too. ■

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

IV. Approximation Algorithm for the Static Social-Aware S_DT Placement Problem

In this section, we deal with the static social-aware S_DT placement problem. We first propose an approximation algorithm for it. We then analyze the approximation ratio and time complexity of the proposed algorithm.

A. Overview of the Approximation Algorithm

The idea behind the proposed algorithm is as follows. We first obtain a potential solution $\mathbb{S}$, which is a set of S_DTs deployed in cloudlets with violations on the memory capacities of cloudlets and global privacy budgets of requests. We then partition set $\mathbb{S}$ into two disjoint subsets S_1 and S_2 such that S_DTs in neither of them has any violations on global privacy budgets of requests, and choose one from S_1 and S_2 with a larger utility gain by denoting it as S'. We further partition set S' into two disjoint subsets S_3 and S_4 such that neither of them has any violations on memory capacities of cloudlets. We finally choose the one from S_3 and S_4 with a larger utility gain as the final solution to the problem.

B. Approximation Algorithm

It is observed that the deployment of a S_DT consumes memory resource and a privacy budget. Referring to the global privacy budget constraint (6), we define *the privacy consumption ratio* $\sigma(\lambda^l)$ of placing the lth S_DT λ^l as follows.

$$\sigma(\lambda^l) = \frac{\epsilon(\lambda^l)}{B(\lambda^l)}, \tag{9}$$

where $\epsilon(\lambda^l)$ is the consumed privacy budget of λ^l by Eq. (1).

To guide the deployment of S_DTs, we adopt a metric - the ratio $\rho(\lambda^l)$ for deploying the lth S_DT λ^l, with

$$\rho(\lambda^l) = \frac{u(\lambda^l)}{m(\lambda^l) \cdot \sigma(\lambda^l)}, \tag{10}$$

where $u(\lambda^l)$ is the utility of deploying λ^l calculated by Eq. (3), and $m(\lambda^l)$ is the memory resource consumed by λ^l.

The approximation algorithm proceeds iteratively as follows. Let Λ be the set of all candidate S_DTs of requests with $\Lambda = \{\lambda_{r,n} \mid r \in R, n \in N_r\}$. The set of deployed S_DTs is $\mathbb{S} = \emptyset$ initially. Denote by $\mathbb{S}^{l-1}$ the set of the first $l-1$ S_DTs placed before placing the lth S_DT, where $\mathbb{S}^l = \mathbb{S}^{l-1} \cup \{\lambda^l\}$.

In each iteration, we identify a S_DT $\lambda_{r,n} \in \Lambda \backslash \mathbb{S}^{l-1}$ as λ^l with the largest $\rho(\lambda^l)$ in Eq. (10), while updating $\mathbb{S}^l = \mathbb{S}^{l-1} \cup \{\lambda^l\}$. We partition $\mathbb{S}$ into two subsets S_1 and S_2 through examining the privacy budget consumption of requests, and determine at which cloudlet to deploy λ^l through examining the memory resource consumption of cloudlets as follows.

Let $\mathcal{B}_r(\mathbb{S}^l)$ be the accumulative privacy budget consumption of request r by deploying S_DTs in $\mathbb{S}^l$. For each identified S_DT λ^l, if the consumed privacy budget of request r by $\mathbb{S}^l$ is greater than its privacy budget B_r, i.e., $\mathcal{B}_r(\mathbb{S}^l) > B_r$, we put λ^l into set S_1; otherwise ($\mathcal{B}_r(\mathbb{S}^l) \leq B_r$), we put λ^l into set S_2. Also, if $\mathcal{B}_r(\mathbb{S}^l) \geq B_r$, we will no longer consider request r by removing its rest S_DTs from Λ, i.e., $\Lambda = \Lambda \backslash \{\lambda_{r,n} \mid n \in N_r\}$. S_1 and S_2 are disjoint and $\mathbb{S} = S_1 \cup S_2$.

Algorithm 1 Approximation Algorithm for the Static Social-Aware S_DT Placement Problem

Input: An MEC network $G = (V, E)$, a set $\mathcal{N}$ of IoT devices, and a set R of DT service requests.

Output: Maximize the utility gain of deploying S_DTs in cloudlets.

$\mathbb{S}^0 \leftarrow \emptyset$; $\hat{\mathbb{S}} \leftarrow \emptyset$; $S_1 \leftarrow \emptyset$; $S_2 \leftarrow \emptyset$; $S_3 \leftarrow \emptyset$; $S_4 \leftarrow \emptyset$;
$\mathbb{V} \leftarrow V$; $\Lambda \leftarrow \{\lambda_{r,n} \mid r \in R, n \in N_r\}$; $l \leftarrow 1$;
while $\mathbb{V} \neq \emptyset$ **or** $\Lambda \backslash \mathbb{S}^{l-1} \neq \emptyset$ **do**
  Identify a S_DT $\lambda_{r,n} \in \Lambda \backslash \mathbb{S}^{l-1}$ as λ^l with the largest $\rho(\lambda^l)$ in Eq. (10); $\mathbb{S}^l \leftarrow \mathbb{S}^{l-1} \cup \{\lambda^l\}$;
  if $\mathcal{B}_r(\mathbb{S}^l) > B_r$ **then**
    $S_1 \leftarrow S_1 \cup \{\lambda^l\}$;
  else
    $S_2 \leftarrow S_2 \cup \{\lambda^l\}$;
  end if;
  if $\mathcal{B}_r(\mathbb{S}^l) \geq B_r$ **then**
    $\Lambda \leftarrow \Lambda \backslash \{\lambda_{r,n} \mid n \in N_r\}$;
  end if;
  Identify a cloudlet $v \in \mathbb{V}$ with the largest residual memory resource, and place λ^l to cloudlet v;
  if $\mathcal{M}_v(\mathbb{S}^l) > M_v$ **then**
    $\hat{\mathbb{S}} \leftarrow \hat{\mathbb{S}} \cup \{\lambda^l\}$;
  end if;
  if $\mathcal{M}_v(\mathbb{S}^l) \geq M_v$ **then**
    $\mathbb{V} \leftarrow \mathbb{V} \backslash \{v\}$;
  end if;
  $l \leftarrow l + 1$;
end while;
$S' \leftarrow \arg\max_{S \in \{S_1, S_2\}} \sum_{\lambda^l \in S} u(\lambda^l)$;
$S_3 \leftarrow \hat{\mathbb{S}} \cap S'$;
for each S_DT $\lambda^l \in S_3$ **do**
  if the assigned cloudlet of λ^l has no capacity violation by S' **then**
    $S_3 \leftarrow S_3 \backslash \{\lambda^l\}$;
  end if
end for;
$S_4 \leftarrow S' \backslash S_3$;
return $\arg\max_{S \in \{S_3, S_4\}} \sum_{\lambda^l \in S} u(\lambda^l)$.

Note that $\mathbb{S}$ is partitioned into two sets S_1 and S_2, and denote the one with the larger utility as S'. Because S' is a subset of $\mathbb{S}$, a S_DT in $\hat{\mathbb{S}}$ may not cause capacity violation on a cloudlet by S' anymore. We then obtain S_3 as follows. We first let $S_3 = \hat{\mathbb{S}} \cap S'$. For each S_DT λ^l in S', if the cloudlet in which λ^l is allocated has no capacity violation by S', we then remove λ^l from S_3.

Let $S_4 = S' \backslash S_3$. S' is further partitioned into two disjoint subsets S_3 and S_4 with $S' = S_3 \cup S_4$. We claim that deploying S_DTs in either S_3 or S_4 causes no violation on global privacy budget constraints of requests and memory capacity constraints on cloudlets by Lemma 3. We finally choose one from S_3 and S_4 with the larger utility as the final solution to the static social-aware S_DT placement problem.

The detailed algorithm is shown in `Algorithm 1`.

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

C. Algorithm Analysis

Lemma 1: Suppose that Algorithm 1 terminates when all requests run out of global privacy budget. Given a potential solution $\mathbb{S}$ delivered by Algorithm 1, let $\mathbb{S}^{opt}$ be the set of placed S_DTs in the optimal solution to the static social-aware S_DT placement problem. Let $\mathbb{S}_r^{opt}$ be the set of deployed S_DTs for request r by $\mathbb{S}^{opt}$ with $\mathbb{S}^{opt} = \cup_{r\in R}\mathbb{S}_r^{opt}$. Similarly, let $\mathbb{S}$ be the potential solution delivered by Algorithm 1 with $\mathbb{S} = \cup_{r\in R}\mathbb{S}_r$, where $\mathbb{S}_r$ is the set of deployed S_DTs for request r. Then, (i) $\rho(\lambda^l) \geq \rho(\lambda^*), \forall r \in R, \forall \lambda^l \in \mathbb{S}_r, \forall \lambda^* \in \mathbb{S}_r^{opt} \setminus \mathbb{S}_r$; and (ii) $\sum_{\lambda^l\in\mathbb{S}} u(\lambda^l) \geq \frac{m_{min}}{m_{max}} \cdot \sum_{\lambda^*\in\mathbb{S}^{opt}\setminus\mathbb{S}} u(\lambda^*)$, where m_{max} and m_{min} are the maximum and minimum amounts of memory resource consumed among all S_DTs, respectively.

Proof: (i) If $\mathbb{S}_r^{opt} \setminus \mathbb{S}_r = \emptyset$, the lemma follows. Otherwise, because the S_DT identified by Algorithm 1 has the largest $\rho(\lambda^l)$ at each iteration, and the potential solution $\mathbb{S}$ allows resource violations. We thus have $\rho(\lambda^l) \geq \rho(\lambda^*)$, $\forall r \in R, \forall \lambda^l \in \mathbb{S}_r$, $\forall \lambda^* \in \mathbb{S}_r^{opt} \setminus \mathbb{S}_r$.

(ii) Let $\lambda^*_{r,max} = \arg\max_{\lambda^*\in\mathbb{S}_r^{opt}\setminus\mathbb{S}_r} \rho(\lambda^*),\ \forall r \in R$, then

$$\sum_{\lambda^l\in\mathbb{S}} u(\lambda^l) = \sum_{r\in R}\sum_{\lambda^l\in\mathbb{S}_r} u(\lambda^l) = \sum_{r\in R}\sum_{\lambda^l\in\mathbb{S}_r} \rho(\lambda^l)\cdot m(\lambda^l)\cdot\sigma(\lambda^l) \quad (11)$$

$$\geq \sum_{r\in R}\sum_{\lambda^l\in\mathbb{S}_r} \rho(\lambda^*_{r,max})\cdot m(\lambda^l)\cdot\sigma(\lambda^l) \quad (12)$$

$$\geq m_{min}\cdot\sum_{r\in R}\rho(\lambda^*_{r,max})\cdot\frac{\sum_{\lambda^l\in\mathbb{S}_r}\epsilon(\lambda^l)}{B_r} \quad (13)$$

$$\geq m_{min}\cdot\sum_{r\in R}\rho(\lambda^*_{r,max})\cdot\frac{\sum_{\lambda^*\in\mathbb{S}_r^{opt}\setminus\mathbb{S}_r}\epsilon(\lambda^*)}{B_r} \quad (14)$$

$$\geq m_{min}\cdot\sum_{r\in R}\sum_{\lambda^*\in\mathbb{S}_r^{opt}\setminus\mathbb{S}_r}\rho(\lambda^*)\cdot\frac{\epsilon(\lambda^*)}{B_r}$$

$$= m_{min}\cdot\sum_{r\in R}\sum_{\lambda^*\in\mathbb{S}_r^{opt}\setminus\mathbb{S}_r}\frac{u(\lambda^*)}{m(\lambda^*)\cdot\sigma(\lambda^*)}\cdot\frac{\epsilon(\lambda^*)}{B_r}$$

$$\geq \frac{m_{min}}{m_{max}}\cdot\sum_{r\in R}\sum_{\lambda^*\in\mathbb{S}_r^{opt}\setminus\mathbb{S}_r} u(\lambda^*)$$

$$\geq \frac{m_{min}}{m_{max}}\cdot\sum_{\lambda^*\in\mathbb{S}^{opt}\setminus\mathbb{S}} u(\lambda^*), \quad (15)$$

where Eq. (11) holds by the definition of $\rho(\lambda^l)$, i.e., Eq. (10). Ineq. (12) holds by (i) and the definition of $\lambda^*_{r,max}$. Ineq. (13) holds by Eq. (9). Ineq. (14) holds because Algorithm 1 terminates when the privacy budget consumed of each request r by $\mathbb{S}$ is no less than its global privacy budget B_r, while no request has its global privacy budget violated by the optimal solution. Ineq. (15) holds due to the definition of $\lambda^*_{r,max}$. ■

Lemma 2: Suppose Algorithm 1 terminates when all cloudlets run out of resources. Let $\mathbb{S}$ and $\mathbb{S}^{opt}$ be the potential solution by Algorithm 1 and the optimal solution to the static social-aware S_DT placement problem, respectively. Then,

$$\sum_{\lambda^l\in\mathbb{S}} u(\lambda^l) \geq \frac{\theta_{min}}{\theta_{max}}\sum_{\lambda^*\in\mathbb{S}^{opt}} u(\lambda^*), \quad (16)$$

where $\theta(\lambda^l) = \frac{u(\lambda^l)}{m(\lambda^l)}$, and θ_{max} and θ_{min} are the maximum and minimum values of $\theta(\lambda^l)$, respectively.

Proof:

$$\sum_{\lambda^l\in\mathbb{S}} u(\lambda^l) = \sum_{\lambda^l\in\mathbb{S}}\theta(\lambda^l)\cdot m(\lambda^l)$$

$$\geq \theta_{min}\cdot\sum_{\lambda^l\in\mathbb{S}} m(\lambda^l) \geq \theta_{min}\cdot\sum_{\lambda^*\in\mathbb{S}^{opt}} m(\lambda^*)$$

$$\geq \frac{\theta_{min}}{\theta_{max}}\sum_{\lambda^*\in\mathbb{S}^{opt}}\theta(\lambda^*)\cdot m(\lambda^*)$$

$$\geq \frac{\theta_{min}}{\theta_{max}}\sum_{\lambda^*\in\mathbb{S}^{opt}} u(\lambda^*) \quad (17)$$

where Ineq. (17) holds because the memory resource consumption of each cloudlet by $\mathbb{S}$ is no less than its capacity, while the optimal solution causes no resource violation. ■

Lemma 3: The solution delivered by Algorithm 1 for the static social-aware S_DT placement problem causes no violation on global privacy budget constraints of requests and no violation on memory capacity constraints on cloudlets.

Proof: Referring to Algorithm 1, if a request r has its global privacy budget fully utilized or has its global privacy budget constraint violated ($\mathcal{B}_r(\mathbb{S}^l) \geq B_r$), the request is no longer to be considered. Therefore, S_1 accommodates at most one S_DT for each request, while S_2 accommodates S_DTs, causing no violation on the global privacy budget constraints of requests. Assuming the privacy budget of a request r is sufficient for deploying a single S_DT of any candidate IoT device in N_r, we can observe that deploying S_DTs in either S_1 or S_2 causes no violation on privacy budget constraints of requests.

Denote by S' the set between S_1 and S_2 with the larger utility, which is further partitioned into S_3 and S_4. Especially, S_3 accommodates S_DTs which cause memory capacity violations on cloudlets, and S_3 has at most one S_DT assigned to a cloudlet, because a cloudlet is no longer to be considered when its capacity is fully utilized or violated. Meanwhile, S_4 accommodates S_DTs causing no violation on the memory capacities of cloudlets. Assuming the memory capacity on each cloudlet suffices for any single S_DT deployment, we observe the final solution (S_3 or S_4) causes neither violations on global privacy budgets of requests nor violations on memory capacity of cloudlets. ■

Theorem 2: Given an MEC network $G = (V, E)$, a set $\mathcal{N}$ of IoT devices, a set R of requests, each request $r \in R$ has a candidate set of IoT devices N_r for its S_DT deployment. There is an approximation algorithm, Algorithm 1, for the static social-aware S_DT placement problem with an approximation ratio of $\frac{1}{4}\cdot\min\left\{\frac{m_{min}}{m_{max}+m_{min}}, \frac{\theta_{min}}{\theta_{max}}\right\}$, and the algorithm takes $O(|R|^2\cdot|N|^2_{max} + |R|\cdot|N|_{max}\cdot|V|)$ time, where m_{max} and m_{min} are the maximum and minimum amounts of memory resource consumed by any S_DT, respectively. $\theta(\lambda^l) = \frac{u(\lambda^l)}{m(\lambda^l)}$ with θ_{max} and θ_{min} the maximum and minimum values of $\theta(\lambda^l)$, respectively, and $|N|_{max}$ is the maximum value of N_r.

Proof: We analyze the approximation ratio by distinguishing two cases. Case 1. Algorithm 1 terminates when all requests

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

run out of global privacy budgets; and Case 2. `Algorithm 1` terminates when all cloudlets run out of resources.

Case 1. By Lemma 1, we have

$$\sum_{\lambda^l \in \mathbb{S}} u(\lambda^l) \geq \frac{m_{min}}{m_{max}} \cdot \sum_{\lambda^* \in \mathbb{S}^{opt} \setminus \mathbb{S}} u(\lambda^*). \tag{18}$$

Then, the value of the optimal solution is

$$\begin{aligned}\sum_{\lambda^* \in \mathbb{S}^{opt}} u(\lambda^*) &\leq \sum_{\lambda^l \in \mathbb{S}} u(\lambda^l) + \sum_{\lambda^* \in \mathbb{S}^{opt} \setminus \mathbb{S}} u(\lambda^*) \\ &\leq \left(1 + \frac{m_{max}}{m_{min}}\right) \cdot \sum_{\lambda^l \in \mathbb{S}} u(\lambda^l), \quad \text{by Ineq. (18)} \\ &= \left(\frac{m_{max} + m_{min}}{m_{min}}\right) \cdot \sum_{\lambda^l \in \mathbb{S}} u(\lambda^l). \end{aligned} \tag{19}$$

The final solution value delivered by `Algorithm 1` is

$$\begin{aligned}&\max\left\{\sum_{\lambda^l \in S_3} u(\lambda^l), \sum_{\lambda^l \in S_4} u(\lambda^l)\right\} \\ &\geq \frac{1}{2} \cdot \max\left\{\sum_{\lambda^l \in S_1} u(\lambda^l), \sum_{\lambda^l \in S_2} u(\lambda^l)\right\} \geq \frac{1}{4} \sum_{\lambda^l \in \mathbb{S}} u(\lambda^l) \\ &\geq \frac{1}{4} \cdot \frac{m_{min}}{m_{max} + m_{min}} \cdot \sum_{\lambda^* \in \mathbb{S}^{opt}} u(\lambda^*), \quad \text{by Ineq. (19).} \end{aligned} \tag{20}$$

Case 2. By Lemma 2, we have

$$\sum_{\lambda^l \in \mathbb{S}} u(\lambda^l) \geq \frac{\theta_{min}}{\theta_{max}} \sum_{\lambda^* \in \mathbb{S}^{opt}} u(\lambda^*). \tag{21}$$

Similarly, the value of the final solution is

$$\begin{aligned}&\max\left\{\sum_{\lambda^l \in S_3} u(\lambda^l), \sum_{\lambda^l \in S_4} u(\lambda^l)\right\} \\ &\geq \frac{1}{2} \cdot \max\left\{\sum_{\lambda^l \in S_1} u(\lambda^l), \sum_{\lambda^l \in S_2} u(\lambda^l)\right\} \geq \frac{1}{4} \sum_{\lambda^l \in \mathbb{S}} u(\lambda^l) \\ &\geq \frac{1}{4} \cdot \frac{\theta_{min}}{\theta_{max}} \sum_{\lambda^* \in \mathbb{S}^{opt}} u(\lambda^*), \quad \text{by Ineq. (21).} \end{aligned} \tag{22}$$

Combining Ineq. (20) and (22), we have

$$\begin{aligned}&\max\left\{\sum_{\lambda^l \in S_3} u(\lambda^l), \sum_{\lambda^l \in S_4} u(\lambda^l)\right\} \\ &\geq \frac{1}{4} \cdot \min\left\{\frac{m_{min}}{m_{max} + m_{min}}, \frac{\theta_{min}}{\theta_{max}}\right\} \cdot \sum_{\lambda^* \in \mathbb{S}^{opt}} u(\lambda^*). \end{aligned} \tag{23}$$

The rest is to analyze the time complexity of `Algorithm 1`. There are at most $|R| \cdot |N|_{max}$ S_DTs to be deployed, i.e., there are at most $|R| \cdot |N|_{max}$ iterations in the proposed algorithm. Within each iteration, `Algorithm 1` identifies a S_DT with the largest $\rho(\lambda^l)$ and also identifies a cloudlet with the largest residual memory resource for the deployment of the S_DT. This takes $O(|R| \cdot |N|_{max} + |V|)$ time. Thus, the running time of `Algorithm 1` is $O(|R|^2 \cdot |N|^2_{max} + |R| \cdot |N|_{max} \cdot |V|)$. The theorem then follows. ■

V. Online Algorithm for the Dynamic Social-Aware S_DT Placement Problem

In this section, we deal with the dynamic social-aware S_DT placement problem over a given time horizon, where service requests arrive one by one without the knowledge of future arrivals, each incoming request must be responded by either accepting or rejecting it immediately. We propose an online algorithm for the problem, by applying the primal-dual dynamic updating technique [12]. We also analyze the performance of the proposed algorithm through analyzing its competitive ratio and time complexity.

A. Overview of the Online Algorithm

The general strategy for this problem proceeds as follows. We first formulate an ILP solution to the offline version of the dynamic social-aware S_DT placement problem, and let **P1** be the ILP solution. Denote by **P2** the linear relaxation of the ILP (**P1**), and **P3** is the dual of **P2**. A feasible solution to **P3** returns a feasible solution to **P2**, and a feasible solution for the dynamic social-aware S_DT placement problem is then derived from the solution to **P2**.

B. Online Algorithm

We provided an ILP formulation to the static social-aware S_DT placement problem in (4) in the previous section. We here adopt a binary variable y_r to denote whether request r is admitted ($y_r = 1$) or not ($y_r = 0$), and provide a modified ILP formulation for the offline version of the dynamic social-aware S_DT placement problem.

$$\textbf{P1}: \text{ Maximize } \sum_{r \in R} \sum_{n \in N_r} \sum_{v \in V} u_{r,n} \cdot x_{r,n,v}, \tag{24}$$

subject to:

$$\sum_{r \in R} \sum_{n \in N_r} m_{r,n} \cdot x_{r,n,v} \leq M_v, \qquad \forall v \in V \tag{25}$$

$$\sum_{n \in N_r} \sum_{v \in V} \epsilon_{n,n_r} \cdot x_{r,n,v} \leq B_r \cdot y_r, \qquad \forall r \in R \tag{26}$$

$$\sum_{v \in V} x_{r,n,v} \leq y_r, \qquad \forall r \in R, \forall n \in N_r \tag{27}$$

$$x_{r,n,v} \in \{0,1\}, \qquad \forall r \in R,\ \forall n \in N_r,\ \forall v \in V \tag{28}$$

$$y_r \in \{0,1\}, \qquad \forall r \in R. \tag{29}$$

If request r is admitted, Constraint (26) ensures that the global privacy budget on each request is not violated, while Constraint (27) ensures that each S_DT is deployed in at most one cloudlet. Otherwise (request r is rejected with $y_r = 0$), both Constraint (26) and (27) ensure that none of the S_DTs of request r is placed.

The Linear Program (LP) for **P2** is formulated as follows.

$$\textbf{P2}: \text{ Maximize } \sum_{r \in R} \sum_{n \in N_r} \sum_{v \in V} u_{r,n} \cdot x_{r,n,v}, \tag{30}$$

subject to:

(25), (26), and (27),

$$y_r \leq 1, \qquad \forall r \in R \tag{31}$$

$$x_{r,n,v} \geq 0, \qquad \forall r \in R,\ \forall n \in N_r,\ \forall v \in V \tag{32}$$

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

$$y_r \geq 0, \quad \forall r \in R, \tag{33}$$

where constraints (27) and (31) ensure that $x_{r,n,v} \leq 1$.

The dual **P3** of **P2** is formulated as follows.

$$\textbf{P3}: \text{Minimize } \sum\nolimits_{v\in V} M_v \cdot \alpha_v + \sum\nolimits_{r\in R} \mu_r, \tag{34}$$

subject to:

$$m_{r,n} \cdot \alpha_v + \epsilon_{n,n_r} \cdot \beta_r + \gamma_{r,n} - u_{r,n} \geq 0, \quad \forall r \in R,\ \forall n \in N_r,\ \forall v \in V \tag{35}$$

$$\mu_r - B_r \cdot \beta_r - \sum_{n\in N_r} \gamma_{r,n} \geq 0, \quad \forall r \in R \tag{36}$$

$$\alpha_v \geq 0,\ \beta_r \geq 0,\ \gamma_{r,n} \geq 0,\ \mu_r \geq 0, \quad \forall r \in R,\ \forall n \in N_r,\ \forall v \in V, \tag{37}$$

where α_v, β_r, $\gamma_{r,n}$ and μ_r are dual variables, referring to Constraints (25), (26), and (27), (31), respectively. Also, the dual variables are non-negative as indicated in Constraint (37).

From (35), we can see $\forall r \in R$,

$$\sum_{n\in N_r}\sum_{v\in V}\gamma_{r,n} \geq \sum_{n\in N_r}\sum_{v\in V}(u_{r,n} - m_{r,n}\cdot\alpha_v - \epsilon_{n,n_r}\cdot\beta_r). \tag{38}$$

We have $\forall r \in R$,

$$|V| \cdot \sum_{n\in N_r} \gamma_{r,n} \geq |V| \cdot \sum_{n\in N_r} (u_{r,n} - \epsilon_{n,n_r} \cdot \beta_r) - \sum_{n\in N_r}\sum_{v\in V} m_{r,n} \cdot \alpha_v \tag{39}$$

Therefore, $\forall r \in R$,

$$\sum_{n\in N_r} \gamma_{r,n} \geq \sum_{n\in N_r} (u_{r,n} - \epsilon_{n,n_r}\cdot\beta_r) - \frac{1}{|V|} \cdot \sum_{n\in N_r}\sum_{v\in V} m_{r,n}\cdot\alpha_v. \tag{40}$$

We can see $\sum_{n\in N_r} u_{r,n} = 1$ by the utility definition (3). Combining Ineq. (36) and (40), we have $\forall r \in R$,

$$\mu_r \geq 1 + \left(B_r - \sum_{n\in N_r} \epsilon_{n,n_r}\right)\cdot\beta_r - \frac{1}{|V|}\cdot\sum_{n\in N_r}\sum_{v\in V} m_{r,n}\cdot\alpha_v. \tag{41}$$

Because $\beta_r \geq 0$ is a non-negative dual variable, we set $\beta_r = 0$, and have $\forall r \in R$,

$$\mu_r \geq 1 - \frac{1}{|V|} \cdot \sum_{n\in N_r}\sum_{v\in V} m_{r,n} \cdot \alpha_v. \tag{42}$$

For the sake of convenience, we define the constant ζ_r as follows.

$$\zeta_r = \frac{1}{|V|} \cdot \sum_{n\in N_r} m_{r,n}. \tag{43}$$

Ineq. (42) becomes

$$\mu_r \geq 1 - \zeta_r \cdot \sum_{v\in V} \alpha_v. \tag{44}$$

We claim that feasible values of $\gamma_{r,n}$ can be found through carefully setting values of α_v and μ_r, subject to Ineq. (44). As a result, a feasible solution to **P3** with feasible values of dual variables α_v, β_r, $\gamma_{r,n}$ and μ_r can be obtained, which is shown in Lemma 4.

We now propose an online algorithm for the dynamic social-aware S_DT placement problem, i.e., **P1**, through updating variables in the primal and dual programming simultaneously. Initially, we set the value of $\alpha_v = 0$ for each cloudlet $v \in V$. We design an admission control policy for each arrived request r as follows.

If $1 - \zeta_r \cdot \sum_{v\in V} \alpha_v \leq 0$, request r is rejected, while we set μ_r as 0, and the values of α_v, $\forall v \in V$, do not change. Otherwise, request r is admitted, and we set μ_r as follows.

$$\mu_r = 1 - \zeta_r \cdot \sum_{v\in V} \alpha_v. \tag{45}$$

In the following, we show the admission of request r by updating the value of α_v. Given the residual capacities on cloudlets, we first place the S_DTs of request r by invoking `Algorithm 1`, i.e., identify the decision variables $\{x_{r,n,v}, \mid \forall n \in N_r,\ \forall v \in V\}$. Then, the amount $\mathbb{M}_{r,v}$ $\left(= \sum_{n\in N_r} m_{r,n} \cdot x_{r,n,v}\right)$ of memory resource in cloudlet v is consumed, by deploying S_DTs in cloudlet v for request r. For each cloudlet $v \in V$ with any S_DT deployed for request r, we update its α_v as follows.

$$\alpha_v = \alpha_v \cdot \left(1 + \frac{\zeta_r \cdot \mathbb{M}_{r,v}}{M_v \cdot \sum_{n\in N_r} m_{r,n}}\right) + \frac{\zeta_r \cdot \mathbb{M}_{r,v}}{M_v \cdot \sum_{n\in N_r} m_{r,n}}. \tag{46}$$

We can see that at least one S_DT is deployed by `Algorithm 1` for request r. We have shown the solution delivered by `Algorithm 1` for the static social-aware S_DT placement problem causes no violations on memory capacities on cloudlets in Lemma 3, due to the assumption that the memory capacity on each cloudlet suffices for any single S_DT deployment. However, the residual memory resource of a cloudlet may not be able to accommodate any single S_DT deployment when a request arrives in this dynamic social-aware S_DT placement problem. Therefore, the proposed online algorithm is likely to cause violations on memory capacities on cloudlets, which will be bounded later in Lemma 6.

The online algorithm for the dynamic social-aware S_DT placement problem is detailed in `Algorithm 2`.

C. Algorithm Analysis

The rest is to analyze the competitive ratio and time complexity of `Algorithm 2`.

Lemma 4: Given feasible dual variables α_v and μ_r, subject to Ineq. (44), there always exist feasible dual variables β_r and $\gamma_{r,n}$ to deliver a feasible solution to the dual problem **P3**.

Proof: (1) Recall that the dual variables α_v, β_r, $\gamma_{r,n}$ and μ_r are non-negative, as indicated in Constraint (37).

Constraint (36) with $\beta_r = 0$ means $\sum_{n\in N_r} \gamma_{r,n} \leq \mu_r$, where μ_r is non-negative, and Ineq. (44) is derived from Constraints (35) and (36) by setting $\beta_r = 0$. We can see that given feasible α_v and μ_r subject to Ineq. (44), there always exists a feasible value of $\gamma_{r,n}$ meeting Constraints (35) and (36). Hence, we can identify feasible values of $\gamma_{r,n}$ and β_r when updating α_v and μ_r, subject to Ineq. (44), and deliver a feasible solution to the dual **P3**. ■

Let ζ_{max} and ζ_{min} be the maximum and minimum values of ζ_r by (43). Let M_{max} and M_{min} be the maximum and minimum memory capacities on any cloudlet, respectively. Let

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2 Online Algorithm for the Dynamic Social-Aware S_DT Placement Problem

Input: An MEC network $G=(V,E)$, a set $\mathcal{N}$ of IoT devices, and a sequence R of arriving DT service requests without the knowledge of future request arrivals.

Output: Maximize the utility gain of deploying S_DTs in cloudlets of DT service requests in an online manner.

1: $\alpha_v \leftarrow 0, \forall v \in V$;
2: **while** a service request r arrives **do**
3: **if** $1-\zeta_r \cdot \sum_{v\in V} \alpha_v \leq 0$ **then**
4: Reject request r, and $\mu_r \leftarrow 0$;
5: **else**
6: Admit request r, and update μ_r by Eq. (45);
7: Construct a problem instance **P1** for the single request r, and invoke `Algorithm 1` to obtain a solution to deploy S_DTs for request r, with the residual memory capacities on cloudlets;
8: **for** each cloudlet $v \in V$ with any S_DT placement for request r **do**
9: Update α_v by Eq. (46);
10: **end for**;
11: **end if**;
12: **end while**.

m_{max} and m_{min} be the maximum and minimum amounts of memory resources consumed by any S_DT. Denote by $|N|_{max}$ the maximum number of candidate IoT devices of a request. Denote by $z_{r,v}$ a binary variable. When $z_{r,v}=1$, it means that at least a S_DT is deployed in cloudlet v for request r; otherwise $z_{r,v}=0$.

Lemma 5: When updating the dual variable α_v by `Algorithm 2`, we have

$$\alpha_v \geq \left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)^{\sum_{r\in R} z_{r,v}} -1; a \quad (47)$$

$$\alpha_v < \frac{1}{\zeta_{min}}\cdot\left(1+\frac{\zeta_{max}}{M_{min}}\right)+\frac{\zeta_{max}}{M_{min}}. \quad (48)$$

Proof: We first show that Ineq. (47) holds by induction. The right-hand side of Ineq. (47) is 0 before requests arrive. Therefore, the induction hypothesis holds with $\alpha_v=0$ initially. Let $\alpha_v(before)$ and $\alpha_v(after)$ represent the values of α_v before and after the arrival of request r'. We show that Ineq. (47) holds by distinguishing two cases as follows.

Case 1. α_v is not updated, i.e., $z_{r',v}=0$, because of the rejection of DT service request r', or admitting r' with none of its S_DT deployed in cloudlet v. Because α_v is not updated, i.e., $\alpha_v(after)=\alpha_v(before)$, then

$$\begin{aligned}\alpha_v(after)&=\alpha_v(before)\\&=\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)^{\sum_{r\in R\setminus\{r'\}} z_{r,v}}-1\\&\geq\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)^{\sum_{r\in R\setminus\{r'\}} z_{r,v}}\cdot\\&\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)^{z_{r',v}}-1,\ \text{by} z_{r',v}=0\\&=\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)^{\sum_{r\in R} z_{r,v}}-1.\end{aligned} \quad (49)$$

Case 2. α_v is updated, i.e., $z_{r',v}=1$, because of admitting DT service request r' with at least one S_DT deployed in cloudlet v. Referring to the update function (46),

$$\begin{aligned}\alpha_v(after)&=\alpha_v(before)\cdot\left(1+\frac{\zeta_{r'}\cdot\mathbb{M}_{r',v}}{M_v\cdot\sum_{n\in N_{r'}} m_{r',n}}\right)\\&+\frac{\zeta_{r'}\cdot\mathbb{M}_{r',v}}{M_v\cdot\sum_{n\in N_{r'}} m_{r',n}}\\&\geq\alpha_v(before)\cdot\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)\\&+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\\&\geq\left(\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)^{\sum_{r\in R\setminus\{r'\}} z_{r,v}}-1\right)\\&\cdot\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)\\&+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}},\ \text{by Ineq. (47)}\\&\geq\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)^{\sum_{r\in R\setminus\{r'\}} z_{r,v}}\\&\cdot\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)^{z_{r',v}}-1,\\&\text{by } z_{r',v}=1\\&=\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)^{\sum_{r\in R} z_{r,v}}-1.\end{aligned} \quad (50)$$

We then show that Ineq. (48) holds. α_v is 0 before requests arrive, and a new request r is admitted with $1-\zeta_r\cdot\sum_{v\in V}\alpha_v>0$. Then

$$1-\zeta_r\cdot\sum_{v\in V}\alpha_v>0\Rightarrow\sum_{v\in V}\alpha_v<\frac{1}{\zeta_r}\Rightarrow\alpha_v<\frac{1}{\zeta_{min}}.$$

This means if $\alpha_v\geq\frac{1}{\zeta_{min}}$, α_v will not be updated; otherwise α_v may be updated by function (46).

$$\begin{aligned}\alpha_v&=\alpha_v\cdot\left(1+\frac{\zeta_r\cdot\mathbb{M}_{r,v}}{M_v\cdot\sum_{n\in N_r} m_{r,n}}\right)+\frac{\zeta_r\cdot\mathbb{M}_{r,v}}{M_v\cdot\sum_{n\in N_r} m_{r,n}}\\&<\frac{1}{\zeta_{min}}\cdot\left(1+\frac{\zeta_{max}}{M_{min}}\right)+\frac{\zeta_{max}}{M_{min}},\end{aligned} \quad (51)$$

where Ineq. (51) holds, because the consumed memory resource $\mathbb{M}_{r,v}$ on a cloudlet v of deploying S_DTs of request r is at most that of deploying S_DTs of all candidate IoT devices of request r, i.e., $\mathbb{M}_{r,v}\leq\sum_{n\in N_r} m_{r,n}$. ■

Lemma 6: The memory resource violation on each cloudlet in the solution delivered by `Algorithm 2` for **P1** is upper bounded by ψ, where $\psi=\frac{|N|_{max}\cdot m_{max}}{M_{min}}\cdot\frac{\ln\left(\frac{1}{\zeta_{min}}\cdot\left(1+\frac{\zeta_{max}}{M_{min}}\right)+\frac{\zeta_{max}}{M_{min}}+1\right)}{\ln\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)}-1$.

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

Proof: From Lemma 5, we have

$$\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)^{\sum_{r\in R} z_{r,v}}-1 < \frac{1}{\zeta_{min}}\cdot\left(1+\frac{\zeta_{max}}{M_{min}}\right)+\frac{\zeta_{max}}{M_{min}}.$$

We then have

$$\sum\nolimits_{r\in R} z_{r,v} < \frac{\ln\left(\frac{1}{\zeta_{min}}\cdot\left(1+\frac{\zeta_{max}}{M_{min}}\right)+\frac{\zeta_{max}}{M_{min}}+1\right)}{\ln\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)}$$

The memory resource utilized in cloudlet v is

$$\sum_{r\in R}\sum_{n\in N_r} m_{r,n}\cdot x_{r,n,v} \le |N|_{max}\cdot m_{max}\cdot\sum_{r\in R} z_{r,v}$$
$$< |N|_{max}\cdot m_{max}\cdot\frac{\ln\left(\frac{1}{\zeta_{min}}\cdot\left(1+\frac{\zeta_{max}}{M_{min}}\right)+\frac{\zeta_{max}}{M_{min}}+1\right)}{\ln\left(1+\frac{\zeta_{min}\cdot m_{min}}{M_{max}\cdot|N|_{max}\cdot m_{max}}\right)}.$$

Following Constraint (25), the violation on the memory capacity of a cloudlet thus is no more than ψ. ■

Lemma 7: Denote by $\mathcal{A}_1$ and $\mathcal{A}_3$ the values of the solutions by `Algorithm 2` for **P1** and **P3**, respectively, we have $\frac{1+\zeta_{max}}{u_{min}}\cdot\mathcal{A}_1 \ge \mathcal{A}_3$, where u_{min} is the minimum utility gain of deploying a S_DT by (3), and ζ_{max} is the maximum value of ζ_r by (43).

Proof: Because $\mathcal{A}_1 = \mathcal{A}_3 = 0$, the claim holds initially. Let $\Delta\mathcal{A}_1$ and $\Delta\mathcal{A}_3$ be the value differences of $\mathcal{A}_1$ and $\mathcal{A}_3$ after the rejection or admission of DT service request r, respectively. We show that the claim holds, through showing $\frac{1+\zeta_{max}}{u_{min}}\cdot\Delta\mathcal{A}_1 \ge \Delta\mathcal{A}_3$ for each request r as follows.

Case 1. Request r is rejected. Because $\Delta\mathcal{A}_1 = \Delta\mathcal{A}_3 = 0$, we have $\frac{1+\zeta_{max}}{u_{min}}\cdot\Delta\mathcal{A}_1 \ge \Delta\mathcal{A}_3$.

Case 2. Request r is admitted. We can see $\Delta\mathcal{A}_1$ is the utility gain by request r, and $\Delta\mathcal{A}_3 = \sum_{v\in V} M_v\cdot\Delta\alpha_v + \mu_r$ by the objective function (34), where $\Delta\alpha_v$ is the value difference of α_v before and after the updating of α_v. By the updating functions (46) and (45) of α_v and μ_r, we have

$$\begin{aligned}\Delta\mathcal{A}_3 &= \sum\nolimits_{v\in V} M_v\cdot\Big(\frac{\zeta_r\cdot\mathbb{M}_{r,v}}{M_v\cdot\sum_{n\in N_r} m_{r,n}}\cdot\alpha_v \\ &\quad+\frac{\zeta_r\cdot\mathbb{M}_{r,v}}{M_v\cdot\sum_{n\in N_r} m_{r,n}}\Big)+\mu_r \\ &= \zeta_r\cdot\frac{\sum_{v\in V}\mathbb{M}_{r,v}\cdot\alpha_v}{\sum_{n\in N_r} m_{r,n}}+\zeta_r\cdot\frac{\sum_{v\in V}\mathbb{M}_{r,v}}{\sum_{n\in N_r} m_{r,n}} \\ &\quad+1-\zeta_r\cdot\sum\nolimits_{v\in V}\alpha_v \\ &\le 1+\zeta_r \qquad (52)\\ &\le \frac{1+\zeta_r}{u_{min}}\cdot\Delta\mathcal{A}_1 \\ &\le \frac{1+\zeta_{max}}{u_{min}}\cdot\Delta\mathcal{A}_1, \qquad (53)\end{aligned}$$

where (52) holds as $\sum_{v\in V}\mathbb{M}_{r,v} \le \sum_{n\in N_r} m_{r,n}$, i.e., $\zeta_r\cdot\frac{\sum_{v\in V}\mathbb{M}_{r,v}}{\sum_{n\in N_r} m_{r,n}} \le \zeta_r$. As $\mathbb{M}_{r,v} \le \sum_{n\in N_r} m_{r,n}$, we have $\zeta_r\cdot\frac{\sum_{v\in V}\mathbb{M}_{r,v}\cdot\alpha_v}{\sum_{n\in N_r} m_{r,n}} \le \zeta_r\cdot\sum_{v\in V}\alpha_v$. Ineq. (53) holds, because the utility gain by admitting request r is no less than the minimum utility gain of deploying a S_DT, i.e., u_{min}. ■

Theorem 3: Given an MEC network $G=(V,E)$, a set $\mathcal{N}$ of IoT devices, a sequence R of arriving DT service requests without the knowledge of future request arrivals over a given time horizon, each arrived request $r\in R$ has a candidate set of IoT devices N_r for its S_DT deployment. There is an online algorithm `Algorithm 2` with a provable competitive ratio of $\frac{u_{min}}{1+\zeta_{max}}$ for the dynamic social-aware S_DT placement problem. The violation of memory capacity on any cloudlet in the solution is no larger than ψ, where ψ is defined in Lemma 6, and the algorithm takes $O(|N|^2_{max}+|N|_{max}\cdot|V|)$ time per request $r\in R$.

Proof: Denote by $\mathcal{OPT}(P1)$, $\mathcal{OPT}(P2)$ and $\mathcal{OPT}(P3)$ the optimal solution values of **P1**, **P2**, and **P3**, respectively. Let $\mathcal{A}_1$ and $\mathcal{A}_3$ be the solution values delivered by `Algorithm 2` for **P1** and **P3**, respectively. We have $\frac{1+\zeta_{max}}{u_{min}}\cdot\mathcal{A}_1 \ge \mathcal{A}_3$ by Lemma 7. Because `Algorithm 2` delivers a feasible solution to the minimization problem **P3** by Lemma 4, then $\mathcal{A}_3 \ge \mathcal{OPT}(P3)$. Because **P3** is the dual of **P2**, then $\mathcal{OPT}(P3) \ge \mathcal{OPT}(P2)$ by the weak duality. As **P2** is the relaxation of the maximization problem **P1**, we have $\mathcal{OPT}(P2) \ge \mathcal{OPT}(P1)$. Thus, $\mathcal{A}_3 \ge \mathcal{OPT}(P1)$. Then, $\mathcal{A}_1 \ge \mathcal{A}_3/\left(\frac{1+\zeta_{max}}{u_{min}}\right) \ge \mathcal{OPT}(P1)\cdot\left(\frac{u_{min}}{1+\zeta_{max}}\right)$.

The analysis on memory capacity violation is given in Lemma 6. We can see that there is no violation on the global privacy budget of any admitted request as shown in Lemma 3.

The time complexity of `Algorithm 2` for examining each incoming request r is dominated by invoking `Algorithm 1`, which is $O(|N|^2_{max}+|N|_{max}\cdot|V|)$, due to Theorem 2. ■

VI. Performance Evaluation

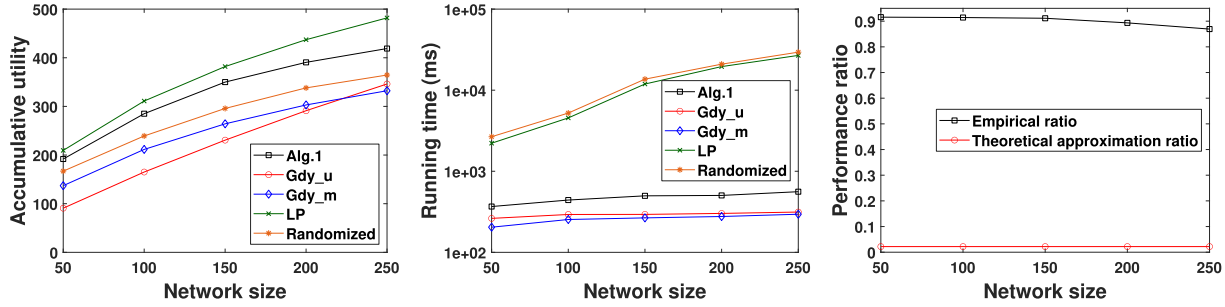
In this section, we first evaluated the performance of `Alg. 1` for the static S_DT placement problem and the impacts of important parameters on its performance. We then evaluated the performance of `Alg. 2` for the dynamic S_DT placement problem and the impacts of important parameters on its performance.

A. Experimental Settings

Consider an MEC network with the number of APs (and their co-located cloudlets) ranging from 50 to 250. The topology of each network is generated by the GT-ITM tool [13]. The memory capacity on each cloudlet is drawn between 6,400 MB and 10,240 MB [39]. The amount of the allocated memory of a container for implementing a S_DT ranges from 128 MB to 1,024 MB [39]. There are 1,000 IoT devices in the MEC network, and there are 1,000 service requests. Each request is issued by the user of a random IoT device, while the number of candidate IoT devices of a request ranges from 10 to 20, and the candidate IoT devices are set randomly. The global privacy budget on each request is set within $[20,40]$. The values of trust q_{n_1,n_2} is randomly drawn within $[0.1,0.9]$. The interaction intensiveness l_{n_1,n_2} between two IoT devices n_1 and n_2 is randomly drawn within $[0.1,0.9]$. Parameters τ and κ in Eq. (1) are set as 0.5 and 10, respectively [9].

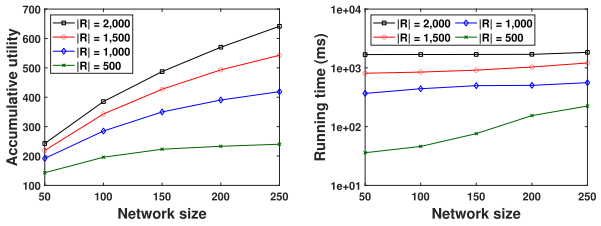
We first evaluated `Algorithm 1`, referred to as `Algorithm 1`, for the static social-aware S_DT placement problem against the following three benchmarks.

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.



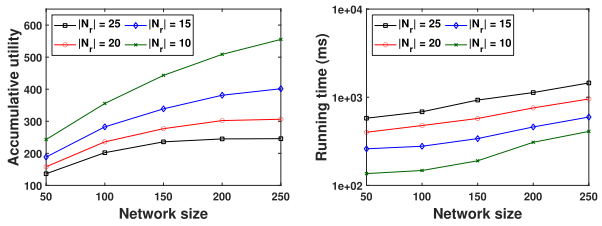
(a) Utility (b) Running time (c) Performance ratios by Alg.1

Fig. 2. Performance of different algorithms for the static social-aware S_DT placement problem.



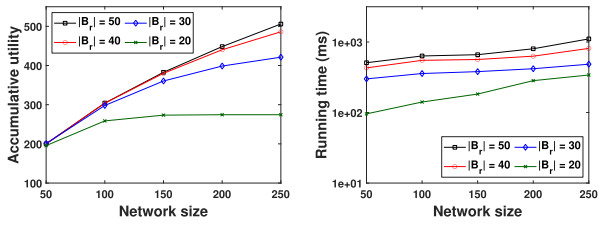
(a) Utility (b) Running time

Fig. 3. Impact of the number $|R|$ of requests on the performance of Alg. 1.



(a) Utility (b) Running time

Fig. 4. Impact of the number $|N_r|$ of candidate IoT devices for each request r on the performance of Alg. 1.



(a) Utility (b) Running time

Fig. 5. Impact of the global privacy budget B_r for each request r on the performance of Alg. 1.

- Gdy_u: it greedily identifies a S_DT with the maximum utility and its request has enough residual privacy budget for the identified S_DT in each iteration. The chosen S_DT then is deployed in a cloudlet with enough residual memory resource. This procedure continues until no more S_DTs can be identified or deployed in any cloudlet.
- Gdy_m: similar to Gdy_u, it identifies a S_DT with the smallest memory resource consumption iteratively.
- LP: it is the relaxed Linear Program (LP) solution by ILP (4), where $x_{r,n,v}$ is a real number between 0 and 1, and the solution delivered by LP is an upper bound on the optimal solution of the static social-aware S_DT placement problem.
- Randomized: it is the randomized algorithm in [37], which first obtains a fractional solution through solving the relaxed LP by ILP (4). It then rounds the fractional solution into an integral solution, based on their calculated probabilities, without causing any resource violation.

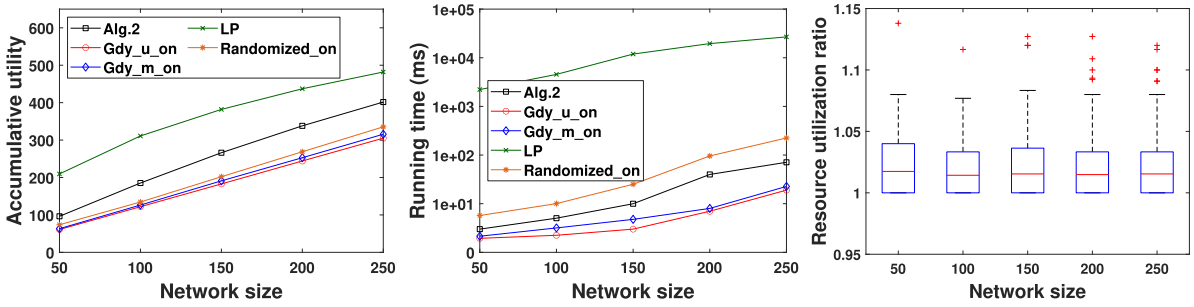
We then studied the performance of Algorithm 2, referred to as Alg. 2, for the dynamic social-aware S_DT placement problem, against four benchmarks: Randomized_on, Gdy_u_on, Gdy_m_on and LP, where Randomized_on, Gdy_u_on, and Gdy_m_on, which are the online versions of Randomized, Gdy_u and Gdy_m, respectively. This means that Randomized_on, Gdy_u_on and Gdy_m_on invoke Randomized, Gdy_u and Gdy_m respectively to admit each incoming DT service request. We can see that the solution of LP is the upper bound on the optimal solution of the dynamic social-aware S_DT placement problem.

The value in each figure is the mean of 30 different network instances with the same size. The running time of each algorithm is obtained by a desktop with an Octa-Core Intel(R) Xeon(R) CPU @ 2.20 GHz, 32G RAM. Unless otherwise specified, we adopt the above-mentioned parameters by default.

B. Performance Evaluation of Algorithms for the Static Social-Aware S_DT Placement Problem

We first studied the performance of Alg. 1 for the static social-aware S_DT placement problem against four comparison algorithms Gdy_u, Gdy_m, LP and Randomized, by varying network size from 50 to 250. Fig. 2 plots the performance curves of the five comparison algorithms, from which it can be seen when network size is 250, the utility of Alg. 1 is 86.9% of that of LP, but outperforms Gdy_u, Gdy_m and Randomized by 21.1%, 26.2%, and 14.9%, respectively. The rationale is that Alg. 1 jointly considers the privacy budget and memory resource for deploying S_DTs in maximizing the total utility gain. Fig. 2(b) indicates that Alg. 1 takes a longer running time, compared with those of Gdy_u and Gdy_m, because the solution by Alg. 1 is achieved through a series of refinements on the initial solution. Among the comparison algorithms, Randomized for the LP takes the longest running time. Fig. 2(c) shows the theoretical approximation ratio and the empirical ratio by Alg. 1, where the theoretical approximation ratio is analyzed

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.



(a) Accumulative utility (b) Running time (c) Resource usage ratios by Alg.2

Fig. 6. Performance evaluation of different algorithms for the dynamic social-aware S_DT placement problem.

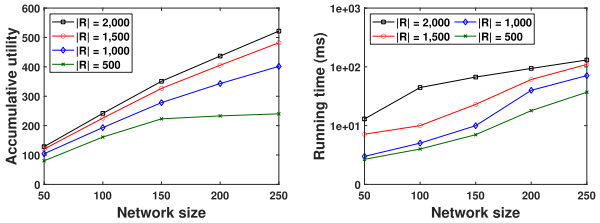
in Theorem 2, and the empirical ratio is the accumulative utility by Alg. 1 divided by that of LP. We can observe from Fig. 2(c) that the empirical ratio of Alg. 1 is 0.869, which is significantly larger than its theoretical approximation ratio, as the theoretical analysis on the approximation ratio is the most conservative one, and obtained in the worst case. Notice that the analytical approximation ratio is determined by the heterogeneous resource environments - the difference between either the ratio of m_{min} to m_{max}, and the ratio of θ_{min} to θ_{max}, by Theorem 2. It is observed that the empirical performance of algorithm Alg. 1 is significantly better than its analytical counterpart, as evidenced by Fig. 2.

We then investigated the impacts of important parameters on the performance of Alg. 1 over the given time horizon.

(a) We investigated the impact of the number $|R|$ of requests on the performance of Alg. 1, by varying $|R|$ from 500 to $2,000$. It is evidenced from Fig. 3(a) that the utility in the solution delivered by Alg. 1 when $|R| = 500$ is 37.4% of that by itself when $|R| = 2,000$, assuming that the network size is 250. Fig. 3(b) demonstrates Alg. 1 with $2,000$ requests takes the longest running time. The justification is that more utilities can be obtained with a large number of requests, while taking more time to examine the requests.

(b) We studied the impact of the number $|N_r|$ of candidate IoT devices for each request r on the performance of Alg. 1, by choosing $|N_r| = 10$, 15, 20, and 25, respectively. As seen from Fig. 4(a), the utility obtained by Alg. 1 when $|N_r| = 25$ is 44.2% of that by itself when $|N_r| = 10$, assuming that the network size is 250. This is because of the utility definition (3), i.e., the utility gain of selecting an IoT device for a request depends on the total interaction intensiveness of all the candidate IoT devices of the request. Fig. 4(b) implies that a larger value of $|N_r|$ leads to more running time due to more candidate IoT devices for requests to be examined.

(c) We evaluated the impact of the global privacy budget B_r demanded by each request r on the performance of Alg. 1. Fig. 5 plots the performance curves of Alg. 1 when $B_r = 20, 30, 40$ and 50, respectively. It is observed from Fig. 5(a) that the utility by Alg. 1 with $B_r = 20$ is 54.3% of that by itself with $B_r = 50$, assuming that the network size is 250. This is because a large global privacy budget can be allocated to select more IoT devices for providing DT data for each request, with a larger B_r. Also, Alg. 1 with $B_r = 50$ achieves the similar performance by itself with $B_r = 40$. This is justified by that the memory resource capacity constraint is the bottleneck, when the global privacy budgets of requests are large. Fig. 5(b) shows that a larger B_r leads to more running time, because of managing larger global privacy budgets of requests to select more IoT devices to provide DT data.



(a) Utility (b) Running time

Fig. 7. Impact of the number $|R|$ of requests on the performance of Alg. 2 over a given time horizon.

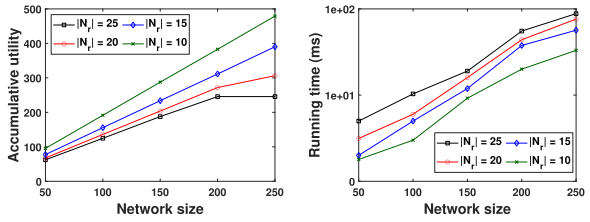
C. Performance Evaluation of Algorithms for the Dynamic Social-Aware S_DT Placement Problem

We then investigated the performance of Alg. 2 for the dynamic social-aware S_DT placement problem against four benchmarks: Gdy_u_on, Gdy_m_on, LP and Randomized_on, by varying network size from 50 to 250. Fig. 6(a) shows that when network size is set at 250, the performance of Alg. 2 is around 83.3% of that of LP - the upper bound on the optimal value in terms of utility. Alg. 2 also outperforms Gdy_u_on, Gdy_m_on and Randomized_on by 31.6%, 27.3% and 19.8%, respectively. This is because Alg. 2 adopts the primal-dual dynamic updating technique to establish an efficient admission control policy to handle each incoming DT service request, which takes more running time than that of Gdy_u_on and Gdy_m_on, as evidenced in Fig. 6(b). The resource usage ratios of cloudlets by Alg. 2 is shown in Fig. 6(c), where the resource usage ratio of a cloudlet is its consumed memory resource to its memory capacity, and we can see the memory capacity of any cloudlet is violated by no greater than 13.8%.

The rest is to study the impacts of important parameters on the performance of Alg. 2 over the given time horizon.

We investigated the impact of the number $|R|$ of requests on the performance of Alg. 2, by changing the number of requests from 500 to $2,000$. Fig. 7(a) demonstrates that the utility delivered by Alg. 2 when $|R| = 500$ is 46.1% of that by itself when $|R| = 2,000$, with the network size of 250. This is due to the fact that Alg. 2 intends to admit DT service requests with less resource consumption referring to its

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.



(a) Utility (b) Running time

Fig. 8. Impact of the number $|N_r|$ of candidate IoT devices for each request r on the performance of Alg. 2.

admission control policy, thereby obtaining more accumulative utility gain with the dynamic arrivals of requests. Also, it takes more time to examine more requests as shown in Fig. 7(b).

We studied the impact of the number $|N_r|$ of candidate IoT devices for each request r on the performance of Alg. 2, with $|N_r|$ from 10 to 25. Similar to the phenomena in Fig. 4, we can see from Fig. 8(a) that the utility obtained by Alg. 2 when $|N_r| = 25$ is 51.3% of that by itself when $|N_r| = 10$, when the network size is 250. This is because the utility gain of selecting an IoT device for the S_DT deployment of a request intends to decrease with the increasing number of candidate IoT devices of the request, by the utility definition (3).

VII. Conclusion

In this paper, we investigated the novel social-aware service provisioning in DT-assisted MEC environments. We first introduced a differential privacy-based federated learning framework for admitting service requests. Built upon the framework, we then formulated two social-aware S_DT placement problems under static and dynamic settings of service request arrivals. For the static version of the problem, we provided an ILP formulation when the problem size is small or medium, otherwise we developed a performance-guaranteed approximation algorithm. For the dynamic version of the problem, we devised an online algorithm with a provable competitive ratio, by leveraging the primal-dual dynamic updating technique. Finally, we conducted simulations to evaluate the performance of the proposed algorithms. Simulation results show that the proposed algorithms outperform their counterparts, improving the comparison baselines by at least 14.9%.

References

[1] S. AbdulRahman, S. Otoum, O. Bouachir, and A. Mourad, "Management of digital twin-driven IoT using federated learning," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3636–3649, Nov. 2023.

[2] X. Ai, W. Liang, and C. Liu, "Joint optimization of model retraining and inference services in DT-assisted edge computing," *IEEE Trans. Netw.*, vol. 34, pp. 1804–1819, 2026.

[3] X. Ai, W. Liang, Y. Zhang, and W. Xu, "Fidelity-aware inference services in DT-assisted edge computing via service model retraining," *IEEE Trans. Services Comput.*, vol. 18, no. 4, pp. 2089–2102, Jul. 2025.

[4] (2025). *AWS Lambda*. [Online]. Available: https://aws.amazon.com/lambda/

[5] (2025). *Azure*. [Online]. Available: https://azure.microsoft.com/en-us/services/functions/

[6] I. Bistritz and N. Bambos, "Power is knowledge: Distributed and throughput optimal power control in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 32, no. 6, pp. 4722–4734, Dec. 2024.

[7] O. Chukhno, N. Chukhno, G. Araniti, C. Campolo, A. Iera, and A. Molinaro, "Optimal placement of social digital twins in edge IoT networks," *Sensors*, vol. 20, no. 21, p. 6181, Oct. 2020.

[8] O. Chukhno, N. Chukhno, G. Araniti, C. Campolo, A. Iera, and A. Molinaro, "Placement of social digital twins at the edge for beyond 5G IoT networks," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23927–23940, Dec. 2022.

[9] M.-S. Chung, C.-H. Wang, D.-N. Yang, G.-S. Lee, W.-T. Chen, and J.-P. Sheu, "SIoT selection, clustering, and routing for federated learning with privacy-preservation," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 5274–5279.

[10] L. Cui, Y. Qu, S. Yu, L. Gao, and G. Xie, "A trust-grained personalized privacy-preserving scheme for big social data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[11] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, Aug. 2014.

[12] M. X. Goemans and D. P. Williamson, "The primal-dual method for approximation algorithms and its application to network design problems," in *Approximation Algorithms for NP-Hard Problems*. Boston, MA, USA: PWS Publishing, 1997, pp. 144–191.

[13] (2019). *GT-ITM*. [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/

[14] L. Jiang, H. Zheng, H. Tian, S. Xie, and Y. Zhang, "Cooperative federated learning and model update verification in blockchain-empowered digital twin edge networks," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11154–11167, Jul. 2022.

[15] J. Li et al., "Mobility-aware utility maximization in digital twin-enabled serverless edge computing," *IEEE Trans. Comput.*, vol. 73, no. 7, pp. 1837–1851, Jul. 2024.

[16] J. Li et al., "Wait for fresh data? Digital twin empowered IoT services in edge computing," in *Proc. MASS*, Sep. 2023, pp. 397–405.

[17] J. Li et al., "AoI-aware service provisioning in edge computing for digital twin network slicing requests," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 14607–14621, Dec. 2024.

[18] J. Li et al., "AoI-aware user service satisfaction enhancement in digital twin-empowered edge computing," *IEEE/ACM Trans. Netw.*, vol. 32, no. 2, pp. 1677–1690, Apr. 2024.

[19] J. Li, J. Wang, W. Liang, S. Das, and Q. Chen, "Improving the freshness of digital twins in edge computing," in *Proc. WASA*, 2025, pp. 74–86.

[20] J. Li, J. Wang, Q. Chen, Y. Li, and A. Y. Zomaya, "Digital twin-enabled service satisfaction enhancement in edge computing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2023, pp. 1–10.

[21] J. Li, J. Wang, W. Liang, X. Jia, and A. Y. Zomaya, "Inference service fidelity maximization in DT-assisted edge computing," *IEEE Trans. Mobile Comput.*, vol. 25, no. 1, pp. 1352–1366, Jan. 2026.

[22] J. Li, J. Wang, W. Liang, J. Wu, Q. Chen, and Z. Xu, "Social-aware DT-assisted service provisioning in serverless edge computing," in *Proc. 20th Int. Conf. Mobility, Sens. Netw. (MSN)*, Dec. 2024, pp. 374–381.

[23] J. Li et al., "Maximizing user service satisfaction for delay-sensitive IoT applications in edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 5, pp. 1199–1212, May 2022.

[24] Y. Li, D. Zeng, L. Gu, K. Wang, and S. Guo, "On the joint optimization of function assignment and communication scheduling toward performance efficient serverless edge computing," in *Proc. IEEE/ACM 30th Int. Symp. Quality Service (IWQoS)*, Jun. 2022, pp. 1–9.

[25] Z. Liang, H. Chen, Y. Liu, and F. Chen, "Data sensing and offloading in edge computing networks: TDMA or NOMA?," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4497–4508, Jun. 2022.

[26] X. Lin, J. Wu, J. Li, W. Yang, and M. Guizani, "Stochastic digital-twin service demand with edge response: An incentive-based congestion control approach," *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2402–2416, Apr. 2023.

[27] T. Luo, M. Pan, P. Tholoniat, A. Cidon, R. Geambasu, and M. Lécuyer, "Privacy budget scheduling," in *Proc. USENIX OSDI*, 2021, pp. 55–74.

[28] Y. Ma, W. Liang, M. Huang, W. Xu, and S. Guo, "Virtual network function service provisioning in MEC via trading off the usages between computing and communication resources," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2949–2963, Oct. 2022.

[29] T. Öncan, "A survey of the generalized assignment problem and its applications," *INFOR, Inf. Syst. Oper. Res.*, vol. 45, no. 3, pp. 123–142, Aug. 2007.

[30] L. Pan, L. Wang, S. Chen, and F. Liu, "Retention-aware container caching for serverless edge computing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2022, pp. 1069–1078.

[31] S. Pasteris, S. Wang, M. Herbster, and T. He, "Service placement with provable guarantees in heterogeneous edge computing systems," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 514–522.

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.

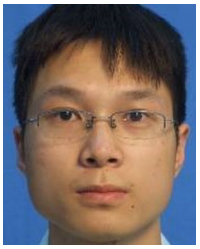
[32] Y. Ren, S. Guo, B. Cao, and X. Qiu, "End-to-end network SLA quality assurance for C-RAN: A closed-loop management method based on digital twin network," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4405–4422, May 2024.
[33] X. Shang, Y. Mao, Y. Liu, Y. Huang, Z. Liu, and Y. Yang, "Online container scheduling for data-intensive applications in serverless edge computing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2023, pp. 1–10.
[34] M. Vaezi et al., "Digital twins from a networking perspective," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23525–23544, Dec. 2022.
[35] Y. Wang, Z. Su, S. Guo, M. Dai, T. H. Luan, and Y. Liu, "A survey on digital twins: Architecture, enabling technologies, security and privacy, and future prospects," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 14965–14987, Sep. 2023.
[36] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.
[37] K. Xiao, S. Yang, F. Li, L. Zhu, X. Chen, and X. Fu, "Making serverless not so cold in edge clouds: A cost-effective online approach," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8789–8802, Sep. 2024.
[38] G. Xu et al., "Trust2Privacy: A novel fuzzy trust-to-privacy mechanism for mobile social networks," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 72–78, Jun. 2020.
[39] Z. Xu, Y. Fu, Q. Xia, and H. Li, "Enabling age-aware big data analytics in serverless edge clouds," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun.*, May 2023, pp. 1–10.
[40] Z. Xu et al., "Efficient and fault tolerant data stream processing with uncertain data rates in serverless edge computing," *IEEE Trans. Services Comput.*, vol. 19, no. 1, pp. 295–308, Jan. 2026.
[41] Z. Xu et al., "Stateful serverless application placement in MEC with function and state dependencies," *IEEE Trans. Comput.*, vol. 72, no. 9, pp. 2701–2716, Sep. 2023.
[42] Z. Yao, S. Xia, Y. Li, and G. Wu, "Cooperative task offloading and service caching for digital twin edge networks: A graph attention multi-agent reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3401–3413, Nov. 2023.
[43] J. Zhang et al., "Efficient throughput maximization in dynamic rechargeable networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 3, pp. 2254–2268, Mar. 2024.
[44] Q. Zhang, F. Liu, and C. Zeng, "Online adaptive interference-aware VNF deployment and migration for 5G network slice," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 2115–2128, Oct. 2021.
[45] Y. Zhang, W. Liang, Z. Xu, and X. Jia, "Mobility-aware service provisioning in edge computing via digital twin replica placements," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 11295–11311, Dec. 2024.
[46] L. Zhao et al., "A digital twin-assisted intelligent partial offloading approach for vehicular edge computing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3386–3400, Nov. 2023.
[47] X. Zhou et al., "Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3191–3211, Oct. 2023.

Jing Li received the B.Sc. and Ph.D. degrees (Hons.) from The Australian National University in 2018 and 2022, respectively. He is currently a Post-Doctoral Fellow with City University of Hong Kong. His research interests include edge computing, the Internet of Things, digital twin, network function virtualization, and combinatorial optimization.

Jianping Wang (Fellow, IEEE) received the B.S. and M.S. degrees in computer science from Nankai University, Tianjin, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from The University of Texas at Dallas in 2003. She is currently a Professor with the Department of Computer Science, City University of Hong Kong. Her research interests include cloud computing, service oriented networking, edge computing, and network performance analysis.

Weifa Liang (Fellow, 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 The Australian National University in 1998. He is currently a Full Professor with the Department of Computer Science, City University of Hong Kong. Prior to joining City University of Hong Kong, he was a Full Professor with The Australian National University. His research interests include design and analysis of routing protocols for wireless ad hoc and sensor networks, mobile edge computing (MEC), the Internet of Things and digital twins, machine learning and LLM, network function virtualization (NFV), design and analysis of parallel and distributed algorithms, approximation and online algorithms, and graph theory. He currently serves as an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS. He is also a Distinguished Contributor of the IEEE Computer Society.

Jie Wu (Fellow, IEEE) received the Ph.D. degree in computer engineering from Florida Atlantic University, Boca Raton, FL, USA, in 1989. He was a Distinguished Professor with Florida Atlantic University. He is currently the Director of the Center for Networked Computing and a Laura H. Carnell Professor with Temple University. He is also the Director of International Affairs with the College of Science and Technology. He was the Chair of the Department of Computer and Information Sciences from Summer 2009 to 2016 and an Associate Vice Provost for International Affairs from Fall 2015 to Summer 2017. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He is a Fellow of AAAS. He was a recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award. He was the General Co-Chair of IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, and the Program Co-Chair of IEEE INFOCOM 2011 and CCF CNCC 2013. He was a Program Director with the National Science Foundation. He serves on several editorial boards, including IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON SERVICE COMPUTING, *Journal of Parallel and Distributed Computing*, and *Journal of Computer Science and Technology*.



Quan Chen (Member, IEEE) received the B.S., master's, and Ph.D. degrees from the School of Computer Science and Technology, Harbin Institute of Technology, China. He was a Post-Doctoral Research Fellow with the Department of Computer Science, Georgia State University. He is currently an Associate Professor with the School of Computers, Guangdong University of Technology. His research interests include routing and scheduling algorithms in wireless networks and sensor networks.

Zichuan Xu (Member, IEEE) received the B.Sc. and M.E. degrees in computer science from Dalian University of Technology, China, in 2008 and 2011, respectively, and the Ph.D. degree in computer science from The Australian National University in 2016. From 2016 to 2017, he was a Research Associate with the Department of Electronic and Electrical Engineering, University College London, U.K. He is currently a Full Professor and a Ph.D. Advisor with the School of Software, Dalian University of Technology. His research interests include mobile edge computing, server less computing, network function virtualization, algorithmic game theory, and optimization problems.

Authorized licensed use limited to: Fujian Normal University. Downloaded on July 06,2026 at 06:20:14 UTC from IEEE Xplore. Restrictions apply.