Social-Aware DT-Assisted Service Provisioning in Serverless Edge Computing

Jing Li†, Jianping Wang†, Weifa Liang†, Jie Wu¶, Quan Chen§, and Zichuan Xu$
† Department of Computer Science, City University of Hong Kong, Hong Kong, P. R. China
¶ Department of Computer and Information Sciences, Temple University, Philadelphia, USA
§ Guangdong University of Technology, Guangzhou, 510006, P. R. China
$ School of Software, Dalian University of Technology, Dalian, 116621, P. R. China
Emails: {jing.li, jianwang, weifa.liang}@cityu.edu.hk, jiewu@temple.edu, quan.c@gdut.edu.cn, z.xu@dlut.edu.cn

***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 in network edges, the recent advances in Serverless Edge Computing (SEC) 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 the orchestration of DTs. In this paper, we design a differential privacy-based federated learning framework to build a DT network for DT services in response to user DT service requests in SEC, thereby enhancing the Quality of Services (QoS). To this end, we first formulate a novel social-aware problem for placing DTs in an SEC network, and show its NP-hardness. We then provide an Integer Linear Program (ILP) solution to the problem when the problem size is small; otherwise, we design an approximation algorithm with a provable approximation ratio. We finally evaluate the algorithm performance through simulations. Simulation results demonstrate the proposed algorithm is promising, which improves by no less than 21.1% of the performance of benchmarks.**

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

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. 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 [19].

The work by Jing Li, Jianping Wang and Weifa Liang was supported by Hong Kong Research Grants Council (RGC) under the Collaborative Research Fund (CRF) grant C1042-23GF. The work by Jing Li was supported by the Post-Doc Fellowship Award from the RGC of the Hong Kong Special Administrative Region, China (Project No. CityU PDFS2425-1S02). The work by Weifa Liang was supported by Hong Kong RGC under CityU HK Grant No: 7005845, 8730094, 9043510, and 9380137, respectively.

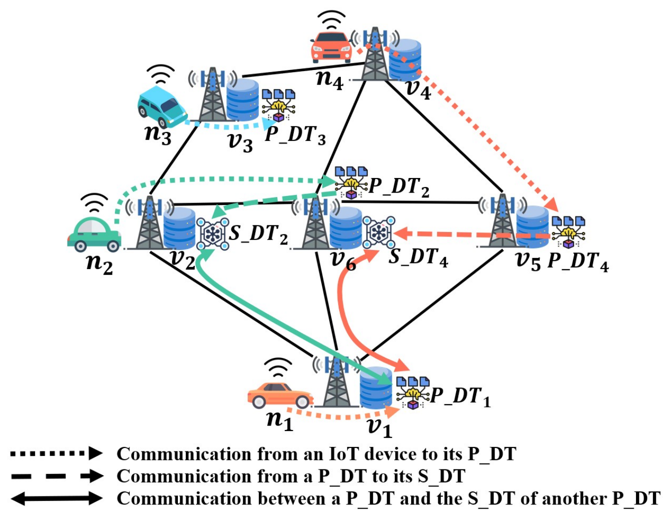


Fig. 1. An illustrative example of a DT network for a request in an SEC network where each Access Point (AP) has a co-located cloudlet. Each IoT device has a *Primary DT* (P_DT) deployed in a cloudlet, i.e., P_DT_1, P_DT_2, P_DT_3 and P_DT_4 of IoT devices n_1, n_2, n_3 and n_4 are deployed in cloudlets 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.

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 [12], [25]. Albeit with these benefits of MEC, it raises issues about resource scarcity and inefficient resource management [11]. Taking advantages of the container technology, serverless computing provides fine-grained resource allocation with high elasticity for MEC [23]. In this context, Serverless Edge Computing (SEC) integrates MEC and serverless computing, and has emerged as a crucial research area and endows DTs with new vigors [18].

With the accelerated penetration of DTs, the DT network paradigm is expected to deliver comprehensive DT services to users, through grouping a collection of DTs and analyzing their global information [19]. However, building a DT network confronts significant challenges, such as the rising 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 [10]. Nevertheless, federated learning still suffers from potential leakages of private data, through examining the differences of model 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 [21].

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 of DTs [4], [5]. On behalf of their physical counterparts, DTs keep traces of social relationships of IoT devices, which provides further insights into how to build 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 [6]. 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 SEC network, where each request is based on a DT network. 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 [20], 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, an S_DT is implemented by a short-lived container that contains partial features of the IoT device, which means an S_DT requires much less resources compared with its P_DT. The model of an 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.

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. Each of such an 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 [6]. Each service request has a *global privacy budget* to bound its privacy leakage [16], 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 the interaction intensiveness among IoT devices, which is also indicated by their social relationships.

Providing social-aware DT services in an SEC network poses several challenges. First, 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 in cloudlets for each request in an SEC network, considering limited resource capacities on cloudlets? Finally, 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?

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 SEC. 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, a novel optimization problem for placing S_DTs in an SEC network is formulated, and an approximation algorithm for the problem with the guaranteed performance is devised.

The main contributions of this paper are as follows.

- We formulate a novel social-aware S_DT placement problem for DT-enabled service provisioning, considering data privacy, and illustrate the NP-hardness of the problem.
- We propose an Integer Linear Program (ILP) solution for the problem when the problem size is small, and devise an approximation algorithm for the problem with a provable approximation ratio.
- We evaluate the algorithm performance by simulations. The results demonstrate that the proposed approximation algorithm is promising, which improves the performance of the benchmarks by no less than 21.1%.

The remainder of the paper is arranged as follows. Section II surveys related works on providing social-aware DT services in SEC. Section III shows the system model and problem definition with an ILP formulation. Section IV proposes an approximation algorithm for the social-aware S_DT placement problem. Section V evaluates the algorithm performance, and Section VI concludes the paper.

II. Related work

Serverless computing endows network service provisioning with great elasticity, which has been extended to edge environments, and Serverless Edge Computing (SEC) is receiving a surge of attention recently [18], [19], [23], [24]. Shang *et al.* [18] formulated a problem for data flow and container placement in SEC networks, and introduced an online algorithm to mitigate both operational costs and service delays. Xu *et al.* [23] 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 SEC. However, none of the aforementioned works considered DT services in an SEC.

In recent years, DT techniques applied in edge environments have been extensively investigated [12]–[15], [19], [20], [25]. Li *et al.* [13] 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. Yao *et al.* [25] 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. A few recent studies take social relationships into consideration for providing DT services [4], [5], [26]. Chukhno *et al.* [4] 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.* [5] also examined the dynamic placements of social-aware DTs, and provided a heuristic algorithm to mitigate the communication delay. Zhao *et al.* [26] 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 allays user privacy worries and offers a promising distributed machine learning architecture for enabling DT services [1], [10]. 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.* [10] 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. 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.

In contrast to the aforementioned works, in this paper we study social-aware DT placements in an SEC network for DT-assisted service provisioning. We develop a differential privacy-based federated learning framework to construct a DT network for each request admission, and devise an efficient approximation algorithm for the social-aware DT placement problems with a provable approximation ratio.

III. Preliminaries

A. System model

Consider an SEC network $G = (V, E)$, consisting of a set V of Access Points (APs) and a set E of links interconnecting APs. Each AP has a co-located cloudlet, and we use notion $v \in V$ 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 l_{n_1,n_2} the trust of IoT device n_1 to IoT device n_2 with $0 \leq l_{n_1,n_2} \leq 1$ [6]. A larger value of l_{n_1,n_2} indicates device n_1 trusts on device n_2 with higher fidelity. Denote by λ_{n_1,n_2} the interaction intensiveness of IoT device n_1 with IoT device n_2, with $0 \leq \lambda_{n_1,n_2} \leq 1$ [5], where a larger λ_{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. A federated learning framework for constructing a DT network for each request

Now 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. 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 [20].

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 SEC 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 [24], according to serverless platforms [2], [3], because the amount of data loaded to the memory is related to the amount of memory assigned to the container [23]. Denote by $m_{r,n}$ the amount of memory resource needed in a container to deploy an 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 an 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 SEC network.

C. Differential Privacy and privacy budget constraint

The model of an 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 [21].

Definition 1: ((ϵ, δ)-differential privacy [8]): 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 are 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 one data indicates that the data will receive greater privacy protection. In terms of the DT data privacy issue, an S_DT adopts the differential privacy strategy through adding the Gaussian noise to model parameters to avoid revealing the original data [21].

We assume that each request $r \in R$ has a global privacy budget B_r to bound its privacy leakage [16]. The required privacy protection level can be obtained based on social relationships (trust) among IoT devices [22]. Following [6], [7], the privacy budget for data communication from IoT device n_1 to IoT device n_2 is calculated as follows.

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

where l_{n_1,n_2} is the value of the trust from IoT device n_1 to device n_2 with $0 \leq l_{n_1,n_2} \leq 1$, τ is a constant with $0 \leq \tau \leq 1$ to avoid the denominator to be zero, and $\kappa > 0$ is a tuning parameter to scale the allocated privacy budget.

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. The 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, and 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. Then we define the utility gain $u_{r,n}$ of selecting an IoT device n from N_r as

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

where $\lambda_{n_r,n}$ is the intensiveness that IoT device n_r interacts with IoT device n, with $0 \leq \lambda_{n_r,n} \leq 1$ [5], indicating the importance of the DT data of IoT device n for n_r.

E. Problem definition

Definition 2: Given an SEC 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 S_DT deployment with a given global privacy budget B_r. *The 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.

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 cloudlet $v \in V$ for request r, and $x_{r,n,v} = 0$ otherwise. The ILP of the 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:

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

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

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

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

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.

F. NP-hardness of the defined problem

Theorem 1: The social-aware S_DT placement problem for service provisioning in an SEC is NP-hard.

The NP-hardness of the social-aware S_DT placement problem can be shown through a polynomial reduction from the generalized assignment problem, which is NP-hard [17]. The detailed proof is omitted due to space limitation.

IV. Approximation algorithm for the social-aware S_DT placement problem

In this section, we deal with the social-aware S_DT placement problem by proposing an approximation algorithm, and its core idea is as follows. We first obtain a potential solution $\mathbb{S}$, i.e., a set of S_DTs deployed in cloudlets, which allows violations on memory capacities of cloudlets and global privacy budgets on requests. We then partition set $\mathbb{S}$ into two disjoint subsets S_1 and S_2 respectively, where the S_DTs in either S_1 or S_2 cause no violation on global privacy budgets of requests. We choose one from S_1 and S_2 with a larger utility gain and denote this set as S'. We further partition S' into two disjoint subsets S_3 and S_4 respectively, and either S_3 or S_4 causes no violations on memory capacities on cloudlets. We finally choose one from S_3 and S_4 with a larger utility gain, which serves as the final solution to the problem.

It is observed that the deployment of any 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 $\rho(\lambda^l)$ 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 an 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 ($\mathcal{B}_r(\mathbb{S}^l) \le B_r$), we put λ^l into set S_2. Also, if $\mathcal{B}_r(\mathbb{S}^l) \ge B_r$, 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 $\mathbb{S} = S_1 \cup S_2$.

We now identify a cloudlet for the deployment of λ^l. Specifically, the candidate set of cloudlets is $\mathbb{V} = V$ initially. We then identify a cloudlet $v \in \mathbb{V}$ with the largest residual memory resource for deploying λ^l. Let $\mathcal{M}_v(\mathbb{S}^l)$ be the accumulative memory resource consumption of cloudlet v, via placing S_DTs in $\mathbb{S}^l$. For each identified S_DT λ^l, if the consumed memory resource of the assigned cloudlet of λ^l after its deployment is greater than its capacity M_v, i.e., $\mathcal{M}_v(\mathbb{S}^l) > M_v$, we put λ^l into set S_3. Also, if the consumed memory resource of cloudlet v_k is no less than its capacity M_v after deploying S_DT λ^l, i.e., $\mathcal{M}_v(\mathbb{S}^l) \ge M_v$, we remove cloudlet v from $\mathbb{V}$ with $\mathbb{V} \leftarrow \mathbb{V} \backslash \{v\}$, i.e., the cloudlet is removed from further consideration. It can be seen that S_3 is a set of S_DTs which cause capacity violations on cloudlets, and each cloudlet has at most one associated S_DT in S_3. This procedure continues until either the set of to-be-considered requests becomes empty (i.e., all requests run out of global privacy budgets) or the set of to-be-considered cloudlets becomes empty (i.e., all cloudlets run out of resources).

Note that $\mathbb{S}$ has been partitioned into two sets S_1 and S_2, and one of them with the larger utility is identified as S'. Because S' is a subset of $\mathbb{S}$, an S_DT in S_3 may not cause capacity violation on a cloudlet by S' any more. We then refine S_3 as follows. We first update $S_3 = S_3 \cap S'$. 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 $S_4 = S' \backslash S_3$, and S' is now partitioned into two disjoint sets 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, which will be shown in Lemma 3. We finally choose S_3 or S_4 with the larger utility as the final solution to the social-aware S_DT placement problem. The detailed algorithm is shown in Algorithm 1.

Algorithm 1 Approximation algorithm for the social-aware S_DT placement problem

Input: An SEC network $G = (V, E)$, and a set R of DT service requests.
Output: Maximize the utility gain of deploying S_DTs in cloudlets.

$\mathbb{S}^0 \leftarrow \emptyset$; $S_1 \leftarrow \emptyset$; $S_2 \leftarrow \emptyset$; $S_3 \leftarrow \emptyset$; $S_4 \leftarrow \emptyset$;
$\mathbb{V} \leftarrow V$; $\Lambda \leftarrow \{\lambda_{r,n} \mid r \in R, n \in N_r\}$; $l \leftarrow 1$;
while $\mathbb{V} \neq \emptyset$ **or** $\Lambda \backslash \mathbb{S}^{l-1} \neq \emptyset$ **do**
  Identify an 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 $\mathcal{B}_r(\mathbb{S}^l) \ge B_r$ **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**
    $S_3 \leftarrow S_3 \cup \{\lambda^l\}$;
  end if;
  if $\mathcal{M}_v(\mathbb{S}^l) \ge M_v$ **then**
    $\mathbb{V} \leftarrow \mathbb{V} \backslash \{v\}$;
  end if;
  $l \leftarrow l + 1$;
end while;
$S' \leftarrow \arg\max_{S \in \{S_1, S_2\}} \sum_{\lambda^l \in S} u(\lambda^l)$; $S_3 \leftarrow S_3 \cap S'$;
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 $\arg\max_{S \in \{S_3, S_4\}} \sum_{\lambda^l \in S} u(\lambda^l)$;

A. Algorithm analysis

Lemma 1: Suppose that Algorithm 1 terminates when all requests run out of global privacy budgets, 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 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) $\rho(\lambda^l) \ge \rho(\lambda^*), \forall r \in R, \forall \lambda^l \in \mathbb{S}_r$, $\forall \lambda^* \in \mathbb{S}_r^{opt} \setminus \mathbb{S}_r$; and (ii) $\sum_{\lambda^l \in \mathbb{S}} u(\lambda^l) \ge \frac{m_{min}}{m_{max}} \cdot \sum_{\lambda^* \in \mathbb{S}^{opt} \setminus \mathbb{S}} u(\lambda^*)$, 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 $\rho(\lambda^l) \ge \rho(\lambda^*)$, $\forall r \in R, \forall \lambda^l \in \mathbb{S}_r$, $\forall \lambda^* \in \mathbb{S}_r^{opt} \setminus \mathbb{S}_r$.

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

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

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

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

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

$$\geq m_{min}\cdot\sum_{r\in R}\sum_{\lambda^*\in\mathbb{S}_r^{opt}\setminus\mathbb{S}_r}\rho(\lambda^*)\cdot\frac{\epsilon(\lambda^*)}{B_r} \quad (15)$$

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

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

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 that `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 optimal solution to the 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

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

$$\geq\theta_{min}\cdot\sum_{\lambda^l\in\mathbb{S}}m(\lambda^l)\geq\theta_{min}\cdot\sum_{\lambda^*\in\mathbb{S}^{opt}}m(\lambda^*) \quad (17)$$

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

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 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 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. Similarly, we can show deploying S_DTs in either S_3 or S_4 causes no memory capacity violations on cloudlets, and 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 SEC 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 social-aware S_DT placement problem with an approximation ratio of $\frac{1}{4}\cdot\min\{\frac{m_{min}}{m_{max}+m_{min}},\frac{\theta_{min}}{\theta_{max}}\}$, 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. $\theta(\lambda^l)=\frac{u(\lambda^l)}{m(\lambda^l)}$, θ_{max} and θ_{min} are the maximum and minimum values of $\theta(\lambda^l)$, 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 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

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

Then, the value of the optimal solution is

$$\sum_{\lambda^*\in\mathbb{S}^{opt}}u(\lambda^*)\leq\sum_{\lambda^l\in\mathbb{S}}u(\lambda^l)+\sum_{\lambda^*\in\mathbb{S}^{opt}\setminus\mathbb{S}}u(\lambda^*)$$

$$\leq(1+\frac{m_{max}}{m_{min}})\cdot\sum_{\lambda^l\in\mathbb{S}}u(\lambda^l),\quad\text{by Ineq. (18)}$$

$$=(\frac{m_{max}+m_{min}}{m_{min}})\cdot\sum_{\lambda^l\in\mathbb{S}}u(\lambda^l). \quad (19)$$

The final solution value delivered by `Algorithm` 1 is

$$\max\{\sum_{\lambda^l\in S_3}u(\lambda^l),\sum_{\lambda^l\in S_4}u(\lambda^l)\}$$

$$\geq\frac{1}{2}\cdot\max\{\sum_{\lambda^l\in S_1}u(\lambda^l),\sum_{\lambda^l\in S_2}u(\lambda^l)\}\geq\frac{1}{4}\sum_{\lambda^l\in\mathbb{S}}u(\lambda^l)$$

$$\geq\frac{1}{4}\cdot\frac{m_{min}}{m_{max}+m_{min}}\cdot\sum_{\lambda^*\in\mathbb{S}^{opt}}u(\lambda^*),\quad\text{by Ineq. (19).} \quad (20)$$

Case 2. By Lemma 2, we have

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

Similarly, the value of the final solution is

$$\max\{\sum_{\lambda^l\in S_3}u(\lambda^l),\sum_{\lambda^l\in S_4}u(\lambda^l)\}$$

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

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

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

The detailed analysis of the time complexity of `Algorithm` 1 is omitted due to space limitation. ∎

V. Performance Evaluation

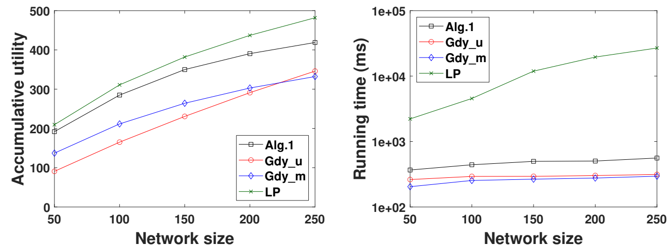
A. Experimental settings

Consider an SEC 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 [9]. The memory capacity on each cloudlet is drawn between $6,400$ MB and $10,240$ MB [23]. The amount of the allocated memory of a container for implementing an S_DT ranges from 128 MB to $1,024$ MB [23]. There are $1,000$ IoT devices in the SEC 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 the trust l_{n_1,n_2} and the interaction intensiveness λ_{n_1,n_2} are randomly drawn within $[0.1, 0.9]$. Parameters τ and κ in Eq. (1) are set as 0.5 and 10, respectively [6]. 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. We evaluated `Algorithm` 1, referred to as `Alg.1`, for the social-aware S_DT placement problem against the following benchmarks.

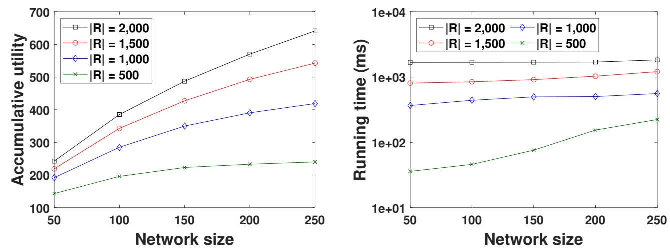
- `Gdy_u`: it greedily identifies an S_DT with the maximum utility and its request has enough residual privacy budget for the identified S_DT in each iteration. The chosen S_DT then is deployed in a cloudlet with enough residual memory resource. This procedure continues until no more S_DTs can be identified or deployed in any cloudlet.
- `Gdy_m`: similar to `Gdy_u`. It identifies an S_DT with the smallest memory resource consumption iteratively.
- `LP`: 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 social-aware S_DT placement problem.

B. Algorithm performance evaluation

We first studied the performance of `Alg.1` against `Gdy_u`, `Gdy_m` and `LP` for the social-aware S_DT placement problem, with the network size from 50 to 250. Fig. 2 shows the utility



(a) Utility (b) Running time
Fig. 2. Performance of different algorithms.



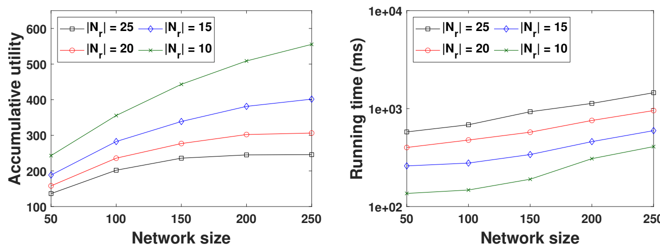
(a) Utility (b) Running time
Fig. 3. Impact of the number $|R|$ of requests on the performance of `Alg.1`.

gain and running time of the algorithms. When the network size is 250, the utility of `Alg.1` is 86.9% of that of `LP`, which outperforms `Gdy_u` and `Gdy_m` by 21.1% and 26.2%, respectively. The rationale is that `Alg.1` jointly considers the consumed privacy budget and memory resource for deploying S_DTs to optimize the total utility gain. Fig. 2(b) indicates that `Alg.1` takes more running time than that of `Gdy_u` and `Gdy_m`, because `Alg.1` first obtains a potential solution and then refines the potential solution to obtain the final solution.

We then investigated the impact of the number $|R|$ of requests on the performance of `Alg.1`, by varying the value of $|R|$ from 500 to $2,000$. Evidenced by Fig. 3(a), the utility 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.

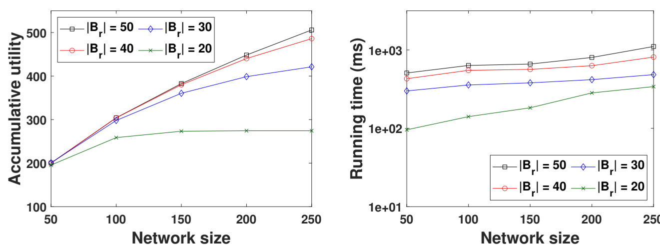
We also evaluated the impact of the number $|N_r|$ of candidate IoT devices for each request r on the performance of `Alg.1`, with $|N_r| = 10, 15, 20$ and 25, respectively. As seen from Fig. 4(a), the utility obtained by `Alg.1` when $|N_r| = 25$ is 44.2% of that by itself when $|N_r| = 10$, assuming 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) illustrates a larger value of $|N_r|$ leads to more running time, due to examining more candidate IoT devices for requests.

We finally evaluated the impact of the global privacy budget B_r of 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



(a) Utility (b) Running time

Fig. 4. Impact of the number $|N_r|$ of candidate IoT devices of 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`.

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.

VI. Conclusion

In this paper, we investigated social-aware service provisioning in DT-assisted SEC environments through DT placements. We first introduced a differential privacy-based federated learning framework for admitting service requests. Built upon the framework, we formulated a social-aware S_DT placement problem. We then provided an ILP formulation for the problem when the problem size is small or medium, otherwise we developed a performance-guaranteed approximation algorithm for it. Finally, we evaluated the algorithm performance via simulations. The simulation results indicate the proposed algorithm is promising, which outperforms its counterparts by at least 21.1%.

References

[1] S. Abdulrahman, S. Otoum, O. Bouachir and A. Mourad. Management of digital twin-driven IoT using federated learning. *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3636-3649, 2023.

[2] AWS Lambda. https://aws.amazon.com/lambda/. Accessed in February 2024.

[3] Azure. https://azure.microsoft.com/en-us/services/functions/. Accessed in February 2024.

[4] 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, 2020.

[5] 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 of Things Journal*, vol. 9, no. 23, pp. 23927 – 23940, 2022.

[6] 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. *Proc of ICC'22*, IEEE, 2022.

[7] L. Cui, Y. Qu, S. Yu, L. Gao, and G. Xie. A trust-grained personalized privacy-preserving scheme for big social data. *Proc. of ICC'18*, IEEE, 2018.

[8] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3 – 4, pp. 211 – 407, 2013.

[9] GT-ITM. http://www.cc.gatech.edu/projects/gtitm/, 2019.

[10] 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 of Things Journal*, vol. 9, no. 13, pp. 11154 – 11167, 2022.

[11] J. Li, S. Guo, W. Liang, J. Wang, Q. Chen, W. Xu, K. Wei, and X. Jia. Mobility-aware utility maximization in digital twin-enabled serverless edge computing. *IEEE Transactions on Computers*, vol. 73, no. 7, pp. 1837 – 1851, 2024.

[12] J. Li, S. Guo, W. Liang, J. Wu, Q. Chen, Z. Xu, W. Xu, and J. Wang. Wait for fresh data? digital twin empowered IoT services in edge computing. *Proc of MASS'23*, IEEE, pp. 397 – 405, 2023.

[13] J. Li, J. Wang, Q. Chen, Y. Li, and A. Y. Zomaya. Digital twin-enabled service satisfaction enhancement in edge computing. *Proc of INFOCOM'23*, IEEE, 2023.

[14] J. Li, S. Guo, W. Liang, J. Wang, Q. Chen, Z. Hong, Z. Xu, W. Xu, and B. Xiao. AoI-aware service provisioning in edge computing for digital twin network slicing requests. *IEEE Transactions on Mobile Computing*, to be published, 2024, doi: 10.1109/TMC.2024.3449818.

[15] J. Li, S. Guo, W. Liang, J. Wang, Q. Chen, Z. Xu, and W. Xu. AoI-aware user service satisfaction enhancement in digital twin-empowered edge computing. *IEEE/ACM Transactions on Networking*, vol. 32, no. 2, pp. 1677–1690, 2024.

[16] T. Luo, M. Pan, P. Tholoniat, A. Cidon, R. Geambasu, and M. Lécuyer. Privacy budget scheduling. *USENIX OSDI'21*, pp. 55 – 74, 2021.

[17] T. Öncan. A survey of the generalized assignment problem and its applications. *Information Systems and Operational Research*, vol. 45, no. 3, pp. 123 – 142, 2007.

[18] X. Shang, Y. Mao, Y. Liu, Y. Huang, Z. Liu, and Y. Yang. Online container scheduling for data-intensive applications in serverless edge computing. *Proc of INFOCOM'23*, IEEE, 2023.

[19] M. Vaezi, K. Noroozi, T. D. Todd, D. Zhao, G. Karakostas, H. Wu, and X. Shen. Digital twins from a networking perspective. *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23525 – 23544, 2022.

[20] 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 of Things Journal*, vol. 10, no. 17, pp. 14965 – 14987, 2023.

[21] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor. Federated learning with differential privacy: algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454 – 3469, 2020.

[22] G. Xu, B. Liu, L. Jiao, X. Li, M. Feng, K. Liang, L. Ma, and X. Zheng. Trust2Privacy: a novel fuzzy trust-to-privacy mechanism for mobile social networks. *IEEE Wireless Communications*, vol. 27, no. 3, pp. 72 – 78, 2020.

[23] Z. Xu, Y. Fu, Q. Xia, and H. Li. Enabling age-aware big data analytics in serverless edge clouds. *Proc of INFOCOM'23*, IEEE, 2023.

[24] Z. Xu, L. Zhou, W. Liang, Q. Xia, W. Xu, W. Ren, H. Ren, and P. Zhou. Stateful serverless application placement in MEC with function and state dependencies. *IEEE Transactions on Computers*, vol. 72. no. 9, pp. 2701 – 2716, 2023.

[25] 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 Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3401 – 3413, 2023.

[26] L. Zhao, Z. Zhao, E. Zhang, A. Hawbani, A. Y. Al-Dubai, Z. Tan, and A. Hussain. A digital twin-assisted intelligent partial offloading approach for vehicular edge computing. *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3386 – 3400, 2023.