Information Sciences 505 (2019) 562–570

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Privacy-aware service placement for mobile edge computing via federated learning

Yongfeng Qian[a], Long Hu[b,*], Jing Chen[c,*], Xin Guan[d,*], Mohammad Mehedi Hassan[e,f], Abdulhameed Alelaiwi[e,f]

[a] School of Computer Science, China University of Geosciences, Wuhan 430074, China
[b] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[c] School of Computer, Wuhan University, Wuhan 430072, China
[d] School of Data Science and Technology, Heilongjiang University, Harbin, China
[e] Research Chair of Smart Technologies, King Saud University, Riyadh 11543, Saudi Arabia
[f] College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

A R T I C L E I N F O

*Article history:*
Received 2 March 2019
Revised 15 July 2019
Accepted 18 July 2019
Available online 18 July 2019

*Keywords:*
Service placement
Edge cloud
Privacy preserving
Federated learning

A B S T R A C T

Mobile edge clouds can offer delay-sensitive services to users by deploying storage and computing resources at the network edge. Considering resource-limited edge server, it is impossible to deploy all services on the edge clouds. Thus, many existing works have addressed the problem of arranging services on mobile edge clouds for better quality of services (QoS) to users. In terms of existing service placement strategies, the historical data of requesting services by users are collected to analyze. However, those historical data belong to users' sensitive information, without appropriate privacy preserving measures may hinder the implementation of traditional service arrangement strategies. Service placement with considering users' privacy and limited resources of mobile edge clouds, is an extremely urgent problem to be solved. In this paper, we propose a privacy-aware service placement (PSP) scheme to address the problem of service placement with privacy-awareness in the edge cloud system. The purpose of PSP mechanism is to protect users' privacy and provide better QoS to users when obtaining services from mobile edge clouds. Specifically, whether service placement on mobile edge clouds or not is modeled as a 0–1 problem. Then, a hybrid service placement algorithm is proposed that combines a centralized greedy algorithm and distributed federated learning. Compared with other optimization schemes, the simulation experiments show that PSP algorithm could not only protect users' privacy but also meet users' service demands through mobile edge clouds.



## 1. Introduction

With the rapid development of Internet of Things, there is an increasing demand of computing and storage resources in terms of mobile applications and services, such as virtual reality (VR) and augmented reality (AR) [10,12,17,28,29]. In order to overcome the limitations of battery capacity and computing ability of mobile devices, those services can be offloaded to

* Corresponding authors.
*E-mail addresses:* yfqian@cug.edu.cn (Y. Qian), hulong@hust.edu.cn (L. Hu), chenjing@whu.edu.cn (J. Chen), guanxin@hlju.edu.cn (X. Guan), mmhassan@ksu.edu.sa (M.M. Hassan), aalelaiwi@ksu.edu.sa (A. Alelaiwi).

https://doi.org/10.1016/j.ins.2019.07.069
0020-0255/

remote clouds to process [16] [23]. Due to a long transmission distance, however, it brings high latency for users to wait when obtaining services from remote clouds [21] [19]. In this case, it provides low quality of service (QoS) to mobile users since the delay of waiting for services is too large to endure [14,26].

As a novel distributed cloud framework, mobile edge clouds have the capability of computing, storage, and communication at the network edge, which can handle computing-intensive and delay-sensitive services offloaded to edge clouds [4,30]. This is because edge clouds with computing and storage resources are close to users (i.e., one hop usually), which brings low delay [8]. Therefore, edge computing can not only guarantee high QoS, but also provide low delay for mobile users [20,33].

Based on some advanced technologies [5,6], such as software-defined network (SDN) and network function virtualization (NFV), mobile edge computing is changing services interaction pattern to mobile users [3,27]. Existing researches about mobile edge computing can be roughly divided into three categories: (i) *task offloading* [7,25], which tries to identify an optimal collaborative task offloading scheme to decide where to process the task, such as mobile devices, edge clouds, or remote clouds; (ii) *resource allocation* [35], which aims to design a resource allocation strategy to allocate resources to maximize resource utilization in terms of limited computing, storage, and communication resources of edge clouds. (iii) *services placement* [14], whose objective is to explore an optimal service placement scheme to maximize QoS of mobile users. Specially, services placement on mobile edge clouds can further reduce latency of users to obtain services [22,31]. Taking scene rendering of VR as an example, scene rendering and popular videos should be arranged on mobile edge clouds in advance to reduce the delay when mobile users require VR services.

Considering limited computing, storage and communication resources of mobile edge clouds, it is unrealistic to place all services on mobile edge clouds, which are different from remote clouds with abundant resources. Therefore, it is necessary to consider which services can be placed on mobile edge clouds, i.e., designing an optimal service placement scheme. Service placement has attracted increasing interests during the last few years. For example, Chen *et al.* [7] design an opportunistic task offloading model based on cloudlets, which realizes the tradeoff between cost and delay. Hao *et al.* [13] introduce an optimal service placement scheme under limited storage resources on mobile edge clouds, which aims to minimize consumed energy. Xu *et al.,* [32] give a joint service placement and task offloading scheme which takes users' required services and tasks to be solved into consideration simultaneously.

However, above schemes ignore users' privacy [11] while do not consider limited computing, communication, and storage resources of mobile edge clouds when to design service placement strategies. Specifically, existing service placement strategies depend on the degree of users' preference toward services. For example, when users prefer one service to a high degree, the quantity of their demands on this service is large. When users prefer one service to a low degree, the quantity of their demands on this service is small. However, degrees of users' preference on services belong to theirs privacy information [15,34], which may include users' locations [1], environments and personalized demands that users do not want to reveal [2,18]. However, the existing placement strategies do not care users' privacy which is a compelling but less studied problem that needs to be addressed. Furthermore, existing privacy preservation schemes towards content caching are not applicable, since service demands are more complex than content demands. For example, service demands relate to storage, communication, and computing resources of mobile edge clouds [9,24]. In addition, service demands become more personalized and involve more users' privacy. Therefore, it is necessary to explore a privacy-aware service placement strategy.

In this paper, we design an optimal service placement strategy in terms of limited computing, storage, and communication resources of mobile edge clouds, while preserving users' privacy simultaneously. Specifically, we first explore how to maximize the satisfaction of users' services demands under limited resources (i.e., computing, storage, and communication resources) of mobile edge clouds. Second, we build users' preference model and use federated learning to estimate users' service demands while protecting their privacy. Finally, the simulation experiment results show that the privacy-aware service placement algorithm can not only protect users' privacy but also meet more users' service demands from mobile edge clouds comparing with traditional service placement strategy. The main contributions of this paper can be summarized as follows.

- We propose a privacy-aware service placement scheme for mobile edge clouds. To the best of our knowledge, this is the first paper that studied joint users' privacy preserving and service placement in mobile edge clouds system. Specially, considering users' privacy and limited communication, computing, and storage resources of mobile edge clouds, we can obtain which services can be placed on mobile edge clouds.
- We formulate the privacy-aware service placement problem as a 0–1 problem. Furthermore, federated learning is utilized to preserve mobile users' privacy, which can train users' preference model locally.
- The experiment results show that the algorithm can not only protect users' privacy but also realize better performance than existing task arrangement strategy.

The rest of this paper is organized as follows. We present the system model and problem formulation in Section 2. The privacy-aware service placement scheme is described in Section 3. Our simulation results and discussions are given in Section 4. Finally, Section 5 concludes this paper.

## 2. System model and problem formulation

In this section, we will introduce the system model and problem formulation.

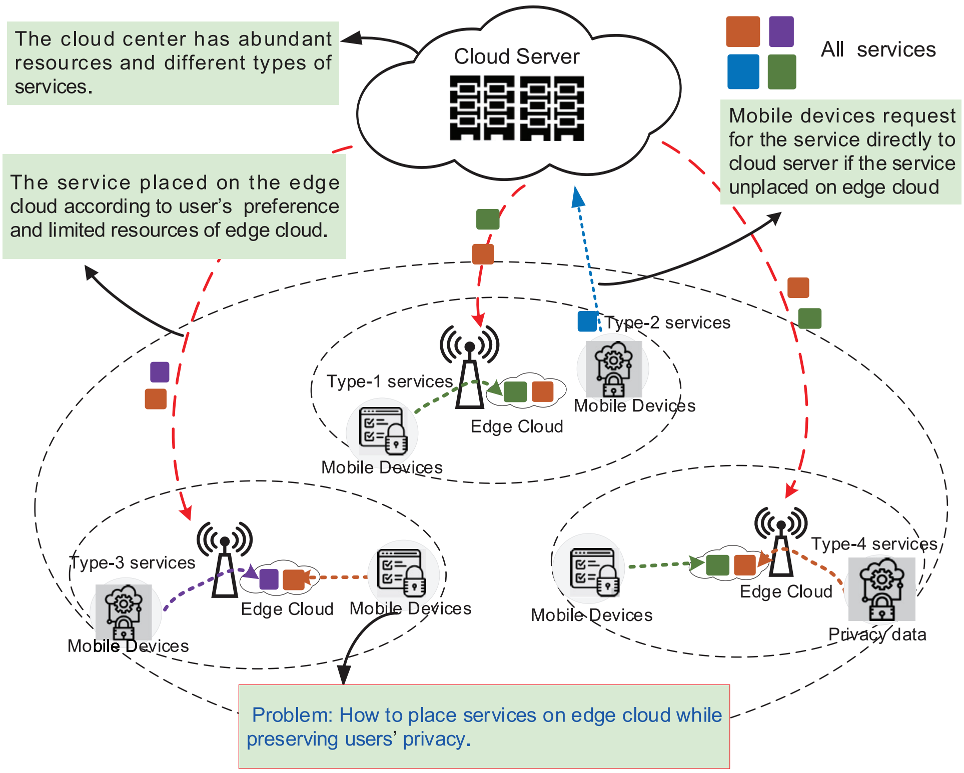



**Fig. 1.** Illustration of privacy-aware service placement on the edge cloud.

### 2.1. System overview

Without loss of generality, we assume the system proposed in this paper has one mobile edge cloud and several mobile devices. Let $\mathcal{N}$ denote the set of mobile devices, indexed by $\mathcal{N} = \{1, 2, \cdots, N\}$. Each mobile device can communicate with the edge cloud through wireless channel. The edge cloud has computing, storage, and communication resources. Let the computing power of this edge cloud (i.e., the maximum CPU frequency) be $C$, which can be utilized to process services. Assume the storage capacity of this edge cloud is $S$, which can be used to store data relevant to services. Denote $B^u$ and $B^d$ as the bandwidth of the uplink and downlink respectively, which describe uploading and downloading services required for users.

As mentioned above, the computing, storage, and communication resources of the edge cloud are limited, where not all services can be placed. Therefore, it is necessary to determine which services can be placed on edge cloud to maximally satisfy service demands of users. As shown in Fig. 1, edge clouds can satisfy users services requirements if placed those services on edge clouds in advance, such as Type-4 services. If the services requested by users are not cached on the edge cloud in advance, however, users need to obtain services from remote clouds through cellular network, such as Type-2 service. Therefore, how to place services on edge clouds is a crucial problem needed to be addressed. Furthermore, in order to preserving users' preference towards different services, the service arrangement strategy should also take privacy protection into consideration. In summary, how to place services while preserving uses' privacy is the main problem in this paper.

### 2.2. Service model

Assume the set of services is $\mathcal{K} = \{1, 2, \cdots, K\}$. The computing, storage, and communication resources required for every service are different. For example, a video-related service requires more communication and storage resources. A VR service demands more computing, storage, and communication resources. Let $c_k, s_k, r_k^u, r_k^d$ denote the $k$th service's demands for computing, storage, and communication resource, where $c_k$ is the computing space needed by service $k$, $s_k$ is the storage space needed by service $k$, and $r_k^u$ and $r_k^d$ are the bandwidth of uplink and downlink needed by service $k$.

Let $a_k \in \{0, 1\}$ denote the service placement variable which indicates whether service $k$ is placed at the edge cloud ($a_k = 1$) or not ($a_k = 0$). If $a_k = 1$, users can obtain service $k$ from this edge cloud. In contrary, if $a_k$ is equal to 0, users can not obtain service $k$ from the edge cloud, which may bring long delay because users will obtain services from remote clouds.

Furthermore, the following vector can denote the service arrangement strategy.

$$\mathbf{a} = (a_k \in \{0, 1\} : k \in \mathcal{K}) \tag{1}$$

### 2.3. Users'preference model

The times that user $n$ requests for service $k$ is $d_{n,k}$. It should be noticed that the times of the user's requests are relevant to users' preference on services. If one user prefers a service, then this user will often request this service. If this user does not like the service, then the service will not be requested by this user. Therefore, it is necessary to learn the users' preference model to obtain the times of requests. Specifically, the following formula denotes the request times $d_{n,k}$.

$$d_{n,k} = f_k(x(n)) \tag{2}$$

where $x(n) \in \mathbb{R}^m$, which contains users' context information, such as their ages and locations. Without loss of generality, the vector $x(n, k)$ is standardized, that is, $x(n, k) \in [0, 1]^m$. $f_n(\cdot)$ denotes the nonlinear relationship between the request times and users' preference.

We utilize sigmoid function to estimate user's preference toward service $k$ as follows.

$$p_{n,k} = p_{\theta_n}[y(k) = 1|x(n)] = \frac{1}{1 + e^{-(\theta_n^T x(n))}} \tag{3}$$

where $y(k) = 1$ if and only if the service $k$ belongs to user's preferred category. $\theta_n$ is the weight parameter, and $p_{n,k}$ is the user's preference degree of service $k$. Moreover, the probability of all users' requests on the service $k$ can be obtained.

$$p_k = \frac{1}{N}\sum_{n=1}^{N} p_{n,k} \tag{4}$$

Furthermore, the times of the users' requests for services are relevant to their personalized demands. Thus, the edge cloud can forecast users' demands at the next moment and update the corresponding service placement strategy based on the users' preference model. For example, the edge cloud can replace the services requested for few times to be those services requested for many times. In addition, updating the services arranged on edge clouds need backhaul to retrieve services from the cloud terminal, which will include some costs.

### 2.4. Problem formulation

Based on the above discussions, we can know that the computing, storage, and communication resources of the edge cloud are limited. When to design service placement strategy $a_k$, the following constraints should be satisfied in terms of the limited resources of the edge cloud.

(i) The storage capacity needed by services placed on the edge cloud should be less than the maximum storage capacity of the edge cloud.

$$C1 : \sum_{k=1}^{K} a_k s_k \le S \tag{5}$$

(ii) The services running on the edge cloud should be no more than the total computing capability of the edge cloud.

$$C2 : \sum_{n=1}^{N}\sum_{k=1}^{K} d_{n,k} a_k c_k \le C \tag{6}$$

(iii) The uplink and downlink communication bandwidth of services on the edge cloud should not exceed the maximum communication bandwidth of the edge cloud.

$$C3 : \sum_{n=1}^{N}\sum_{k=1}^{K} d_{u,k} a_k r_k^u \le B^u \tag{7}$$

$$C4 : \sum_{n=1}^{N}\sum_{k=1}^{K} d_{u,k} a_k r_k^d \le B^d \tag{8}$$

Regarding the service placement scheme, the objective is to maximize the quantity of services from the edge cloud while protecting users' privacy. Specifically, the problem can be described as follows:

$$\mathbf{P1} : \underset{\mathbf{a_k^t}}{\text{maximize}} \ \frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N_t} D_n^t(a_k^t) \tag{9}$$

$$\text{subject to} \ \ C1 - C4 \tag{10}$$

$$C5 : a_k \in \{0, 1\}. \tag{11}$$

where the objective function is to maximize users' services demands from the edge cloud. The first constraint indicates that arranged services should not exceed the maximum storage capacity of the edge cloud. The second constraint means that the computing capability needed for the services should not exceed the maximum computing capability of the edge cloud. The third constraint means that the uplink bandwidth needed by all services should not exceed the uplink bandwidth of the edge cloud. The fourth constraint means that the downloading bandwidth needed by all services should not exceed the downlink bandwidth of the edge cloud. The fifth constraint is to describe whether the service is placed on the edge cloud or not.

## 3. The privacy-aware service placement scheme

To preserving users' privacy, we utilize distributed federated learning to learn users' preferences on services. Then, we will use a greedy algorithm to realize the service placement scheme.

### 3.1. Federated learning for users preference

We use federated learning to learn users' preferences on services while protecting users' privacy. This is because federated learning trains the preference model locally without submitting users' privacy data to the edge cloud, which protects users' privacy and obtains the users' preference model simultaneously. Specifically, mobile users can use the data collected by themselves to train the preference model and then offload parameters to the edge cloud to update. After updating parameters, the edge cloud gives feedback to the mobile device. The mobile device trains it again according to the new parameters and obtains the users' preferences model.

Then, we will show how to use federated learning. First, we utilize mobile equipments to update parameters. The logistics loss function of the $n$th user is defined to be $l(x_n, y(k); \theta_n)$, which can be denoted as follows.

$$l(x_n, y_k; \theta_n) = -y_k \log(p_{n,k}) - (1 - y_k) \log(1 - p_{n,k}) \tag{12}$$

where $y_k \in \{0, 1\}$. We use gradient descent method to solve this loss function. Therefore, when it is iterated in the $j$th time, we can get the following equation.

$$\theta_n^{(j+1)} = \theta_n^{(j)} - \eta^{(j)} g^{(j)}, \quad j = 1, 2, \cdots \tag{13}$$

where $g^{(j)} = \nabla_{\theta_n} l(x_n, y_k; \theta_n)$ is the gradient vector about $\theta_n$ and $\eta^{(j)}$ is learning rate. The iteration termination condition of the $n$th mobile devices is as follows.

$$||\theta_n^{(j+1)} - \theta_n^{(j)}|| \le \epsilon \tag{14}$$

where $\epsilon$ is a small integer. Through the above analysis, we can obtain $N$ parameters $\theta_1$, $\theta_2$, $\cdots$, $\theta_N$.

Then, the edge cloud can update the above parameters. Specifically, the updating strategy is given below.

$$\theta^{(j)} = \frac{1}{N} \sum_{i=1}^{N} \theta_i^{(j)} \tag{15}$$

In terms of the above equation, the parameters are trained uniformly. Therefore, users only need to send the trained results to the edge cloud, instead of sending all the users' privacy data to it. In this way, users' privacy can be preserved and a training model can be obtained.

### 3.2. A Greedy Algorithm for service placement

As for the optimization problem P1, it is a 0–1 optimization problem. Based on the knapsack problem, we can get this optimization problem is an NP-hard problem. In order to solve this problem, we use a greedy algorithm in this paper. The greedy algorithm is shown below. First, the service that can maximally improve the objective function can be placed on the edge cloud. Then, the secondary service is placed on the edge cloud. The rest can be done in the same manner until the computing, communication, or storage resources of all services reach the upper constraint of the edge cloud.

## 4. Simulations and performance analysis

In this section, we will show experimental results and corresponding analysis about the privacy-aware service placement (PSP) scheme proposed in this paper and other traditional service placement strategies, i.e., random services placement strategy and popular services placement strategy.

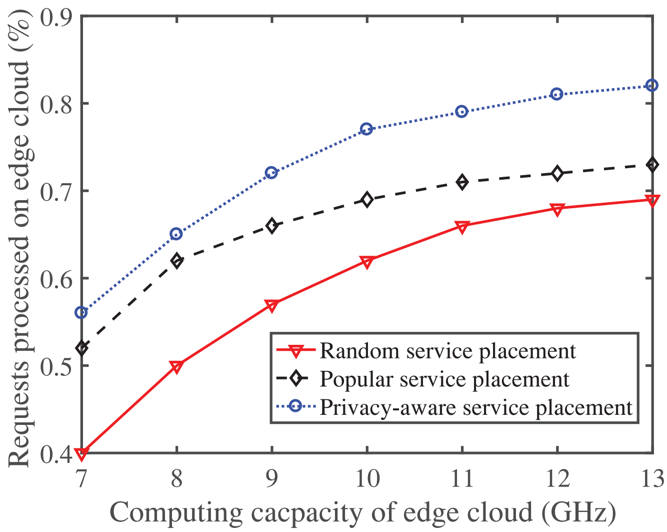



**Fig. 2.** The impact of edge cloud computing capacity on the percentage of requests processed on edge cloud.

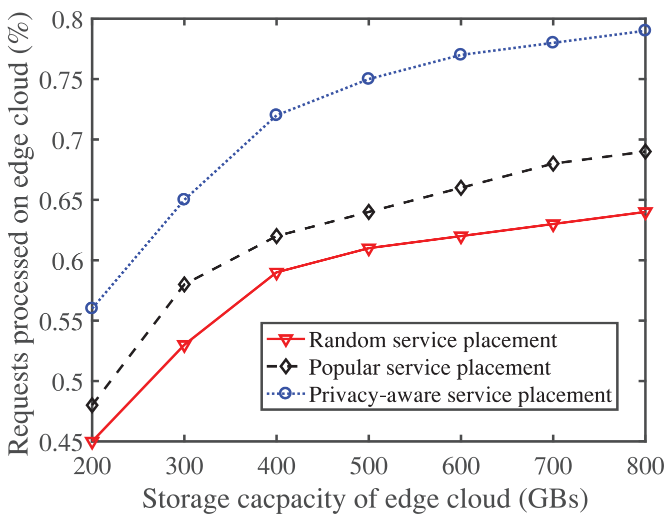
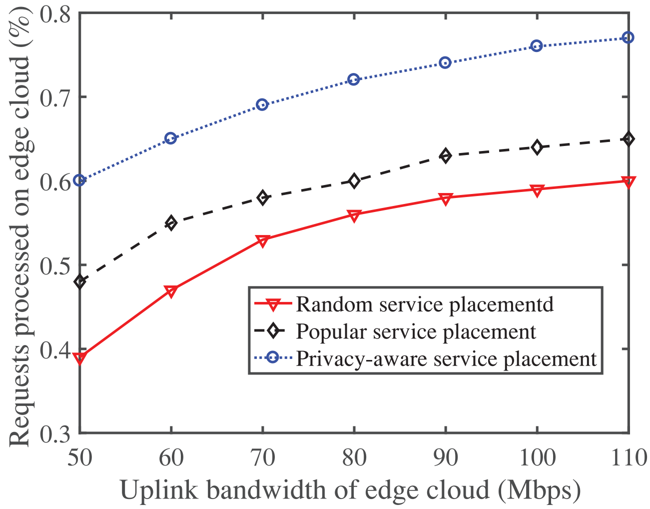



**Fig. 3.** The impact of edge cloud storage capacity on the percentage of requests processed on edge cloud.

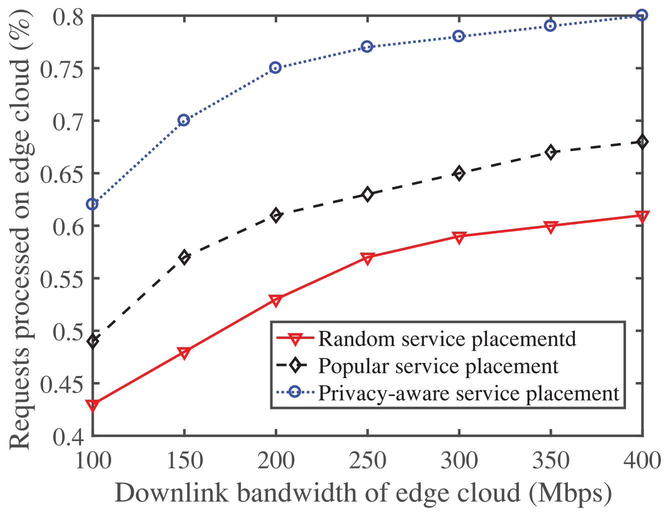
### *4.1. Simulation setting*

In this section, we evaluate a system that contains one edge cloud and several mobile users. To show the relation between users' demands and users' context, the following parameters are set. The number of users contained is $N = 10208$. The number of services is $K = 23$. Users' context information included age, gender, nationality, occupation, education, and equipment type. Actually, the information involves users' privacy and users are not willing to offload those data to the edge cloud. Services' information includes game-type applications among others. In addition, we assume that the services requested by users can be placed on edge cloud. Federated learning is used to learn users' demand patterns of services.

The computing, communication, and storage resources owned by the edge cloud and needed by services are as follows. In terms of the edge cloud, the computing capacity of the edge cloud is set to $C = 10$ GHz. The storage capacity of the edge cloud is $S = 500$ GBs. The uplink bandwidth of the edge cloud is $B^u = 75$ Mbps, and the downlink bandwidth of the edge cloud is $B^d = 250$ Mbps. As for services, assume that the computing capacity $c_k$ of service $k$ is randomly chosen from [0.1,0.5] GHz. The storage capacity $s_k$ of the service $k$ is randomly chosen from [20,100] GBs. The uplink bandwidth $r_k^u$ of the service $k$ is randomly chosen from [1,5] Mbps. The downlink bandwidth $r_k^d$ of the service $k$ is randomly chosen from [1,20] Mbps.




(a)




(b)

**Fig. 4.** The impact of edge cloud bandwidth capacity on the percentage of requests processed on the edge cloud: (a) The impact of edge cloud uplink bandwidth capacity on the percentage of requests processed on the edge cloud; (b) The impact of edge cloud downlink bandwidth capacity on the percentage of requests processed on the edge cloud.

### *4.2. Algorithm comparison*

The proposed PSP algorithm is compared with the following two algorithms (i.e., random service placement strategy and popular service placement strategy).

- *Random service placement strategy*, which chooses the service randomly from the services' library every time until the computing, communication, or storage resources of all services placed reaches the upper limit of the edge cloud.
- *Popular service placement strategy*, which estimates the request pattern of each service based on users preference and places the services on the edge cloud in order according to the request pattern until the computing, communication, or storage resources of all services placed reaches the upper limit of the edge cloud.

### 4.3. Performance analysis

In this section, the impact of the computing, storage, and communication resources of the edge cloud are analyzed about different service placement schemes. From Fig. 2 to Fig. 4, the x-coordinate indicates the computing, storage and communication resource of the edge cloud respectively, and the y-coordinate indicates the percentage of the requests processed on the edge cloud.

The impact of the computing ability of the edge cloud on service placement strategy is first analyzed. Fig. 2 shows that, with the improvement of the computing ability of the edge cloud, the percentage of services' requests processed on the edge cloud increases for the three service arrangement strategies. This is because the edge cloud can contain more services and meet more service requests after the computing ability of the edge cloud is improved. Furthermore, the PSP scheme in this paper is much better than random services placement strategy and popular services placement strategy. This is because the PSP scheme considers users' privacy, users' requests pattern, and the limited computing, storage, and communication resources of the edge cloud.

Then, the influence of the edge cloud's storage ability on service placement strategy is analyzed. Fig. 3 shows that with the improvement of the edge cloud's storage ability, the percentage of requests processed on the edge cloud increases. Compared with the influence of computing capacity on the service, the influence in storage capacity on service placement is greater. Furthermore, similar with Fig. 2, with the increasement of storage capacity, the PSP scheme is better than the other two schemes.

Finally, Figs. 4(a) and 4(b) show the influence of uplink bandwidth and downlink bandwidth on different services placement schemes, respectively. From those two sub-figures, it can be seen that with the increase of uplink bandwidth and downlink bandwidth, more services requirements by users can be processed on the edge cloud.

## 5. Conclusion

In this paper, we first analyze services placement schemes on the edge cloud considering limited computing, storage, and communication resources and model it to be a 0–1 optimization problem. Second, we use distributed federated learning to learn users' preferences and a greedy algorithm is used to solve the optimization problem. Based on this, PSP scheme is proposed that can not only protect users' privacy but also decrease the load of services on the remote cloud. Finally, simulation results showed that the proposed PSP scheme is more effective compared to other schemes. For future work, we will consider using PSP for several edge clouds.

## Conflict of Interest

The authors declare that they have no conflicts of interest to this work. They declare that they do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

## Acknowledgement

The authors are grateful to the Deanship of Scientific Research at King Saud University for funding this work through the Vice Deanship of Scientific Research Chairs: Chair of Smart Technologies.

## References

[1] J. Chen, K. He, Q. Yuan, M. Chen, R. Du, Y. Xiang, Blind filtering at third parties: an efficient privacy-preserving framework for location-based services, IEEE Trans. Mobile Comput. 17 (11) (2018) 2524–2535.
[2] J. Chen, K. He, Q. Yuan, R. Du, L. Wang, Batch identification game model for invalid signatures in wireless mobile networks, IEEE Trans. Mobile Comput. 16 (6) (2017) 1530–1543.
[3] J. Chen, C. Wang, Z. Zhao, K. Chen, R. Du, G.-J. Ahn, Uncovering the face of android ransomware: characterization and real-time detection, IEEE Trans. Inf. Forensics Secur. 13 (5) (2018) 1286–1300.
[4] L. Chen, J. Xu, Task offloading and replication for vehicular cloud computing: a multi-armed bandit approach, (2018) arXiv:1812.04575.
[5] M. Chen, Y. Hao, Label-less learning for emotion cognition, IEEE Trans. Neural Netw. Learn. Syst. (2019). doi: 10.1109/TNNLS.2019.2929071.
[6] M. Chen, Y. Hao, H. Gharavi, V.C. Leung, Cognitive information measurements: a new perspective, Inf. Sci. (2019). https://arxiv.org/abs/1907.01719.
[7] M. Chen, Y. Hao, C.-F. Lai, D. Wu, Y. Li, K. Hwang, Opportunistic task scheduling over co-located clouds in mobile environment, IEEE Trans. Serv. Comput. 11 (3) (2016) 549–561.
[8] M. Chen, Y. Hao, K. Lin, Z. Yuan, L. Hu, Label-less learning for traffic control in an edge network, IEEE Netw. 32 (6) (2018) 8–14.
[9] M. Chen, F. Herrera, K. Hwang, Cognitive computing: architecture, technologies and intelligent applications, IEEE Access 6 (2018) 19774–19783.
[10] M. Chen, W. Li, G. Fortino, Y. Hao, L. Hu, I. Humar, A dynamic service migration mechanism in edge cognitive computing, ACM Trans. Internet Technol. (TOIT) 19 (2) (2019) 30.
[11] M. Chen, Y. Qian, J. Chen, K. Hwang, S. Mao, L. Hu, Privacy protection and intrusion avoidance for cloudlet-based medical data sharing, IEEE Trans. Cloud Comput. (2016), doi:10.1109/TCC.2016.2617382.
[12] M. Chen, J. Zhou, G. Tao, J. Yang, L. Hu, Wearable affective robot, IEEE Access 6 (2018) 64766–64776.
[13] Y. Hao, M. Chen, L. Hu, M.S. Hossain, A. Ghoneim, Energy efficient task caching and offloading for mobile edge computing, IEEE Access 6 (2018) 11365–11373.
[14] T. He, H. Khamfroush, S. Wang, T. La Porta, S. Stein, It's hard to share: Joint service placement and request scheduling in edge clouds with sharable and non-sharable resources, in: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2018, pp. 365–375.
[15] H. Jiang, C. Cai, X. Ma, Y. Yang, J. Liu, Smart home based on wifi sensing: a survey, IEEE Access 6 (2018) 13317–13325.

[16] S. Josilo, G. Dan, Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks, IEEE Trans. Mobile Comput. 18 (1) (2019) 207–220.
[17] J. Li, Z. Ming, M. Qiu, G. Quan, X. Qin, T. Chen, Resource allocation robustness in multi-core embedded systems with inaccurate information, J. Syst. Archit. 57 (9) (2011) 840–849.
[18] Y. Li, W. Dai, Z. Ming, M. Qiu, Privacy protection for preventing data over-collection in smart city, IEEE Trans. Comput. 65 (5) (2016) 1339–1350.
[19] K. Lin, M. Chen, J. Deng, M.M. Hassan, G. Fortino, Enhanced fingerprinting and trajectory prediction for IoT localization in smart buildings, IEEE Trans. Autom. Sci. Eng. 13 (3) (2016) 1294–1307.
[20] K. Lin, J. Luo, L. Hu, M.S. Hossain, A. Ghoneim, Localization based on social big data analysis in the vehicular networks, IEEE Trans. Industr. Inf. 13 (4) (2017) 1932–1940.
[21] K. Lin, J. Song, J. Luo, W. Ji, M.S. Hossain, A. Ghoneim, Green video transmission in the mobile cloud networks, IEEE Trans. Circuits Syst. Video Technol. 27 (1) (2016) 159–169.
[22] Y. Ma, Y. Hao, M. Chen, J. Chen, P. Lu, A. Košir, Audio-visual emotion fusion (avef): a deep efficient weighted approach, Inf. Fusion 46 (2019) 184–192.
[23] J.L.D. Neto, S.-Y. Yu, D.F. Macedo, M.S. Nogueira, R. Langar, S. Secci, et al., Uloof: a user level online offloading framework for mobile edge computing, IEEE Trans. Mobile Comput. 17 (11) (2018) 2660–2674.
[24] S. Penghao, H. Yuxiang, L. Julong, T. Le, M. Chen, Tide: time-relevant deep reinforcement learning for routing optimization, Future Gener. Comput. Syst. 99 (2019) 401–409.
[25] L. Pu, L. Jiao, X. Chen, L. Wang, Q. Xie, J. Xu, Online resource allocation, content placement and request routing for cost-efficient edge caching in cloud radio access networks, IEEE J. Sel. Areas Commun. 36 (8) (2018) 1751–1767.
[26] Y. Qian, Y. Jiang, J. Chen, Y. Zhang, J. Song, M. Zhou, M. Pustišek, Towards decentralized IoT security enhancement: a blockchain approach, Comput. Electr. Eng. 72 (2018) 266–273.
[27] Y. Qian, Y. Zhang, X. Ma, H. Yu, L. Peng, Ears: Emotion-aware recommender system based on hybrid information fusion, Inf. Fusion 46 (2019) 141–146.
[28] M. Qiu, Z. Chen, J. Niu, Z. Zong, G. Quan, X. Qin, L.T. Yang, Data allocation for hybrid memory with genetic algorithm, IEEE Trans. Emerg. Topics Comput. 3 (4) (2015) 544–555.
[29] M. Qiu, Z. Ming, J. Li, J. Liu, G. Quan, Y. Zhu, Informer homed routing fault tolerance mechanism for wireless sensor networks, J. Syst. Archit. 59 (4–5) (2013) 260–270.
[30] D. Wang, Y. Peng, X. Ma, W. Ding, H. Jiang, F. Chen, J. Liu, Adaptive wireless video streaming based on edge computing: opportunities and approaches, IEEE Trans. Serv. Comput. (2018), doi:10.1109/TSC.2018.2828426.
[31] L. Wang, L. Jiao, T. He, J. Li, M. Mühlhäuser, Service entity placement for social virtual reality applications in edge computing, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 468–476.
[32] J. Xu, L. Chen, P. Zhou, Joint service caching and task offloading for mobile edge computing in dense networks, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 207–215.
[33] P. Yang, N. Zhang, S. Zhang, L. Yu, J. Zhang, X.S. Shen, Content popularity prediction towards location-aware mobile edge caching, IEEE Trans. Multimedia 21 (4) (2018) 915–929.
[34] P. Zhao, J. Li, F. Zeng, F. Xiao, C. Wang, H. Jiang, Illia: Enabling *k*-anonymity-based privacy preserving against location injection attacks in continuous lbs queries, IEEE Internet Things J. 5 (2) (2018) 1033–1042.
[35] T. Zhao, I.-H. Hou, S. Wang, K. Chan, Red/led: an asymptotically optimal and scalable online algorithm for service caching at the edge, IEEE J. Sel. Areas Commun. 36 (8) (2018) 1857–1870.