Privacy-Preserving QoS Forecasting in Mobile Edge Environments

Pengcheng Zhang, *Member, IEEE*, Huiying Jin, Hai Dong, *Member, IEEE*,
Wei Song, *Member, IEEE*, and Athman Bouguettaya, *Fellow, IEEE*

Abstract—Mobile Edge Computing is an emerging technology offering low latency responses by deploying edge servers near mobile devices. We propose a novel privacy-preserving QoS forecasting approach – Edge-Laplace QoS (QoS forecasting with Laplace noise in mobile Edge environments) to address the challenges of user mobility and information leakage encountered by QoS forecasting in mobile edge environments. Edge-Laplace QoS is able to accurately and efficiently forecast Quality of Service (QoS) of various Web Services, while effectively protecting user privacy in mobile edge environments. We employ an improved differential privacy method to add dynamic disguises to the original QoS data in the edge environment to protect user data privacy. A collaborative filtering method is adopted to retrieve similar users' accessing records based on geographic locations of their accessed servers for QoS forecasting. We conduct a set of experiments using several public network data sets. The results show that the efficiency of Edge-Laplace QoS is superior to traditional forecasting approaches. Edge-Laplace QoS is also validated to be more suitable for edge environments than traditional privacy-preserving approaches.

Index Terms—Moving edge, mobile devices, quality of service, user privacy, geographic location, differential privacy, fast edge forecasting.

1 Introduction

Mobile devices are increasingly being used as the medium of choice to conduct business activities [1]. Likewise, cloud computing is by and large the way businesses and individuals manage their data and conduct their computations [2], [3]. Web services are frequently used as the main abstraction to organize and access resources on the Web [4]. Web service developers typically publish similar functionalities pertaining to different Web services [5]. Quality of Service (QoS) is usually the non-functional criterion that is used to select Web services with similar functionalities [6], [7]. Common QoS parameters include *response time*, *throughput*, *cost*, etc. [8]. Online personalized QoS forecasting provides an important basis for users to select and recommend suitable Web services in a dynamic network environment [9], [10], [11], [12]. Mobile Edge Computing (MEC) is an emerging technology [13]. It offers short response time and fast processing speed by deploying edge servers close to mobile devices to ensure the delivery of reliable services. However, the use of mobile edge servers poses new challenges including data privacy and security [14]. In this instance, collaborative filtering techniques running on edge servers are used to speed up the selection process and avoid going back to the cloud servers.

There has been a large body of research in the area of Web services QoS prediction via collaborative filtering [12], [15], [16], [17], [18]. Most collaborative filtering methods predict un-invoked service QoS by using similar users' historical data. The major research directions in this area include personalized recommendation [19], local neighborhood matrix factorization [20], context sensitive [21] and privacy-preserving QoS prediction [22]. However, these approaches are largely suitable for services in traditional environments [15]. They will probably meet two major challenges in the mobile edge environment: 1) *context-sensitively and accurately forecasting edge service QoS values*, and 2) *preventing users' personal QoS information disclosure in the collaborative filtering process* [23]. The following scenario explains the two challenges.

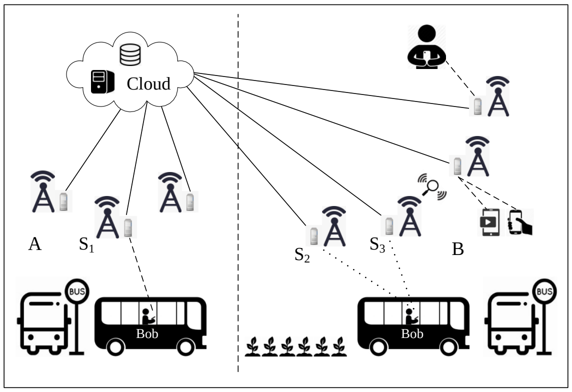


Fig. 1: Service invocation scenario in mobile edge computing

Assume that a user *Bob* who is on a bus and has been watching a *YouTube* video by accessing edge server S_1 in

- P. Zhang and H. Jin are with the College of Computer and Information, Hohai University, Nanjing, China
E-mail: pchzhang@hhu.edu.cn; 367046895@qq.com
- H. Dong is with the School of Science, Royal Melbourne Institute of Technology, Melbourne, VIC 3001, Australia
E-mail: hai.dong@rmit.edu.au
- W. Song is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
E-mail: wsong@njust.edu.cn
- A. Bouguettaya is with the School of Computer Science, The University of Sydney, NSW, Australia
E-mail: athman.bouguettaya@sydney.edu.au

Manuscript received XXX XX, XXXX; revised XXX XX, XXXX.

edge region A. As Fig. 1 depicts, *Bob*'s bus is moving to edge region B, where *YouTube* can be accessed from several different edge servers (i.e., S_2 and S_3). Now the question is to choose the right edge server enabling *Bob* to continuously access the same *YouTube* video with similar quality (e.g., resolution). This requires us to forecast the average QoS of these edge servers for provisioning the *YouTube* video service during the time period that *Bob* is in edge region B. Now these servers do not contain the historical QoS data of *Bob* for accessing the *YouTube* video. The traditional way of forecasting is to reference *Bob*'s historical QoS data for accessing the video in other servers, e.g., the historical QoS of the video provisioning in server S_1. However, the historical QoS data in server S_1 is no longer valid for the forecasting. This results from the change of the edge region and the corresponding environment (e.g., bandwidth or network traffic). Therefore, these servers need to reference their local users' historical QoS data with respect to the video provisioning. This however causes the leakage of these users' private information.

We summarize the problems of Web service QoS forecasting in the mobile edge environment as follows:

i). Traditional QoS forecasting approaches may cause users' personal QoS information leakage. Users in the same region of an edge usually invoke services from the same base station. Therefore, these users can be viewed as being affiliated to similar edge environments (i.e., similar available bandwidth and network traffic). Similarly, users in different regions may be affiliated to different edge environments [13]. Real-time movement of user locations leads to changes in the edge environment and invalidates historical data, with more mobile users invoking services on the edge. This makes it necessary to constantly search for similar users with the purpose of QoS forecasting. This process lead to the leakage of many users' personal QoS information. Therefore, QoS forecasting based on privacy-preserving has significant research impact.

ii). Context-sensitive similar user searching is a bottleneck for QoS forecasting in the mobile edge environment. During the forecasting process, it is difficult to find similar users due to the real-time differences in regional environments (i.e., base stations, available bandwidth or network traffic) and service data (i.e., response time, throughput, reliability or cost). At present, most similarity calculation methods are relatively static and purely based on historical data. They do not consider the impact of environmental and real time differences on selecting similar users. Therefore, these methods are inapplicable for the edge environment.

We propose a novel privacy-preserving Web service QoS forecasting approach in the mobile edge environment, abbreviated as Edge-Laplace QoS (QoS forecasting with Laplace noise in Mobile Edge Environments). An edge region is divided into several geographic locations to obtain the precise edge QoS data information and adapt to the dynamic edge environment. The improved differential privacy method with the function of constant noise value updating is used to disguise the dynamic and variable edge-end services, and realize the QoS forecasting with edge-end privacy protection. In general, the contributions of this paper mainly include the following three aspects:

- *We design a privacy-preserving solution for personal QoS information protection in the mobile edge environment.* Laplace noise is added into the context-sensitive QoS data to realize the edge disguising, where the noise is continuously and dynamically generated by the differential privacy method [24]. The Laplace mechanism smartly uses the disguised data for user similarity calculation, which not only effectively protects user privacy but also ensures a high forecasting accuracy relatively.
- *We devise a novel QoS forecasting method for the mobile edge environment.* A collaborative filtering based method [15] is employed to search for similar users in close distances and obtain their disguised QoS data for forecasting. This method locates the edge server being accessed by the queried user as the center and adopts the distances between the center and other servers as radii. The scope is continuously expanded accordingly to find similar users in other edge servers to obtain their historical QoS accessing records. Every time when similar users are searched, the up-to-date values of the service attributes will be acquired from the users to ensure the data freshness.
- *We particularly design a series of experiments for evaluating the proposed Edge-Laplace QoS approach based on pubic data sets.* The experiments validate the influence of edge region partition and noise updating on forecasting and the effectiveness of scope expansion to find similar users in the mobile edge environment. The experimental results also show that Edge-Laplace QoS achieves the goal of user privacy protection, while ensuring higher forecasting accuracy compared with other approaches.

The structure of the paper is organized as follows. Section 2 surveys state-of-the-art QoS forecasting approaches and some privacy protection studies, and also discusses their limitations. Section 3 gives the background used by our approach. Section 4 presents the details of our approach. Section 5 analyzes the experimental results based on several public data sets. Section 6 concludes the paper and plans our future work.

2 RELATED WORK

Existing QoS forecasting approaches can primarily be categorized into personalized recommendation based [16], [17], local neighborhood matrix factorization based [20], context-sensitive based [21], and privacy protection based [22], [25], [26], [27] according to the underlying theories.

Collaborative filtering Web service QoS forecasting approaches collect QoS information by means of user contribution mechanisms. They use hybrid methods to predict QoS information. Zheng et al. [17] designed a WSRec algorithm. WSRec calculates different information weights based on user similarity and service similarity. It establishes a novel hybrid recommendation method. The QoS forecasting method based on local neighborhood matrix factorization combines domain knowledge of artificial intelligence. It proposes a two-level selection mechanism. This method reduces the impact of data information missing to a certain extent. Nevertheless, it is only designed for traditional environments without considering the dynamic feature of mobile edge computing. Wei et al. [20] proposed a personalized QoS forecasting approach. This approach can identify a group of neighbors that have high correlations with the target user. It establishes an extension matrix based on geographic information. However, it only considers local heterogeneous resources. This is unsuitable for a distributed environment. The context-aware QoS forecasting method considers the complexity of service invocation. It simulates the interactions between users and service environments. Wu et al. [21] made full use of implicit and explicit context factors in QoS data. They proposed a context-dependent matrix factorization QoS forecasting method. The proposed approach only considers the spatial context factors. It cannot be applied to the forecasting environment considering time and other factors.

Privacy protection oriented QoS forecasting approaches have the function of users' raw data protection. Qi et al. [22] employed a local sensitive hash privacy protection based service recommendation approach to obtain a balance between service accuracy, privacy protection and high efficiency. This approach is only suitable for single-dimensional service quality. However, service quality is usually dynamic and multidimensional. Zhang et al. [25] adopted a scalable big data multi-dimensional anonymization approach for distributed systems. This approach can significantly improve scalability and time-efficiency of the multidimensional scheme over existing approaches. Nevertheless, this approach mainly targets big data mining platforms. It is unsuitable for user service quality forecasting and recommendation. Shahriar et al. [26] proposed a personalized Web service recommendation approach enhanced by location-aware privacy protection. A privacy protection protocol was proposed. It realizes privacy protection by encrypting QoS and hiding locations. However, the forecasting accuracy is unpromising. Liu et al. [27] devised a QoS forecasting method based on differential privacy. This method can disguise the original data by adding noises. It is only suitable for the general environment. It cannot be applied to the edge environment with short timeliness and high change rates. The edge server is deployed between a mobile client and a neighboring mobile server in the mobile edge environment. When the mobile web browser sends a request to the URL page, the edge server first intercepts it to analyze the user behavior to improve the QoS. Therefore, how to accurately predict QoS values before invoking services is an important issue in mobile edge service recommendation. Wang et al. [23] proposed a QoS forecasting service recommendation technique based on collaborative filtering in a mobile edge environment. This method directly uses user's historical data to forecast without considering the user's privacy.

Existing privacy protection studies also focus on security verification, assurance and certification [28], [29]. Lin et al. [30] developed a comprehensive mobile provable data possession scheme to determine the integrity and availability of outsourced data. This scheme bases on a hash tree structure and a boneh-lynn-shacham short signature scheme to support the dynamic and stateless outsourcing verification. Searchable encryption technologies emerge to address the problem of searching ciphertext in cloud servers. Sun et al. [31] proposed an attribute-based keyword search scheme with an efficient user revocation function. This scheme allows multiple data owners to independently encrypt their data and outsource their data to cloud servers. Data users can generate their own search functions without relying on third-party agencies. The locations of mobile users are dynamic in mobile edge computing. Handover authentication is an authentication transfer technology to address authentication of users with high mobility. He et al. [32] recently introduced a handover authentication protocol for mobile wireless networks. The protocol employs identity-based public key cryptography to meet the security and privacy requirements of the handover authentication. Current assurance techniques increasingly rely on model-based verification. However, these techniques fall short on provisioning sound solutions for continuous evaluation on the validity and correctness of their assessment. He et al. [33] presented a trustworthy cloud certification scheme based on continuous model verification. This scheme considers modeling time, execution probability, configuration constraints and attack flows based on service execution traces.

There is no privacy protection aware QoS prediction approach in the MEC environment at present, according to our literature survey.

3 PRELIMINARIES

3.1 Mobile Edge Computing

Mobile edge computing (MEC) is regarded as the key technology and architectural concept of transition to 5G [34]. MEC is promoting the convergence of cloud computing platforms and mobile networks in traditional centralized data centers. This is enabled by "sinking" services and functions originally located in cloud data centers to the edges of mobile networks, and providing computing, storage, network and communication resources at the edges of mobile networks [35]. Additionally, mobile network operators can open more network information and network congestion control functions to third-party developers through the MEC technology, and allow them to provide users with more applications and services. The architecture of edge computing is shown in Fig. 2.

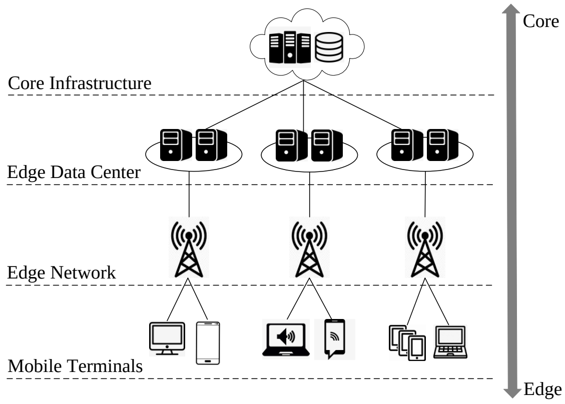


Fig. 2: Architecture of edge computing

3.2 Differential Privacy

Differential privacy is different from traditional cryptosystems in terms of strict attack modes. It gives a strict quantitative definition for privacy leakage. Users can obtain maximum privacy protection while ensuring data availability basing on the concept of differential privacy. The biggest advantage of this technique is that, the noise required for interference is independent on the original data, although the data is distorted after processing. A higher degree of privacy protection can be implemented by inserting a small amount of noises into the data [36]. The differential privacy method is still regarded as the most strict and robust privacy protection mode among many similar techniques, such as *k*-anonymization and *l*-diversity privacy protection. This results from its solid mathematical argumentation basis [24]. Laplace mechanism is a way of implementing differential privacy [37]. We employ Laplace mechanism in our approach to generate dynamic noises for the original QoS data in the edge environment to protect user data privacy.

3.2.1 Security Definition

If data sets *D1* and *D2* differ in at most one element, and the service set $S \subseteq$ range *(k)*, then a random function K defines ϵ - the differential privacy. It can be expressed as:

$$\frac{Pr[K(D_1 \in S)]}{Pr[K(D_2 \in S)]} \leq exp(\epsilon) \tag{1}$$

where *Pr[.]* indicates a probability space. Generally the privacy parameter ϵ is greater than 0. A higher ϵ yields a stronger privacy guarantee.

The implementation of differential privacy is featured with data randomness, e.g., Laplace noise. This is due to the fact that differential privacy is defined in probabilities.

3.2.2 Laplace Mechanism

Differential privacy was originally proposed by Dwork et al. [37]. It is achieved by adding random noise to a distribution function, such as Laplace distribution. A random variable satisfies Laplace distribution if it satisfies the probability density function distribution of the following equation:

$$f\ (x|\mu, b) = \frac{1}{2b} exp(-\frac{|x-\mu|}{b}) \tag{2}$$

where μ and b are respectively the positional parameter and the scale parameter. We make μ = 0 to simplify the function calculation. This distribution can then be viewed as a symmetric exponential distribution with standard deviation $\sqrt{2}b$. The distribution of the probability density function with different b values is shown in Fig. 3.

We use Laplace distribution to increase the noise. Let b = $\Delta f/\epsilon$, the noise can be expressed as:

$$Laplace(\Delta f/\epsilon) \tag{3}$$

where Δf is the global sensitivity that represents the extreme difference between attribute value column vectors [24], and ϵ is the privacy parameter.

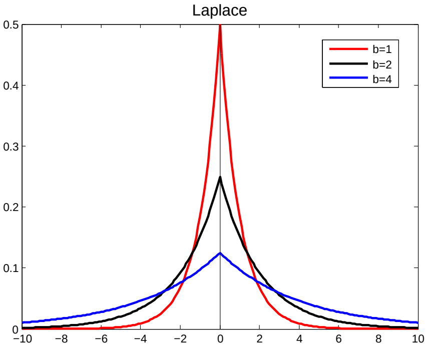


Fig. 3: Probability density function

3.3 Similarity Measure Method

Given a recommendation system consisting of m users and n services, the relationship between users and services is usually represented by a matrix of $m \times n$, in which each record $r_{m,n}$ represents a vector of QoS attribute values, such as *response time*, *throughput*, *failure rate*, etc.

Pearson correlation coefficient (PCC) [38], [39] is one of the most common similarity calculation methods in recommendation systems. It is easy to implement and has high possibility to obtain high precision. We use PCC in our approach for user similarity calculation based on the disguised data. PCC is often used to define the similarity between users u and v based on their invoked services in user-based collaborative filtering of Web services. The calculation formula[1] is:

$$Sim(u,v) = \frac{\sum_{s_i \in S}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{s_i \in S}(r_{u,i} - \bar{r}_u)^2 \sum_{s_i \in S}(r_{v,i} - \bar{r}_v)^2}} \tag{4}$$

where service set *S* is a collection of services invoked jointly by user u and user v, $r_{u,i}$ and $r_{v,i}$ respectively denote the QoS value of service *i* invoked by user u and user v, and $\bar{r}_u$ and $\bar{r}_v$ respectively represent the average value of service set *S* invoked by user u and user v. Service similarity *Sim(u,v)* is within the interval [0,1] in this definition. The larger the value, the greater the similarity [17].

PCC tends to overestimate the similarity between users with similar characteristics in the actual calculation process. This is because the base number of their invoked services is small and they have similar QoS history records on commonly invoked Web services [40]. In order to solve such problems, this research adopts the improved similarity weight method to reduce the impact of a few co-invoked services. An equation for different user similarity calculation is defined as follows to improve PCC:

$$Sim'(u,v) = \frac{2 \times |I_u \cap I_v|}{|I_u| + |I_v|} Sim(u,v) \tag{5}$$

where I_u and I_v separately represent the number of services invoked by user u and user v. A new similarity value is calculated according to the number of services respectively invoked by user u and user v and the number of same services invoked by user u and user v.

[1] https://www.spss-tutorials.com/pearson-correlation-coefficient

4 THE EDGE-LAPLACE QOS APPROACH

The workflow of Edge-Laplace QoS is outlined in Section 4.1. The three steps of Edge-Laplace QoS are introduced in details in Section 4.2, Section 4.3 and Section 4.4.

4.1 Overview of Edge-Laplace QoS

We propose a privacy-preservation oriented QoS forecasting method (Edge-Laplace QoS) in the mobile edge environment. Edge-Laplace QoS works towards the goals of privacy preservation and accurate and efficient edge QoS forecasting. The system workflow is shown in Fig. 4. It is mainly divided into three steps:

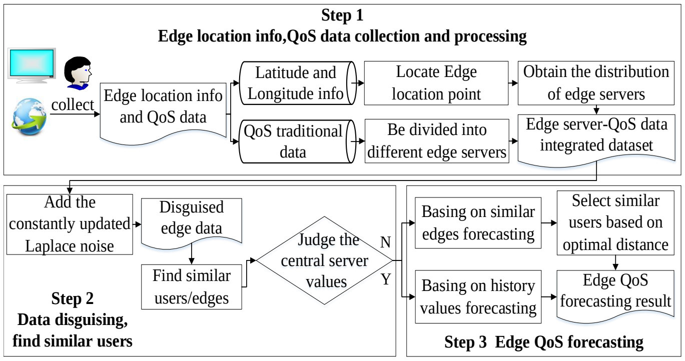

Step 1
Edge location info,QoS data collection and processing
collect
Edge location info and QoS data
Latitude and Longitude info
Locate Edge location point
Obtain the distribution of edge servers
QoS traditional data
Be divided into different edge servers
Edge server-QoS data integrated dataset
Add the constantly updated Laplace noise
Disguised edge data
Find similar users/edges
Judge the central server values
N
Y
Step 2
Data disguising, find similar users
Basing on similar edges forecasting
Select similar users based on optimal distance
Basing on history values forecasting
Edge QoS forecasting result
Step 3 Edge QoS forecasting

Fig. 4: Edge-Laplace QoS overview

1) *Edge location information and QoS data collection and processing.* First, the distribution of the edge servers is obtained according to the latitude and longitude values of their geographic locations. Next, the traditional data set is fused with the edge server data to form the user-service integrated data set in the edge environment. Here we use the scenario of *Bob* in Fig. 1 as an example. First, we obtain the locations of the two edge regions. Next, we fuse the QoS attribute values of the services accessed by users in the two regions (e.g., the QoS data of *Bob* watching the YouTube video) together with the edge server locations to form a user-service integrated edge data set.
2) *Data disguising and similar user searching.* The updated noise value is added into the original data to obtain the disguised edge data set after obtaining the edge data set in Step 1. Next, similar users are retrieved in the circled area centered at the edge server being accessed with the continually expanded radius. This step will be terminated once the prediction error reaches its minimum. The historical data of the users who previously accessed the central server is therefore retrieved. In *Bob*'s example, the updated noise value is added into the QoS data in edge region B. The servers in edge region B can search for other users' disguised historical QoS data with respect to the YouTube video provisioning, e.g., continuous diffusion searching within a radius of 100 m. *Bob* can only observe other users' disguised data in edge region B during this process.
3) *Edge QoS forecasting.* The forecasting mode is determined by whether or not the central server contains the historical data. If it does, the forecasting is based on the historical data of the central server using the collaborative filtering technique; otherwise the historical data of the similar users obtained in Step 2 will be used to predict QoS based on the collaborative filtering method. In *Bob*'s example, the forecasting will base on the historical data of Server S_2, if S_2 contains other users' historical QoS data with respect to the YouTube video provisioning; otherwise the forecasting will base on the result obtained in Step 2, namely other users' historical QoS data with respect to the YouTube video provisioning from servers within a radius of 100 m.

4.2 Data Collection and Preprocessing

Let us take the user *Bob* watching a YouTube video on a bus as an example. Edge server S_1 in edge region A and edge server S_2 in edge region B respectively log some quality data about this video service, e.g., resolution and response time. It is required to collect these data from these dispersed servers. In this paper, the complete edge geographical location information and QoS attribute data are first collected by employing the method in [17]. We download the open source data set based on the link provided in [17], followed by data preprocessing. The data preprocessing consists of the following steps.

4.2.1 Edge Region Division

Region division is implemented upon latitude and longitude values. It is mainly divided into three steps. **Step 1**: we target the latitude and longitude values of the geographic locations in the data set. **Step 2**: the non-repeated positions are selected as the locations of the edge servers. **Step 3**: the edge region is formed according to the density distribution of the edge servers.

The edge region partition is explained in terms of our experimental data set. First, the edge servers are positioned according to the latitude and longitude values of the geographic locations in the data set. For example, (-86.9162, 40.4249) and (-122.2536, 37.8668) are in North America, (9.1833,48.7667) and (-1.6743,48.112) are in Europe, and (114.1667,22.25) and (139.69,35.69) are in Asia. Next, we select the *North America* data set which contains the largest number of edge servers as the experimental data set. The *North America* data set contains 87 edge server locations formed by 174 sets of latitude and longitude values.

Finally, the edge regions are formed according to the density distribution of the 87 edge servers. The distribution of *North America* is shown in Fig. 5. The distribution of our another experimental data set – the *Shanghai* Telecom data set is shown in Fig. 6.

4.2.2 Traditional QoS Data Set Processing

We do not need to consider the impact of environmental factors on data set for QoS prediction in traditional environments. However, the existing QoS data set is inapplicable for this research. This is because the edge environment has the characteristics of fast response and real-time dynamics. We use the following method to integrate the QoS attribute values with the edge servers. First, the data of each QoS attribute is organized in the form of a two-dimensional matrix, where the rows correspond to services and the

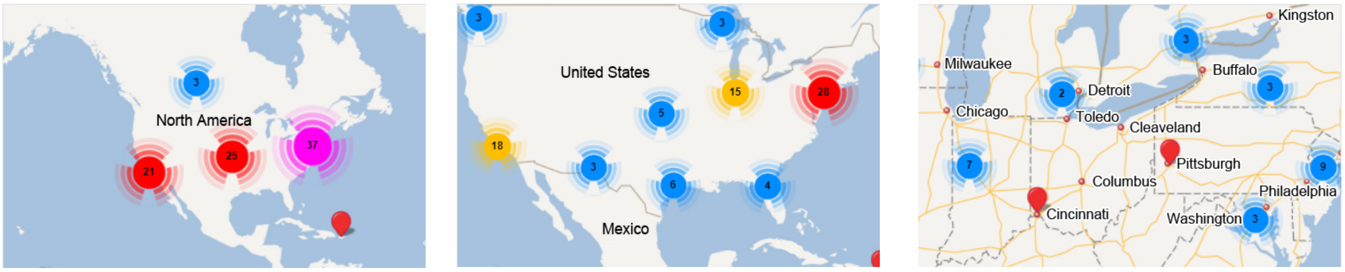


Fig. 5: Chart of *North America* area distribution

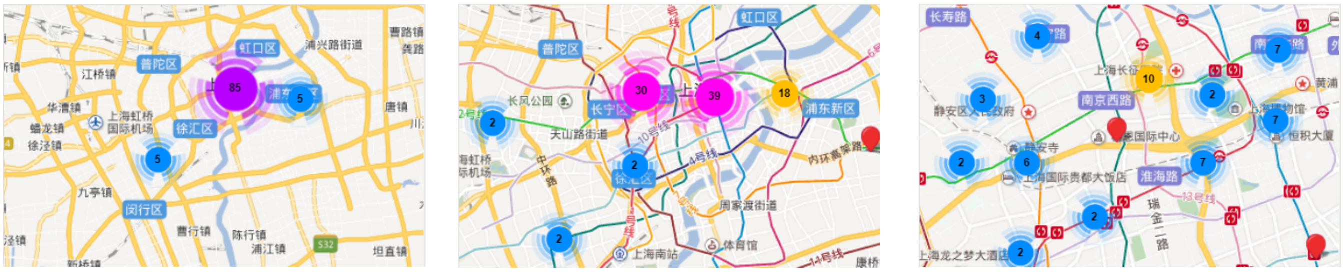


Fig. 6: Chart of *Shanghai* area distribution

columns correspond to service users. Each column refers to the QoS attribute values of the service set invoked by a single user, which is treated as a column vector. Second, this matrix is divided into s sub-matrices which might be single-column or multi-column matrices according to users' actual service accessing records in each edge server, where s is the number of edge servers. This sub-matrix setting can be used to locate the exact average QoS value of a specific service that is provisioned over a specific edge server and accessed by a specific user during a specific visit in the past. The following shows an example of the matrix division, where the data is obtained from our experimental data set. R is a matrix of accessing records of response time for all the edge servers. We identify t edge servers in the data set.

$$R = \begin{bmatrix} 0.4270 & 2.4930 & 1.0230 & \cdots & 0.4280 \\ 0.6520 & 0.6880 & 0.5380 & \cdots & 0.6120 \\ 0.6420 & 1.0530 & 0.8190 & \cdots & 0.6280 \\ 0.3690 & 0.3970 & 0.5580 & \cdots & 0.4730 \\ 0.1970 & 0.1920 & 0.2620 & \cdots & 0.2340 \\ 0.1830 & 0.2060 & 0.2650 & \cdots & 0.2330 \end{bmatrix}$$

The divided edge sub-matrix R' is shown below after QoS attribute values are merged with t edge servers. The number of columns in each sub-matrix represents the number of user accessing records in each edge server.

$$R'_1 = \begin{bmatrix} 0.4270 \\ 0.6520 \\ 0.6420 \\ 0.3690 \\ 0.1970 \\ 0.1830 \end{bmatrix} R'_2 = \begin{bmatrix} 2.4930 & 1.0230 \\ 0.6880 & 0.5380 \\ 1.0530 & 0.8190 \\ 0.3970 & 0.5580 \\ 0.1920 & 0.2620 \\ 0.2060 & 0.2650 \end{bmatrix} \cdots R'_t = \begin{bmatrix} 0.4280 \\ 0.6120 \\ 0.6280 \\ 0.4730 \\ 0.2340 \\ 0.2330 \end{bmatrix}$$

4.2.3 Integrated Edge Service-QoS Data Set

The edge servers with geographic locations and the edge sub-service set are fused to form an integrated edge service-QoS data set after the edge region partitioning. The user set in an edge server is $U = \{u_1, u_2, ..., u_n\}$ and the accessed edge service set is $S = \{s_1, s_2, ..., s_k\}$, in which n and k are all positive integers. The QoS attributes of the services in our research mainly include *response time* and *throughput*. The edge service data set thus consists of response time-user matrices and throughput-user matrices, the sizes of which are $k * n$. The processed user service set has more precise edge features, closer attribute associations, and more accurate and reliable edge forecasting.

4.3 Noise Adding and Similar Users Searching

We collect the user *Bob*'s QoS data via Section 4.2. In this section, we add noises to the data for privacy protection. We then search for similar users to forecast QoS values. The forecasting is applicable when *Bob* just moves to a new edge region where there is no his previous accessing record.

Laplace mechanism in differential privacy focuses on adding random noise values to disguise the original data by training the privacy parameter ϵ (equation (3)). The random number generated by the column vectors (defined in Section 4.2.2) is invariable, when the noise value is added to the columns in the traditional differential privacy method. The fixed random number in the mobile edge environment, however, cannot satisfy the dynamic characteristic of QoS. Our improved differential privacy method takes the QoS data set in the mobile edge environment as input and continuously updates the random numbers generated by the column vectors in the data set. The dynamic random number is able to enhance privacy protection by adapting to the mobile edge environment. In the training process, we continuously adjust the privacy parameter to achieve its optimal value, when the prediction error is minimum. The disguised data is closest to the real value, when the minimum prediction error is achieved. We also use dynamic Laplace noise to further improve the integrated edge service-QoS data set, and achieve the goal of accurate and efficient QoS forecasting in the mobile edge environment.

The input of Laplace mechanism is the original data set $g(x)$, and the output is X. b in the Laplace distribution function is $\Delta f/\epsilon$, and its formula is:

$$X = g(x) + Laplace(\Delta f/\epsilon) \tag{6}$$

where Δf is the extreme difference between two data set vectors. Its definition in the integrated edge service-QoS data set is:

$$\Delta f = max(r_{u,i} - r_{u,j}) \tag{7}$$

where $r_{u,i}$ and $r_{u,j}$ respectively represent the QoS values of service i and service j invoked by user u. In simpleness, ϵ-differential privacy of user u is achieved by the following equation:

$$R_{u,i} = r_{u,i} + Laplace(\Delta f/\epsilon) \tag{8}$$

In the process of adding noise, the random number x in the Laplace distribution function is updated continuously to update the noise value $Laplace(\Delta f/\epsilon)$.

Let us take the edge QoS attribute value rt in the experimental data as an example. We record the noise-free original QoS data when a user accesses a service on an edge server as $rt = [rt_1, rt_2, \cdots, rt_k]$, where rt_k is a collection of the single QoS attribute values for a service k. $R_k = max(rt_k) - min(rt_k)$ represents the range of a single service QoS vector. The attribute values of response time $RT = [RT_1, RT_2, \cdots, RT_k]$ are formed by adding the noise value to the original data to achieve data privacy protection, where $RT_k = zscore(rt_k) + Laplace(R_k/\epsilon)$.

The raw data is disguised by the improved differential privacy algorithm during the first stage. Then we use the disguised data to find similar users in addition to achieving the goal of user privacy protection. Liu et al. [27] proposed a method for QoS forecasting by adding Laplace noise to raw data in the traditional environment. In this paper, we heuristically optimize this QoS forecasting approach to make it adapt to the edge environment.

The data in the edge environment has the characteristics of real-time and quick updating. It is necessary for a user who is accessing an edge server to consider whether or not this edge server environment is similar to his/her own edge server environment during the process of searching for similar users. Usually the edge environment in the same region is relatively similar while the edge environment in different regions is relatively different. Therefore, distance is a decisive factor in the process of similar user searching. Users are highly mobile in comparison to edge servers. The former may switch among different servers to invoke services. Therefore, we set the edge server that the user is accessing as the center and the nearest integer distance between the center and the closest edge server as the initial radius. We incrementally increase the searching distance based on the initial radius (i.e. initial radius * n $(n = 1, 2...)$) to find similar users in other servers. The edge server accessed by a user U_n is denoted as b with latitude and longitude values (α_1, β_1), and the edge server accessed by other users to be searched is b_i with latitude and longitude values (α_2, β_2). We can use the following typical Geographical Distance algorithm [41], [42] to calculate the distance between edge servers.

$$dis(b, b_i) = 2 * 6371 * asin(sqrt(hav(\theta))) \tag{9}$$

with

$$hav(\theta) = sin^2(\frac{\beta_1 - \beta_2}{2}) + cos\beta_1 cos\beta_2 sin^2(\frac{\alpha_1 - \alpha_2}{2}) \tag{10}$$

The number of similar users may be further improved by increasing the distance after getting the distance between the user U_n and other edge servers.

We first judge whether there are historical user accessing records in the edge server b in the process of finding similar users. The details of the user similarity measurement are described in Step 2 of Fig. 4. If there are similar users, it means that there is historical data in the edge server b, and the forecasting will be realized by the collaborative filtering method based on historical values. Otherwise, it means that there is no historical data, so we need to continuously find similar servers by increasing the distance. The process of searching range expansion will stop, when the prediction error first reaches to its minimum.

4.4 Edge QoS Forecasting

For the user *Bob*, when he accesses an edge server in a new edge region, the video service quality forecasting is realized based on 1) the historical QoS values of similar users in this edge server if these historical values exist in the server, or 2) the historical QoS values of similar users in other edge servers within a certain range.

Users need to perform $z - score$ standardization processing on QoS values before data disguising in order to eliminate the difference between user data and guarantee forecasting accuracy. The calculation formula of $z - score$ is as follows:

$$q_{u,i} = (r_{u,i} - \overline{r}_u)/\omega_u \tag{11}$$

where $\overline{r}_u$ and ω_u respectively denote the mean value and the standard deviation of QoS vector r_u. The standardized QoS data has zero mean and unit variance. The user data is disguised based on the standardized data, which can be expressed as:

$$Q_{u,i} = q_{u,i} + Laplace(\Delta f/\epsilon) \tag{12}$$

The discrete degree of noise addition is determined by continuously adjusting the weights of privacy parameters ϵ. ϵ impacts the Laplace distribution function (equation (2)) in the process of noise addition. It is known that $b = \Delta f/\epsilon$, in which b is inversely proportional to ϵ, and $f(x)$ is inversely proportional to b, so $f(x)$ is proportional to ϵ. That is to say, the smaller ϵ is, the smaller the added noise value is, the smaller the privacy constraint is, the less the limitation of the original data is, and the more accurate the forecasting result is.

Users can only observe the disguised data but not the real data in the process of similarity calculation after data disguising, where $Q_{u,i}$ is used to calculate the similarity. According to $z - score$ standardization, standard deviation $\omega_u = \sqrt{\sum_{s_i \in S}(r_{u,i} - \overline{r}_u)^2/I_u}$ (I_u is the number of service i invoked by user u) is substituted into equation (4) and further simplified to equation (13):

$$Sim(u, v) = \frac{\sum_{s_i \in S}(r_{u,i} - \overline{r}_u)(r_{v,i} - \overline{r}_v)}{\omega_u \omega_v \sqrt{I_u I_v}} \tag{13}$$

Similarly, substituting $q_{u,i} = (r_{u,i} - \overline{r}_u)/\omega_u$ in z standardization into the above formula can obtain:

$$Sim(u,v) = \frac{\sum_{s_i \in S} q_{u,i} q_{v,i}}{\sqrt{I_u I_v}} \tag{14}$$

That is, the calculation of $Similarity$ is simplified. To describe the process more clearly, we denote two raw QoS data vectors as $a = (a_1, a_2, \cdots, a_n)$ and $b = (b_1, b_2, \cdots, b_n)$ respectively. The corresponding disguised QoS data vectors are $A = (A_1, A_2, \cdots, A_n)$ and $B = (B_1, B_2, \cdots, B_n)$. The scalar product between two vectors remains the same, although the data are disguised. It is proved as follows:

$$\begin{aligned} AB &= \sum_{i=1}^{n} A_i B_i \\ &= \sum_{i=1}^{n} (a_i + Laplace(\Delta f_a/\epsilon_a))(b_i + Laplace(\Delta f_b/\epsilon_b)) \\ &= \sum_{i=1}^{n} (a_i b_i + Laplace(\Delta f_a/\epsilon_a) Laplace(\Delta f_b/\epsilon_b)) \\ &+ \sum_{i=1}^{n} (a_i Laplace(\Delta f_b/\epsilon_b) + b_i Laplace(\Delta f_a/\epsilon_a)) \end{aligned} \tag{15}$$

where a_i and $Laplace(\Delta f_b/\epsilon_b)$ are two independent vectors, and $Laplace(\Delta f_b/\epsilon_b)$ satisfies the symmetric exponential distribution of the Laplace probability density function, when $\mu = 0$. These make $\sum a_i Laplace(\Delta f_b/\epsilon_b) \approx 0$, $\sum b_i Laplace(\Delta f_a/\epsilon_a) \approx 0$, $\sum Laplace(\Delta f_a/\epsilon_a) Laplace(\Delta f_b/\epsilon_b) \approx 0$. Finally, It can be proved that $AB = \sum a_i b_i = ab$. The following equation can be obtained by substituting it into Equation (14):

$$Sim(u,v) \approx \frac{\sum_{s_i \in S} Q_{u,i} Q_{v,i}}{\sqrt{I_u I_v}} \tag{16}$$

The similarity calculation can still be performed with the disguised data while the real data is hidden. $Sim(u,v)$ is between the intervals [0,1]. The larger the value, the higher the similarity between two users.

Algorithm 1 Edge forecasting method by Laplace

```
Require: User $u$ accesses Edge Server $b$, Nearby edge server $b_i$, Service $s$ invoked jointly, Disguised QoS Data $ds$;
Ensure: Edge QoS forecasting value
$v$ as other accessing user in edge server $b$;
$v_i$ as accessing user in edge server $b_i$;
Calculating distance of $b$ and $b_i$;
Judging the number of $v$ when $dis == 0$;
if number($v$)>0 then
    Similar userset.add(user($v$)), $v++$;
    QoS value forecasting;
else
    find other similar user $v_i$ in $b_i$;
    $dis++$;
    $v_i++$;
    The search process will stop when the minimum prediction error appears;
    Find similar userset $v_i$ via $ds$ of $s$ within $dis$ $\lambda$;
    Best distance $\lambda$ recommendation;
    Get similar userset $v_i$;
    QoS value forecasting;
end if
```

Based on Algorithm 1 and equation (15), the QoS value of service i that is invoked by user u can be predicted directly by the following formula:

$$q'_{u,i} = \overline{Q}_u + \frac{\sum_{v \in Sim_u} Sim(u,v)(Q_{v,i} - \overline{Q}_v)}{\sum_{v \in Sim_u} Sim(u,v)} \tag{17}$$

where $\overline{Q}_u$ and $\overline{Q}_v$ are the average disguised QoS values of the service sets respectively invoked by user u and v, and $Q_{v,i}$ represents the disguised value of service i invoked by user v. After the QoS forecasting, the anti-standardization process is performed to get the value before the *z-score* standardization according to the *mean(u)* and standard deviation *std(u)* of the original data of user u. This makes the Edge-Laplace QoS forecasting approach more accurate.

5 EVALUATION

In this section, a set of dedicated experiments are performed to explore the following basic research questions:

- RQ1: What is the best privacy parameter value that can effectively protect user privacy while guaranteeing prediction accuracy for Edge-Laplace QoS?
- RQ2: What is the effect of searching distance on edge prediction in Edge-Laplace QoS?
- RQ3: How does Top-k recommendation affect the efficiency of edge prediction?
- RQ4: Is Edge-Laplace QoS more efficient than the state-of-the-art traditional forecasting approaches?

5.1 Data Set

We use two data sets to ensure the completeness of the experiment and the effectiveness of the method. The first part of the data [2] is the real-world QoS evaluation results of 5825 Web services used by 339 users and the geographic location information of the service users derived from the open source data set. The above Web service QoS data mainly includes two QoS attributes – *response time (RT)* and *throughput (TP)*. We use the historical data of these two attributes to predict the future attribute values. In this data set, each set of user information contains IP address, geographic location, etc. We use Baidu Map [3] to locate the edge regions based on the geographic location information, and select the *North America* region that contains the largest amount of data as an experimental data set. The edge server information of different countries in *North America* data set is shown in Table 1.

TABLE 1: *North America* dataset information

Server ID	IP Address	Country	Latitude	Longitude
5	128.31.1.13	United States	42.3646	-71.1028
16	142.104.21.241	Canada	48.4202	-123.3671
57	136.145.115.194	Puerto Rico	18.25	-66.5

The second part of the data [4] [23], [43] mainly comes from more than 7.2 million records generated by 9481 mobile phones that access 3233 base stations in the *Shanghai* Telecom System. The base station location information

[2] https://github.com/wsdream/wsdream-dataset
[3] http://map.yanue.net/
[4] http://sguangwang.com/TelecomDataset.html

mainly includes two attributes: *latitude* and *longitude*, which are used to locate the edge servers. We randomly select the records of 339 mobile users with unique IDs to perform the experiment. The edge server information of different districts in the *Shanghai* data set is shown in Table 2.

TABLE 2: *Shanghai* dataset information

Server ID	**Historical access Times**	**District**	**Latitude**	**Longitude**
3	0	Huangpu	31.241131	121.487911
22	14	Pudong	31.240874	121.518086
68	4	Hongkou	31.256232	121.498254

Our integrated edge service-QoS data set contains two edge regions, including 87 edge servers in *North America* and 95 edge servers in *Shanghai*. Their division and distribution are described and illustrated in Section 4.2.1. The numbers of edge servers with historical data in *North America* and *Shanghai* are 60 and 67. There are 174 and 339 sets of QoS accessing records (*RT* and *TP*) in *North America* and *Shanghai*.

5.2 Metrics

The experiment intuitively compares the forecasting performance between several candidate approaches based on *MAE* (Mean Absolute Error) and *RMSE* (Root Mean Square Error):

$$MAE = \frac{\sum \left|q_{u,i} - q'_{u,i}\right|}{N} \tag{18}$$

where $q_{u,i}$ is the true value of service i, $q'_{u,i}$ is the forecasting value of service i, and N is the number of predicted services. *MAE* can better reflect the actual forecasting error and accuracy.

$$RMSE = \sqrt{\frac{\sum (q_{u,i} - q'_{u,i})^2}{N}} \tag{19}$$

RMSE can indicate the relative error rates and reflect the stability of forecasting.

5.3 Experimental Procedure

It is expected that the experiments can prove that the proposed edge QoS forecasting approach, featured with 1) adding constantly updated Laplace noise to the edge data set and 2) using the distance method to find similar users, can effectively improve the forecasting efficiency and protect the user privacy. The experiments are implemented in a computer system with Intel(R) Core(TM) i5-8250U CPU @1.60GHz, 8.00GB RAM, Windows 10, and matlab 7.13 .

We design the following comparative approaches after preprocessing the *North America* regional and *Shanghai* regional edge data sets.

- *Trad-LUMEAN*: A privacy protection oriented QoS mean value forecasting approach based on Laplace mechanism. It employs the traditional cloud environment data set. The centralized data processing is implemented on the data set, i.e., similar users searching is not affected by distance, regions and other environmental factors.
- *Trad-LUPCC*: A Laplace privacy protection oriented QoS forecasting approach based on user similarity. It employs the traditional cloud environment data set.
- *Edge-Laplace QoS*: A privacy protection oriented QoS forecasting approach with Laplace noise. It bases on the mobile edge environment data set.
- *Edge-NonLaplace QoS*: A QoS forecasting approach based on collaborative filtering. It bases on the mobile edge environment data set.

We adjust the privacy parameter and dynamically update the random number to add noise value to the original data. We then select the parameter with the minimum *RMSE* value to carry on the follow-up experiment. Next, we analyze the influence of the distance between edge servers on the forecasting results and attempt to discover the best distance and the optimal Top-k similar users, on which the forecasting can obtain higher precision. This method is demonstrated to have better forecasting accuracy and privacy protection effect compared with the traditional collaborative filtering forecasting approaches. The experimental steps are briefly described as follows:

(1) We add the continuously updated Laplace noise to the edge data set to achieve a better trade-off between forecasting accuracy and privacy protection effect. The Laplace noise is updated by increasing the privacy parameter from 0.5 to 5. The increment is 0.5 each time.

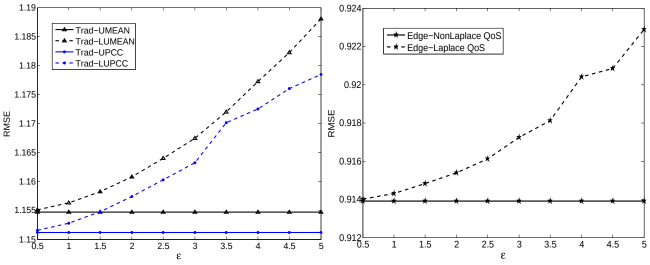
(2) We retrieve similar users based on the service attribute values that are disguised by the noise addition. We treat the edge server accessed by the user as the center of the circle and continuously increase the radius to find similar users in the surrounding edge servers. The *North America* users use a kilometer as the measuring unit and the *Shanghai* users use a meter as the measuring unit according to the scale of the experimental data. The best search distance is obtained according to the lowest error value (i.e., MAE or RMSE) for the predicted value. The experimental servers are randomly selected among three edge servers – with null accessing records, medium accessing records or maximum accessing records.

(3) We employ the collaborative filtering method to predict the QoS attribute values. We first determine whether there are other recorded users in the edge server being accessed by the user, i.e., the central server. If the result is positive, the QoS attribute values are predicted based on the historical QoS attribute values; otherwise the QoS attribute values are predicted based on the historical data of similar users in the edge servers within the best distance obtained by Step 2 of Fig. 4. The number of Top-k similar users in the edge environment is recommended for the collaborative filtering based prediction after the forecasting is completed.

5.4 Experimental Results

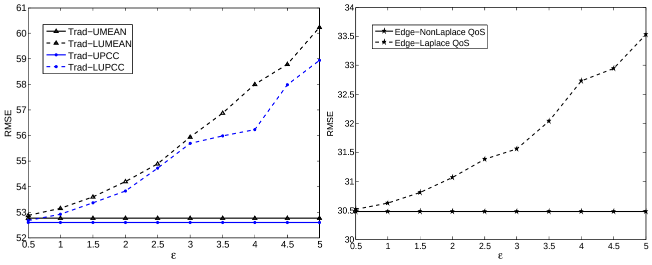
5.4.1 Impact of Privacy Parameter

The *RT* and *TP* forecasting results on the traditional data set and the edge data set are respectively shown in Fig. 7 and Fig. 8. The results show that the RMSE values in the edge environment are significantly smaller than those in the traditional environment. In addition, the trend of the error curve shows that, the dispersion degree of the noise value becomes larger and the forecasting error rises, when the privacy parameter ϵ increases. In summary, the final forecasting result is more accurate when the privacy parameter value is smaller, where the smaller value indicates a looser privacy constraint and less restriction on the original data. Accordingly the data is disguised with a privacy parameter of 0.5 in the following experiment.



(a) (b)

Fig. 7: Forecasting performance of RT on RMSE: (a) in traditional environment, (b) in edge environment.



(a) (b)

Fig. 8: Forecasting performance of TP on RMSE: (a) in traditional environment, (b) in edge environment.

5.4.2 Impact of Searching Distance

Our next experiment is to assess how distance between two edge servers (the central and the surrounded) impacts the forecasting accuracy. Two groups of three edge servers with different properties (i.e., null, medium and maximum accessing records) are randomly picked respectively from the *North America* and *Shanghai* data sets to perform the experiments. Taking *TP* data as an example, the experiment of MAE and RMSE with varied distance is first carried out in the *North America* data set.

Fig. 9(a) and 9(b) show variations of the forecasting error values with the increasing distance for searching the similar users for the user that accesses Edge Server 3 in the *North America* data set. In the *North America* data set, there is no historical accessing record in Edge Server 3. It can be seen that the error values both reduce when the distance is increased from 20 km to 40 km. They reach a reliably low state in 40 km. Therefore, the best distance for similar user searching is 40 km when accessing Edge Server 3. Table 3 shows the distribution of similar edge servers and users in different distances from Edge Server 3.

TABLE 3: Distribution of similar servers and users for *North America* Edge Server 3

Distance / Number	20km	30km	40km	50km	60km
Similar edge server	2	4	8	9	10
Similar user	4	7	14	16	17

Edge Server 30 locally contains two similar historical user accessing records, which is close to the average storage per server. The error values of Fig. 9(c) and 9(d) generally keep on rising. The forecasting accuracy reaches the highest near 10 km. There is no other edge server within 10 km. Therefore, Edge Server 30 obtains the best forecasting result based on its own historical data. Edge Server 70 has the largest number (seven) of historical user accessing records in the data set. From Fig. 9(e) and 9(f), it can be seen that the variations of the error values are relatively smaller when the distance is within 45 km. Both MAE and RMSE obtain the minimum values when the distance is 15 km. There is only Edge Server 70 itself within 15 km according to the statistics.

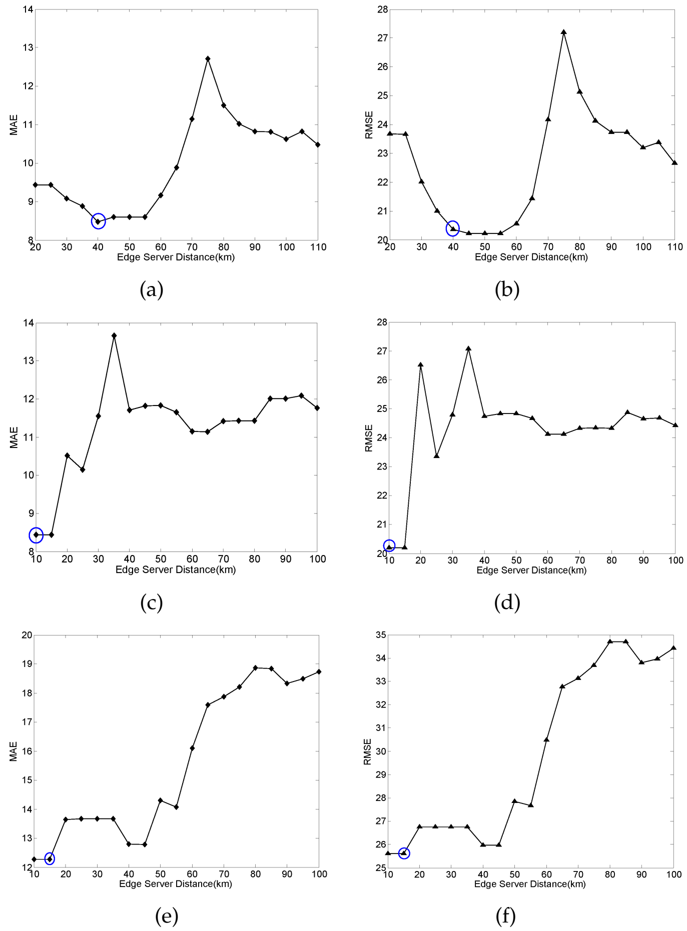
Similarly, three edge servers are randomly picked from the *Shanghai* data set. The experimental results are shown in Fig. 10. Among the three servers, Edge Server 3 stores zero historical user accessing record, Edge Server 36 contains four historical user records, close to the average in the data set, and Edge Server 16 has fourteen user records, which is the largest. From Fig. 10(a) and 10(b), it can be seen that the optimal distance for Edge Server 3 to find similar users is 1300 m. Table 4 shows that there are twenty-three similar edge servers and seventy-four similar users within 1300 m.

TABLE 4: Distribution of similar servers and users for *Shanghai* Edge Server 3

Distance / Number	900m	1100m	1300m	1500m	1700m
Similar edge server	13	19	23	29	30
Similar user	41	62	74	101	102

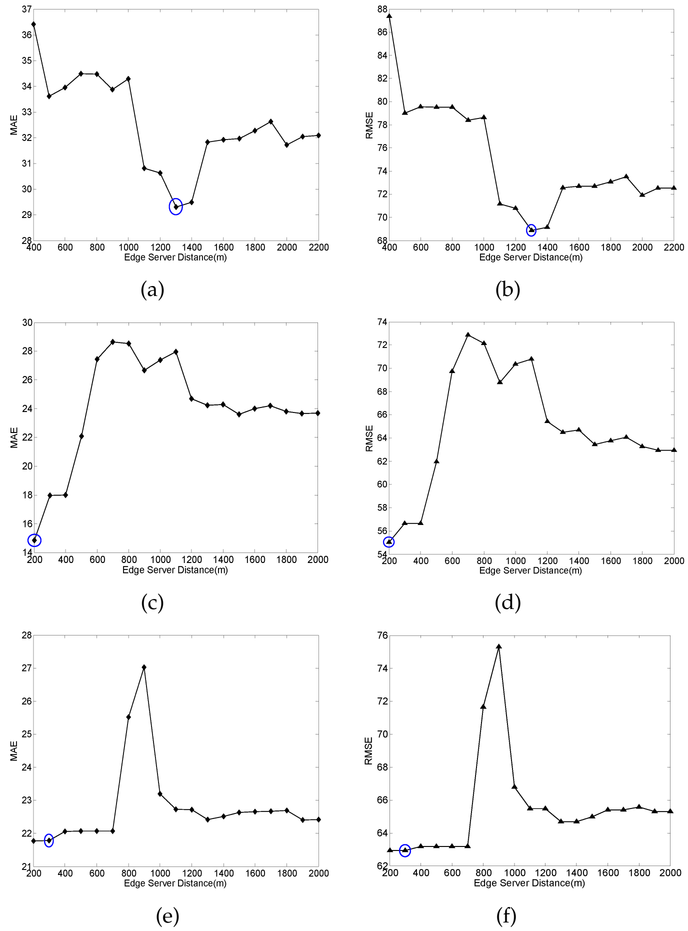
Fig. 10(c) and 10(d) show that the forecasting error values are smallest within the initial distance (200 m). It can be concluded Edge Server 36 achieves the best forecasting result based on its own historical data, where there are four similar users. Fig. 10(e) and 10(f) show that Edge Server 16 obtains both the minimum error values in 300 m. There is only itself within 300 m according to the statistics.

These two groups of experiments show that the privacy-preserving forecasting based on the historical data of the queried edge server can achieve the best forecasting accuracy, when there are historical records in the server; the best result relies on the geographic distribution of the similar edge servers and users, when there is no historical record in the queried server.



(a) (b) (c) (d) (e) (f)

Fig. 9: Forecasting performance of the *North America* Edge Servers on MAE and RMSE with increasing distance: Forecasting MAE of (a) Edge Server 3, (c) Edge Server 30, and (e) Edge Server 70, and Forecasting RMSE of (b) Edge Server 3, (d) Edge Server 30, and (f) Edge Server 70.



(a) (b) (c) (d) (e) (f)

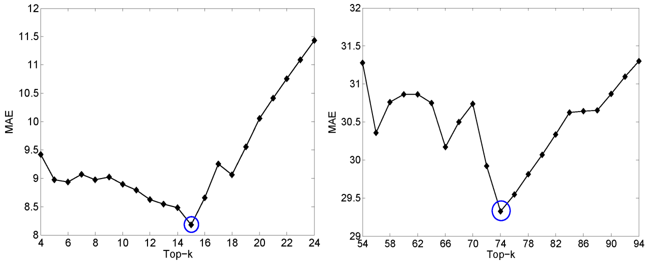
Fig. 10: Forecasting performance of the *Shanghai* Edge Servers on MAE and RMSE with increasing distance: Forecasting MAE of (a) Edge Server 3, (c) Edge Server 36, and (e) Edge Server 16, and Forecasting RMSE of (b) Edge Server 3, (d) Edge Server 36, and (f) Edge Server 16.

5.4.3 Impact of Top-k Recommendation

The experiments on *North America* Edge Server 3 and *Shanghai* Edge Server 3 prove that the appropriate *Top-k* similar user recommendation makes the forecasting more accurate when accessing the edge servers without historical data. We conduct experiments to assess how the number of similar users influence the QoS forecasting accuracy of the above two edge servers, given the optimal distances for similar user searching obtained in the above experiments. Tables 3 and 4 list the number of similar edge servers and users of the two edge servers in certain distances. 14 and 74 are respectively the number of similar users in the two edge servers' optimal searching distances obtained in the previous section. We conduct the following experiment to find the exact numbers of similar users that minimize the prediction error. We take 14 and 74 as the midpoints, and increase and decrease the number of similar users from the midpoints according to the degree of user density. The recommended *Top-k* in the *North America* experiment is in the range [4,24], and that in the *Shanghai* experiment is in the range [54,94]. The error values varied with the growing Top-k are shown in Fig. 11. It can clearly be observed that the optimal *Top-k* for the *North America* experiment is 15 and for the *Shanghai* experiment is 74.

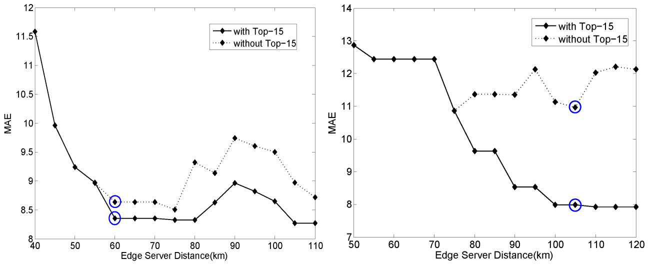
The following experiments are based on these obtained optimal *Top-k* values to verify if the *Top-k* recommendation can improve prediction accuracy. Two edge servers without historical accessing records are randomly picked from each of the two data sets, which are Edge Server 13 and 19 in *North America* and Edge Server 59 and 60 in *Shanghai*.

As shown in Fig. 12, the abscissa is the distance between the accessed edge server and the surrounding edge servers, and the ordinate is the MAE of the prediction result of the QoS attribute *TP*. The *Top-k* user recommendation method refers to the forecasting based on k users with the highest similarity among surrounding edge servers, while the non-user recommendation is based on all users in the surrounding edge servers. If the number of similar users does not reach the k-value in some search distances, the *Top-k* method will select as many users as possible. Obviously, the number of similar users reaches 15 in the search distance of 55 km when accessing *North America* Edge Server 13. The error value tends to be stable near 60 km and the error value with Top-15 method is significantly smaller than that without Top-15. The number of similar users is 15 in the search distance of 75 km when accessing *North America* Edge Server 19. The best forecasting performance is achieved in the search distance of 105 km.



(a) (b)

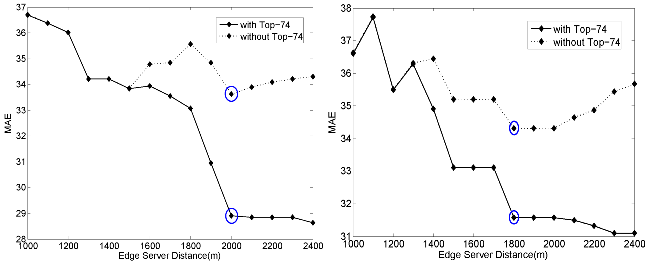
Fig. 11: Forecast for the number of Top-k on MAE: (a) access *North America* edge server, (b) access *Shanghai* edge server.



(a) (b)

Fig. 12: Effect of with Top-15 for different *North America* edge servers: (a) Edge Server 13, (b) Edge Server 19.

The distance-aware forecasting error values of *Shanghai* Edge Servers 59 and 60 are shown in Fig. 13. The forecasting error values with Top-74 are significantly smaller than those without Top-74. Hence, it can be concluded that *Top-k* similar user recommendation is an important factor to achieve high forecasting accuracy in the privacy-based QoS forecasting approach in the edge environment.



(a) (b)

Fig. 13: Effect of with Top-74 for different *Shanghai* edge servers: (a) Edge Server 59, (b) Edge Server 60.

5.4.4 Comparison Experiment

We compare the performance among the Laplace-based QoS average prediction approach in the traditional environment (Trad-LUMEAN), the user similarity-based Laplace QoS forecasting approach in the traditional environment (Trad-LUPCC), the collaborative filtering-based QoS forecasting approach in the mobile edge environment (Edge-NonLaplace QoS), and the privacy protection oriented QoS forecasting approach based on Laplace distribution in the mobile edge environment (Edge-Laplace QoS). We respectively select 10%, 20% and 50% (i.e., the matrix density) of the data from *Shanghai* and *North America* data sets as training data. We use the remainders as testing data. The training data is used to compute the similarity between two users via their jointly invoked services. The testing data is used to verify the prediction accuracy of our approach. The results are shown in Table 5 and Table 6.

Table 5 shows the variations on MAE and RMSE of the RT data with the increasing matrix density when accessing the *Shanghai* edge servers. It can clearly be seen that the forecasting error value is smaller in the edge environment than that in traditional environments. Therefore, it is more accurate to invoke services in the edge environment. We find that the similarity between edge servers becomes stable with the increasing matrix density by observing the horizontal change of the table. The error value of the proposed Edge-Laplace QoS forecasting approach steadily decreases and is slightly larger than that of the Edge-NonLaplace approach, when the matrix density increases from 10% to 50%.

The approach is applied to forecast the TP data in the *North America* edge servers to further verify the generality of the approach. Table 6 shows the error statistics of the forecasting results on different matrix densities. It again indicates that the forecasting effectiveness in the edge environment is obviously better than that in the traditional environment with the increase of the matrix density. In addition, the error value of the proposed Edge-Laplace QoS approach steadily decreases. The error value is close to the error value of the Edge-NonLaplace QoS approach.

6 CONCLUSIONS AND FUTURE WORK

Existing QoS forecasting approaches cannot cater for the demand in mobile edge computing on time sensitivity, high user mobility and information leakage prevention. We propose a novel privacy-preserving QoS forecasting approach for the edge environment named Edge-Laplace QoS.

In the future, first, we plan to investigate the changing trend of QoS attribute values in the edge service-user data sets for more accurate prediction. Second, the current approach only tunes the privacy parameter in the Laplace mechanism. Actually some other parameters of the Laplace distribution function can also be optimized.

ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China under Grant No. 61572171 and No. 61761136003, the Natural Science Foundation of Jiangsu Province under Grant No. BK20191297, and the Fundamental Research Funds for the Central Universities under Grant No. 2019B15414.

This work was also made possible by NPRP9-224-1-049 grant from the Qatar National Research Fund (a member of Qatar Foundation) and DP160100149 grant from Australian

TABLE 5: Accuracy comparison of RT data forecasting results

	Methods	Matrix Density=10%		Matrix Density=20%		Matrix Density=50%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
Response Time	Trad-LUMEAN	1.5115	2.3100	1.3326	2.1317	1.2142	1.9147
	Trad-LUPCC	1.0463	1.8484	0.9076	1.6614	0.7384	1.3323
	Edge-NonLaplace QoS	0.5008	0.8364	0.4459	0.7461	0.3286	0.6871
	Edge-Laplace QoS	**0.6565**	**0.9952**	**0.5092**	**0.7654**	**0.4003**	**0.7076**

TABLE 6: Accuracy comparison of TP data forecasting results

	Methods	Matrix Density=10%		Matrix Density=20%		Matrix Density=50%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
Throughput	Trad-LUMEAN	11.2937	23.6692	9.0800	18.5116	7.0253	17.5147
	Trad-LUPCC	10.3923	15.4455	8.8030	16.5170	6.4092	14.6721
	Edge-NonLaplace QoS	7.7251	14.8957	6.3688	12.8500	5.1731	11.2815
	Edge-Laplace QoS	**7.8009**	**14.9343**	**6.7608**	**12.8904**	**5.8509**	**11.5083**

Research Council. The statements made herein are solely the responsibility of the authors.

References

[1] M. La Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices," *IEEE communications surveys and tutorials*, vol. 15, no. 1, pp. 446–471, 2013.

[2] W. Song, F. Chen, H.-A. Jacobsen, X. Xia, C. Ye, and X. Ma, "Scientific workflow mining in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2979–2992, 2017.

[3] S. M. M. Fattah, A. Bouguettaya, and S. Mistry, "A CP-Net based qualitative composition approach for an iaas provider," in *International Conference on Web Information Systems Engineering*, pp. 151–166, Springer, 2018.

[4] Q. Yu, X. Liu, A. Bouguettaya, and B. Medjahed, "Deploying and managing web services: issues, solutions, and directions," *The VLDB Journal–The International Journal on Very Large Data Bases*, vol. 17, no. 3, pp. 537–572, 2008.

[5] L. Chen, Y. Feng, J. Wu, and Z. Zheng, "An enhanced QoS prediction approach for service selection," in *2011 IEEE International Conference on Services Computing*, pp. 727–728, IEEE, 2011.

[6] L. Grunske, "Specification patterns for probabilistic quality properties," in *2008 ACM/IEEE 30th International Conference on Software Engineering*, pp. 31–40, 2008.

[7] P. Zhang, H. Jin, H. Dong, and W. Song, "M-BSRM: Multivariate bayesian runtime QoS monitoring using point mutual information," *IEEE Transactions on Services Computing, 2019, DOI: 10.1109/TSC.2019.2952604*.

[8] W.-T. Tsai, X. Zhou, Y. Chen, and X. Bai, "On testing and evaluating service-oriented software," *Computer*, vol. 41, no. 8, pp. 40–46, 2008.

[9] S. Liu and X. Meng, "Approach to network services recommendation based on mobile users¡ location," *J. Softw*, vol. 25, no. 11, pp. 2556–2574, 2014.

[10] J. Yin, L. Wei, S. Deng, Y. Li, Z. Wu, and N. Xiong, "Colbar: A collaborative location-based regularization framework for QoS prediction," *Information Sciences*, vol. 265, no. 5, pp. 68–84, 2014.

[11] P. He, J. Zhu, Z. Zheng, J. Xu, and M. R. Lyu, "Location-based hierarchical matrix factorization for web service recommendation," in *2014 IEEE International Conference on Web Services*, pp. 297–304, 2014.

[12] X. Chen, X. Liu, Z. Huang, and H. Sun, "Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation," in *IEEE International Conference on Web Services*, pp. 9–16, 2010.

[13] Z. Li, R. Xie, L. Sun, and T. Huang, "A survey of mobile edge computing," *Telecommunications Science*, no. 1, p. 11, 2018.

[14] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[15] Z. Li, Z. Bin, L. Ying, G. Yan, and Z. Zhi-Liang, "A web service QoS prediction approach based on collaborative filtering," in *Services Computing Conference*, pp. 725–731, 2010.

[16] Q. Xie, K. Wu, J. Xu, P. He, and M. Chen, "Personalized context-aware QoS prediction for web services based on collaborative filtering," in *International Conference on Advanced Data Mining and Applications*, pp. 368–375, Springer, 2010.

[17] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware web service recommendation by collaborative filtering," *IEEE Transactions on Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.

[18] P. Zhang, H. Jin, H. Dong, W. Song, and L. Wang, "LA-LMRBF: Online and long-term web service QoS forecasting," *IEEE Transactions on Services Computing, 2019, DOI: 10.1109/TSC.2019.2901848*.

[19] L. Shao, J. Zhang, Y. Wei, and J. Zhao, "Personalized QoS prediction forweb services via collaborative filtering," in *IEEE International Conference on Web Services*, pp. 439–446, 2007.

[20] L. Wei, J. Yin, Y. Li, and Z. Wu, "Efficient web service QoS prediction using local neighborhood matrix factorization," *Engineering Applications of Artificial Intelligence*, vol. 38, pp. 14–23, 2015.

[21] H. Wu, K. Yue, B. Li, B. Zhang, and C.-H. Hsu, "Collaborative QoS prediction with context-sensitive matrix factorization," *Future Generation Computer Systems*, vol. 82, pp. 669–678, 2018.

[22] L. Qi, H. Xiang, W. Dou, C. Yang, Y. Qin, and X. Zhang, "Privacy-preserving distributed service recommendation based on locality-sensitive hashing," in *IEEE International Conference on Web Services*, pp. 49–56, 2017.

[23] S. Wang, Y. Zhao, L. Huang, J. Xu, and C.-H. Hsu, "QoS prediction for service recommendations in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 134–144, 2019.

[24] C. Dwork, "Differential privacy," *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.

[25] X. Zhang, L. Qi, W. Dou, Q. He, C. Leckie, K. Ramamohanarao, and Z. Salcic, "Mrmondrian: Scalable multidimensional anonymisation for big data privacy preservation," *IEEE Transactions on Big Data*, vol. PP, no. 99, pp. 1–1, 2017.

[26] S. Badsha, X. Yi, I. Khalil, D. Liu, S. Nepal, E. Bertino, and K.-Y. Lam, "Privacy preserving location-aware personalized web service recommendations," *IEEE Transactions on Services Computing*, 2018.

[27] S. Liu, A. Liu, Z. Li, G. Liu, J. Xu, L. Zhao, and K. Zheng, "Privacy-preserving collaborative web services QoS prediction via differential privacy," in *Asia-Pacific Web*, pp. 200–214, 2017.

[28] M. Anisetti, C. Ardagna, E. Damiani, and G. Polegri, "Test-based security certification of composite services," *ACM Transactions on the Web (TWEB)*, vol. 13, no. 1, pp. 1–43, 2018.

[29] S. Lins, P. Grochol, S. Schneider, and A. Sunyaev, "Dynamic certification of cloud services: trust, but verify!," *IEEE Security and Privacy*, vol. 14, no. 2, pp. 66–71, 2016.

[30] C. Lin, Z. Shen, Q. Chen, and F. T. Sheldon, "A data integrity verification scheme in mobile cloud computing," *Journal of Network and Computer Applications*, vol. 77, pp. 146–151, 2017.

[31] W. Sun, X. Liu, W. Lou, Y. T. Hou, and H. Li, "Catch you if you lie to me: Efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 2110–2118, IEEE, 2015.

[32] D. He, S. Zeadally, L. Wu, and H. Wang, "Analysis of handover authentication protocols for mobile wireless networks using identity-based public key cryptography," *Computer Networks*, vol. 128, pp. 154–163, 2017.

[33] M. Anisetti, C. A. Ardagna, E. Damiani, N. El Ioini, and F. Gaudenzi, "Modeling time, probability, and configuration constraints for continuous cloud service certification," *Computers and Security*, vol. 72, pp. 234–254, 2018.

[34] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.

[35] W. Shi, H. Sun, J. Cao, Q. Zhang, and W. Liu, "Edge computing-an emerging computing model for the internet of everything era," *Journal of computer research and development*, vol. 54, no. 5, pp. 907–924, 2017.

[36] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*, 2008.

[37] C. Dwork, F. Mcsherry, and K. Nissim, "Calibrating noise to sensitivity in private data analysis," in *Conference on Theory of Cryptography*, pp. 265–284, 2006.

[38] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction in speech processing*, pp. 1–4, Springer, 2009.

[39] A. M. Neto, A. C. Victorino, I. Fantoni, and D. E. Zampieri, "Real-time dynamic power management based on pearson's correlation coefficient," in *2011 15th International Conference on Advanced Robotics (ICAR)*, pp. 304–309, IEEE, 2011.

[40] M. R. McLaughlin and J. L. Herlocker, "A collaborative filtering algorithm and evaluation metric that accurately model the user experience," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 329–336, ACM, 2004.

[41] S. Ramachandran, O. Deshpande, C. Roseman, N. Rosenberg, M. Feldman, and L. C. Sforza, "Support from the relationship of genetic and geographic distance in human populations for a serial founder effect originating in africa," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 44, pp. 15942–15947, 2005.

[42] R. W. Sinnott, "Virtues of the haversine," *Sky and Telescope*, vol. 68, no. 2, article 159, p. 158, 1984.

[43] Y. Li and S. Wang, "An energy-aware edge server placement algorithm in mobile edge computing," in *2018 IEEE International Conference on Edge Computing (EDGE)*, pp. 66–73, IEEE, 2018.

Pengcheng Zhang received the Ph.D. degree in computer science from Southeast University in 2010. He is currently an associate professor with the College of Computer and Information, Hohai University, Nanjing, China, and was a visiting scholar at San Jose State University, USA. His research interests include software engineering, services computing and data science. He has published in premiere or famous computer science journals, such as *IEEE Transactions on Services Computing*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Big Data*, *IEEE Transactions on Emerging Topics in Computing*, *Information and Software Technology*, *Journal of System and Software*, and *Software: Practice and Experience*. He was the co-chair of IEEE AI Testing 2019 conference. He served as technical program committee member on various international conferences. He is a member of the IEEE.

Huiying Jin is a PHD candidate with the College of Computer and Information, Hohai University, Nanjing, China. She received her bachelor degree in Software Engineering from Yangzhou University, Yangzhou, China in 2017. Her current research interests include services computing and data mining. She has published in international journals such as *IEEE Transactions on Services Computing* and *Information and Software Technology*.

Hai Dong is a Lecturer at School of Science in RMIT University, Melbourne, Australia. He was previously a Vice-Chancellor's Research Fellow in RMIT University. He received a PhD from Curtin University of Technology, Perth, Australia. He has published a monograph and over 80 research publications in international journals and conferences, such as Communications of the ACM, IEEE Transactions on Services Computing, IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Electronics, Journal of Computer and System Science, World Wide Web, ICSOC, ICWS, etc. He received the best paper award in ICSOC 2016. His primary research interests include: Services Computing, Distributed Systems, Cyber Security, Artificial Intelligence and Data Mining. He is a member of the IEEE and the ACM.

Wei Song received his Ph.D. degree from Nanjing University, China. He is a full professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China, and was a visiting scholar at Technische Universität München, Germany. His research interests include data science and engineering, software engineering and methodology, program analysis and testing, services and cloud computing, and process analysis and mining. He was invited to the Schloss Dagstuhl Seminar "Integrating Process-Oriented and Event-Based Systems" held in August, 2016. He has published in premiere computer science journals such as *IEEE Transactions on Big Data*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Dependable and Secure Computing*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Services Computing*, and *IEEE Transactions on Software Engineering*, as well as premiere international conferences such as ESEC/FSE, ASE. He is a member of the IEEE.

Athman Bouguettaya is Professor and Head of School of Computer Science at the University of Sydney, Australia. He received his PhD in Computer Science from the University of Colorado at Boulder (USA) in 1992. He was previously Science Leader in Service Computing at CSIRO ICT Centre, Canberra. Australia. Before that, he was a tenured faculty member and Program director in the Computer Science department at Virginia Tech. He is or has been on the editorial boards of several leading journals including, the IEEE Transactions on Services Computing, IEEE Transactions on Knowledge and Data Engineering, ACM Transactions on Internet Technology, ACM Computing Surveys, and VLDB Journal. He has published more than 250 books, book chapters, and articles in journals and conferences in the area of databases and service computing (e.g., the IEEE TKDE, the ACM TWEB, WWW Journal, VLDB Journal, SIGMOD, ICDE, VLDB, and EDBT). He was the recipient of several federally competitive grants in Australia (e.g., ARC) and the US (e.g., NSF, NIH). He is a Fellow of the IEEE and a Distinguished Scientist of the ACM.