# QoS Prediction for Service Recommendations in Mobile Edge Computing

Shangguang Wang[1], Yali Zhao[1], Lin Huang[1], Jinliang Xu[1], Ching-Hsien Hsu[2*]
[1]State Key Laboratory of Networking and Switching Technology; [2]Department of Computer Science and Information
[1]Engineering Beijing University of Posts and Telecommunications; [2]Chung Hua University
[1]Beijing 100876, China; [2]Hsinchu 707, Taiwan
{sgwang; zhaoyali2015; huanglin; jlxu}@bupt.edu.cn; chh@chu.edu.tw

**Abstract**—Mobile edge computing is an emerging technology that provides services within the close proximity of mobile subscribers by edge servers that are deployed in each edge server. Mobile edge computing platform enables application developers and content providers to serve context-aware services (such as service recommendation) by using real time radio access network information. In service recommendation system, quality of service (QoS) prediction plays an important role when mobile devices or users want to invoke services that can satisfy user QoS requirements. However, user mobility (e.g., from one edge server to another) often makes service QoS prediction values deviate from actual values in traditional mobile networks. Unfortunately, many existing service recommendation approaches fail to consider user mobility. In this paper, we propose a service recommendation approach based on collaborative filtering and make QoS prediction based on user mobility. This approach initially calculates user or edge server similarity and selects the Top-K most-similar neighbors, predicts service QoS, and then makes service recommendation. We have implemented our proposed approach with experiments based on Shanghai Telecom datasets. Experimental results show that our approach can significantly improve on the accuracy of service recommendation in mobile edge computing.

**Keywords**—*Mobile Edge Computing, QoS, Service Recommendation, Edge server Similarity*

## 1. INTRODUCTION

Mobile edge computing is an emerging technology that provides Web and cloud services within the close proximity of mobile subscribers. Traditional telecom network operators perform traffic control flow (forwarding and filtering of packets), but in mobile edge computing, edge servers are also deployed in each edge server. It also enables application developers and content providers to serve QoS-aware service recommendation based user context information by using real time radio access network information [1], [2]. In Mobile edge computing environment, edge server is deployed in between the mobile client and server near mobile proximity. For example, when a mobile web browser sends a request for a URL page, the response from the server is first intercepted at the edge server, since it can device information and analyze users behavior to improve services[1]. Based on the growing popularity of mobile devices, a large number of mobile services have been developed that run on mobile devices and often are invoked by people accessing edge servers in mobile edge computing [3]. Thus, it is important to know which mobile services have better QoS values for performance optimization. Hence, how to predict the QoS values accurately before services are invoked is a very important issue for service recommendation in mobile edge computing.

As it is well known that service QoS data are notably more volatile, and mobile devices often roam in mobile environment [4], [5]. Due to the mobility of mobile devices, history QoS data of mobile services in an edge server will fail when mobile devices move in another edge server and theservices QoS data in the new edge server is empty. To explain changes of edge server for mobile users easily, we present two types of edge server definition:

***Definition 1: old edge server*** – This refers to the edge server from which the active user adopted services before he moved out of its radio coverage.

***Definition 2: new edge server*** – This refers to the current edge server after the active user moved from the radio coverage of the old edge server. The active user adopts services by accessing this new edge server.

Although many QoS-aware service recommendation approaches [6], [7], [8] have been proposed in traditional Internet environments, they often fail to make accurate service recommendation in mobile edge computing because two problems exist that decrease service recommendation accuracy:

1) Volatility of QoS data. One active user invokes the same service many times, and QoS value is different each time. For example, one active user named Sam watches a movie on his mobile phone; the movie can be smooth one time but freeze the next time because of volatile QoS data. The above phenomenon is common in real life.

2) Mobility of active users. An active user often moves around, and edge servers change according to the location of the active user [9] in mobile edge computing. Suppose that Sam often uses service from an old edge server. When using one video service on his mobile phone, its response time is 100 msec on average when the host server running the service is deployed in the old edge server. When Sam roams in a new edge server, if the video service remains invoked, traditional service recommendation approaches often monitor its historical QoS data in the old edge server, and obtained response time remains 100 msec. However, its real response time will be different because Sam is located change.

Based on research and experiments with existing service recommendation approaches such as [10], [11], [8], and [12], [13], we found that these approaches caused large errors in mobile edge computing because of user mobility. User mobility results in changing user locations and data volatility. These large errors are introduced in detail as follows:

[*] Corresponding author

1) Mobility of user locations. In mobile edge computing, users invoke services by accessing different edge servers based on their dynamically changing locations. Because of user mobility, edge server handoff will be frequent [4], [5]. Therefore, history QoS data of users in the old edge server will likely be invalid when user the location changes significantly, and the QoS data of users in the new edge server are absent. Therefore, we should consider the mobility of users and it is important to learn how to predict the user QoS data from new edge servers.
2) Volatility of mobile networks. Because of the volatility of mobile environment, if you use QoS data for invoking the same service one time, the QoS prediction value cannot reflect the real situation of the QoS. Therefore, QoS prediction values for services will cause larger errors based on one-time QoS data.
3) Volatility of the same services at different invoked times. The QoS data for invoking the same services at a different time by one user are volatile. Calculating similarity between users based on the original QoS data is not reliable. If we do not preprocess the original QoS data, they will cause larger errors when calculating similarity between users.

Different from traditional service recommendation approaches, we first predict QoS values by reducing the influence of the above three factors. We then perform QoS prediction based on collaborative filtering and make service recommendations based on user mobility. Our approach was inspired by the following two cases; i.e., when users roam in a new edge server, if there are users in the new edge server and they invoke the service, then we can predict the QoS value based on their historical data. Otherwise, we use other user historical data from other edge servers at which they invoke the service. Finally, we conduct several experiments to verify our prediction accuracy based on the real-world environments.

The remainder of this paper is organized as follows: Section 2 shows our related work, and Section 3 introduces a motivation scenario. Section 4 presents our service recommendation approach based on user similarity and edge server similarity. Section 5 describes the implementation of our experiments and performance comparisons. Section 6 draws conclusion for our paper.

## 2. RELATED WORK

We have reviewed many Web service recommendation studies based on collaborative filtering algorithms, such as [14], [8], [15], [16], [7], and [17]. A few classic studies on the subject exist, including [11] and [18]. For example, Shao et al. [11] proposed an approach based on collaborative filtering to perform similarity mining and make predictions for users based on their experiences. The approach contains two steps. First, they calculate the similarity between each two consumers with their historical QoS data. Then, they predict the unused service QoS for consumers based on user similarity. They propose an approach to predicting user similarity by considering user similarity and user history service QoS experiences that is very important for subsequent research on QoS prediction. Zheng et al. [18] presented a Web service recommender system called WSRec to collect Web service QoS information from the real-world environment. Based on QoS data collected by WSRec, they proposed an effective and novel hybrid collaborative filtering algorithm to predict Web service QoS value. The approach to predicting QoS value considers both user similarity and item similarity to improve prediction accuracy. The above research presented many approaches [10],[19],[20],[15]. These studies are very meaningful and focus much effort on improving the accuracy of service recommendation. However, these approaches are only appropriate to predict QoS in traditional Internet environments and will result in large deviations or fail in mobile edge computing.

To address the above problems, many studies have been performed in traditional mobile network. They consider mobile location [12], [8], [21], mobile service [22], [23], [21], [24], or mobile networks [25], [26], [27] to adapt to mobile Internet environments. For mobile location, Chen et al. [12] proposed a location-based Web service recommendation, which employs both Web service history QoS values and user locations to make personalized QoS predictions. In their paper, they present a conception named similar region and retrieve approximate user locations by their IP addresses. They select the most similar region to predict QoS value, but similar regions are few, and the approach cannot adapt to mobile Internet environments. For a mobile recommendation, Zheng et al. [21] modeled user location-activity relationships with a tensor representation and proposed a regularized tensor and matrix decomposition solution to address the sparse data problem to adapt to mobile information retrieval. They retrieve the data of many users and apply collaborative filtering to find like-minded users and like-patterned activities at different locations, but the approach only considers the location of users, not service locations. Samba et al. [27] proposed an approach based on machine learning to predict throughput using data related to the context of the user, which refers to factors such as radio channel quality, speed, and distance from the edge server. The approach considers mobile networks, but no detailed algorithm is presented for QoS prediction.

The proposed useful approaches mentioned above will yield accurate service recommendation on the traditional Internet for Web service. However, in mobile edge computing, these approaches will make result in a large deviation or fail. To address this problem, we propose an approach that considers user location and data volatility in mobile edge computing.

## 3. MOTIVATION

Suppose that Sam often use one service on his mobile phone by accessing an edge server b1 with response time (e.g., one QoS property) of 100 msec on average. As Fig. 1 depicts, Sam now travels to another edge server b2, and he want to use the same service. Then how to predict the QoS value of the service become an important issue by accessing the edge server b2. If the predicted value is less than 100 msec, this means Sam can still use the service; otherwise the service will be migrated to the edge server b2 or other services are recommended.

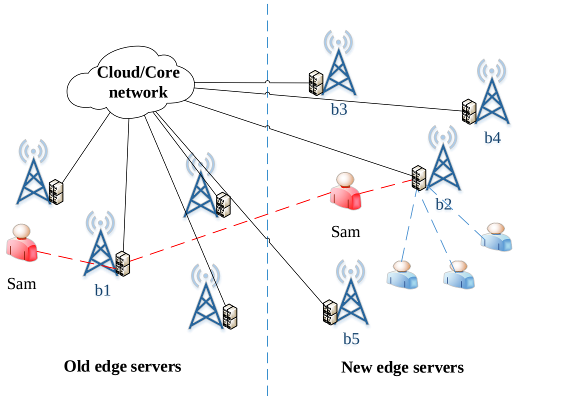


Fig. 1. Service recommendation in mobile edge computing

For solving the problem, traditional approaches [11], [18], [10], [15], [28], [29] often monitor the history QoS data of the edge server b1 for Sam and the obtained response time would remain 100 msec. However, the edge server that Sam accesses and the mobile networks that support Sam have changed. As Fig. 1 depicts, Sam invokes the service by accessing edge server b2 instead of edge server b1; thus, the response time must be different.

We recognize that some QoS properties, such as response time and throughput, are highly related to the network environments of the edge server near which the user is located [12], [27]. On edge server b2, many other people invoke services, and Sam can ask those people who invoke the same service as he does to obtain the response time. However, if no one has invoked the same service, how can we predict the QoS value for Sam?

Our motivating problem is to make more-accurate QoS predictions for service recommendation in mobile edge computing.

To reach this goal, several challenges must be addressed. 1) How can we redefine the CF (collaborative filtering) algorithm to adapt it for QoS prediction when considering edge server information? 2) How can we perform service recommendation in mobile edge computing?

## 4. OUR APPROACH

Motivated by the above analysis, we propose an approach based on the CF algorithm to predict user QoS data by weakening the volatility of QoS data and considering the mobility of users. In our approach, based on the QoS data after normalization, we initially calculate user or edge server similarity. If the service invoked by an active user exists in the QoS data of new edge server, we calculate the similarity between users. If not, we should find other similar edge servers for the active user; therefore, we propose an algorithm to calculate the similarity between edge servers by adopting the Pearson Correlation Coefficient (PCC). Then, we select Top-K users or edge servers. Finally, we predict the QoS value of the active user based on our approach for service recommendation.

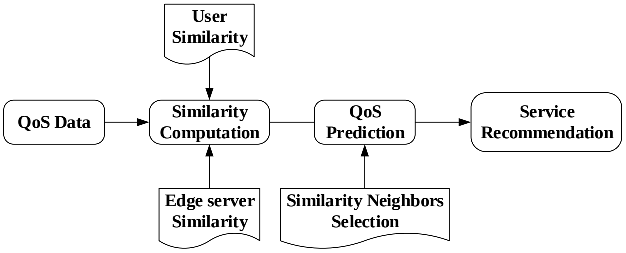


Fig. 2. Procedure for Our Approach

As shown in Fig. 2, our proposed approach consists of three steps as follows:

1) Similarity Computation. This step contains a user and edge server similarity computation to calculate the similarity between an active user and other users or to calculate the similarity of the edge server with other edge servers to obtain the set of similar users or edge servers for an active user.
2) QoS prediction. With similar users or edge servers selected out by similarity neighbor selection for active users, we can predict the QoS value.
3) Service Recommendation. Based on predicted QoS values, we make service recommendations to meet user requirements.

### 4.1 Similarity Computation

With the development of 4G/5G network, an increasing number of edge servers have been built, and users move around in mobile network environments [30]. Hence, when we predict the QoS data of service *s* invoked by active user *u*, we must consider the current edge server of active user *u*. Based on service *s* being invoked by active user *u*, we can divide the situation into two cases, as follows:

**Case 1.** *Service s invoked by active user u has history QoS data in the new edge server, i.e., users in the control of the new edge server have adopted the same service s as active user u; therefore, the history QoS data of service s have been stored in the new edge server.*

**Case 2.** *Service s invoked by active user u has no history QoS data in the new edge server, i.e., users in the control of the new edge server have not adopted the same service s as active user u; therefore, there is no history QoS data about service s in the new edge server.*

Many recommender systems [31], [32] have used PCC [33], [34] to calculate similarity because PCC can be implemented easily and can achieve high accuracy. In the next section, we use PCC to calculate similarities between two users and two edge servers.

#### 4.1.1 User Similarity Computation

In this paper, let $q_{u,s}^{t}$ represent the history QoS value when user *u* repeatedly invokes service *s* ($s$=1,2,3,…) at the *t*-th time ($t$=1,2,3,…), and $q_{u,s}^{'t}$ represent the QoS value after min-max normalization [35].

For situations such as Case 1 describes, the QoS history in the old edge server of active user *u* is invalid and cannot be used for user similarity computation. However, we can use the history QoS data of the same service that active user u adopts. We calculate the active user *u* similarity to other users who are in the control of the new edge server based on the PCC with the following equation:

$$sim_{u,v} = \frac{1}{S}\sum\nolimits_{s=1}^{S} \frac{\sum\nolimits_{t=1}^{T}(q^{'t}_{u,s} - E_{u,s})(q^{'t}_{v,s} - E_{v,s})}{\sqrt{\sum\nolimits_{t=1}^{T}(q^{'t}_{u,s} - E_{u,s})^2}\sqrt{\sum\nolimits_{t=1}^{T}(q^{'t}_{v,s} - E_{v,s})^2}} \tag{1}$$

with

$$E_{u,s} = \frac{\sum\nolimits_{t=1}^{T} q^{'t}_{u,s}}{T} \tag{2}$$

$$q^{'t}_{u,s} = \frac{q^{t}_{u,s} - Q^{\min}_{u,s}}{Q^{\max}_{u,s} - Q^{\min}_{u,s}} \tag{3}$$

where $sim_{u,v}$ represents the similarity between user *u* and user *v* on the same service *s* ( $s \in S_u \cap S_v$ ), which both user *u* and user *v* have commonly invoked. *S* is the total number of the same invoked services, and $E_{u,s}$ is the average value of service *s* invoked by user *u* for *T* times. $q^{'t}_{u,s}$ is the QoS value after min-max normalizing the history QoS value. We use $Q_{u,s}(Q_{u,s} = (q^1_{u,s}, q^2_{u,s}, ..., q^t_{u,s}, ..., q^n_{u,s}))$ to represent user *u* invoking service *s n* times. $Q^{\min}_{u,s}, Q^{\max}_{u,s}$ represent the min and max QoS value for $Q_{u,s}$ , respectively.

### 4.1.2 Edge Server Similarity Computation

For the situation described in Case 2, i.e., service *s* invoked by active user *u* has no history QoS data in the new edge server, we must find edge servers similar to the old edge server.

Based on the PCC, we propose an algorithm to find similar edge servers for a new edge server accessed by an active user. The algorithm is similar to the user similarity computation except that the edge server similarity computation employs the similarity between edge servers instead of between service users. The similarity computation between edge server $b_1$ and edge server $b_2$ can be calculated with the following equation:

$$sim_{b_1,b_2} = \frac{1}{S}\sum\nolimits_{s=1}^{S} \frac{\sum\nolimits_{u=1}^{U}(P^{u}_{b_1,s} - E_{b_1,s})(P^{u}_{b_2,s} - E_{b_2,s})}{\sqrt{\sum\nolimits_{u=1}^{U}(P^{u}_{b_1,s} - E_{b_1,s})^2}\sqrt{\sum\nolimits_{u=1}^{U}(P^{u}_{b_2,s} - E_{b_2,s})^2}} \tag{4}$$

with

$$P^{u}_{b_1,s} = \frac{\sum\nolimits_{t=1}^{T} q^{'t}_{u,s}}{T} \tag{5}$$

$$E_{b_1,s} = \frac{\sum\nolimits_{u=1}^{U} P^{u}_{b_1,s}}{U} \tag{6}$$

where $sim_{b_1,b_2}$ is the similarity between edge server *b₁* and edge server *b₂*. $P^{u}_{b_1,s}$ represents the average QoS value of service *s* invoked by active user *u* in edge server *b₁*. $E_{b_1,s}$ represents the average QoS value of service *s* invoked in edge server *b₁*, where *S* denotes the total number of same services invoked in edge servers *b₁* and *b₂*.

### 4.1.3 Significance Weighting

If two users or edge servers have similar QoS histories on a few of the same invoked services, then using the PCC will overestimate the similarities of service users or edge servers [10], [36]. To address this problem, we employ a significance weight to reduce the influence of a small number of invoked similar services. An enhanced PCC for the similarity computation between different service users is defined in the following equation:

$$sim^{'}_{u,v} = \frac{2 \times |S_u \cap S_v|}{|S_u| + |S_v|} sim_{u,v} \tag{7}$$

where $sim^{'}_{u,v}$ represents the new similarity value, $|S_u \cap S_v|$ is the number of services invoked by both users, and $|S_u|$ and $|S_v|$ are the number of services invoked by user *u* and user *v* respectively.

Similar to the user similarity computation, an enhanced PCC between different edge servers is defined as follows:

$$sim^{'}_{b_1,b_2} = \frac{2 \times |S_{b_1} \cap S_{b_2}|}{|S_{b_1}| + |S_{b_2}|} sim_{b_1,b_2} \tag{8}$$

where $|S_{b_1} \cap S_{b_2}|$ is the number of services that invoked in both edge server *b₁* and edge server *b₂*, and $|S_{b_1}|$ and $|S_{b_2}|$ are the numbers of services invoked in edge server *b₁* and edge server *b₂*, respectively.

## 4.2 QoS Prediction

Based on the above user and edge server similarity computations, we propose a user-based similarity selection and distance-based similarity selection approach, respectively. Then, we predict QoS value of services for active users.

### 4.2.1 Similar Neighbor Selection

An important step for making accurate QoS value prediction is to select similar neighbors, because dissimilar neighbors will decrease prediction accuracy. We will introduce two algorithms for Case 1 and Case 2, respectively.

1) User-based Similarity Selection

For the situation of an active user *u* as in Case 1, we must select Top-K most-similar users for user *u*. We use the enhanced Top-K algorithms to rank the users based on PCC similarities and select the Top-K most-similar users for making QoS value predictions. Different from traditional Top-K algorithms, the enhanced Top-K algorithm excludes users with PCC similarities less than or equal to 0.

The Top-K similar user set of user *u* is

$$S_u(u_i) = \{u_1, u_2, ..., u_i, ..., u_K\}, i = 1, 2, ..., K \tag{9}$$

where *i* represents the ordinal of similar users of user *u*. The enhanced user-based Top-K set of similar users can be found with the following equation:

$$S'_u(u_i) = \{u_i \in S_u(u_i), sim'_{u,u_i} > 0, u_i \neq u\} \quad (10)$$

2) Distance-based Similarity Selection

For the situation of an active user *u* as in Case 2, we should find similar edge servers for new edge server *b*, which is accessed by active user *u*. Every edge server has a radio coverage area, and the distance cannot be too great because the resulting area might contain noise and thus degrade prediction performance.

Motivated by the situation, we propose a distance-based enhanced Top-K selection strategy. The strategy (i.e., Eq.11) considers the edge server distribution density around *b*.

We define a parameter $\lambda$ that represents the distance between other edge servers and *b*, and we select similar edge servers for *b* within distance $\lambda$.

Based on the above analysis, the distance-based set of similar edge servers can be found with the following equation:

$$S_b(b_i) = \{b_1, b_2, ..., b_i, ...b_B\}, d(b,b_i) < \lambda \quad (11)$$

where *i* represents the ordinal of similar edge servers of *b*, and *B* represents the total edge servers of *b* in the range of $\lambda$, which will be analyzed in Section 4.5 in detail. $d(b,b_i)$ represents the distance between edge server *b* and $b_i$ based on great circle distances using the haversine [37]. We calculate the distance using a typical Geographic Distance algorithm [38], in which the distance $d(b,b_i)$ between edge server *b* and $b_{i\ is}$ specified by (latitude, longitude) coordinates ($\alpha_1$, $\sigma_1$) and ($\alpha_2$, $\sigma_2$).

$$d(b,b_i) = 2R\arctan(\frac{\sqrt{hav(\theta)}}{\sqrt{1-hav(\theta)}}) \quad (12)$$

with

$$hav(\theta) = \sin^2(\frac{\sigma_1-\sigma_2}{2}) + \cos\sigma_1\cos\sigma_2\sin^2(\frac{\alpha_1-\alpha_2}{2}) \quad (13)$$

where $\theta$ is a central angle between the two edge servers and R is the radius of the Earth, which we assume to be 6371 km.

Based on the above, we select Top-K most-similar edge servers for *b*. Similar to user-based, the Top-K similar edge server set of *b* can obtained with the following equation:

$$S'_b(b_i) = \{b_i \in S_b(b_i), sim'_{b,b_i} > 0, b_i \neq b\} \quad (14)$$

Thus, dissimilar neighbors with negative correlations and the null intersection neighbors will be discarded from the similar neighbor sets.

### 4.2.2 QoS Prediction for Active Users

Based on the similarity between every pair of users and edge servers and the most-similar users or edge servers, we finally predict the QoS value for active users with Algorithm 1.

**Algorithm 1**: QoS Prediction Algorithm

```
INPUT : User u ; Service s; Edge Server b; QoS Data ds
OUTPUT : Prediction QoS Value
BEGIN :
b_ds  as the QoS data for edge server b from ds
//divides situation into Case 1 and Case 2
isServiceExist == false;
WHILE i < b_ds.length()
  IF service(i) == s;
    isServiceExist = true ;
      BREAK;
  i++;
RETURN isServiceExist;
// find similar users for user u
Case 1: IF isServiceExist == true;
  similar_userset = null;
  WHILE i < b_ds.length()
    IF user(i) invoked s via b
     similar_userset.add(user(i));
    i++;
  Top_K(similar_userset);
RETURN prediction QoS value;
// find similar edge servers for b
Case 2: IF isServiceExist == false;
 similar_edgeserver = =null;
 WHILE i<ds.length
  IF any user invoked s via b(i)
    similar_ edgeserverset.add(b(i));
    i++;
 bestedgeserverset == null;
 WHILE i < similar_ edgeserverset.length()
// find best similar edge servers with distance within λ
 IF Distance(b,similar_ edgeserverset(i))< λ
 best edgeserverset.add(similar_ edgeserverset(i));
     i++;
  Top_K(edgeserverset);
RETURN prediction QoS value;
END
```

Based on Algorithm 1, whether or not services that are invoked by active users exist in the QoS history data in the new edge server, we predict QoS for active users as follows:

a) Based on the user similarity, we predict the QoS value of active user *u* with the following equation:

$$pred_{user}(u,s) = \bar{u} + \frac{\sum_{u_i \in S'_u(u_i)} sim'_{u,u_i}(E_{u_i,s} - \bar{u_i})}{\sum_{u_i \in S'_u(u_i)} sim'_{u,u_i}} \quad (15)$$

with

$$\bar{u_i} = \frac{1}{S}\sum_{s=1}^{S} E_{u_i,s} \quad (16)$$

where $\bar{u}$ is the average of QoS values of different services invoked by active user *u*. $\bar{u_i}$ represents the average QoS values of different services invoked by similar user $u_i$, and *S* is the total number of services invoked by user $u_i$.

b) Based on the edge server similarity, service *s* invoked by active user *u* has no history QoS data in the new edge server. Therefore, we find the Top-K similar edge servers for the new edge server. Because the history QoS value of the active *u* in the old edge server fails for predicting and based on the Top-K similar edge servers, we propose an approach to predict the QoS value for service *s* invoked by active user *u* as follows:

$$pred_{edgeserver}(u,s) = \frac{\sum_{b_i \in S_b^{'}(b_i)} sim_{b,b_i}^{'} * E_{b_i,s}}{\sum_{b_i \in S_b^{'}(b_i)} sim_{b,b_i}^{'}} \qquad (17)$$

with

$$E_{b_i,s} = \frac{\sum_{u=1}^{U} P_{b_i,s}^{u}}{U} \qquad (18)$$

where $E_{b_i,s}$ represents the QoS expectation of service *s* invoked in edge server $b_i$ .

### 4.3 Service Recommendation

Based on the above-predicted QoS values of services for an active user, when the service can meet the active user QoS requirements, mobile edge computing platform recommend that the active user can still use the service from the old edge server, otherwise the platform recommend the active user to use other service or migrate the service to the new edge servers.

## 5. EXPERIMENTS

In this section, we perform experiments to verify the performance of our approach and compare the results with other CF methods. Our experiments are intended to 1) validate the rationality of our proposed approach; 2) compare our approach with other CF methods; and 3) analyze parameters of our approach to achieve optimum performance.

### 5.1 Experiments Setup

We adopt the QoS dataset to validate our prediction approach, and we conduct experiments by employing eclipse 4.5 and JDK 1.8. Based on previous work, our experiments primarily contain two parts: 1) compare our approach with other known methods; 2) study the optimal parameter $\lambda$ and the effect of parameter Top-K in our approach.

### 5.2 Dataset

#### 5.2.1 Dataset Description

In our experiments, we adopt a hybrid dataset that is a mixture of the Shanghai Telecom and WSDream datasets [10]. The Shanghai Telecom dataset contains Internet information about 6357 service invocations on 3233 base station. Note that we call base station as edge server in the following experiment. Fig. 3 allows a distinct understanding of the distribution of edge servers for Shanghai Telecom. The WSDream dataset describes real-world QoS evaluation results, including both response time and throughput values, obtained from 142 users on 4500 web services over 64 times as a 142×4500 user-service matrix.

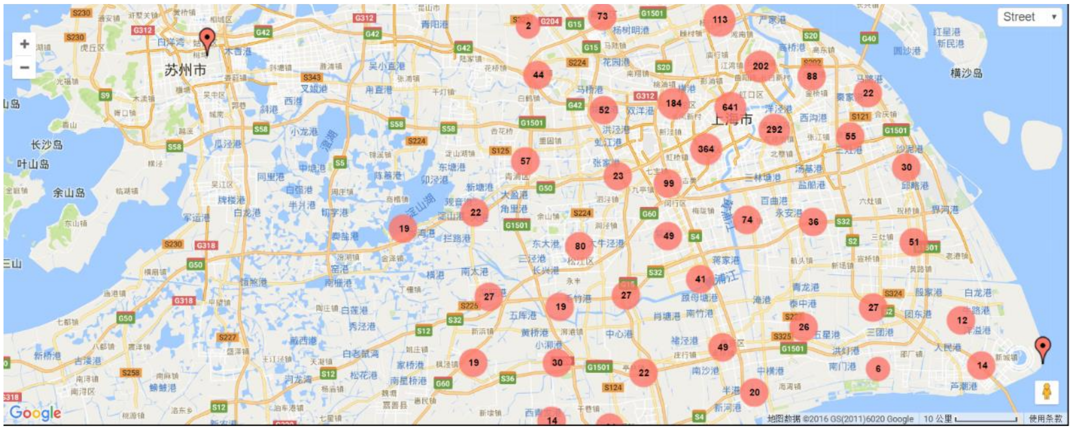


Fig. 3. Distribution of edge servers for Shanghai Telecom

Fig. 3 shows the distribution of 3233 edge servers. The number represents the number of edge servers within range of the red circle. Fig. 3 illustrates that the edge server distribution is dense in Shanghai.

For the Shanghai Telecom dataset, Table 1 describes the Internet information for user 27 as an example. For edge server 106, user 27 invokes services 211 times. However, user 27 invokes services in edge server 214 only 1 time. User 27 invoked services in 18 edge servers altogether. Table 1 shows that one user invokes services by accessing different edge servers and many times in the same edge server. Additionally, the total number of edge servers that the user accesses shows that the handoff between edge servers is frequent. Next, we mix the WSDream and Shanghai Telecom

Table 1. Shanghai Telecom Internet Information for User 27

| **User ID** | 27 | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Edge server ID** | 211 | 212 | 152 | 107 | 230 | 227 | 106 | 214 | 215 | 222 | 223 | 97 | 199 | 221 | 224 | 108 | 152 | 121 |
| **Invoked Times** | 2 | 9 | 2 | 3 | 2 | 2 | 211 | 1 | 1 | 1 | 13 | 3 | 1 | 126 | 2 | 2 | 8 | 2 |

datasets by considering the characteristics of both datasets, i.e., we take real response time from the WSDream dataset as the response time when the user invokes service by accessing an edge server in the Shanghai Telecom dataset.

### 5.2.2 Dataset Mixture

Because of user mobility, we must consider the location of edge servers that are accessed by active users. To adapt to mobile networks environments, we need a dataset comprising QoS data on services by accessing edge servers for users. However, no dataset meets our requirements; a dataset cannot adapt real-world mobile network environments if it is obtained by simulation experiments.

Following many studies on data fusion [39], [40], [41], we introduce an approach on how to mix QoS data and edge server data in this section.

QoS data come from the WSDream dataset[10] and have been normalized to reduce volatility. The edge server data originate from Shanghai Telecom; they contain 6358 users and 3233 edge servers. Every user has a unique ID and multiple edge servers; every edge server owns a detailed location.

Based on the above analysis of two datasets, we fill QoS data into edge server data following these rules:

1) Same user ID in Shanghai Telecom dataset with one edge server. We take one user's QoS data on the same service from the WSDream dataset;
2) If edge server changes, we take one user's QoS data on another service from WSDream dataset;
3) If user ID changes, we take another user's QoS data from WSDream.

Following the above data mixture rule, we obtain a new hybrid dataset that contains both QoS data and edge server locations.

## 5.3 Accuracy Metrics

Mean absolute error (MAE) and root mean squared error (RMSE) are commonly used to measure the difference between values predicted by a model or estimator and real values. We adopt MAE and RMSE to measure the prediction accuracy of our approach by making comparisons with other methods. MAE is defined as follows:

$$MAE = \frac{\sum \left| P_{u,s} - \bar{P}_{u,s} \right|}{N} \tag{19}$$

where $P_{u,s}$ is the predicted QoS value, and $\bar{P}_{u,s}$ denotes the real QoS value of service $s$ invoked by active user $u$.

RMSE is defined as follows:

$$RMSE = \sqrt{\frac{\sum (P_{u,s} - \bar{P}_{u,s})^2}{N}} \tag{20}$$

where $N$ is the number of predicted values. The smaller the values of MAE and RMSE are, the more accurate the prediction.

## 5.4 Performance Comparisons

We compare the performance of our approach with other approaches. The other approaches are as follows:

1) UMEAN: employs the average QoS performance of the current service user on other web services;
2) IMEAN: employs the average QoS performance of the web service observed by other service users;
3) UPCC: user-based prediction algorithm using PCC; employs similar users for service recommendation [42], [11];
4) IPCC: item-based prediction algorithm using PCC; employs similar web services for service recommendation [31];
5) WSRec: WSRec [10] combines user-based and item-based methods to predict QoS values for service recommendation.

In this experiment, we randomly select an active user from our dataset to make service recommendation in terms of response time and compare our approach with other methods. The experimental results are shown in Table 2.

Table 2. Accuracy Comparison

| | Methods | Matrix Density=10% | | Matrix Density=20% | | Matrix Density=50% | |
|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| Response Time | UMEAN | 0.8783 | 1.8531 | 0.8699 | 1.8642 | 0.8433 | 1.8478 |
| | IMEAN | 0.7004 | 1.5593 | 0.6805 | 1.5305 | 0.6625 | 1.5165 |
| | UPCC | 0.5866 | 1.3580 | 0.5197 | 1.2751 | 0.3995 | 1.1554 |
| | IPCC | 0.6403 | 1.3766 | 0.5189 | 1.2669 | 0.3680 | 1.1913 |
| | WSRec | 0.5431 | 1.2351 | 0.4987 | 1.1254 | 0.4885 | 1.0993 |
| | **OUR APPROACH** | **0.5213** | **1.1896** | **0.4826** | **1.0247** | **0.3556** | **0.9783** |

Table 2 shows that the MAE and RMSE values of our approach become smaller as the user-service matrix density increases from 10% to 50%, because similarities between users or edge servers become steadier as the amount of data increases. When the matrix density is set as10% or 20%, the MAE and RMSE values of our approach are slightly smaller than with other approaches. Therefore, our approach's accuracy will decrease when the matrix density is sparse.

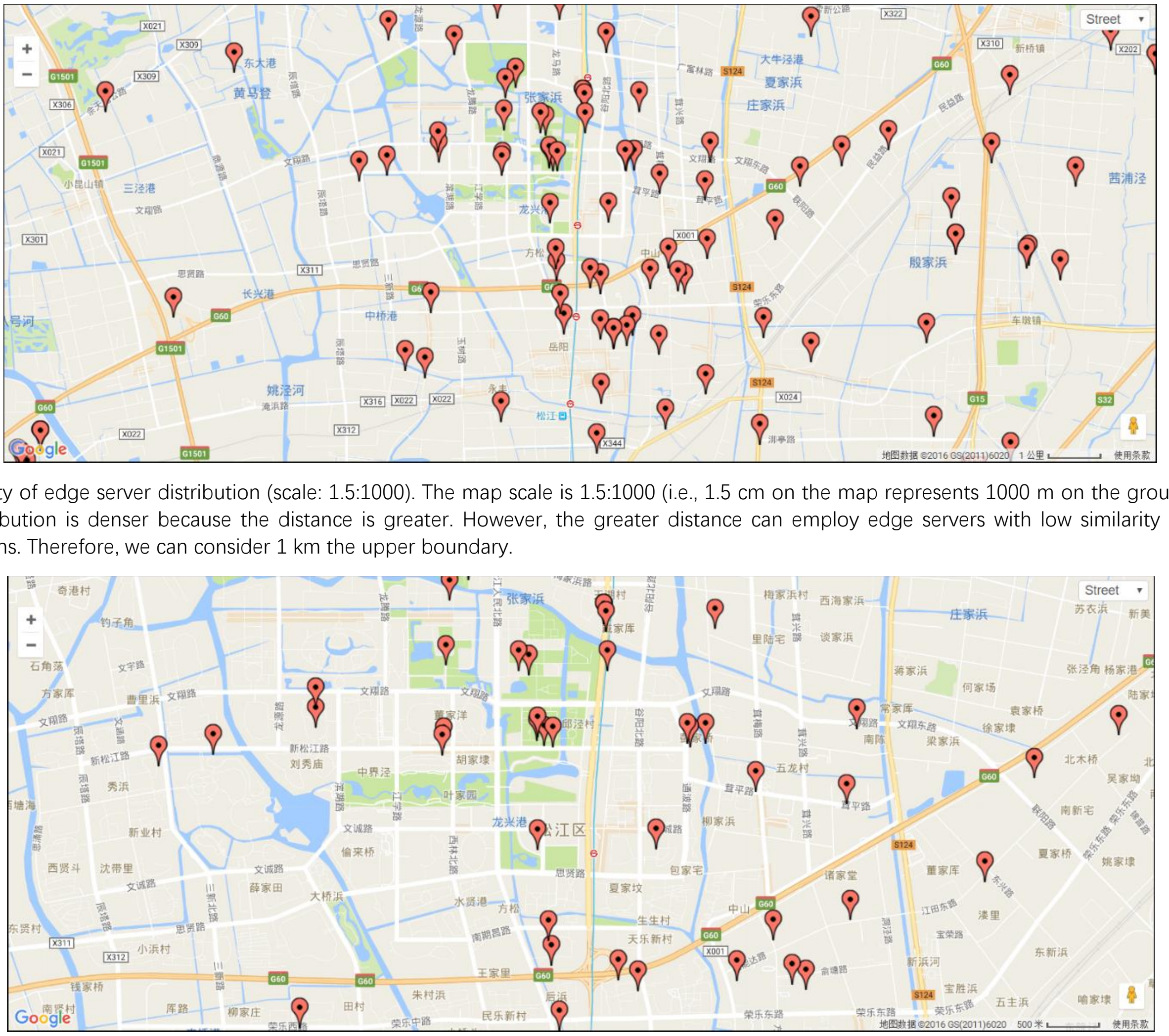
When the matrix density increases to 50%, the MAE and RMSE values are smaller than other approaches, indicating that the prediction accuracy can be improved by increasing the matrix density. Thus, our approach is more accurate than are all of the other methods in terms of response time because the other approaches cannot consider volatility and mobility, which is necessary to adapt to the mobile edge computing environment.

## 5.5 Effect of parameter $\lambda$

### 5.5.1 Is a smaller $\lambda$ better?

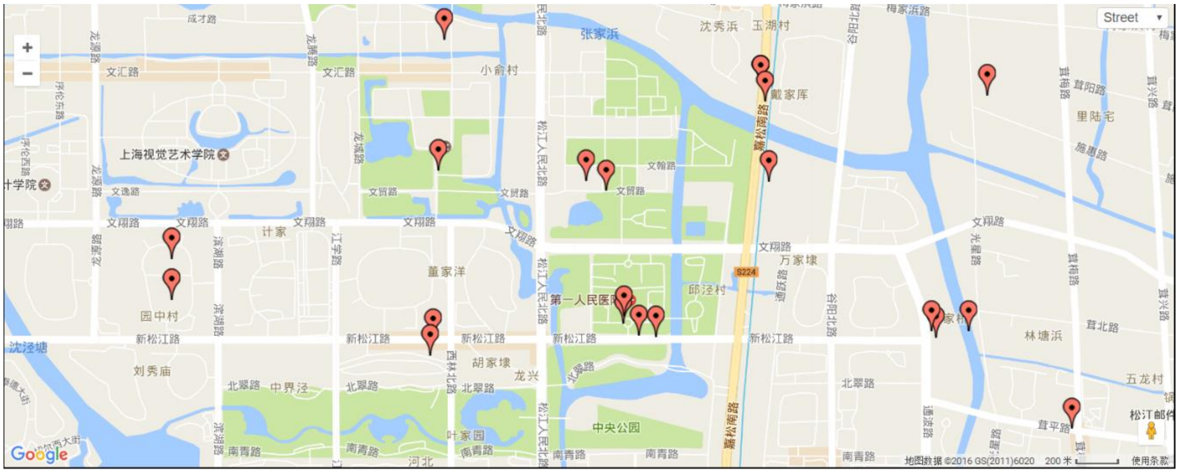
In this section, we use Google Maps API, that is, EasyMapMaker, to display the density of edge server distribution to analyze the better fitting distance between similar edge servers. EasyMapMaker can map Excel or other spreadsheet data onto a Google map and avoids manually plotting multiple locations on a map. We apply all edge servers from the list onto a map. Fig. 4 shows the density of edge servers based on different distances between edge servers.

We must define the boundary to select similar edge servers because largest or smallest distance is inappropriate. From Fig. 4, $\lambda$ is not better when smaller; we can find a better fitting boundary by analyzing the density of edge server distribution. To calculate the optimum $\lambda$, we design the following experiment.

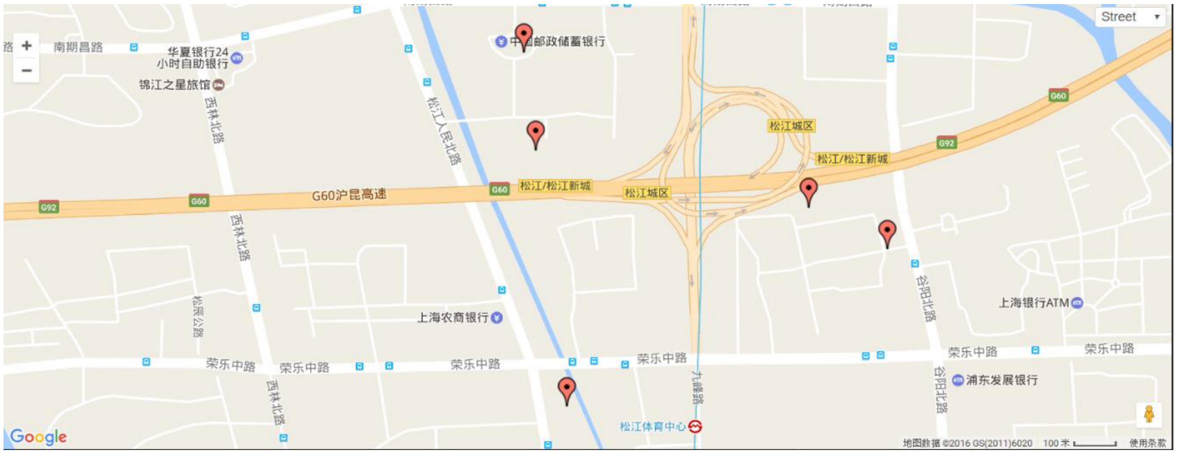

Street
Google
地图数据 ©2016 GS(2011)6020 1 公里
使用条款
Street
Google
地图数据 ©2016 GS(2011)6020 500 米
使用条款

(a) Density of edge server distribution (scale: 1.5:1000). The map scale is 1.5:1000 (i.e., 1.5 cm on the map represents 1000 m on the ground), and the distribution is denser because the distance is greater. However, the greater distance can employ edge servers with low similarity to make predictions. Therefore, we can consider 1 km the upper boundary.

(b) Density of edge server distribution (scale: 1.5:500). The map scale is 1.5:500. The distribution becomes sparser, but we can obviously find similar edge server clusters. Therefore, we consider 500 m the middle boundary of the parameter analysis.



(c) Density of edge server distribution (scale: 1.5:200). The map scale is 1.5:200. We can find nearby edge servers based on the distance between edge servers.



(d) Density of edge server distribution (scale: 1.5:100). The map scale is 1.5:100. We can find few near edge servers. Therefore, we consider 200 m the lower boundary.

Fig. 4. Different scales on maps influence the density of edge server distribution. A smaller $\lambda$ is not better, because a smaller $\lambda$ can employ few or no edge servers to make predictions.

### 5.5.2 Optimum setting of the $\lambda$ parameter

Table 3. Similar Edge server Density Distribution

| **User ID** | **Ranges of Distances between Similar Edge servers** | | | | |
|---|---|---|---|---|---|
| | 5 km | 2 km | 1 km | 0.5 km | 0.2 km |
| **7** | 267 | 52 | 12 | 2 | 1 |
| **29** | 613 | 121 | 29 | 5 | 1 |
| **132** | 653 | 194 | 67 | 25 | 2 |

To study the optimum value of parameter $\lambda$ for Case 2, i.e., the number of similar edge servers needed to provide a relatively accurate recommendation, we study one edge server for an active user at a time.

We have speculated that the optimum $\lambda$ is related to a similar edge server density distribution. Therefore, we initially study the distribution of similar edge servers for an active user before making recommendations and observe the distances between similar edge servers and the edge server for an active user. Table 3 shows the results for three active users randomly selected from the dataset. Each active user is in the coverage of an edge server, and User ID in Table 3 denotes the id in the hybrid dataset. For each of three active users, we calculate how many similar edge servers exist in different distance ranges.

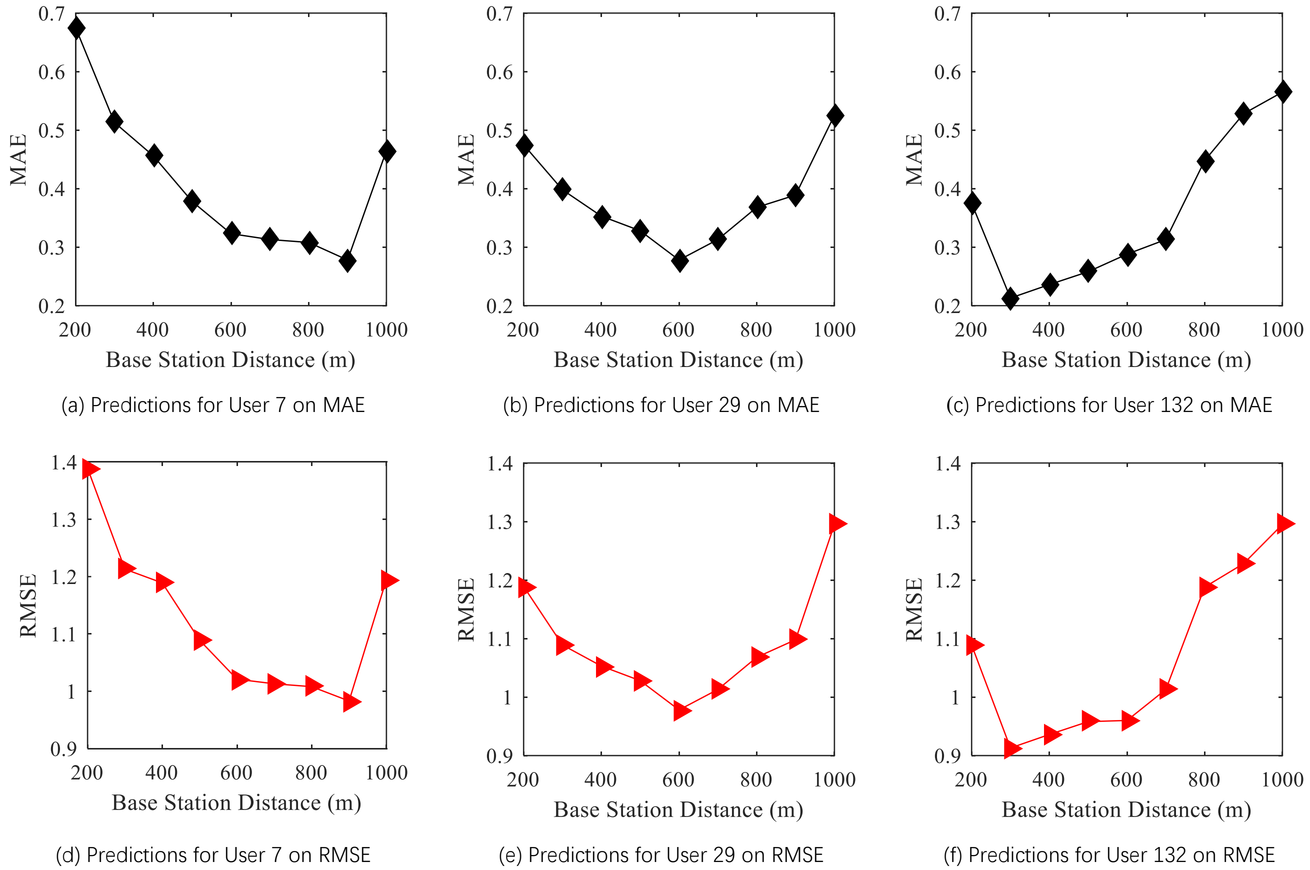
Table 3 shows that the greater the range of distances, the greater the number of similar edge servers. Similarly, a smaller range of distances implies fewer stations. When the range of distance is less than 0.2 km, the number of similar edge servers is less than 2. Using dissimilar or smallest similar edge servers to predict the missing value will significantly reduce prediction accuracy. Hence, we make predictions for one active user in the range between 0.2 km and 1 km to examine which distance ranges provided the most accurate service recommendations.

From Fig. 5a and Fig. 5d, the most accurate prediction for user 7 comes from $\lambda$ = 900 m. User 7 has 12 similar edge servers when distance is less than 1 km. Therefore, the number of similar edge servers for user 7 is less than 12, which contributes to the prediction accuracy.

Form Fig. 5b and Fig. 5e, the most accurate prediction for user 29 comes from $\lambda$ = 600 m. User 29 has 29 similar edge servers when distance is less than 1 km, and 9 edge servers when distance becomes less than 0.5 km. Therefore, the number of similar edge servers that contribute to prediction accuracy is slightly greater than 9.

User 132 has 25 similar edge servers when distance is less than 0.5 km, and the most accurate prediction comes from $\lambda$ = 300 m (see Fig. 5c and Fig. 5f). Combining Table 3, Fig. 5c and Fig. 5d, we can obtain an approximate number less than 10.

Fig. 5 and Table 3 illustrate that the optimum distance is determined by similar edge server density distribution. If the distribution is denser, the distance will be smaller. For different active users, the optimum $\lambda$ value is different. High performance can be achieved by setting $\lambda$ to include similar edge servers whose number is approximately 10 in our hybrid dataset.



(a) Predictions for User 7 on MAE

(b) Predictions for User 29 on MAE

(c) Predictions for User 132 on MAE

(d) Predictions for User 7 on RMSE

(e) Predictions for User 29 on RMSE

(f) Predictions for User 132 on RMSE

Fig. 5. Effect of parameter $\lambda$ for Different Active Users on response time. The best $\lambda$ employs approximately 10 similar edge servers, which contributes to accurate prediction.

### 5.6 Effect of Enhanced Top-K

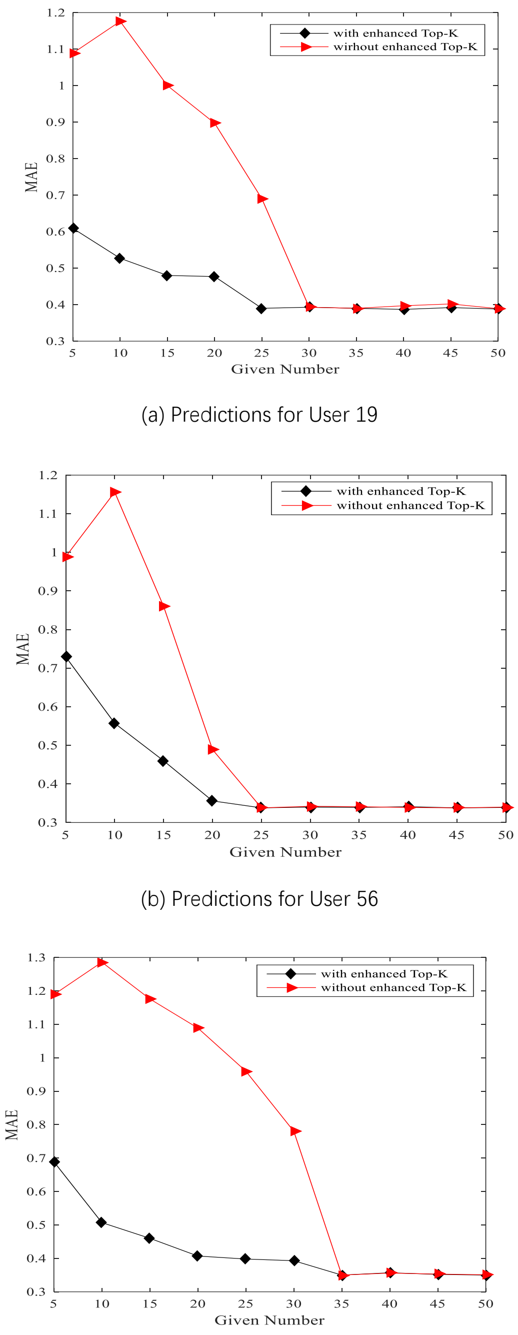
We study the effect of enhanced Top-K for Case 1, i.e., using an enhanced Top-K algorithm, dissimilar users with negative PCC values from the Top-K similar neighbors are excluded.

We do experiments on two versions for three active users who are selected randomly. One version employs enhanced Top-K, whereas another does not. For example, Fig. 6 shows that our approach with the enhanced Top-K outperforms without the enhanced Top-K on response time with the increasing number of service users (i.e., the given number).

Our experiments set Top-K as 10. Fig. 6 shows that the prediction values without enhanced Top-K for User 19, User 56, and User 176 are volatile because they might include negative similar users to make a prediction, which will greatly decrease prediction accuracy.

Fig. 6a shows that differences between the two versions in MAE decrease when the given number increases; when the given number is set as 35, the two versions almost overlap.

Fig. 6b and Fig. 6c show the same trend as shown in Fig. 6a. However, the difference between two versions decreases to 0 when the given number is set as 25 for User 56 and the given number is set as 35 for User 176, respectively. Therefore, Top-K can be set to be a large value to obtain optimal performance based on different active users in our approach.



(a) Predictions for User 19

(b) Predictions for User 56

(c) Predictions for User 176

Fig. 6. Effect of enhanced Top-K for different active users on response time. The given number represents the number of service users. The experimental setting is Top-K=10 and given number is from 5 to 50.

## 6. CONCLUSION and FUTURE WORK

Different from traditional service recommendations base on QoS prediction, our approach considers user mobility and data volatility to adapt to mobile edge computing environments. Based on a real-world hybrid dataset, our experimental results show that prediction accuracy outperforms other approaches in mobile edge computing environments. In this paper, our approach initially calculates user or edge server similarity depending upon users' changing locations, selects the Top-K most-similar neighbors to decrease data volatility, and finally makes service recommendation based on QoS prediction.

Although our approach improves service recommendation accuracy in mobile edge computing, it possesses a few limitations: 1) the prediction accuracy of our approach decreases when the matrix density is less than 10%. 2) The similarity computation between edge servers might produce a large error when the distribution density of edge servers is sparser. Thus, our future work will focus on solving these two limitations. We will consider on the anticipatory network model [43] related to user mobility prediction. For example, including a predictive migration model into an anticipatory network may increase the degree of adaptability of the overall service recommendation in the network in our future work.

To provide further clues regarding the QoS and how the physical mobility data correlate with the Telecom data, our future work will investigate the statistics of transition patterns of users migrating between the edge stations based on traffic and public transport data. Another topic worth further research appears cognitive aspects of service recommendation [13], such as impact of time when the service is recommended after entering the new edge station, as well as patterns and probabilities of rational vs. non-rational user reactions to service recommendations.

## ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation of China (Grant No. 61472047). The helpful suggestions from the anonymous reviewers are gratefully acknowledged, and I also offer thanks to Lubao Wang whose discussion and comment helped greatly.

### REFERENCES

[1] A. Ahmed and E. Ahmed, A survey on mobile edge computing, in: 10th International Conference on Intelligent Systems and Control (ISCO), 2016, pp. 1-8.

[2] L. Chiaraviglio, F. Cuomo, A. Gigli, M. Maisto, Y. Zhou, Z. Zhifeng, Z. Honggang, A reality check of Base Station Spatial Distribution in mobile networks, in: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016, pp. 1065-1066

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, Edge Computing: Vision and Challenges, IEEE Internet of Things Journal, 5(2016) 637-646.

[4] S. Tekinay, B. Jabbari, Handover and channel assignment in mobile cellular networks, IEEE Communications Magazine, 29 (1991) 42-46.

[5] R. Arshad, H. ElSawy, S. Sorour, T.Y. Al-Naffouri, M.-S. Alouini, Handover management in dense cellular networks: A stochastic geometry approach, arXiv preprint arXiv:1604.08552, (2016).

[6] Z. Zheng, H. Ma, M.R. Lyu, I. King, QoS-Aware Web Service Recommendation by Collaborative Filtering, IEEE Transactions on Services Computing, 4 (2011) 140-152.

[7] Y. Jiang, J. Liu, M. Tang, X. Liu, An effective web service recommendation method based on personalized collaborative filtering, in: 2011 IEEE International Conference on Web Services (ICWS), 2011, pp. 211-218.

[8] W. Lo, J. Yin, S. Deng, Y. Li, Z. Wu, Collaborative web service qos prediction with location-based regularization, in: 2012 IEEE International

Conference on Web Services (ICWS), 2012, pp. 464-471.

[9] N. Ekiz, T. Salih, S. Kucukoner, K. Fidanboylu, An overview of handoff techniques in cellular networks, International journal of information technology, 2 (2005) 132-136.

[10] Z. Zheng, H. Ma, M.R. Lyu, I. King, Qos-aware web service recommendation by collaborative filtering, Services Computing, IEEE Transactions on, 4 (2011) 140-152.

[11] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, H. Mei, Personalized qos prediction for web services via collaborative filtering, in: 2007 IEEE International Conference on Web Services (ICWS), 2007, pp. 439-446.

[12] X. Chen, Z. Zheng, Q. Yu, M.R. Lyu, Web service recommendation via exploiting location and QoS information, IEEE transactions on parallel and distributed systems, 25 (2014) 1913-1924.

[13] A.M.J Skulimowski, Congnitive Content Recommendation in Digital Knowledge Repositories – a Survey of Recent Trends, in: 2017 International Conference on Artificial Intelligence and Soft Computing (ICAISC), 2017

[14] Y. Ma, S. Wang, P.C.K. Hung, C.H. Hsu, Q. Sun, F. Yang, A Highly Accurate Prediction Algorithm for Unknown Web Service QoS Values, IEEE Transactions on Services Computing, 9 (2016) 511-523.

[15] Z. Zheng, M.R. Lyu, Collaborative reliability prediction of service-oriented systems, in: 2010 ACM/IEEE International Conference on Software Engineering, 2010, pp. 35-44.

[16] X. Chen, X. Liu, Z. Huang, H. Sun, Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation, in: 2010 IEEE International Conference on Web Services (ICWS), 2010, pp. 9-16.

[17] S. Wang, Y. Ma, B. Cheng, y. f, R. Chang, Multi-Dimensional QoS Prediction for Service Recommendations, IEEE Transactions on Services Computing (2016).

[18] Z. Zheng, H. Ma, M.R. Lyu, I. King, Wsrec: A collaborative filtering based web service recommender system, in: 2009 IEEE International Conference on Web Services (ICWS), 2009, pp. 437-444.

[19] Z. Zheng, Y. Zhang, M.R. Lyu, Distributed qos evaluation for real-world web services, in: 2010 IEEE International Conference on Web Services (ICWS), 2010, pp. 83-90.

[20] Z. Zheng, H. Ma, M.R. Lyu, I. King, Collaborative web service qos prediction via neighborhood integrated matrix factorization, IEEE Transactions on Services Computing, 6 (2013) 289-299.

[21] V.W. Zheng, B. Cao, Y. Zheng, X. Xie, Q. Yang, Collaborative Filtering Meets Mobile Recommendation: A User-Centered Approach, in: AAAI, 2010, pp. 236-241.

[22] C. Zhang, L. Zhang, G. Zhang, QoS-Aware Mobile Service Selection Algorithm, Mobile Information Systems, 2016 (2016).

[23] R. Verma, A. Srivastava, A novel web service directory framework for mobile environments, in: 2014 IEEE International Conference on Web Services (ICWS), 2014, pp. 614-621.

[24] L. Wang, Q. Sun, S. Wang, Y. Ma, J. Xu, J. Li, Web Service QoS Prediction Approach in Mobile Internet Environments, in: 2014 IEEE International Conference on Data Mining Worjshop (ICDWM), 2014, pp. 1239-1241.

[25] W.-S. Soh, H.S. Kim, QoS provisioning in cellular networks based on mobility prediction techniques, IEEE Communications Magazine, 41 (2003) 86-92.

[26] S.H. Shah, K. Nahrstedt, Predictive location-based QoS routing in mobile ad hoc networks, in: 2002 IEEE International Conference on Communications (ICC), 2002, pp. 1022-1027.

[27] A. Samba, Y. Busnel, A. Blanc, P. Dooze, G. Simon, Throughput Prediction in Cellular Networks: Experiments and Preliminary Results, in: CoRes 2016, 2016.

[28] S. Wang, Z. Zheng, Z. Wu, M.R. Lyu, F. Yang, Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation Systems, IEEE Transactions on Services Computing, 8 (2015) 755-767.

[29] M. You, X. Xin, W. Shangguang, L. Jinglin, S. Qibo, Y. Fangchun, QoS evaluation for web service recommendation, China Communications, 12 (2015) 151-160.

[30] L. Chiaraviglio, F. Cuomo, A. Gigli, M. Maisto, Y. Zhou, Z. Zhao, H. Zhang, A reality check of base station spatial distribution in mobile networks, in: IEEE Infocom, 2016.

[31] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: 2001 ACM international conference on World Wide Web, 2001, pp. 285-295.

[32] T. Hofmann, Latent semantic models for collaborative filtering, ACM Transactions on Information Systems, 22 (2004) 89-115.

[33] J. Benesty, J. Chen, Y. Huang, I. Cohen, Pearson correlation coefficient, in: Noise reduction in speech processing, Springer, 2009, pp. 1-4.

[34] P. Sedgwick, Pearson's correlation coefficient, Bmj, 345 (2012).

[35] S. Patro, K.K. Sahu, Normalization: A Preprocessing Stage, arXiv preprint arXiv:1503.06462, (2015).

[36] M.R. McLaughlin, J.L. Herlocker, A collaborative filtering algorithm and evaluation metric that accurately model the user experience, in: 2004 international ACM SIGIR conference on Research and development in information retrieval, 2004, pp. 329-336.

[37] B. Shumaker, R. Sinnott, Astronomical computing: 1. Computing under the open sky. 2. Virtues of the haversine, Sky and telescope, 68 (1984) 158-159.

[38] S. Ramachandran, O. Deshpande, C.C. Roseman, N.A. Rosenberg, M.W. Feldman, L.L. Cavalli-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, 102 (2005) 15942-15947.

[39] F. Castanedo, A review of data fusion techniques, The Scientific World Journal, 2013 (2013).

[40] B. Khaleghi, A. Khamis, F.O. Karray, S.N. Razavi, Multisensor data fusion: A review of the state-of-the-art, Information Fusion, 14 (2013) 28-44.

[41] Y. Bar-Shalom, P.K. Willett, X. Tian, Tracking and data fusion, YBS publishing, 2011.

[42] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: 1998 conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 1998, pp. 43-52.

[43] A.M. Skulimowski, Anticipatory network models of multicriteria decision-making processes, International Journal of Systems Science, 45 (2014) 39-59.

**Shangguang Wang** is an associate professor at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT). He received his Ph.D. degree at BUPT in 2011. He has co-authored more than 100 papers, and played a key role at many international conferences and workshops, such as General Chair and TPC Chair. His research interests include Service Computing and Cloud Computing. He is a Senior Member of the IEEE.

**Yali Zhao** received bachelor's degree in computer science and technology from Shandong University, in 2013. Currently, she is a Master Degree Candidate at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests include service computing and edge computing.

**Lin Huang** received the M.E. degree in computer science and technology from the Institute of Network Technology, Beijing University of Posts and Telecommunications, in 2012. Currently, she is a Ph.D. candidate at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests include Reputation measurement, Web service selection.

**Jinliang Xu** received the bachelor's degree in electronic information science and technology from Beijing University of Posts and Telecommunications in 2014. Currently, he is a Ph.D. candidate in computer science at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include Service Computing, Information Retrieval, and Crowdsourcing.

**Ching-Hsien Hsu** is a professor in the department of computer science and information engineering at Chung Hua University, Taiwan. His research includes high performance computing, cloud computing, parallel and distributed systems, and ubiquitous/pervasive computing and intelligence. He has been involved in more than 100 conferences and workshops as various chairs and more than 200 conferences/workshops as a program committee member. He is the editor-in-chief of an international journal on Grid and High Performance Computing and has served on the editorial board for approximately 20 international journals.