Journal of Universal Computer Science, vol. 30, no. 3 (2024), 383-401
submitted: 12/5/2023, accepted: 30/10/2023, appeared: 28/3/2024 *CC BY-ND 4.0*

An SVR-based and Location-aware Method for Mobile QoS Prediction

Lifang Ren
(School of information, Shanxi University of Finance and Economics, Taiyuan 030006, PR China
https://orcid.org/0000-0003-2204-1607, renlf@sxufe.edu.cn)

Jing Li
(School of Computer and Information Technology, Shanxi University, Taiyuan 030006, PR China
https://orcid.org/0009-0009-5594-9153, 946574233@qq.com)

Wenjian Wang
(School of Computer and Information Technology, Shanxi University, Taiyuan 030006, PR China
https://orcid.org/0000-0002-8276-3639, wjwang@sxu.edu.cn, corresponding author)

Abstract: With the rapid development of intelligent mobile communication technology, the number of mobile services and the number of mobile users are both continuously increasing. So, the services used by a user can only account for a very small proportion of the existing services, which results in a sparse user-service quality of service (QoS) matrix. However, QoS is critical for service selection and service recommendation. Therefore, predicting the unknown values of the sparse QoS matrix is essential. However, due to the sparsity of QoS data, the QoS prediction accuracy is difficult to improve. Faced with the problem, this paper intends to utilize the outstanding generalization ability and only support vectors dependent property of support vector regression (SVR) to overcome the difficulty brought by the sparsity of data and predict the unknown QoS more accurately. Moreover, it is evident that in the mobile environment, QoS values are closely related to the locations of the invoking users. Therefore, this paper intends to improve the accuracy of QoS prediction by incorporating not only the information of similar users but also the information of nearby users into feature vectors. On the other hand, the known QoS values of nearby users can be used to roughly estimate the unknown QoS values of the cold-start user, so as to alleviate the cold-start problem to some extent. Thus, a location-aware SVR-based method for QoS prediction (SVR4QP) is proposed. Compared with some classical QoS prediction algorithms, the experimental results show that in $1/3$ of the cases, SVR4QP is moderate; in $1/6$ of the cases, SVR4QP is suboptimal; and in half of the cases, SVR4QP is optimal. Compared with some novel mobile QoS prediction methods, the experimental results show that in $1/4$ of the cases, SVR4QP is moderate; in half of the cases, SVR4QP is suboptimal; and in $1/4$ of the cases, SVR4QP is optimal. All these indicate that SVR4QP has comparatively more accurate mobile QoS prediction.

Keywords: Mobile service, QoS prediction, Location-aware, Support vector regression
Categories: D.2
DOI: 10.3897/jucs.106314

1 Introduction

With the development of mobile communication, mobile applications have become more and more popular. Nowadays, users can use smartphones or other mobile terminal devices to access mobile services anytime, anywhere [Dinh et al. 2013]. Furthermore, the number of devices with service delivery capability is continuously increasing, which indicates that the number of mobile services will be more than that of Web services [Deng et al. 2018]. Moreover, compared with the number of total services, the number of services a user may use is far smaller. Hence for a user, the QoS(quality of service) of most services is unknown. In service selection and service recommendation, QoS prediction plays an important role in providing users with services that can adequately satisfy their QoS requirements. However, in the mobile environment, the network signals differ in different locations. So the QoS experience of a user may be different when invoking the same service at various locations. All these make QoS prediction in the mobile environment meaningful and challenging.

Many research efforts have been devoted to QoS prediction in the mobile environment. The existing studies are mainly CF(collaborative filtering)-based, which can roughly be divided into memory-based and model-based. For example, Wang *et al.* in [Wang et al. 2017] use a memory-based method, Li *et al.* in [Li et al. 2019] and Chen *et al.* in [Chen et al. 2020] both use a model-based method, namely matrix factorization(MF). Although CF-based mobile QoS prediction methods have become dominant, some defects are exposed. For the memory-based methods, QoS values of similar users and similar services are essential for prediction. However, the newcomers have no historical QoS record, so their similar individuals could not be found, i.e., the so-called cold-start problem. In addition, for model-based methods, particularly, the MF-based and tensor factorization (TF)-based method, in order to make the prediction for an unknown QoS value, which needs to fill all the unknown values in the sparse user-service QoS matrix. But, in practice, the user-service QoS matrix is extremely sparse and massive. Hence, MF-based methods are prone to suffer from dimensional disasters.

Due to the fact that the QoS to be predicted are continuous values, QoS prediction problems can essentially be regarded as machine learning regression problems. Additionally, QoS values are not independent of each other, hence, QoS prediction problems are suitable to be modeled as support vector regression (SVR) models [Zhou 2016]. SVR is a regression learner that has outstanding generalization ability, so it is expected that the learned model will have more accurate predictions. Moreover, SVM can learn only based on a few key samples known as support vectors. Hence, the dimension disaster brought by the huge number of users and services may often be avoided. Furthermore, because of the location-sensitive characteristics of mobile service QoS, nearby individuals may have similar QoS, so this paper introduces the location information of both users and services into the QoS prediction model. Thus, this paper considers not only the similar information between users but also the location information of users and services, in an effort to get more accurate QoS prediction. In addition, for the new users and services who have no historical QoS data, their similar individuals could not be found, which brings the cold-start problem to the similarity-based CF methods. However, based on the location sensitivity of mobile QoS, this paper thinks nearby users may receive similar strength of network signal, so they are likely to have similar QoS experiences. Therefore, according to the location of the cold-start user, his/her nearby users can be found. Thus, we can utilize nearby users' historical QoS data to predict the QoS values for the cold-start users, so as to alleviate the cold-start problem to some extent. Consequently, this paper aims at a more effective and more accurate mobile QoS prediction learner noted by SVR4QP (SVR for QoS prediction). The contributions of this paper are summarized as follows.

- The mobile internet is an open environment, users and services are diverse, which poses challenges to QoS prediction. Fortunately, the base learner of SVR4QR, i.e. SVR, has outstanding generalization ability, which makes SVR4QP more adaptable to the diversity of users and services.
- SVR can learn only depending on the support vectors, whose numbers are far less than the overall samples. Hence, it can avoid the dimension disaster brought by numerous users and services, which makes SVR4QP a more effective model.
- In mobile environments, users with close geographical locations commonly receive similar strength of network signal, often indicating similar QoS. Hence, SVR4QP introduces the location information to feature vectors, which allows SVR4QP to better adapt to the location-sensitive characteristic of QoS values in mobile environments, to make more accurate mobile QoS predictions.
- The mobile internet constantly attracts new users and services, so cold-start problems are common. Based on nearby users having similar QoS experiences, SVR4QP finds nearby users or nearby services for cold-start individuals and uses the known QoS values of nearby users or services to roughly estimate the QoS of cold-start individuals, so as to alleviate the cold-start problem.

The remainder of this paper is organized as follows. Section 2 briefly introduces the background knowledge of SVR4QP, including the CF algorithms and the SVR model. Section 3 presents the SVR4QP model, including feature vector extraction, model training, and SVR4QP algorithm. Section 4 describes the experimental results and discussion. Section 5 overviews the related work, and Section 6 concludes the paper.

2 Preliminary

In this section, the background of SVR4QP, including the fundamental of CF and the mechanism of SVM, are briefly introduced.

2.1 Fundamentals of CF

CF has become a chief technology in the recommendation field and is widely used in e-commerce recommendation systems. CF is a recommendation algorithm based on the prediction of users' ratings of services, which is mainly grouped into memory-based CF and model-based CF.

According to the historical rating data, memory-based CF calculates the similarity between users and between items at first and then predicts the active user's rating of the active item based on the active user's historical ratings of the similar items of the active item and the historical ratings of the active user's similar users of the active item. It can be seen that the measurement of similarity is the critical step of the memory-based CF algorithm [Manochandar and Punniyamoorthy 2020]. The most commonly used similarity measures include the Pearson correlation coefficient, Euclidean distance, and cosine similarity.

Based on the historical rating data, model-based CF trains a model that can learn the active user's potential preference pattern and then uses the model to predict the active user's rating of the active item. The commonly used models mainly include matrix factorization-based model [Yang et al. 2021], tensor factorization-based [Neira et al.

2021], and SVM-based models [Ren and Wang 2018], deep neural networks[Yu et al. 2022], etc.

Commonly, CF considers users' historical behaviors and takes advantage of group intelligence, so the recommendation is highly personalized. SVR4QP proposed in this paper is also CF-based, and more specifically, SVR4QP is a model-based CF prediction method. In particular, this paper pays more attention to the location information of users and services, so it is more suitable for mobile computing environments.

2.2 Mechanism of SVR

SVM can be used not only for classification but also for regression, and the SVM used for regression is called SVR. The purpose of regression is to obtain a regression hyperplane $f(x)$ which fits the training set samples as well as possible. Commonly, a loss function is constructed to measure the difference between the predicted values and the sample labels. And, the model $f(x)$ can be determined by minimizing the loss function. SVR sets a margin along the regression hyperplane $f(x)$ with a spacing ε, and the samples falling into the margin are considered correct, so their loss is neglected. An illustration of the linear regression support vector machine is shown in Fig. 1, where $f(x) = w^{\mathrm{T}}x + b$ is the regression hyperplane of the regression model.

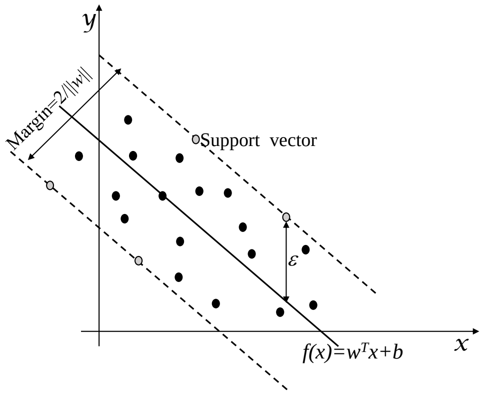


Figure 1: *An illustration of SVR.*

Traditional regression methods consider the prediction results correct if and only if $f(x_i)$ is exactly equal to y_i. However, SVR thinks the prediction results are accurate as long as the difference between $f(x_i)$ and y_i is not too large. Therefore, SVR can be formed as the equation (1).

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 \\ &s.t. \quad |y_i - f(x_i)| < \varepsilon, \forall i \end{aligned} \tag{1}$$

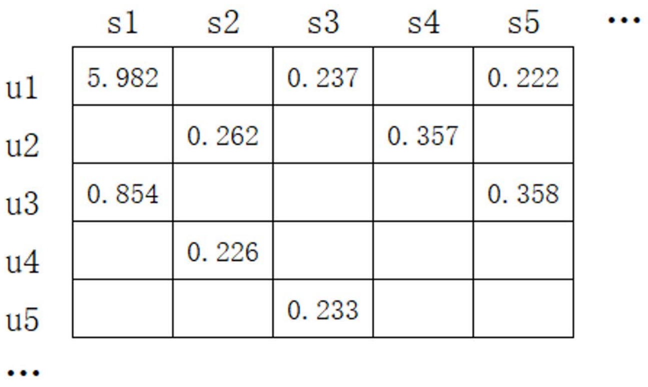
Where ε is the threshold for evaluating whether the regression results are correct. More generally, for nonlinear regression problems, the kernel function can be used to map the data from low-dimensional space to high-dimensional space, so as to realize linear regression in high-dimensional space.

Because statistical learning theory provides SVM with a solid theoretical basis, SVR also shows desirable characteristics in regression. The main advantages of SVR include: (1) SVR remains effective when the dimension of features is greater than the number of samples; (2) SVR can avoid falling into local-optimal solutions, and obtain the global-optimal solution in theory, which will ensure high accuracy; (3) SVR can map a nonlinear problem to a high-dimensional feature space, so as to solve the problem with linear methods, which will ensure its strong generalization ability. Given these advantages, this paper intends to use an SVR-based regression model to solve the mobile QoS prediction problem.

3 Method of SVR4QP

3.1 Problem formulation

As mentioned before, the QoS matrix is sparse. So, the goal of QoS prediction is to determine the unknown value which helps to infer the QoS that the active user may experience by invoking the target service. For example, Fig. 2 is a user-service QoS matrix, denoted by matrix R. Here, for the convenience of explanation, only the first few rows and columns of matrix R were taken, and the sparsity is much smaller than the actual QoS matrix. In this paper, we take the data of response time as the QoS attribute values, and the other QoS attributes are similar to the response time. In matrix R, a row represents a user, and a column represents a service. Hence, the element r_{ij} represents the response time of service s_j used by user u_i. The goal of QoS prediction is to predict the unknown values according to the existing values in the QoS matrix.



	s1	s2	s3	s4	s5	•••
u1	5. 982		0. 237		0. 222	
u2		0. 262		0. 357		
u3	0. 854				0. 358	
u4		0. 226				
u5			0. 233			
•••						

Figure 2: *User-service response time matrix.*

In this paper, the QoS prediction problem will be modeled as an SVR i.e. SVR4QP. Hence, we want to learn a regression hyperplane $f(r) = f(\omega r^{\top} + b)$ which fits the historical QoS data well. Where, r is the feature vector formed by the QoS values of similar users and nearby users, which will be illustrated in Section 3.3; ω and b are parameters of SVR4QP to be learned. If the model is exactly fitted to the data, it often leads to the problem of overfitting. Hence, in practice, the prediction results are considered accurate as long as the absolute error between predicted and real values is not too large. Therefore, SVR4QP aims to find the optimal regression hyperplane that satisfies equation (2).

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\Sigma_{i=1}^{m} l_\epsilon(y_i - f(\omega r^\top + b) \tag{2}$$

Where y_i is the known QoS value representing the real value; $C > 0$ is the penalty factor, and a greater C means a greater penalty on the regression mistakes. Where $l_\epsilon(z)$ is an indicative function defined as equation (3).

$$l_\epsilon(z) = \begin{cases} 0, & if|z| < \varepsilon \\ |z| - \varepsilon, & \text{otherwise} \end{cases} \tag{3}$$

3.2 Calculation of similarity and distance

The theoretical foundation of CF believes that user u and his similar user v have similar QoS values for the same service s. Moreover, considering that nearby users are close in geographic position, who access the mobile internet may be possible through the same base station, this paper thinks the QoS values of the same service s invoked by user u and his nearby users are similar. Therefore, the introduction of the QoS of the nearby users can help to make a more accurate prediction. Therefore, this paper utilizes the information of similar users and nearby users to predict the unknown QoS values.

As shown in equation (4), in this paper, the Pearson correlation coefficient is used to measure the similarity between every two users. Where $I = I_u \bigcap I_v$ represents the set of services invoked by both user u and user v, $r_{u,i}$ represents the QoS value of service i invoked by user u, and $\bar{r}_u$ represents the average QoS value of all services invoked by user u.

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

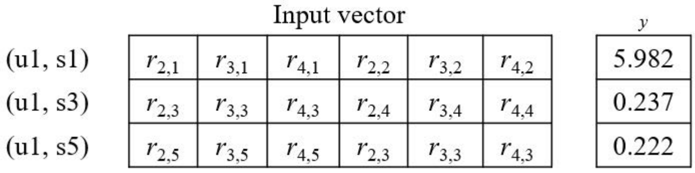
In this paper, the locations of users are used to find the nearby users for an active user. To determine the proximity relationship, it is necessary to calculate the distance between users. And the distance between user u and user v can be measured by Euclidean distance, which can be calculated by equation (5). Where lon$(\cdot)$ and lat$(\cdot) \in (-180, 180]$ represent the longitude and latitude of a location, respectively, and c=111261 is a constant converting the unit of distance measurement unit to meters.

$$d_{u,v} = c \cdot \sqrt{(\text{lon}(u) - \text{lon}(v))^2 + (\text{lat}(u) - \text{lat}(v))^2} \tag{5}$$

3.3 Feature extraction

Consequently, the feature vector can be formed by the information of similar and nearby individuals. Through an example, the following describes how to extract the feature vectors. In the user-service model shown in Fig. 2, suppose u_2 is a nearby user of u_1; u_3 and u_4 are similar users of u_1; s_2 is a nearby service of service s_1; s_4 is a nearby service of s_3; and s_3 is a nearby service of s_5. According to the given nearby and similar information, the known data of user u_1 and their corresponding feature vectors are shown in

Fig. 3. In Fig. 3, $r_{i,j}$ represents the element of user-service QoS matrix on the ith row and the jth column, i.e. QoS values of service s_j invoked by user u_i. Thus, to train the regression model, for the known QoS values, we need to extract their feature vectors. Such as $r_{1,3} = 0.237$, regression value y of (u_1, s_3), the feature vector is formed by the QoS values of u_1's nearby user u_2 invoking s_3, i.e. $r_{2,3}$, u_1's similar users $r_{3,3}$ and $r_{4,3}$ invoking s_3, i.e. $r_{3,3}$ and $r_{4,3}$, and the information of the nearby user and similar users invoking service s_3's similar service s_4, i.e. $r_{2,4}$, $r_{3,4}$ and $r_{4,4}$. Hence, as shown in Fig. 3, the feature vector of (u_1, s_3) is $(r_{2,3}, r_{3,3}, r_{4,3}, r_{2,4}, r_{3,4}, r_{4,4})$.



	Input vector						y
(u1, s1)	$r_{2,1}$	$r_{3,1}$	$r_{4,1}$	$r_{2,2}$	$r_{3,2}$	$r_{4,2}$	5.982
(u1, s3)	$r_{2,3}$	$r_{3,3}$	$r_{4,3}$	$r_{2,4}$	$r_{3,4}$	$r_{4,4}$	0.237
(u1, s5)	$r_{2,5}$	$r_{3,5}$	$r_{4,5}$	$r_{2,3}$	$r_{3,3}$	$r_{4,3}$	0.222

Figure 3: *An example of known data and corresponding feature vectors.*

Due to the sparsity of the user-service matrix, the feature vectors extracted are also sparse, which will lower the accuracy of QoS prediction. In order to overcome the disadvantage, this paper replaces the unknown values in feature vectors with the weighted average of the means of the active user's and active service's known QoS values. That is, if $r_{u,s}$, an element of a feature vector, is unknown, it will be replaced by the average value calculated by equation (6).

$$r_{u,s} = w \cdot \bar{u} + (1 - w) \cdot \bar{s} \tag{6}$$

Where $\bar{u}$ and $\bar{s}$ are the average value of user u's and service's s known QoS values, respectively, and w is the weight allocated to users, which is set to 0.5 in this paper.

As for cold-start users, they do not have historical QoS records, so their similar users could not be found through historical QoS data. However, the geographical location of a cold-start user is easy to obtain. Hence, his/her nearby user can be found. Because nearby users may receive similar strength of network signal, they are likely to have similar QoS experiences. Therefore, historical QoS data of nearby users can be utilized to make a rough prediction for the new user. For example, in the user-service model shown in Fig. 2, if a new user, suppose u_6, and he/she has nearby users u_1 and u_2. User u_6 wants to use service s_3, the QoS $r_{6,3}$ can be predicted by equation (7).

$$\widehat{r}_{6,3} = (r_{1,3} + r_{2,3})/2 \tag{7}$$

In this way, based on the rough QoS predictions, the cold-start user can select the appropriate services. After invoking some services, QoS records are gradually accumulated, and more accurate predictions will be obtained by utilizing the proposed SVR4QP. Similarly, the QoS of cold start services can be roughly predicted. After some actual QoS data are accumulated, SVR4QP can be used for more accurate QoS prediction.

3.4 Model training

After feature extraction, we can obtain the training set of the active user. In practice, too small ε can not ensure all samples fall into the margin, whereas too large ε will bring the bias of regression hyperplane caused by some outliers. Hence, SVR4QP allows each sample to have a relaxation variable ξ_i, which is used to describe the deviation of the sample (r_i, y_i) to the margin. SVR needs not only to maximize the margin but also to minimize the loss. And, the parameters w and b are determined in the optimization process. Thus, the regression hyperplane will serve as the solution to the optimization problem expressed in (8).

$$\begin{aligned} &\min_{w,b} \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i^u + \xi_i^d) \\ s.t.\ &-\varepsilon - \xi_i^d \leq y_i - (\omega^\top r_i + b) \leq \varepsilon + \xi_i^u, \forall i \\ &\xi_i^u \geq 0, \xi_i^d \geq 0, \forall i \end{aligned} \tag{8}$$

Where ξ_i^u and ξ_i^d are the upper bound and the lower bound of the relaxing variable, respectively.

However, in practice, prediction problems are not necessarily linear. In this case, a mapping $\phi(x_i)$ is usually adopted to map the sample space to a higher dimensional space. Then, a kernel function can be used to avoid the inner product operation in high-dimension space in the dual problem of the objective function. Thus, the regression hyperplane can be expressed as shown in equation (9).

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i k(x, x_i) + b \tag{9}$$

Where, α_i is the Lagrange multiplier corresponding to the support vector x_i; and $k(x, x_i) = \phi(x) \cdot \phi(x_i)$ is the kernel function.

In this paper, the Gaussian kernel is adopted as the kernel function, the penalty factor is set as $C = 1$, and the threshold is set as $\varepsilon = 1.0$.

3.5 Solution to SVR4QP

In order to solve the QoS prediction problem, this paper intends to use a regression model to learn the potential patterns hidden in the QoS data. In machine learning, the commonly used regression models include logistic regression, decision tree regression, support vector regression, etc. Logistic regression models require that the features are independent of each other. However, the QoS values are interrelated. Thus logistic regression models are not applicable. For decision tree regression, in practice, it is almost impossible to solve a problem effectively with a single regression tree. Therefore, it is necessary to improve the regression tree with ensemble learning. However, ensemble learning will bring a large amount of calculation. Therefore, this paper adopts SVR to learn the potential patterns with historical QoS data.

SVR4QP extracts feature vectors from the QoS historical record first, then trains the model through an SVR. The main steps of SVR4QP are shown in algorithm 1.

Algorithm 1 SVR-based Qos prediction algorithm

```
Input: User-service QoS matrix $R$;
    The location of each service $(lon(s), lat(s))$, and the location of the users at each invoking $(lon(u), lat(u))$;
    The current location of the active user $(lon(u), lat(u))$.
Output: The predicted QoS values $r_{i,j}$.
1: According to equation (4), calculate the similarities between the active user and the other users and find the top-k similar users to form the similar user set.
2: According to equation (5), calculate the distances between the active user and the other users and find the top-m closer users to form the neighbor user set.
3: According to equation (5), calculate the distances between the active service and the other services and find the top-n closer services to form the neighbor service set.
4: According to the method described in Section 3.3, and based on the users and services selected in previous steps 1, 2, and 3, extract the feature vectors from $R$.
5: Based on the feature vectors, train an SVR model.
6: Using the trained SVR, predict the unknown QoS values.
```

4 Experimental results and discussion

In order to test the performance of SVR4QP, the following experiments were conducted. The experiment environment and dataset are described first. Secondly, the evaluation metrics are set. Then the influences of parameters in the model are studied. Finally, comparison experiments were conducted with some classic algorithms.

4.1 Experiment environment and dataset

The experiments were conducted by Python 3.7. And, the experimental platform runs Windows 10 with an Intel(R) Core(TM) i7-7700 3.60GHz CPU with 16GB RAM. The experimental data in this paper use the real-world QoS Data WS-DREAM [1] dataset 2 released by [Zheng et al. 2010, Zhang et al. 2011]. This dataset collects QoS records of 5825 services from 339 users, including response time and throughput. Besides, the dataset also contains user location information (including the country, the longitude, and the latitude, etc.) and service location information (including the IP of the server, the country, the longitudes, and the latitudes, etc.). The response time data in the dataset is used as the historical QoS data, which is a matrix with a dimension of 339×5825. The purpose of SVR4QP is to predict the unknown values in the historical QoS record. Therefore, this paper randomly deletes some data from the response time matrix to simulate the sparsity of actual QoS data.

This paper makes statistics on the information of WS-DREAMS database 2, and Table 1 shows the statistics including the number of users, the number of services, the number of countries where the users are located, the number of services where the services are deployed, and the average response time.

[1] http://wsdream.com/dataset.html.

Statistical information	value
number of users	339
number of services	5825
number of servers	1021
number of user countries	31
average of response time	1.43s

Table 1: *Statistical information of database.*

Table 2 lists the top-5 countries with the largest number of users. It can be seen from Table 2 that nearly half of the 339 users are located in the United States. Among the 31 countries, the total proportion of users in the top five countries exceeds 70%, which shows that the distribution of users is uneven. Table 3 lists the top 5 servers with the most services. It can be seen from table 3 that the 5825 services are deployed on 1021 servers; The number of services deployed on the top-2 servers accounts for about 20%; The total number of services on the top-5 servers accounts for about 61%. Compared with the distribution of users, the distribution of services is relatively balanced.

Rank	Country	Number of users	Percentage
1	USA	161	47.49%
2	Germany	41	12.09%
3	Japan	16	4.72%
4	Canada	12	3.54%
5	Poland	12	3.54%

Table 2: *Statistics of the top-5 countries with the most users.*

Rank	IP	Number of servers	Percentage
1	96.49.108.226	234	22.92%
2	130.88.98.239	204	19.98%
3	173.201.44.188	71	6.95%
4	70.62.105.158	60	5.78%
5	216.248.159.211	59	5.77%

Table 3: *Statistics of the top-5 servers with the most services.*

4.2 Evaluation metrics

In order to measure the difference between the predicted values and the real values, this paper uses mean absolute error (MAE), root mean square error (RMSE), and normalized mean absolute error (NMAE) are respectively calculated as follows.

$$\text{MAE} = \frac{\sum_{u,i} |rt(u,i) - \widehat{rt}(u,i)|}{N} \tag{10}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i}(rt(u,i) - \widehat{rt}(u,i))^2} \tag{11}$$

$$\text{NMAE} = \frac{N \cdot \text{MAE}}{\sum_{u,i} rt(u,i)} \tag{12}$$

Where N is the total number of predicted QoS values, $rt(u,i)$ represents the actual QoS values of user u invoking service i, and $\widehat{rt}(u,i)$ represents the predicted QoS value of user u invoking service i. The smaller MAE, RMSE, and NMAE mean the smaller differences between the predictive and actual values, that is, the higher the prediction accuracy.

4.3 Parameters Tuning

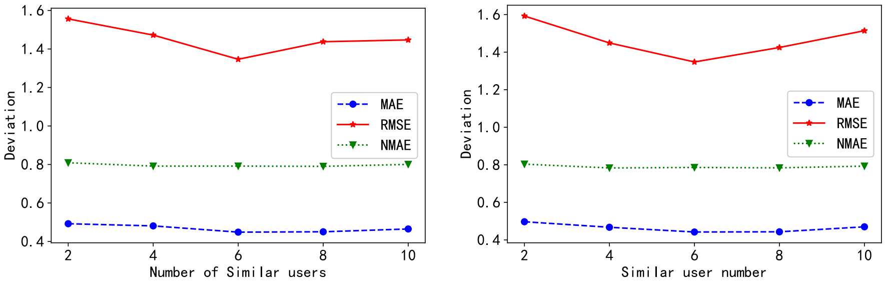
In order to achieve the optimal performance of the model, the parameters of SVR4QP, including the number of similar users, the number of nearby users, and the number of nearby services, are selected by experiments.

4.3.1 Influence of the number of similar users

The different numbers of similar users lead to unequal numbers of features extracted from the user-service QoS matrix, resulting in various training models, which will probably lead to different prediction accuracies. When the number of similar users is too small, maybe some QoS values positively correlating with the QoS values to be predicted will probably result in lower accuracy of QoS prediction. When the number of similar user services is too large, some users' QoS values that have nothing to do with or even a negative correlation with the QoS values of the active user will be put into the feature vector. Then, the prediction accuracy will likewise probably be reduced as well.

Commonly, there are two ways to select a similar user set for an active user. One is to determine a threshold first, and if the similarity between a user and the active user is greater than the given threshold, the user is added to the similar user set of the active user. Another is to sort the similarities between every user and the active user and then select the top n most similar users. The similar user numbers of different users vary greatly, so if the former is adopted, there will be a situation where some users have a large number of similar users, but some users cannot find any similar users. Therefore, the latter is adopted in this paper.

In order to obtain the highest accuracy prediction, Fig. 4 shows the experiments carried out to determine the number of similar users when the density of the response time matrix is 10% and 20%, respectively.



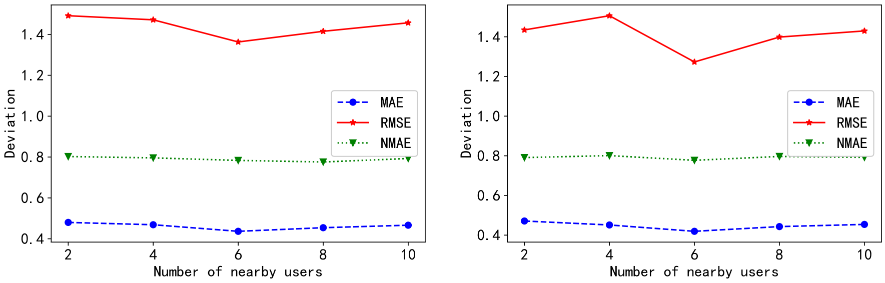
(a) density = 10%. (b) density = 20%.

Figure 4: *Determine the number of similar users.*

As can be seen from Figs. 4 (a) and (b), MAE and RMSE are greatly affected by the number of similar users, but NMAE changes little with the number of users. However, whether the matrix density is 10% or 20%, the three indicators show a downward trend when the number of similar services is less than 6 and an upward trend when it is greater than 6. Therefore, the minimum deviations are obtained when the number of similar users is 6, which illustrates the prediction accuracy is the highest when the number of similar users is 6. Hence, the number of similar users in the subsequent experiments is set to 6.

4.3.2 Influence of the number of nearby users

Likewise, different numbers of nearby users will result in various training models, ultimately bringing different accuracies of QoS prediction. Fig. 5 shows the experiments carried out to determine the number of nearby users when the density of the response time matrix is 10% and 20%, respectively.



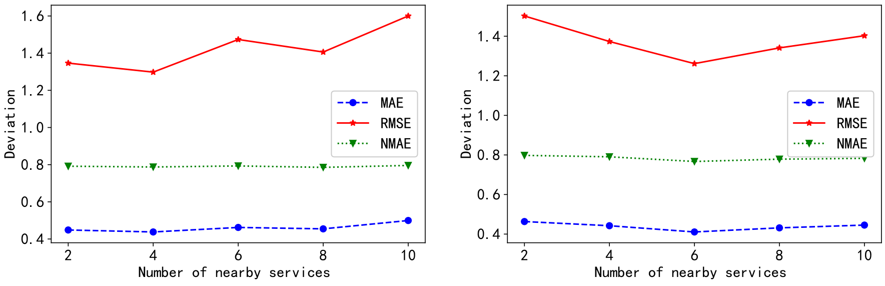
(a) density = 10%. (b) density = 20%.

Figure 5: *Determine the number of nearby users.*

Fig. 5 shows the effect of the number of nearby users on the accuracy when the matrix density is 10% and 20%. As can be seen from Fig. 5, MAE and RMSE are greatly affected by the number of nearby users, and the overall change of NMAE is small. However, no matter whether the matrix density is 10% or 20%, these three indicators generally show a downward trend when the number of nearby users is less than 6 and an upward trend when the number of nearby users is greater than 6. Therefore, when the number of nearby users is 6, the prediction deviations are the smallest and the accuracy is the highest. Hence, the number of nearby users in the subsequent experiments is set to 6.

4.3.3 Influence of the number of nearby services

Compared with the nearby services, similar services have little impact on the accuracy of QoS prediction, so this paper considers only the impact of the number of nearby services on the experimental results. Fig. 6 shows the influence of the number of nearby services on the prediction accuracy when the matrix density is 10% and 20%, respectively.



(a) density = 10%.

(b) density = 20%.

Figure 6: *Determine the number of nearby services.*

In Fig. 6, we can see that when the number of nearby services gradually increases from 2 to 10, NMAE changes little. As the matrix density is 10%, MAE and RMSE show a downward trend when the number of nearby services is less than 4 and an upward trend when it is greater than 4. That is, the minimum deviations are obtained when the number of nearby services is 4. As the matrix density is 20%, MAE and RMSE show a downward trend when the number of nearby services is less than 6 and an upward trend when the number is greater than 6. It illustrates that the minimum deviations are obtained when the number of nearby services is 6. Therefore, in subsequent experiments, when the matrix density is 10%, the number of nearby services is set to 4, and when the matrix density is 20%, the number of nearby services is set to 6.

Methods	**Density=10%**			**Density=20%**		
	MAE	**RMSE**	**NMAE**	**MAE**	**RMSE**	**NMAE**
UMEAN	0.5958	1.3998	1.0915	0.5966	1.2775	1.1403
IMEAN	0.6860	1.7173	1.2482	0.7032	1.6945	1.2744
UPCC	0.5410	1.2802	1.0750	0.4994	1.4077	0.8735
IPCC	0.6184	1.5018	1.1626	0.6055	1.4634	1.2712
UIPCC	0.5104	**1.2479**	0.8702	0.4992	**1.2160**	0.9170
NMF	0.6183	1.8957	1.0012	0.5614	1.2609	**0.7812**
NN	0.5655	1.4232	1.0324	0.5813	1.3433	1.1465
FM	0.5743	1.3486	-	0.5114	1.2503	-
SVR4QP	**0.4727**	1.3339	**0.7968**	**0.4634**	1.2838	0.8393

Table 4: *Comparison of MAE/RMSE/NMAE of different methods.*

4.4 Comparison experiments

4.4.1 Comparison with classical algorithms

In order to verify the effectiveness of SVR4QP, the prediction accuracies are compared between SVR4QP and the following classical algorithms.

UMean, the mean of users, this algorithm takes the average of QoS values of all services invoked by the active user as the prediction QoS.

IMean, mean of items, this algorithm takes the average of all historical QoS values of the active service as the prediction QoS.

UPCC, is a user-based collaborative filtering algorithm, which uses the PCC to find similar users first, then based on the QoS values of the active services invoked by similar users, calculates the prediction of the unknown QoS values [Jia et al. 2015].

IPCC, is an item-based collaborative filtering algorithm, that uses the PCC to find similar services at first, and then based on the QoS values of similar services invoked by the active user, calculates the prediction of the unknown QoS values [Zhang et al. 2009].

UIPCC, which is a user and item-based collaborative filtering algorithm, in essence, which through a balance factor combines the results of IPCC and IPCC as the prediction value [Kant and Mahara 2018].

NMF, a nonnegative matrix factorization algorithm, is an algorithm improved from matrix factorization, which requires the factors to be nonnegative [Zheng et al. 2013, Arora et al. 2016].

NN, is a neural network-based regression model for QoS prediction.

FM, is a factorization machine-based model proposed in the literature [Rendle 2012], which aims at dealing with the data sparsity problem in prediction.

Except for FM, we programmed the above algorithms using the optimal parameters set in the original paper. The results of FM are derived from the literature [Tang et al. 2019], where there are only 2 evaluation metrics, MSE and RMSE. Table 4 shows the experimental results of SVR4QP and the above-mentioned classic algorithms on the WS-Dream database 2.

It can be seen from table 4 that, as a whole, the prediction results of the model-based method (NMF, NN, FM, SVR4QP) are better than the memory-based algorithms (UPCC, IPCC, UIPCC) and far better than mean-based algorithms (UMean and IMean).

As can be seen from table 4, when the matrix density is 10%, compared with other methods, SVR4QP achieves the minimum MAE and NMAE. And, the MAE of SVR4QP is less than the secondary minimum MAE by 0.038, and the NMAE is less than the secondary minimum NMAE by 0.0734; UIPCC achieves the minimum RMSE, which is 0.086 lower than that of SVR4QP. When the matrix density is 20%, SVR4QP obtains the minimum MAE, which is 0.0358 lower than the secondary minimum MAE; UIPCC obtains the minimum RMSE, which is 0.0678 lower than SVR4QP; NMF obtains minimum NMAE, which is 0.058 lower than SVR4QP which obtains the secondary minimum NMAE. To sum up, in $1/3$ of the cases, SVR4QP is moderate; in $1/6$ of the cases, SVR4QP is suboptimal; and in half of the cases, SVR4QP is optimal. Therefore, in most cases, SVR4SR has the best prediction, which shows the effectiveness of location-aware SVR regression.

4.4.2 Comparison with mobile QoS prediction methods

To further evaluate the prediction accuracy of SVR4QP, experiments are conducted to compare SVR4QP with some mobile QoS prediction methods.

DEDP is a distributed edge differential privacy (DEDP) QoS prediction algorithm, which considering the difference of edge servers improved the user-based CF algorithm [Zhang et al. 2021]. The similarity rate is set to $\alpha = 0.7$.

RFMF is a reputation-aware and federated learning-based method to provide credible and privacy-preserving QoS prediction for mobile edge environments [Zhang et al. 2020].

EQoSP is a QoS prediction method for mobile edge environments, through cluster analysis, which naturally determines the numbers of similar users and similar services [Ren and Wang 2020]. The number of user clusters is $k_1 = 15$, and the number of service clusters is $k_2 = 7$.

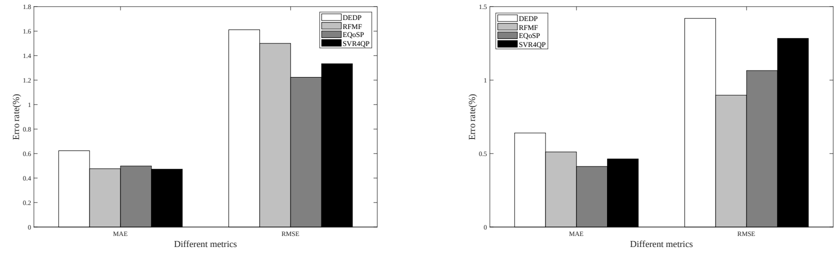
In this section, we compare the MAE and RMSE of QoS predictions when the matrix densities are 10% and 20%, and Fig. 7 shows the comparison results. From Fig. 7 we can see, as a whole, compared with these novel methods, the prediction accuracy of SVR4QP is qualified. Fig. 7(a) shows the MAE and RMSE of the different methods when the density of the QoS matrix is 10%. From Fig. 7(a), we can see when the QoS matrix is 10%, the MAEs of REMF, EQosP, and SVR4QP are about the same and better than DEDP, and SVR4QP is the best by a slight margin. With respect to the RMSE, SVR4QP is the suboptimal. Fig. 7(b) shows the MAE and RMSE of the different methods when the density of the QoS matrix is 20%. From Fig. 7(b), we can see the MAE of SVR4QP is a bit worse than the optimal, and the RMSE is slightly worse than the suboptimal. To sum up, in $1/4$ of the cases, SVR4QP is moderate; in half of the cases, SVR4QP is suboptimal; and in $1/4$ of the cases, SVR4QP is the best. Therefore, we think SVR4QP has comparatively better prediction accuracy in mobile environments.

5 Related work

So far, many methods have been proposed for personalized QoS prediction. The existing QoS prediction approaches are mainly based on the CF, which can be roughly divided into memory-based approaches and model-based approaches.

5.1 Memory-based approach

Memory-based methods assume that the QoS value of the active user invoking the active service is similar to the QoS values of the user's similar group invoking the active service, and is similar to the QoS values of the similar services invoked by the active users.



(a) density = 10%.

(b) density = 20%.

Figure 7: *Prediction accuracy comparison with novel methods.*

In such methods, researchers mainly modify the measurement method of similarity to select more similar users or services, so as to improve the accuracy of QoS prediction. Such as Wang *et al.* in [Wang et al. 2017] min-max normalizes the history QoS value at first and then uses PCC (Pearson correlation coefficient) to calculate the similarity of users. When computing PCC similarity, Liu and Chen in [Liu and Chen 2019] employ the entropy information of similar users or services to improve prediction accuracy.

Generally, the memory-based approach is theoretically easy to understand and implement, but, in practice, it is easy to encounter the difficulty of cold-start.

5.2 Model-based approach

Model-based methods learn the potential patterns hidden in historical QoS data and then construct a model to predict the unknown QoS data. For example, the commonly used matrix factorization methods [Li et al. 2019, Hernando et al. 2016] factorize the user-service QoS matrix into a product of two low-rank matrices, so some potential factors are learned to assist the prediction of unknown QoS values. Tang *et al.* in [Tang et al. 2019] combined the location information of users and services with the classic factorization machine to predict the unknown QoS of mobile services.

On the whole, compared to memory-based approaches, Model-based approaches can achieve higher accuracy. However, the matrix factorization-based methods need to change the sparse matrix into a dense matrix, especially in practice, the numbers of users and services are large, so the calculation generally takes a long time.

5.3 Mobile QoS prediction

In order to further improve the quality of QoS prediction, there have been many studies focusing on different aspects of mobile environments. For example, Zhang *et al.* in [Zhang et al. 2020] considering the privacy issues of QoS data and the existence of untrusted users, utilize federated learning techniques and developing reputation mechanisms to provide credible and privacy-preserving QoS prediction for mobile edge environments. Mohajer *et al.* in [Mohajer et al. 2022] pay more attention to green networking. In order to maximize the total power utility to meet QoS constraints, they propose a dynamic optimization model to minimize the overall energy consumption of the network and a multi-hop backhauling strategy to effectively use the existing infrastructure of small-cell networks for simultaneous dual-hop transmissions. Dong *et al.* in [Dong et al. 2023] find out that static backhaul infrastructures cannot control severe fluctuating network traffic, and based on the graph theory they propose an adaptive backhaul topology with the ability to adapt to different traffic patterns. Thus, they provide the possibility of effective channel allocation to each backhaul link to meet capacity and QoS demands. Mohajer *et al.* in [Mohajer et al. 2023] considering energy efficiency and fairness assurance propose a dynamic optimization model that maximizes the total uplink/downlink energy efficiency along with satisfying the necessary QoS constraints. Ren and Wang in [Ren and Wang 2020] find out that in the mobile edge computing environment, the mobility of users leads to the switch of access from one edge server to another. Hence, in [Ren and Wang 2020] the QoS of the user using a service in the new location is predicted according to the data provided by similar users and similar edge servers of the new edge server. Zhang and Jin in [Zhang et al. 2021] considering the user has strong mobility and real-time attribute of QoS values in mobile edge environment, propose a privacy protection oriented and Laplace noise-based QoS prediction approach.

Overall, such mobile QoS prediction methods focus not only on QoS but also on other aspects of mobile networks, such as privacy protection [Zhang et al. 2021, Zhang et al. 2020] and energy efficiency [Mohajer et al. 2022, Dong et al. 2023, Mohajer et al. 2023]. Some aim to accurately QoS values prediction [Zhang et al. 2021, Zhang et al. 2020, Ren and Wang 2020], while others aim to meet the user QoS constraints [Mohajer et al. 2022, Dong et al. 2023, Mohajer et al. 2023]. And generally speaking, they have improved the accuracy of predictions to some extent.

Different from the aforementioned approaches, this paper takes account of the influence of the users' and services' locations on the mobile QoS. In so doing, on the one hand, the location information can be used to improve the accuracy of mobile QoS prediction, on the other hand, it can be beneficial to the QoS prediction of cold start users and services.

6 Conclusion

Accurate QoS prediction is a prerequisite for service selection and service recommendation. And, in the mobile environment, QoS values are sensitive to the locations of users and services. Therefore, this paper extracts the feature vectors and combines similar information with the location information first, then transforms the QoS prediction problem into a regression problem of machine learning, and finally proposes a QoS prediction method based on support vector regression, i.e., SVR4QP. Experimental results show that, Compared with the classical algorithms, half of the cases SVR4QP is optimal, and compared with some novel mobile QoS prediction methods, $1/4$ of the cases SVR4QP is optimal and half of the cases SVR4QP if suboptimal. The experimental results demonstrate that SVR4QP has comparatively more accurate mobile QoS prediction.

In practice, users of mobile services may be constantly moving and may also move at very high speeds, such as when using mobile services in a car, where the location awareness of SVR4QP may not keep up. Therefore, in the future, we will consider the impact of the mobility of users in QoS prediction.

Acknowledgements

The work described in this paper was partially supported by the National Natural Science Foundation of China (Nos. U21A20513, 62076154), the Humanities and Social Sciences Research Foundation of the Ministry of Education (No. 23YJAZH118), the Key R&D Program of Shanxi Province (No. 202202020101003), and the Foundation Research Programme of Shanxi Province (No. 202103021224289).

References

[Arora et al. 2016] Arora, S., Ge, R., Kannan, R., Moitra, A.: “Computing a nonnegative matrix factorization-provably”; Siam J Comput, (Jan 2016), 45(4), 1528-1611.

[Chen et al. 2020] Chen, Z., Sun, Y., You, D., Li, F., Shen, L.: “An accurate and efficient web service QoS prediction model with wide-range awareness”; Future Gener. Comp. Syst. 109 (Jan 2021), 275-292.

[Deng et al. 2018] Deng, S., Xiang, Z., Yin, J., Taheri, J., Zomaya, A.: “Composition-driven iot service provisioning in distributed edges”; IEEE Access 6 (Sept 2018), 54258-54269.

[Dinh et al. 2013] Dinh, H., Lee, C., Niyato, D., Wang, P.: “A survey of mobile cloud computing: architecture, applications, and approaches”; Wirel. Commun. Mob. Com. 13, 18 (Dec 2013), 1587-1611.

[Dong et al. 2023] Dong, S., Zhan, J., Hu W., Mohajer, A., Bavaghar, M., Mirzaei, A: “Energy-efficient hierarchical resource allocation in uplink-downlink decoupled NOMA HetNets”; IEEE T Netw Serv Man, (Sep 2023), 20(3), 3380-3395.

[Hernando et al. 2016] Hernando, A., Bobadilla, J.,Ortega F.: “A non-negative matrix factorization for Collaborative Filtering Recommender Systems based on a Bayesian probabilistic model”; Knowl-Based. Syst.,97, (2016), 188-202.

[Jia et al. 2015] Jia, Z., Yang, Y., Gao W., Chen X.: “User-based collaborative filtering for tourist attraction recommendations”; 2015 IEEE international conference on CICT, IEEE, Jabalpur, India, (Dec), 22-25.

[Kant and Mahara 2018] Kant, S. and Mahara, T.: “Merging user and item based collaborative filtering to alleviate data sparsity”; Int J Syst Assur Eng, (Feb 2018), 9(1), 173-179.

[Li et al. 2019] Li, S., Wen, J., Wang, X.: “From reputation perspective: A hybrid matrix factorization for QoS prediction in location-aware mobile service recommendation system”; Mob. Inf. Syst. 2019, 2 (Jan 2019), 1-12.

[Liu and Chen 2019] Liu, J., Chen Y.: “HAP: A Hybrid QoS Prediction Approach in Cloud Manufacturing combining Local Collaborative Filtering and Global Case-based Reasoning”; IEEE T Serv Comput, (Dec 2019), 14(6), 1796-1808.

[Manochandar and Punniyamoorthy 2020] Manochandar, S., Punniyamoorthy, M.: “A new user similarity measure in a new prediction model for collaborative filtering”; Appl. Intell. 51 (Aug 2020), 586-615.

[Mohajer et al. 2022] Mohajer, A., Sorouri, F, Mirzaei, A, Ziaeddini, A, Rad, K J., Bavaghar, M.: “Energy-aware hierarchical resource management and backhaul traffic optimization in heterogeneous cellular networks”; IEEE Syst J, (Dec 2022), 16(4), 5188-5199.

[Mohajer et al. 2023] Mohajer, A., Daliri M. S., Mirzaei, A, Ziaeddini, A, Nabipour, M, Bavaghar M.: “Heterogeneous computational resource allocation for NOMA: Toward green mobile edge-computing systems”; IEEE T Serv Comput, (Apr 2023), 16(2), 1225-1238.

[Neira et al. 2021] Neira, H., Guliany, J., V□asquez, L.: “Multidimension tensor factorization collaborative filtering recommendation”; Proceedings of International Conference on Big Data, Machine Learning and Applications (Mar 2021), 171-178.

[Ren and Wang 2018] Ren, L., Wang, W.: "An SVM-based collaborative filtering approach for Top-N web services recommendation"; Future Gener. Comp. Syst. 78 (2018), 531-543.

[Ren and Wang 2020] Ren, L., Wang, W.: "Method for QoS prediction in mobile edge computing environment"; Journal of Chinese Computer Systems, (Jun 2020), 41(6), 1176-1181.

[Rendle 2012] Rendle, S.: "Factorization machines with libfm"; ACM T Intel Syst Tec, (May 2012) 3(3), 1-22.

[Tang et al. 2019] Tang, M., Liang, W., Yang Y., Xie, J.: "A factorization machine-based QoS prediction approach for mobile service selection"; IEEE Access, 7, (Mar 2019), 32961-32970.

[Wang et al. 2017] Wang, S., Zhao, Y., Huang, L., Xu, J., Hsu, C.: "QoS prediction for service recommendations in mobile edge computing"; J. Parallel Distr. Com. 127, 5 (Sept 2017), 134-144.

[Yang et al. 2021] Yang, E., Huang, Y., Liang, F., Pan, W., Ming,Z.: "FCMF: Federated collective matrix factorization for heterogeneous collaborative filtering"; Knowl-Based. Syst. 220, (2021), 106946.

[Yu et al. 2022] Yu, R., Ye, D., Wang, Z., Zhang, B., Oguti, A., Li, J., Jin, B., Kurdahi, F.: "CFFNN: Cross feature fusion neural network for collaborative filtering"; IEEE T. Knowl. Data En. (Otc 2022), 34(10), 4650-4662.

[Zhang et al. 2009] Zhang, J., Lin, Z., Xiao B., Zhang, C.: "An optimized item-based collaborative filtering recommendation algorithm"; 2009 IC-NIDC, IEEE, Beijing, China, (Nov 2009) , 414-418.

[Zhang et al. 2011] Zhang, Y., Zheng Z., Lyu, M. R: "Exploring latent features for memory-based QoS prediction in cloud computing"; IEEE 30th SRDS, IEEE, Madrid (Oct. 2011), 1-10.

[Zhang et al. 2020] Zhang, Y., Zhang, P., Luo Y., Ji L.: "Towards Efficient, Credible and Privacy-Preserving Service QoS Prediction in Unreliable Mobile Edge Environments"; 39th SRDS, Shanghai, (2020), 309-318.

[Zhang et al. 2021] Zhang, Y., Pan, J., Qi, L., He, Q.: "Privacy-preserving quality prediction for edge-based IoT services"; Future Gener. Comp. Syst., 114, (Jan 2021), 336-348.

[Zheng et al. 2010] Zheng, Z., Zhang, Y., Lyu, M.: "Distributed qos evaluation for real-world web services"; ICWS 2010, Miami Florida (Jul 2010), 83-90.

[Zheng et al. 2013] Zheng, Z., Ma, H., Lyu, M. R, King I.: "Collaborative Web Service QoS Prediction via Neighborhood Integrated Matrix Factorization"; IEEE T Serv Comput, (Jun 2013), 6(3), 289-299.

[Zhou 2016] Zhou, Z.: "Machine Learning"; 2016, Tsinghua university press, China.