TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 08/18 pp315–324
DOI: 10.26599/TST.2021.9010040
Volume 27, Number 2, April 2022

A Truncated SVD-Based ARIMA Model for Multiple QoS Prediction in Mobile Edge Computing

Chao Yan, Yankun Zhang, Weiyi Zhong, Can Zhang, and Baogui Xin*

Abstract: In the mobile edge computing environments, Quality of Service (QoS) prediction plays a crucial role in web service recommendation. Because of distinct features of mobile edge computing, i.e., the mobility of users and incomplete historical QoS data, traditional QoS prediction approaches may obtain less accurate results in the mobile edge computing environments. In this paper, we treat the historical QoS values at different time slots as a temporal sequence of QoS matrices. By incorporating the compressed matrices extracted from QoS matrices through truncated Singular Value Decomposition (SVD) with the classical ARIMA model, we extend the ARIMA model to predict multiple QoS values simultaneously and efficiently. Experimental results show that our proposed approach outperforms the other state-of-the-art approaches in accuracy and efficiency.

Key words: edge computing; QoS prediction; AutoRegressive Integrated Moving Average (ARIMA); truncated Singular Value Decomposition (SVD)

1 Introduction

Because of its benefits in cost reduction, rapid elasticity, on-demand self-service, and optimal resource utilization, cloud computing has become a multi-billion dollar industry that is still increasing worldwide[1]. However, because of the increasing real-time computing demands of cloud users and a massive amount of data generated through the IoT, traditional centralized cloud infrastructure has suffered from problems, such as long latency, jitter, and bandwidth limitation. Mobile Edge Computing (MEC) is a prominent network architecture for solving the aforementioned problems. By shifting a load of cloud computing to the edge of a cellular network, i.e., base stations, MEC helps reduce congestion on mobile networks, decrease latency, and enhance the quality of experience for end-users.

As a multitude of services has been deployed at mobile edge nodes, it is difficult for users to choose optimal services, i.e., services with the highest quality. Personalized service recommendation approaches can provide users with better services. Concretely, a recommendation method, e.g., the Collaborative Filtering (CF) techniques, first predicts the quality value of candidate services, then recommends services with better quality to users. Therefore, the accuracy of Quality of Service (QoS) prediction plays a crucial role in service recommendation.

However, in a MEC environment, QoS prediction faces more challenges than those in the traditional cloud computing environment. First, because MEC uses a wireless transmission medium, its service quality is readily interfered with different factors in the environment. Second, because of the mobility of users, a user may invoke a web service repeatedly through

• Chao Yan and Baogui Xin are with the College of Economic and Management, Shandong University of Science and Technology, Qingdao 266590, China. E-mail: firebird.yan@foxmail.com; xin@sdust.edu.cn.
• Yankun Zhang is with Weifang Key Laboratory of Blockchain on Agricultural Vegetables, Weifang University of Science and Technology, Weifang 262700, China. E-mail: zhangyankunsg@126.com.
• Weiyi Zhong and Can Zhang are with the School of Computer Science, Qufu Normal University, Rizhao 276826, China. E-mail: weiyi_zhong@outlook.com; sdzc1719@126.com.
* To whom correspondence should be addressed.
Manuscript received: 2021-01-29; revised: 2021-05-17; accepted: 2021-05-30

different edge servers. Thus, historical QoS data are stored on different edge nodes. Although some of these historical QoS data are sent to the centralized data center, it is not sufficient to make an accurate QoS prediction.

In practice, a user often invokes a web service repeatedly, which means that a user may have a series of QoS data for the same web service. Thus by exploiting the historical QoS data at different time slots, the problem of QoS prediction can be considered a Temporal Sequence Forecasting (TSF) problem. AutoRegressive Integrated Moving Average (ARIMA)[2] is one of the most popular models in temporal sequence prediction[3]. By merging the autoregressive model and the moving average model with differencing temporal sequence, ARIMA can provide a more accurate prediction for a nonstationary sequence. However, most existing ARIMA models cannot predict multiple sequences simultaneously, they must forecast sequence by sequence, which leads to high-computational costs[4].

In this paper, we apply the ARIMA model to the QoS prediction problem and propose a novel QoS prediction approach based on an extended ARIMA model to improve the accuracy of QoS prediction in an MEC environment. In general, the main contributions of our paper are threefold:

(1) To reduce the effect of noise and accelerate the training of the ARIMA model, we use Singular Value Decomposition (SVD) to obtain a compressed matrix, which still preserves the intrinsic nature of QoS data.

(2) To perform QoS prediction simultaneously, we extend the classical ARIMA model to matrix form and apply it to the learned compressed matrix, which can obtain better efficiency in predicting multiple QoS values.

(3) To validate the feasibility of our approach, we conduct a series of experiments on a real-world QoS dataset. The results show that our proposed approach outperforms other state-of-the-art approaches in accuracy and efficiency.

The remainder of this paper is organized as follows. Section 2 summarizes recent research on QoS prediction in an edge computing environment. In Section 3, we present preliminaries of our approaches, and explain the motivation of our research through an intuitive example. Section 4 introduces the details of our proposed approach. Experimental evaluation is demonstrated in Section 5. Finally, Section 6 concludes our research work.

2 Related Work

Many studies on QoS prediction in an edge computing environment have been conducted in recent years. Wang et al.[5] proposed a QoS prediction approach that considers user mobility and QoS data volatility to adapt to an MEC environment. White et al.[6] introduced the stacked autoencoder model into QoS prediction, which can improve the training efficiency compared to classical matrix factorization, while maintaining the accuracy of QoS prediction. Yin et.al.[7] integrated a convolutional neural network with a matrix factorization model and obtained more stable and accurate results. However, these approaches neglect the historical QoS values at different time slots, which may lead to less accurate results.

To obtain more accurate prediction results, several research works have considered the time factor[8–11]. These research works can be roughly classified into two categories: data-driven approaches and temporal model-based approaches[12]. Data-driven methods view QoS prediction as a missing item problem and typically solve the problem with CF and matrix factorization techniques[13–15]. Yu and Huang[16] treated the temporal quality data as a three-dimensional matrix and made QoS predictions through CF techniques. Qi et al.[17] integrated locality sensitive hashing techniques with CF, which can preserve the privacy of QoS data distributing across different platforms while maintaining prediction accuracy. However, these methods heavily depend on the data sparsity at current time slots.

Temporal model-based approaches typically view QoS values as a time sequence, in which many time series forecasting methods can be used to predict the QoS values. Godse et al.[18] used the ARIMA model to forecast QoS values, then made service selection based on the newly predicted QoS values. Amin et al.[19] incorporated ARIMA with the GARCH model to capture the volatility of QoS data and make more accurate forecasts. However, these methods predict QoS values sequence by sequence, which may lead to high computational costs when predicting a large amount of sequences. Jing et al.[20] represented a multiple temporal sequence as a matrix and generalized the AutoRegressive (AR) model by applying it to the temporal sequence matrix. Shi et al.[4] proposed a temporal sequence prediction method for short temporal sequences. This approach first transforms a multiple temporal sequence

matrix into a high-dimensional tensor, then trains the ARIMA model with learned core tensors. However, these approaches focus on the temporal sequence forecasting, and cannot be applied to QoS prediction directly.

In this paper, we make QoS predictions based on historical QoS data at recent time slots. Furthermore, by incorporating the prominent temporal sequence prediction techniques with QoS prediction methods, we propose a novel QoS prediction approach to improve the QoS prediction accuracy in an MEC environment.

3 Preliminary and Motivation

3.1 Preliminaries

3.1.1 ARIMA

The ARIMA model was originally proposed by Box and Jenkins[21], which is a generalized form of the AutoRegressive Moving Average (ARMA). Both models are designed to predict future values in a time series. Letting y_t denote a series of values at different time points t, the AR part of ARIMA can be formulated as

$$y_t = \sum_{i=1}^{p} \gamma_i y_{t-i} \tag{1}$$

which indicates that the value at t is regressed on its prior p values, where $\gamma_i (i = 1, \ldots, p)$ is the coefficient of AR. Meanwhile, letting ϵ_t denote the white noise error term at t, the Moving Average (MA) part of ARIMA is given by

$$y_t = \mu + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \tag{2}$$

which indicates that the value at t depends linearly on the current and the past q white noise error terms, where μ is the expectation value of y_t, and $\theta_j (j = 1, \ldots, q)$ is the model parameter.

The ARMA model integrates the AR and MA models. Then, it can be written as

$$y_t = \sum_{i=1}^{p} \gamma_i y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t \tag{3}$$

In practice, values in a time series are usually not stationary. To capture the stationary properties of a time sequence, the differencing method is incorporated into the ARMA model, which is then named ARIMA. Letting $\Delta^d y_t$ represent the order-d differencing of y_t, the ARIMA(p, d, q) can be formulated as

$$\Delta^d y_t = \sum_{i=1}^{p} \gamma_i \Delta^d y_{t-i} + \sum_{j=1}^{q} \theta_i \epsilon_{t-j} + \epsilon_t \tag{4}$$

3.1.2 Truncated SVD

The SVD of a given matrix $A \in \mathbf{R}^{m \times n}$ is a factorization of A into three matrices. It can be written as

$$A = U \Sigma V^{\mathrm{T}}, \quad \text{s.t. } UU^{\mathrm{T}} = I, VV^{\mathrm{T}} = I \tag{5}$$

where U is an $m \times m$ unitary matrix, and V is an $n \times n$ unitary matrix. Σ is an $m \times n$ rectangular diagonal matrix, and the diagonal entriy $\delta_i = \Sigma_{ii}$ is called the singular value of A. In practice, the diagonal entries of Σ are in descending order, to ensure that Σ is uniquely determined by A.

In some applications, truncated SVD is adopted to reduce the dimensionality of the matrix. The truncated SVD of a matrix A is given by

$$A = U_r \Sigma_r V_r^{\mathrm{T}} \tag{6}$$

where $r \ll \min(m, n)$, Σ_r is a diagonal matrix composed of the first r singular values from Σ. U_r is an $m \times r$ matrix, and V_r is an $n \times r$ matrix. U_r and V_r correspond to the first r columns of U and V, respectively. If we set $A' = U_r \Sigma_r$, then A' is a compressed matrix of A. A' has many fewer elements than A, while retaining the important features of A.

3.2 Motivation

To demonstrate the motivation of this paper, we present an intuitive example, as shown in Fig. 1. In this example, three edge servers are built-in near the base stations. Numerous services are deployed on the edge servers. For a user named Jim, when he locates near Edge Server 1, he can invoke web services, e.g., Gaode Map and Baidu Map, directly from Edge Server 1, or other services in the cloud center via the edge server. Suppose Jim uses Gaode, Baidu, and Apple Map services on his mobile phone many times through Edge Server 1 and Edge Server 2. If Jim now travels into an area near Edge Server 3, and needs to use a map service, which service should be recommended to him?

Before making a recommendation, we should predict the QoS value of all the map services first. Traditional approaches, e.g., CF and matrix factorization, usually make a prediction based on the latest QoS history data. However, because of the mobility of end-users, users may invoke the same services through different edge servers. Furthermore, as services cached in edge servers often update dynamically, when an end-user invokes the same service in the same location at different time slots, the QoS values may fluctuate sharply. Given these challenges, we present a novel QoS prediction

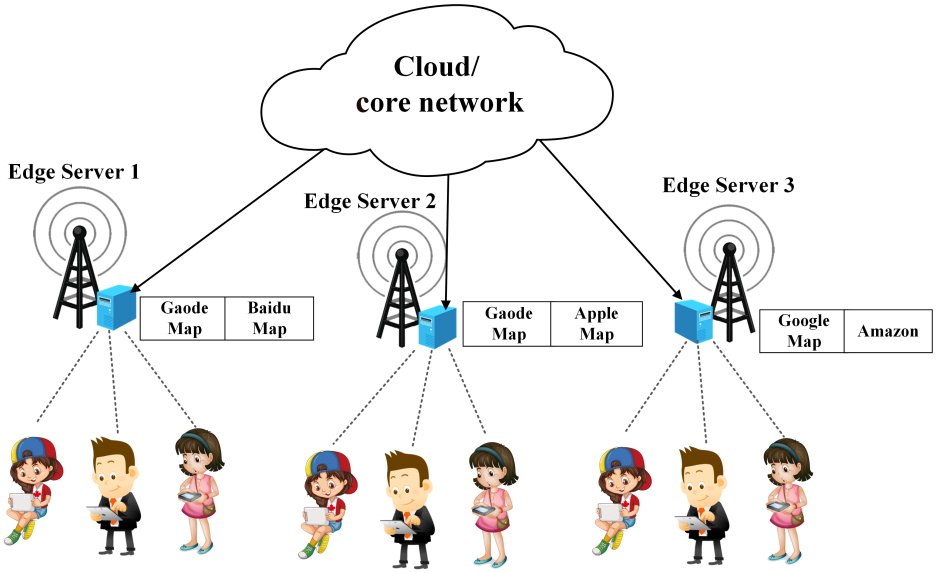


Fig. 1 QoS prediction in an edge computing environment: An intuitive example.

method in this paper, which will be discussed in detail in the next section.

4 An Improved ARIMA Model for QoS Prediction

4.1 Problem formulation

To facilitate the understanding of readers, we first introduce the symbols in this paper.

(1) $U = \{u_1, u_2, \ldots, u_M\}$: The set of users in a mobile edge environment. Here a user may be a mobile user or a user using smart devices, e.g., Apple Watch.

(2) $WS = \{ws_1, ws_2, \ldots, ws_N\}$: The set of web services deployed on the edge server. These web services do not belong to the same service provider.

(3) $q_{i,j,t}$: The corresponding QoS value when user u_i invoked service s_j at time slot t. Note that if u_i did not invoke s_j at time slot t, $q_{i,j,t}$ is set to 0 (invalid).

(4) $\chi \in \mathbf{R}^{M\times N\times T}$: Representing all the QoS values at recent T time slots. Because the QoS value of all the web services invoked by the entire user set at time slot t can be represented by a matrix, χ can be considered a three dimensional matrix, also known as a tensor, as shown in Fig. 2.

4.2 Matricized ARIMA with truncated SVD

We aim to predict all the QoS values at time slots $T+1$ simultaneously. Here, we formulate the problem as a TSF problem. χ can be considered as a sequence of matrices $\chi_1, \chi_2, \ldots, \chi_T$, where χ_t represents the historical QoS matrix at time slot t. Letting $\Delta^d\chi$ denote the d-order

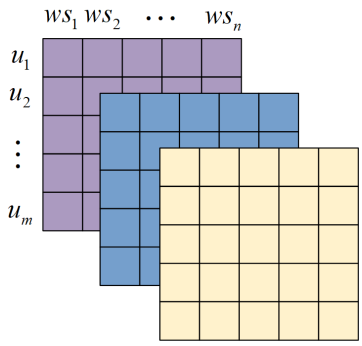


Fig. 2 QoS data representation as a sequence of QoS matrices.

differencing of χ, then

$$\Delta^d\chi = \{\Delta^d\chi_d, \Delta^d\chi_{d+1}, \ldots, \Delta^d\chi_T\} \tag{7}$$

As a user only invokes a tiny proportion of services, the QoS data of a user are very sparse. To reduce the computational and storage cost, we compress the columns of $\Delta^d\chi_t$ through the truncated SVD method, which can be represented as

$$\begin{aligned} &\Delta^d\kappa_t = \Delta^d\chi_t V,\\ &\text{s.t. } VV^{\mathrm{T}} = I \end{aligned} \tag{8}$$

where $V \in \mathbf{R}^{N\times R}$ is an orthogonal factor matrix, and $R \ll N$. $\Delta^d\kappa_t \in \mathbf{R}^{M\times R}$ is the compressed matrix of $\Delta^d\chi_t$, which represents the most important feature of $\Delta^d\chi_t$, but has many fewer elements than $\Delta^d\chi_t$. $\Delta^d\chi_t$ can be recovered by $\Delta^d\kappa_t$ and V,

$$\widehat{\Delta^d\chi_t} = \Delta^d\kappa_t V^{\mathrm{T}} \tag{9}$$

The first goal of our optimization is to minimize the difference between $\Delta^d\chi_t$ and $\widehat{\Delta^d\chi_t}$.

To reduce the computational cost, we incorporate the

compressed matrix instead of the original QoS matrix into the ARIMA model. Therefore, the generalized ARIMA model is defined as follows:

$$\Delta^d \kappa_t = \sum_{i=1}^{p} \gamma_i \Delta^d \kappa_{t-i} - \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t \tag{10}$$

where γ_i and θ_j are the parameters of AR and MA respectively, and ϵ_{t-j} is the random error terms of the past q observations, which are assumed to be independent, identically distributed variables with zero mean. ϵ_t is the prediction error at the current time slot. Therefore, our second goal is to minimize ϵ_t to zero.

On the basis of the two goals of our optimization, the objective function can be defined as follows:

$$\min_{\{\Delta^d \kappa_t, V, \epsilon_{t-j}, \gamma_i, \theta_j\}} \sum_{t=s+1}^{T} \left(\frac{1}{2} \left\| \Delta^d \kappa_t - \Delta^d \chi_t V \right\|_F^2 + \frac{1}{2} \left\| \Delta^d \kappa_t - \sum_{i=1}^{p} \gamma_i \Delta^d \kappa_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \right\|_F^2 \right) \tag{11}$$

where $s = p + d + q$, which is the minimum number of time slots.

We adopt the augmented Lagrangian method, which is widely used in mathematical optimization problems, to minimize the above objective function. We first fix $V, \epsilon_{t-j}, \gamma_i$, and θ_j, compute the partial derivation of the objective function (11) with respect to $\Delta^d \kappa_t$, and equate it to zero. Then, we can obtain the updated formulation of $\Delta^d \kappa_t$ as follows:

$$\Delta^d \kappa_t = \frac{1}{2} \left(\Delta^d \chi_t V + \sum_{i=1}^{p} \gamma_i \Delta^d \kappa_{t-i} - \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \right) \tag{12}$$

Formula (11) with respect to V is

$$\min_{\{V\}} \sum_{t=s+1}^{T} \left(\frac{1}{2} \left\| \Delta^d \kappa_t - \Delta^d \chi_t V \right\|_F^2 \right), \quad \text{s.t. } VV^{\mathrm{T}} = I \tag{13}$$

which is equivalent to the orthogonal Procrustes problem[22]. Then the global optimal solution of Formula (13) is BA^{T}. A and B are the left and right singular vectors of the singular value decomposition of $\sum_{t=s+1}^{T} (\Delta^d \kappa_t^{\mathrm{T}} \Delta^d \chi_t)$, respectively, which is calculated as

$$\sum_{t=s+1}^{T} (\Delta^d \kappa_t^{\mathrm{T}} \Delta^d \chi_t) = A \Sigma B^{\mathrm{T}} \tag{14}$$

The parameters of AR and MA are typically minimized using Yule-Walker method in classical ARIMA. Calculate the partial derivation of Formula (11) with respect to ϵ_{t-j}, and equate it to zero. Then, ϵ_{t-j} can be updated by

$$\epsilon_{t-j} = \frac{\sum_{t=s+1}^{T} \left(\Delta^d \kappa_t - \sum_{i=1}^{p} \gamma_i \Delta^d \kappa_{t-i} + \sum_{k \neq j}^{q} \theta_k \epsilon_{t-k} \right)}{(s+1-T)\theta_j} \tag{15}$$

Formally, we summarize the pseudo-code of the model learning process in Algorithm 1.

4.3 Predicting χ_{T+1}

We calculate the new $\Delta^d \kappa_{T+1}$ by

$$\Delta^d \kappa_{T+1} = \sum_{i=1}^{p} \gamma_i \Delta^d \kappa_{T+1-i} - \sum_{j=1}^{q} \theta_j \epsilon_{T+1-j} \tag{16}$$

Then, we reconstruct the new $\Delta^d \chi_{T+1}$ according to Eq. (9). Finally, we perform inverse d-order differencing for $\Delta^d \chi_{T+1}$ and obtain χ_{T+1}. The pseudo-code of the prediction process can be described in Algorithm 2.

Algorithm 1 Learning $SerPred_{SVD\_ARIMA}$ model

Input: $\chi \in \mathbf{R}^{M \times N \times T}$, p, d, q, and R
Output: $V, \gamma_1, \gamma_2, \ldots, \gamma_p$, and $\theta_1, \theta_2, \ldots, \theta_q$

1: Calculate d-order differencing for $\chi_1, \chi_2, \ldots, \chi_T$, and obtain $\Delta^d \chi_{d+1}, \ldots, \Delta^d \chi_T$
2: Initialize random errors $\epsilon_{t-q}, \epsilon_{t-q+1}, \ldots, \epsilon_{t-1}$
3: Initialize factor matrix $V \in \mathbf{R}^{N \times R}$
4: for $t = p + d + q, \ldots, T$
5: Compute the core tensor $\Delta^d \kappa_t$ of $\Delta^d \chi_t$ by Formula (8)
6: Estimate parameters $\gamma_1, \gamma_2, \ldots, \gamma_p$ of AR and parameters $\theta_1, \theta_2, \ldots, \theta_q$ of MA by Yule-Walker equations
7: Update $\Delta^d \kappa_t$ by Eq. (12)
8: Calculate A and B by Eq. (14)
9: Update $V = BA^{\mathrm{T}}$
10: for $j = 1, 2, \ldots, q$
11: Update ϵ_{t-j} by Eq. (15)
12: Repeat Steps 4 to 11 until convergence
13: Output projection matrix V, and AR and MA parameters $\gamma_1, \ldots, \gamma_p$, and $\theta_1, \ldots, \theta_q$, respectively

Algorithm 2 Predicting QoS value

Output: $\Delta^d \kappa_T, \ldots, \Delta^d \kappa_{T+1-p}, \epsilon_T, \ldots, \epsilon_{T+1-q}$, $\gamma_1, \ldots, \gamma_p, \theta_1, \ldots, \theta_q, V$, and $\Delta^d \chi$
Input: χ_{T+1}

1: Calculate $\Delta^d \kappa_{T+1}$ by Eq. (16)
2: Compute $\Delta^d \chi_{T+1} = \Delta^d \kappa_{T+1} V^{\mathrm{T}}$
3: Carry out inverse d-order differencing for $\Delta^d \chi_{T+1}$ to obtain χ_{T+1}
4: Output χ_{T+1}

5 Experiment

In this section, we conduct a series of experiments to evaluate the proposed $SerPred_{SVD\_ARIMA}$ approach. We investigate our approach on a real-world service quality dataset WS-DREAM[23]. The WS-DREAM dataset contains QoS property values (i.e., response time and throughput) collected from 3889 web services invoked by 142 users over 64 different time slots. As web services and users locate in different countries, we use them to simulate an MEC environment. Experiments are conducted on temporal response time sequences and temporal throughput sequences. Both sequences are split into two parts: QoS values in the last slot are treated as the testing set, and the other QoS values are considered as the training set.

To validate the advantages of our approach, we compare $SerPred_{SVD\_ARIMA}$ with three classical approaches: UPCC[24], IPCC[25], and $SerRec_{time\_LSH}$[17]. Additionally, we adopt the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to evaluate the prediction accuracy of the four methods. The MAE and RMSE are widely used to measure the error between predicted values($\widehat{r_{ij}}$) and real values(r_{ij}).

The MAE is defined as

$$\text{MAE} = \frac{\sum(\widehat{r_{ij}} - r_{i,j})}{N} \tag{17}$$

The RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{\sum(\widehat{r_{ij}} - r_{i,j})^2}{N}} \tag{18}$$

The experiments are conducted on an HP workstation with an Intel Xeon Silver 4210 CPU and 64 GB of RAM. The machine runs Windows 10 and Python 3.6.

5.1 Parameters setting and convergence

First, we study the parameters of our proposed $SerPred_{SVD\_ARIMA}$. By conducting a grid search over the ARIMA model, we determine the best parameter settings ($p = 2, d = 0$, and $q = 3$) for temporal response time sequences, and ($p = 3, d = 0$, and $q = 3$) temporal throughput sequences. Moreover, the dimension of the compressed matrix is set to (142, 800), which means the compression ratio of the columns is approximately 20%.

We investigate the convergence of our approach with respect to the relative error of projection matrices. As shown in Fig. 3, our approach converges quickly, within 20 iterations both on the response time and throughput datasets. Therefore, setting the maximum number of iterations above 20 obtains a sufficient prediction accuracy. Here, we set the maximum number of iterations of our approach to 20 for all the experiments.

5.2 Accuracy comparison with respect to data sparsity

To analyze the effect of the data sparsity of the testing set, we compare the prediction accuracy of four methods on testing sets with different sparsity. The experimental results are shown in Fig. 4.

As our approach makes predictions based on historical QoS data of the last $T-1$ time slots, it shows little variation when data sparsity varies from 30% to 90%. Meanwhile, the MAE and RMSE values of UPCC, IPCC, and $SerRec_{time\_LSH}$ increase as the testing set becomes sparser. The MAE and RMSE values of our approach are always smaller than those of the other three methods when data sparsity varies.

We also compare the time costs of the four QoS prediction approaches, as shown in Fig. 5. This comparison demonstrates that our approach outperforms UPCC, IPCC, and $SerRec_{time\_LSH}$ in efficiency.

5.3 Accuracy comparison with respect to the length of temporal sequence

The length of the temporal sequence (T) has a substantial

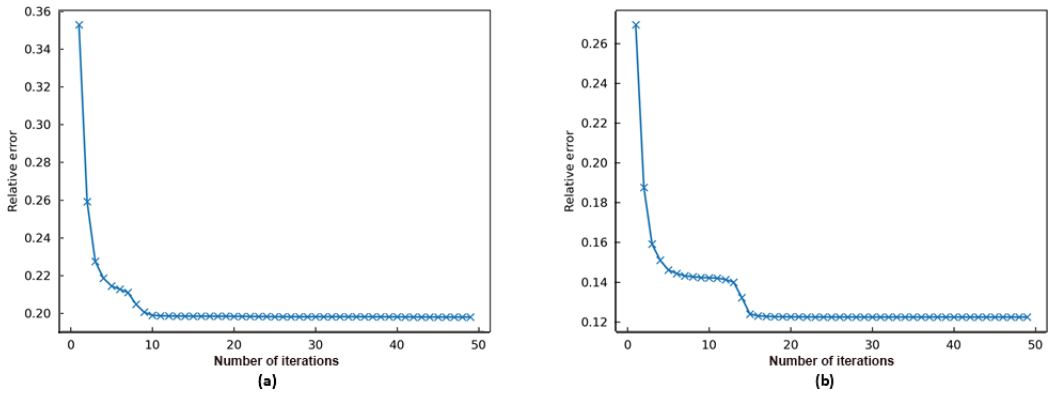


Fig. 3 Convergence curves of $SerPred_{SVD\_ARIMA}$ on (a) response time and (b) throughput datasets.

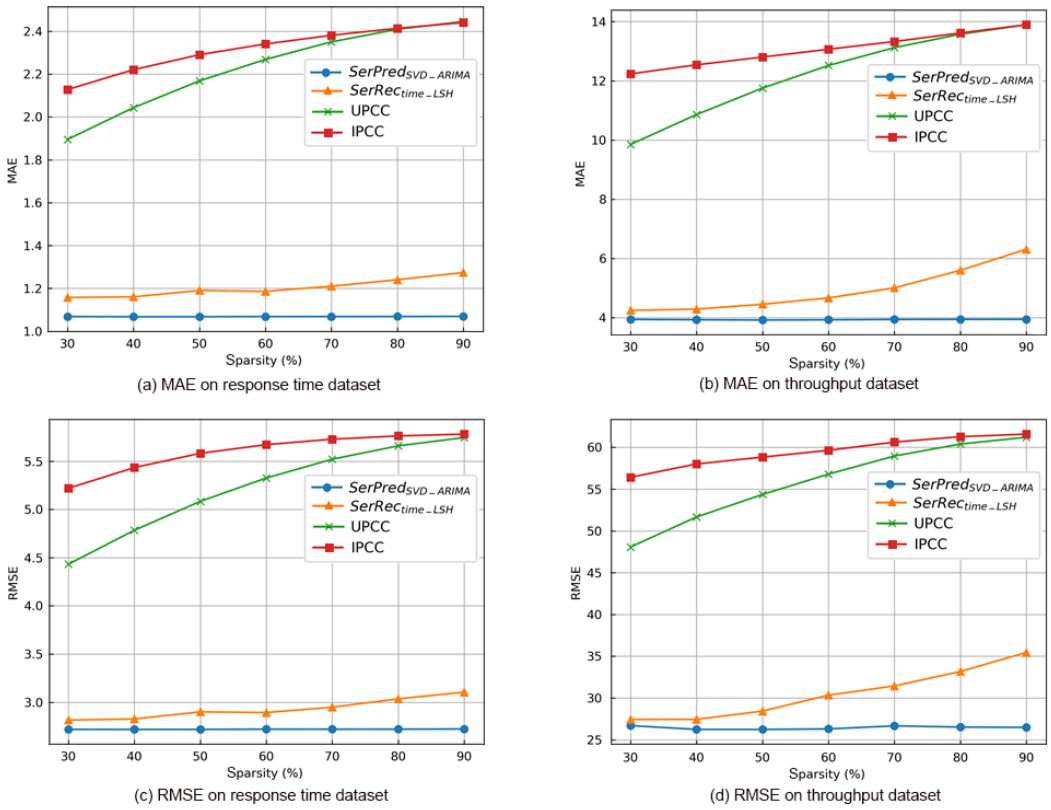


Fig. 4 Prediction accuracy comparison with respect to data sparsity on different datasets.

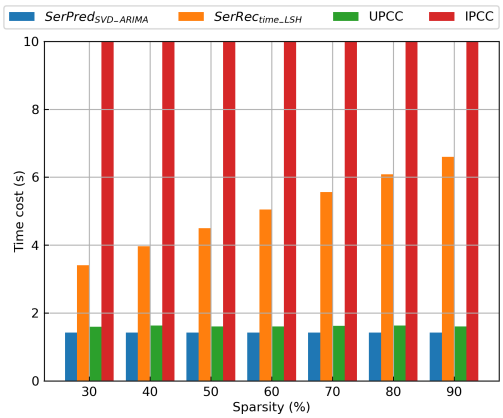


Fig. 5 Time cost comparison with respect to data sparsity. The time cost of IPCC method is too large, so we truncate its bar.

influence on the performance of $SerPred_{SVD\_ARIMA}$ and $SerRec_{time\_LSH}$. To study the relationship between the value of T and the prediction accuracy, we compare the MAE and RMSE values of the four approaches when T varies from 20 to 64. The experimental results are shown in Fig. 6.

For temporal response time sequences, the MAE values of our approach vary slightly with T, while the RMSE values decrease slowly as the number of recent time slots (T) increases. The MAE and RMSE values of our approach are the lowest among the four approaches when T varies from 20 to 64. For temporal throughput sequences, the MAE and RMSE values are approximately stable when T changes. Moreover, our method still obtains more accurate results for different lengths of the temporal throughput sequence. This result is noteworthy because some throughput values change sharply at different time slots, and the MAE and RMSE values of $SerRec_{time\_LSH}$ increase slightly when $T > 50$. In summary, in terms of prediction accuracy, our proposed method remains approximately stable, and outperforms the other three approaches for different lengths of temporal response time and throughput sequences.

Regarding the time cost of the four approaches when T varies from 20 to 64, Fig. 7 shows that the time cost of $SerPred_{SVD\_ARIMA}$ increases with the number of time slots. This is because the computational cost of our proposed approach depends on the value of T. Nonetheless, our method has the lowest time cost among the four approaches.

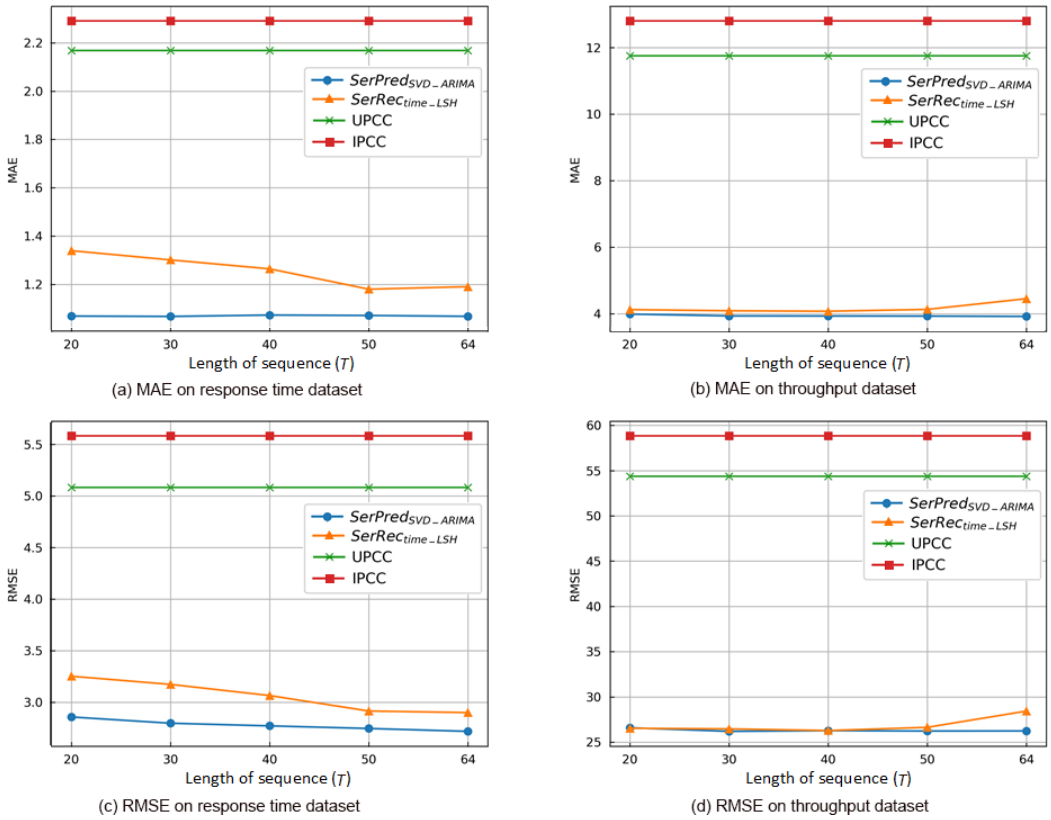


Fig. 6 Prediction accuracy comparison with respect to the number of time slots on different datasets.

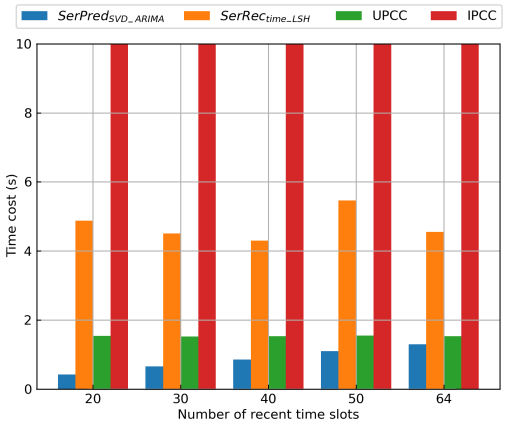


Fig. 7 Time cost comparison with respect to the number of time slots. The time cost of the IPCC method is too high, so we truncate its bar.

6 Conclusion

The quality of web services in an edge computing environment is mainly affected by three factors: (1) the network traffic of the service provider; (2) caching in edge servers; and (3) the mobility of users. Therefore, traditional quality prediction techniques are not suitable in an edge computing environment. In this paper, aiming at predicting multiple temporal QoS sequences simultaneously, we generalize the traditional ARIMA model into matrix mode. Furthermore, to reduce the computation and storage cost of QoS matrices, we use the truncated SVD technique to compress the QoS matrix along the columns and integrate the compressed matrices with the ARIMA model. Finally, by conducting several experiments on a real-world dataset, we find that our proposed approach dramatically improves the QoS prediction accuracy compared to three other classical QoS prediction techniques, and shows approximately stable accuracy with different lengths of temporal sequences.

In future work, we will further refine our algorithm by introducing more optimization goals and context factors, such as those in Refs. [26–33]. In addition, how to improve the recommendation performances by optimizing the network load balance[34–36] is another research topic that requires intensive study.

References

[1] International Data Corporation, Worldwide public cloud services spending forecast to double by 2019, according to IDC, https://www.businesswire.com/news/home/20160121005117/en/Worldwide-Public-Cloud-Services-

Spending-Forecast-to-Double-by-2019-According-to-IDC, 2019.

[2] G. E. P. Box and G. M. Jenkins, Some recent advances in forecasting and control: Part I, *J. Roy. Statist. Soc. Ser. C* (*Appl. Statist.*), vol. 17, no. 2, pp. 91–109, 1968.

[3] C. H. Liu, S. C. H. Hoi, P. L. Zhao, and J. L. Sun, Online ARIMA algorithms for time series prediction, in *Proc. 13th AAAI Conf. Artificial Intelligence*, Phoenix, AZ, USA, 2016, pp. 1867–1873.

[4] Q. Q. Shi, J. M. Yin, J. J. Cai, A. Cichocki, T. Yokota, L. Chen, M. X. Yuan, and J. Zeng, Block hankel tensor ARIMA for multiple short time series forecasting, *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, pp. 5758–5766, 2020.

[5] S. G. Wang, Y. L. Zhao, L. Huang, J. L. Xu, and C. H. Hsu, QoS prediction for service recommendations in mobile edge computing, *J. Parallel Distrib. Comput.*, vol. 127, pp. 134–144, 2019.

[6] G. White, A. Palade, C. Cabrera, and S. Clarke, Autoencoders for QoS prediction at the edge, in *Proc. of 2019 IEEE Int. Conf. Pervasive Computing and Communications*, Kyoto, Japan, 2019, pp. 1–9.

[7] Y. Y. Yin, L. Chen, Y. S. Xu, J. Wan, H. Zhang, and Z. D. Mai, QoS prediction for service recommendation with deep feature learning in edge computing environment, *Mobile Netw. Appl.*, vol. 25, no. 2, pp. 391–401, 2020.

[8] Y. W. Liu, A. X. Pei, F. Wang, Y. H. Yang, X. Y. Zhang, H. Wang, H. N. Dai, L. Y. Qi, and R. Ma, An attention-based category-aware GRU model for the next POI recommendation, *Int. J. Intell. Syst.*, vol. 36, no. 7, pp. 3174–3189, 2021.

[9] Y. Hu, Q. M. Peng, X. H. Hu, and R. Yang, Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering, *IEEE Trans. Serv. Comput.*, vol. 8, no. 5, pp. 782–794, 2015.

[10] L. Y. Qi, C. H. Hu, X. Y. Zhang, M. R. Khosravi, S. Sharma, S. N. Pang, and T. Wang, Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment, *IEEE Trans. Ind. Inform.*, vol. 17, no. 6, pp. 4159–4167, 2021.

[11] X. L. Fan, Y. K. Hu, R. S. Zhang, W. B. Chen, P. Brézillon, and X. L. Fan, Modeling temporal effectiveness for context-aware web services recommendation, in *Proc. 2015 IEEE Int. Conf. Web Services*, New York, NY, USA, 2015, pp. 225–232.

[12] X. Y. Wang, J. K. Zhu, Z. B. Zheng, W. J. Song, Y. H. Shen, and M. R. Lyu, A spatial-temporal QoS prediction approach for time-aware web service recommendation, *ACM Trans. Web*, vol. 10, no. 1, p. 7, 2016.

[13] X. Chen, Z. B. Zheng, X. D. Liu, Z. C. Huang, and H. L. Sun, Personalized QoS-aware web service recommendation and visualization, *IEEE Trans. Serv. Comput.*, vol. 6, no. 1, pp. 35–47, 2013.

[14] H. F. Sun, Z. B. Zheng, J. L. Chen, and M. R. Lyu, Personalized web service recommendation via normal recovery collaborative filtering, *IEEE Trans. Serv. Comput.*, vol. 6, no. 4, pp. 573–579, 2013.

[15] W. Lo, J. W. Yin, S. G. Deng, Y. Li, and Z. H. Wu, Collaborative web service QoS prediction with location-based regularization, in *Proc. 2012 IEEE 19th Int. Conf. Web Services*, Honolulu, HI, USA, 2012, pp. 464–471.

[16] C. Y. Yu and L. P. Huang, A web service QoS prediction approach based on time- and location-aware collaborative filtering, *Serv. Orient. Comput. Appl.*, vol. 10, no. 2, pp. 135–149, 2016.

[17] L. Y. Qi, R. L. Wang, C. H. Hu, S. C. Li, Q. He, and X. L. Xu, Time-aware distributed service recommendation with privacy-preservation, *Inform. Sci.*, vol. 480, pp. 354–364, 2019.

[18] M. Godse, U. Bellur, and R. Sonar, Automating QoS based service selection, in *Proc. 2010 IEEE Int. Conf. Web Services*, Miami, FL, USA, 2010, pp. 534–541.

[19] A. Amin, A. Colman, and L. Grunske, An approach to forecasting QoS attributes of web services based on ARIMA and GARCH models, in *Proc. 2012 IEEE 19th Int. Conf. Web Services*, Honolulu, HI, USA, 2012, pp. 74–81.

[20] P. G. Jing, Y. T. Su, X. Jin, and C. Q. Zhang, High-order temporal correlation model learning for time-series prediction, *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2385–2397, 2019.

[21] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA, USA: Holden-Day, 1976.

[22] J. C. Gower and G. B. Dijksterhuis, *Procrustes Problems*. New York, NY, USA: Oxford University Press, 2004.

[23] Y. L. Zhang, Z. B. Zheng, and M. R. Lyu, WSPred: A time-aware personalized QoS prediction framework for web services, in *Proc. IEEE 22nd Int. Symp. Software Reliability Engineering*, Hiroshima, Japan, 2011, pp. 210–219.

[24] J. S. Breese, D. Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in *Proc. 14th Conf. Uncertainty in Artificial Intelligence*, Madison, WI, USA, 1998, pp. 43–52.

[25] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, GroupLens: An open architecture for collaborative filtering of netnews, in *Proc. 1994 ACM Conf. Computer Supported Cooperative Work*, Chapel Hill North, CA, USA, 1994, pp. 175–186.

[26] A. Guezzaz, Y. Asimi, M. Azrour, and A. Asimi, Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection, *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 18–24, 2021.

[27] H. M. Huang, J. H. Lin, L. Y. Wu, B. Fang, Z. K. Wen, and F. C. Sun, Machine learning-based multi-modal information perception for soft robotic hands, *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 255–269, 2020.

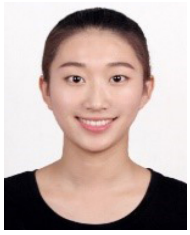
[28] L. Wang, X. Y. Zhang, T. Wang, S. H. Wan, G. Srivastava, S. N. Pang, and L. Y. Qi, Diversified and scalable service recommendation with accuracy guarantee, *IEEE Trans. Comput. Soc. Syst.*, doi: 10.1109/TCSS.2020.3007812.

[29] J. Mabrouki, M. Azrour, G. Fattah, D. Dhiba, and S. El Hajjaji, Intelligent monitoring system for biogas detection based on the Internet of Things: Mohammedia, Morocco City landfill case, *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 10–17, 2021.

[30] L. Y. Qi, X. K. Wang, X. L. Xu, W. C. Dou, and S. C. Li, Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing, *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1145–1153, 2021.
[31] N. Bhardwaj and P. Sharma, An advanced uncertainty measure using fuzzy soft sets: Application to decision-making problems, *Big Data Mining and Analytics*, vol. 4, no. 2, pp. 94–103, 2021.
[32] Y. Khazbak, J. Y. Fan, S. C. Zhu, and G. H. Cao, Preserving personalized location privacy in ride-hailing service, *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 743–757, 2020.
[33] L. N. Wang, X. Y. Zhang, R. L. Wang, C. Yan, H. Z. Kou, and L. Y. Qi, Diversified service recommendation with high accuracy and efficiency, *Knowl.-Based Syst.*, vol. 204, p. 106196, 2020.
[34] Y. P. Fu, Y. S. Hou, Z. F. Wang, X. W. Wu, K. Z. Gao, and L. Wang, Distributed scheduling problems in intelligent manufacturing systems, *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 625–645, 2021.
[35] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts, *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.
[36] N. J. Chen, Z. Wang, R. X. He, J. H. Jiang, F. Cheng, and C. H. Han, Efficient scheduling mapping algorithm for row parallel coarse-grained reconfigurable architecture, *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 724–735, 2021.

Chao Yan received the MEng degree from Chinese Academy of Sciences, China in 2006. He is currently a PhD candidate at Shandong University of Science and Technology, Qingdao, China. His research interests include recommender system and service computing.

Yankun Zhang received the MEng degree from Shandong University of Science and Technology, China in 2011. She is now working in Weifang Key Laboratory of Blockchain on Agricultural Vegetables, Weifang University of Science and Technology, China. Her research interests include recommender system and data mining.



Weiyi Zhong received the BEng degree from Qufu Normal University, China in 2018. She is currently a master student at Qufu Normal University, China. Her research interests include recommender system and service computing.

Can Zhang received the BS degree in statistics from Shandong Finance Institute, China in 2007. She is currently a master student at Qufu Normal University, China. Her current research interests include recommender system and service computing.

Baogui Xin received the PhD degree from Tianjin University, China in 2009. He is currently a professor at Shandong University of Science and Technology, Qingdao, China. He is on the editorial boards of *PLOS One, Frontiers in Physics,* and *Frontiers in Applied Mathematics and Statistics*. His research interests include complex evolution systems, artificial intelligence, fractional order nonlinear systems, optimal decision, and dynamical game theory.