Letter

QoS Prediction Model of Cloud Services Based on Deep Learning

WenJun Huang, PeiYun Zhang, *Senior Member, IEEE*, YuTong Chen, MengChu Zhou, *Fellow, IEEE*, Yusuf Al-Turki, *Senior Member, IEEE*, and Abdullah Abusorrah, *Senior Member, IEEE*

Dear editor,

This letter presents a deep learning-based prediction model for the quality-of-service (QoS) of cloud services. Specifically, to improve the QoS prediction accuracy of cloud services, a new QoS prediction model is proposed, which is based on multi-staged multi-metric feature fusion with individual evaluations. The multi-metric features include global, local, and individual ones. Experimental results show that the proposed model can provide more accurate QoS prediction results of cloud services than several state-of-the-art methods.

Cloud computing provides users with fast and secure cloud services, called "service" for short. With the rapid development of cloud computing, the number of cloud-based services continues to increase. However, it is difficult for users to choose services from lots of candidates to meet their needs. In this case, users must compare their QoS, and then determine the best ones.

QoS can describe non-functional attributes of a service, which is a key indicator often used to evaluate service performance in cloud computing. Due to the uncertainty of user information (such as network status and personal preferences), when different users call the same services, their QoS may differ. Therefore, accurate prediction of QoS values of services is thus required in order to help users choose the most suitable cloud services.

Many methods have emerged to predict QoS, most of which are inspired by collaborative filtering for service recommendation. These methods predict missing QoS values by collecting historical information of users or services. However, they only use information from an original user-service QoS matrix, which may ignore some important factors that affect QoS, such as locations. Differences in user information, service characteristics, and network status lead to different QoS.

With the rapid development of deep learning and computing environments, deep neural network (DNN) technologies have significantly impacted many fields, such as computer vision, data mining, and natural language processing. A DNN has a strong nonlinear fitting ability, which can approximate any nonlinear continuous function. It can extract advanced features from original data after statistical learning on a large amount of data. Thus, it is widely used in many artificial intelligence applications to provide the highest prediction accuracy. DNNs can also be used to accurately predict cloud-service QoS.

Related work: Significant studies have been devoted to solving this problem in recent years. They result in the four main types of methods: Memory-based, model-based, hybrid collaborative filtering, and neural network-based ones.

Memory-based collaborative filtering methods only use an original user-service QoS matrix to predict QoS. Zhang *et al*. [1] propose a QoS prediction method in the field of cloud computing. It learns user features via non-negative matrix factorization (NMF) and utilizes the QoS of similar users to improve prediction accuracy. Although memory-based collaborative filtering methods are easy to implement, they are easily affected by data sparsity. Meanwhile, they have problems such as cold start and poor scalability.

Model-based collaborative filtering methods are widely used to solve the problems mentioned above. Aiming at predicting candidate services for a real-time service adjustment, Zhu *et al*. [2] propose an adaptive matrix factorization method for online QoS prediction.

Hybrid collaborative filtering methods combine memory-based and model-based methods. Since collecting QoS values may cause privacy problems, the studies [3], [4] propose privacy protection strategies to obtain high QoS prediction accuracy while protecting user privacy. These methods offer the advantages of both memory-based and model-based methods. However, they have the problem of high computational complexity.

In recent years, with the development of artificial intelligence, neural networks have been applied to the field of QoS prediction. Using the time correlation of QoS, Xiong *et al*. [5] propose a novel personalized matrix factorization method based on Long Short-Term Memory (LSTM) for online QoS prediction. Chen *et al*. [6] combine an empirical mode decomposition and multivariate LSTM model to propose a hybrid QoS prediction method. However, their network structures and QoS prediction accuracy have much room for improvement in their data preprocessing and feature extraction. This work aims to make such important improvements.

Problem statement: Usually, a user can call multiple cloud services, and a cloud service can be called by different users. As the number of cloud services continues to increase, many services offer similar functions. Users hope to choose a service that meets their needs, which can be achieved by choosing the one with the best QoS from similar services.

After a user calls a cloud service, its QoS value is collected by a cloud system and stored in an original user-service QoS matrix, which is denoted as Q. In Q, rows and columns represent users and services, respectively. Items represent QoS values. $q_{ij} \in Q$ represents the QoS value of service j deployed by user i. Fig. 1 shows the QoS values provided after three users call five services.

	s_1	s_2	s_3	s_4	s_5
u_1	1.984	0.301		0.256	
u_2	6.892		0.276		0.364
u_3		0.264		0.255	0.255

Fig. 1. An original user-service QoS matrix Q.

Given that not all users call all cloud services in a cloud system, Q may have some missing items, which may result in a sparse Q. Due to the similarity among services and among users, missing items can be predicted by using existing/known items in Q. The predicted items are shown in bold in Fig. 2. To accurately predict QoS values, we propose a QoS prediction model based on deep learning, which adopts multi-staged multi-metric feature fusion with individual evaluations for the first time.

Citation: W. J. Huang, P. Y. Zhang, Y. T. Chen, M. C. Zhou, Y. Al-Turki, and A. Abusorrah, "QoS prediction model of cloud services based on deep learning," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 3, pp. 564–566, Mar. 2022.

W. Huang, P. Zhang, and Y. Chen are with Nanjing University of Information Science & Technology, School of Computer and Software, Nanjing, China (e-mail: nt_feifei@163.com; zpy@nuist.edu.cn; hsiaoo@nuist.edu.cn).

M. Zhou is with New Jersey Institute of Technology, Helen and John C. Hartmann Department of Electrical and Computer Engineering, Newark, USA (e-mail: zhou@njit.edu).

Y. Al-Turki and A. Abusorrah are with Center of Research Excellence in Renewable Energy and Power Systems, Department of Electrical and Computer Engineering, Faculty of Engineering, and K. A. CARE Energy Research and Innovation Center, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: yaturki@yahoo.com; aabusorrah@kau.edu.sa).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2021.1004392

	s_1	s_2	s_3	s_4	s_5
u_1	1.984	0.301	**1.3**	0.256	**0.256**
u_2	6.892	**0.268**	0.276	**0.259**	0.364
u_3	**1.905**	0.264	**0.272**	0.255	0.255

Fig. 2. A user-service QoS matrix with predicted items.

Basic concepts:

- Multi-metric features: They include global, local, and individual ones.
- Global feature matrix $\tilde{Q}$: It is generated after performing NMF on an original user-service QoS matrix Q. Global features $\mathcal{H}$ can be extracted from $\tilde{Q}$. $\tilde{q}_{ij}$ is the item at column i and row j in $\tilde{Q}$.
- Local feature matrix G: Based on distance similarity, similar users and similar cloud services are extracted from $\tilde{Q}$, and then G is generated. Local features $\mathcal{L}$ can be obtained from G.
- Individual feature matrices: There are two types of individual feature matrices including matrix U for users and matrix S for cloud services. They are obtained by performing NMF on matrix Q. Individual features $\mathcal{I}$ can be extracted from them.
- Individual evaluation: It comes from matrix $\tilde{Q}$. If the proposed model predicts a QoS value of cloud service j for user i, $\tilde{q}_{ij}$ serves as an individual evaluation.

Proposed prediction model: A new DNN is designed to predict QoS values, which is called multi-staged multi-metric-feature DNN (MM-DNN), as shown in Fig. 3. It has four stages. Multi-metric features are fused in different concatenation layers. Stages 1–3 serve to fuse global, local, and individual features, respectively. In each stage, an individual evaluation is used to modify features, which makes the output more accurate. If these features are input together into MM-DNN at the same time in Stage 1, it may cause a problem of excessive values. Before outputting a final predicted QoS value in Stage 4, an individual evaluation is input to further improve the value. A detailed analysis of the four stages is shown as follows:

Stage 1: Global features are input to the proposed model. Then information with the same size as that of local features is further extracted through L fully connected layers. The features are modified by concatenating an individual evaluation in a concatenation layer. The forward propagation process at this stage can be expressed as

$$\begin{aligned} y_0 &= \mathcal{H}, \\ y_1 &= \varphi(\alpha_1 y_0 + \beta_1), \\ y_2 &= \varphi(\alpha_2 \copyright(y_1, \tilde{q}_{ij}) + \beta_2), \\ y_k &= \varphi(\alpha_k y_{k-1} + \beta_k),\ k \in \{3, 4, \ldots, L\}, \end{aligned} \tag{1}$$

where $\varphi()$ denotes a rectified linear unit, i.e., $\varphi(x) = \max(0, x)$. © is the concatenation operation. y_0 is the input of Stage 1 in MM-DNN. y_2 is obtained through the fully connected layer after concatenating y_1 and $\tilde{q}_{ij}$. y_k is the output of the kth fully connected layer of Stage 1. α_k and β_k represent the weight and bias of the kth fully connected layer, respectively. y_L is the output of Stage 1.

Stage 2: It consists of two concatenation layers and M fully connected layers. Local features are concatenated in a concatenation layer and fed into a fully connected layer. After concatenating an individual evaluation in a concatenation layer, they are learned through fully connected layers. The process is expressed as follows:

$$\begin{aligned} y_{L+1} &= \varphi(\alpha_{L+1}(\copyright(y_L, \mathcal{L})) + \beta_{L+1}), \\ y_{L+2} &= \varphi(\alpha_{L+2} \copyright(y_{L+1}, \tilde{q}_{ij}) + \beta_{L+2}), \\ y_{L+z} &= \varphi(\alpha_L y_{L+z-1} + \beta_{L+z}),\ z \in \{3, 4, \ldots, M\}, \end{aligned} \tag{2}$$

where y_L and $\mathcal{L}$ are the inputs of Stage 2 in MM-DNN, y_{L+1} is obtained through the fully connected layer after concatenating y_L and $\mathcal{L}$, y_{L+2} is obtained through the fully connected layer after concatenating y_{L+1} and $\tilde{q}_{ij}$, y_{L+z} is the output of fully connected layer (L+z), and y_{L+M} is the output of Stage 2.

Stage 3: It contains two concatenation layers and Z fully connected layers. Individual features are connected in a concatenation layer and learned through a fully connected layer. An individual evaluation is then concatenated in a concatenation layer and fed into fully connected layers. The process is expressed as:

$$\begin{aligned} y_{L+M+1} &= \varphi(\alpha_{L+M+1}(\copyright(y_{L+M}, \mathcal{I})) + \beta_{L+N+1}), \\ y_{L+M+2} &= \varphi(\alpha_{L+M+2} \copyright(y_{L+M+1}, \tilde{q}_{ij}) + \beta_{L+M+2}), \\ y_{L+M+b} &= \varphi(\alpha_{L+M+b} y_{L+M+b-1} + \beta_{L+M+b}),\ b \in \{3, 4, \ldots, Z\}, \end{aligned} \tag{3}$$

where y_{L+M} and $\mathcal{I}$ are inputs of Stage 3 in MM-DNN, y_{L+M+1} is obtained through the fully connected layer after concatenating y_{L+M} and $\mathcal{I}$, y_{L+M+2} is obtained through the fully connected layer after concatenating y_{L+M+1} and $\tilde{q}_{ij}$, y_{L+M+b} is the output of the fully connected layer (L+M+b) of MMDNN, and y_{L+M+Z} is the output of Stage 3.

Stage 4: It consists of a concatenation layer and a fully connected layer. The goal of the proposed model is to predict the QoS value of cloud service j for user i. Thus, an individual evaluation $\tilde{q}_{ij}$ is input and connected in the last concatenation layer and learned to further improve the prediction result through the last fully connected layer. The predicted QoS values are then output. The process is expressed as:

$$\begin{aligned} y_{L+M+Z+1} &= \varphi(\beta_{L+M+Z+1}(\copyright(y_{L+M+Z}, \tilde{q}_{ij})) + \beta_{L+M+Z+1}), \\ p_{ij} &= y_{L+M+Z+1}, \end{aligned} \tag{4}$$

where y_{L+M+Z} and $\tilde{q}_{ij}$ are inputs of Stage 4. $y_{L+M+Z+1}$ is obtained through the fully connected layer after concatenating y_{L+M+Z} and $\tilde{q}_{ij}$. $y_{L+M+Z+1}$ is the output of Stage 4. It is also the output of MM-DNN. p_{ij} is a QoS prediction value of cloud service j for user i from the output of MM-DNN.

Experiments: Our experiments use an Intel Core i7-11700KF CPU @ 3.60GHz, NVIDIA GeForce RTX3090 GPU, and Windows 10 64bit. We use Python 3.7 and Pytorch 1.8.0 to realize MM-DNN.

To evaluate the performance of the proposed model, experiments are conducted on a real-world QoS data set of services, which is called WS-DREAM [7]. Let μ be the matrix density:

$$\mu = \xi / |Q| \times 100\%,$$

where ξ is the number of existing items in an original user-service QoS matrix Q. $|Q|$ is the total number of entries in Q. Mean absolute error (MAE) and root mean square error (RMSE) are used as indicators to evaluate prediction accuracy.

The proposed model is compared with the following methods: Probabilistic matrix factorization (PMF) [8], neighborhood-integrated deep matrix factorization (NDMF) [9], and covering-based web service quality prediction via neighborhood-aware matrix factorization (CNMF) [10].

Experimental parameters are set in Table 1. They are obtained

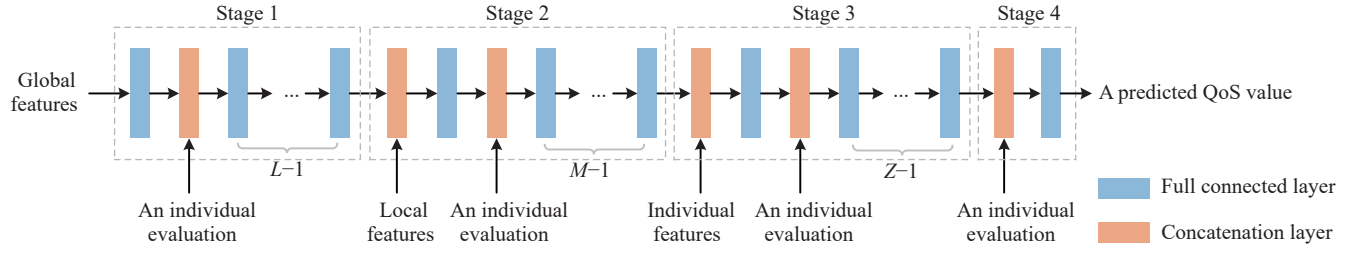

Fig. 3. The structure of MM-DNN.

through lots of experiments. Given four different matrix densities (5%, 10%, 15%, and 20%), the MAE and RMSE of the four methods are compared. Tables 2 and 3 show the MAE and RMSE of the response time and throughput (i.e., two kinds of QoS) of the four methods, respectively. MM-DNN outperforms its peers in terms of prediction accuracy with different matrix densities. The results clearly show that MM-DNN outperforms its peers by 6.1% to 21.2% in response time and by 0.2% to 6.8% in throughput.

Table 1. Experimental Parameters

Parameter	Meaning	Value
J	Number of similar users	25
P	Number of similar cloud services	10
l	Dimension of individual feature vectors	50
L, M, and Z	Number of fully connected layers at Stages 1–3 in MM-DNN	4

Table 2. Comparison of Response Time

Approaches	Density = 5%		Density = 10%		Density = 15%		Density = 20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF	0.5702	1.5458	0.4855	1.3159	0.4503	1.2219	0.4313	1.1691
CNMF	0.5289	**1.3053**	0.4713	1.2373	0.4316	**1.1360**	0.4136	**1.1161**
NDMF	**0.4880**	1.3495	**0.4304**	**1.2349**	**0.3845**	1.1569	**0.3665**	1.1294
MM-DNN	**0.4102**	**1.2165**	**0.3392**	**1.0835**	**0.3261**	**1.0665**	**0.3117**	**1.0321**
Gains	**15.9%**	**6.8%**	**21.2%**	**12.3%**	**15.2%**	**6.1%**	**15.0%**	**7.5%**

Table 3. Comparison of Throughput

Approaches	Density = 5%		Density = 10%		Density = 15%		Density = 20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF	19.1685	58.4231	15.9286	48.1532	14.7298	44.1238	13.9653	41.6633
CNMF	17.2036	**50.6829**	14.7927	**43.6639**	13.9296	**41.9729**	13.3993	40.0503
NDMF	**16.3818**	50.9612	**13.9317**	43.9095	**12.5043**	42.5319	**11.7204**	**39.9431**
MM-DNN	**16.1658**	**50.0102**	**13.3631**	**42.4129**	**12.4415**	**39.1015**	**11.7019**	**37.2234**
Gains	**1.3%**	**1.3%**	**4.1%**	**2.9%**	**0.5%**	**6.8%**	**0.2%**	**6.8%**

Conclusions: This paper presents a QoS prediction model for cloud services based on deep learning and multi-staged multi-metric feature fusion with individual evaluations. A new deep neural network model is constructed to fuse the extracted multi-metric features in multiple stages. At each stage of the model, individual evaluations are used to modify features to improve prediction accuracy. Experimental results show that the proposed method can predict QoS values more accurately than the three compared methods. Our future work plans to use time information to improve the proposed model. Since MM-DNN needs a large amount of data for training, it has limitations when facing a highly dynamic environment. More studies are needed to deal with the related issues [11]–[12].

Acknowledgments: This work was in part supported by the National Natural Science Foundation of China (61872006), the Startup Foundation for New Talents of NUIST, Institutional Fund Projects (IFPNC-001-135-2020), and the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia under grant no. GCV19-37-1441.

References

[1] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring latent features for memory-based QoS prediction in cloud computing," in *Proc. of 2011 IEEE 30th In. Symp. on Reliable Distributed Systems*, pp. 1–10, Nov. 2011.

[2] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Online QoS prediction for runtime service adaptation via adaptive matrix factorization," *IEEE Trans. on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2911–2924, Oct. 2017.

[3] X. Zhu, X. Jing, D. Wu, Z. He, J. Cao, D. Yue, and L. Wang, "Similarity-maintaining privacy preservation and location-aware low-rank matrix factorization for QoS prediction based web service recommendation," *IEEE Trans. on Services Computing*, vol. 14, no. 3, pp. 889–902, May–Jun. 2021.

[4] Y. Zhang, P. Zhang, Y. Luo, and L. Ji, "Towards efficient, credible and privacy-preserving service QoS prediction in unreliable mobile edge environments," in *Proc. of 2020 In. Symp. on Reliable Distributed Systems (SRDS)*, pp. 309–318, Nov. 2020.

[5] R. Xiong, J. Wang, Z. Li, B. Li, and P. Hung, "Personalized LSTM based matrix factorization for online QoS prediction," in *Proc. of 2018 IEEE In. Conf. on Web Services (ICWS)*, pp. 34–41, Sept. 2018.

[6] X. Chen, B. Li, J. Wang, Y. Zhao, and Y. Xiong, " Integrating EMD with multivariate LSTM for time series QoS prediction," in *Proc. of 2020 IEEE In. Conf. on Web Services (ICWS)*, pp. 58–65, Dec. 2020.

[7] Z. Zheng, Y. Zhang, and M. Lyu, "Investigating QoS of real-world web services," *IEEE Trans. on Services Computing*, vol. 7, no. 1, pp. 32–39, Jan.–Mar. 2014.

[8] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," *Advances Neural Information Processing Systems*, vol. 20, no. 1, pp. 1257–1264, Dec. 2007.

[9] G. Zou, *et al*., "NDMF: Neighborhood-integrated deep matrix factorization for service QoS prediction," *IEEE Trans. on Network and Service Management*, vol. 17, no. 4, pp. 2717–2730, Dec. 2020.

[10] Y. Zhang, K. Wang, Q. He, F. Chen, S. Deng, Z. Zheng, and Y. Yang, "Covering-based web service quality prediction via neighborhood-aware matrix factorization, " *IEEE Trans. on Services Computing*, pp. 1–12, Jan. 2019. DOI: 10.1109/TSC.2019.2891517.

[11] W. Yue, Z. Wang, J. Zhang, and X. Liu, "An overview of recommendation techniques and their applications in healthcare," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 4, pp. 701–717, Apr. 2021.

[12] S. Imran, T. Mahmood, A. Morshed, and T. Sellis, "Big data analytics in healthcare − A systematic literature review and roadmap for practical implementation," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 1, pp. 1–22, Jan. 2021.