High-accuracy Reliability Prediction Approach for Blockchain Services under BaaS

Jianlong Xu[1], Zicong Zhuang[1], Kun Wang[1], and Wei Liang[2]

[1]Colleage of Engineering,Shantou University, Shantou, China
[2]College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
{$xujianlong, 19zczhuang, 19kwang$}@stu.edu.cn, weiliang99@hnu.edu.cn

Abstract. With the continuous evolution of service-oriented computingparadigms, blockchain as a service (BaaS) has emerged, which is crucial in the development of blockchain-based applications. To build a high-quality blockchain-based system, users must select highly reliable blockchain services (peers) with excellent quality of service (QoS). However, owing to the large number of services and the sparsity of personalized QoS data, it is difficult to select the optimal services. Hence, we propose a QoS-based blockchain service reliability prediction framework ($BSRPF$) under BaaS. In this framework, we employ a matrix factorization-based method to perform accurate QoS prediction. To validate BSRPF, we conducted experiments based on large-scale real-world data, and the results show that BSRPF achieves high prediction accuracy and outperforms other popular methods.

Keywords: Blockchain, BaaS, Matrix factorization, Reliability prediction

1 Introduction

As an emerging distributed ledger technology, blockchain has received significant attention[1], and various blockchain-based applications are developing rapidly, such as smart contracts[2], Internet of Things[3],and security services[4]. Meanwhile, with the continuous evolution of service-oriented computing (SOC) paradigms, blockchain as a service (BaaS) has emerged, which can improve the productivity of blockchain-based applications development. BaaS is a concept mainly proposed by Microsoft and IBM[5], which aims to execute a certain blockchain node. In BaaS environments, users can quickly design blockchain-based applications by invoking a series of blockchain services (known as blockchain peers). These services are network-based software components that can provide search queries, transaction submissions, data storage, data analysis, and computation services. These services can be either centralized or decentralized to help developers (users) validate their concepts and models more quickly.

However, many services with similar functions exist on the Internet, and the method to select optimal blockchain services to build high-performance blockchain-based systems or applications is the main challenge for users. As shown in Figure 1, for improved performance, users must select suitable blockchain services to form a better blockchain.

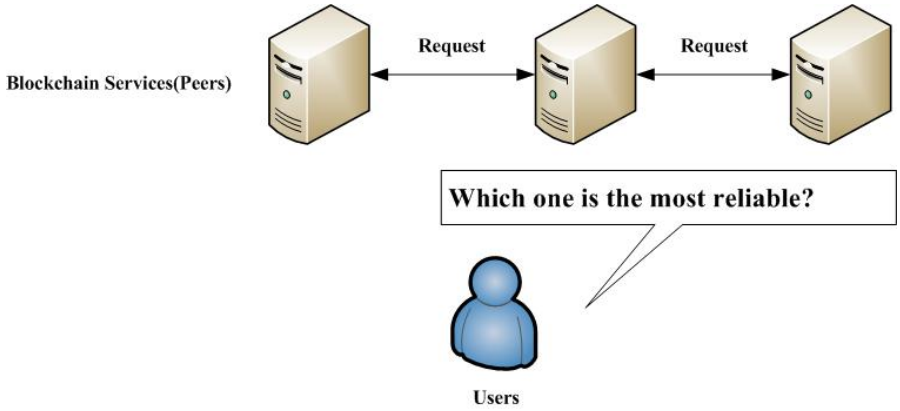

Fig. 1: An Example of Blockchain Services Selection

As a nonfunctional requirement, quality-of-service (QoS) is the most widely used evaluation criterion of optimal services in SOC [6][7]. To obtain the best service for users, one simple solution is to invoke each candidate service to obtain the QoS values (e.g., throughput and latency) of each service and then select the service with the best QoS values. However, this is both time and resource consuming. Meanwhile, owing to the unpredictability of the Internet environment, QoS values may vary for different users when the same service is invoked. Typically, another solution for obtaining unknown QoS is to perform predictions using historical QoS data at the client side, known as personalized QoS prediction[6][7][8]. Therefore, for blockchain service selection, the critical step is to obtain accurate QoS values of candidate services through personalized QoS prediction.

To obtain accurate QoS values, many approaches have been proposed by the service computing community in recent years[9][10]. ~~The~~ popular approaches are based on collaborative filtering, which can be categorized into memory-based collaborative filtering (CF) and model-based CF. As a model-based CF method, matrix factorization (MF) has achieved great success and has been employed for QoS value prediction in web services and cloud services, among others[9][10]. Inspired by the accomplishment of a matrix factorization algorithm for personalized QoS predictions, we herein propose a personalized QoS prediction framework for blockchain services ($BSRPF$) under BaaS. We utilized the QoS values (success rate data), which are from geographically distributed real-world blockchain services, and conducted extensive experiments; the experimental results demonstrated the effectiveness and efficiency of our approach.

The contributions of this paper can be summarized as follows: 1) The problem of QoS personalized prediction for blockchain services is identified and explained; 2) a QoS-based blockchain service reliability prediction framework for blockchain services ($BSRPF$) under BaaS is proposed, and an MF is employed to perform accurate QoS prediction in this framework; 3) BSRPF is compared with other methods and different factors affecting the prediction model for blockchain services are analyzed, in which the results demonstrate the superiority of our method.

The remainder of this paper is organized as follows: Section 2 presents a discussion on related work. Section 3 presents our prediction model BSRPF. Section 4 describes our experiments and results in detail, and Section 5 concludes the paper and presents a discussion of future work.

2 Related Work

This section introduces related work in blockchain reliability prediction, including traditional software reliability research and blockchain-related research.

Regarding traditional software reliability research, QoS prediction has been widely investigated in the past decade. CF methods are the most typical techniques for personalized QoS predictions. The main idea of CF is to determine a group of similar users or services based on the Pearson correlation coefficient (PCC). Subsequently, we predict the results according to the past QoS value. Typically, studies regarding CF starts with memory-based methods. Memory-based CF can be classified into user-[11] and item-based CF[12]. User-based CF searches a set of nearest neighboring users with similar interests using the PCC, and item-based CF calculates the similarity of the items. Zheng et al.[13] proposed a neighborhood-based hybrid model that combines user- and item-based CF approaches. Later, model-based methods emerged. Model-based methods include CF based on a clustering model[14] and a latent semantic model[15]. MF is a model-based CF method that decomposes the user-item scoring matrix into a combination of several parts[16, 17]. Zheng et al.[9]adopted a probability matrix factorization (PMF)-based approach for reliable, personalized predictions.

With regards to blockchain reliability, Xiao et al.[18] proposed a reliability-based evaluation method for circuit units, which avoids security and privacy vulnerable to hardware errors. Zheng et al.[19] proposed an approach to detect Ponzi schemes on blockchain using data-mining and machine-learning methods, which were used to detect Ponzi schemes even at the moment of its creation. Lei et al.[20] presented a reputation-based Byzantine fault tolerance algorithm that incorporates a reputation model to evaluate the operations of each node in a consensus process. Kalodner et al.[21] proposed a multifunctional open-source software platform that supports different blockchain and analysis tasks. It parses the data of a P2P network and original blockchain data, and the analysis results are provided for users to analyze. Inspired by blockchain with distributed ledger technology, Du et al.[22] distributed consensus mechanisms and encryption algorithms, as well as proposed a personalized QoS prediction method for web services based on blockchain-based MF, which was more effective than traditional techniques. In recent years, BaaS has received significant attention from many scholars. For example, Lu et.al [5] proposed a unified blockchain-as-a-service platform, which aims to support both the design and deployment of blockchain-based applications. IBM proposed IBM Hyperledger1 [1]for BaaS deployment solutions.

Inspired by the studies above, we herein study the reliability prediction method for blockchain services.

3 Reliability Prediction Method for Blockchain Services

In this section, the methodology of BSRPF is introduced, including the problem formulation and reliability prediction framework for blockchain services.

3.1 Framework of Blockchain Services Reliability Prediction

We propose a QoS-based blockchain service reliability prediction framework for blockchain services ($BSRPF$) under BaaS, as shown in Fig. 2. Our framework comprises four parts: collection of QoS values, success rate calculation, MF, and reliability prediction.

The framework includes the following main steps:

[1] https://www.ibm.com/blockchain/platform/

1) Users send requests to the blockchain services (peers), and the blockchain services respond to the requests and return the feedback QoS data to the users. The users submit these feedback data to the prediction server.

2) In the prediction server, the data collector collects QoS data regarding the success or failure of the request; next, the success rate data calculation module calculates the success rate based on the submitted data. The calculation results are used form the user-service matrix of the success rate for MF.

3) Because users cannot request all services, this service matrix will not be extremely dense. With these known success rates, we can predict the unknown success rate values based on the MF module.

4) After MF, the request success rates of all users for all services are obtained, and the reliability of each blockchain service (peer) can be calculated by the reliability calculation module.

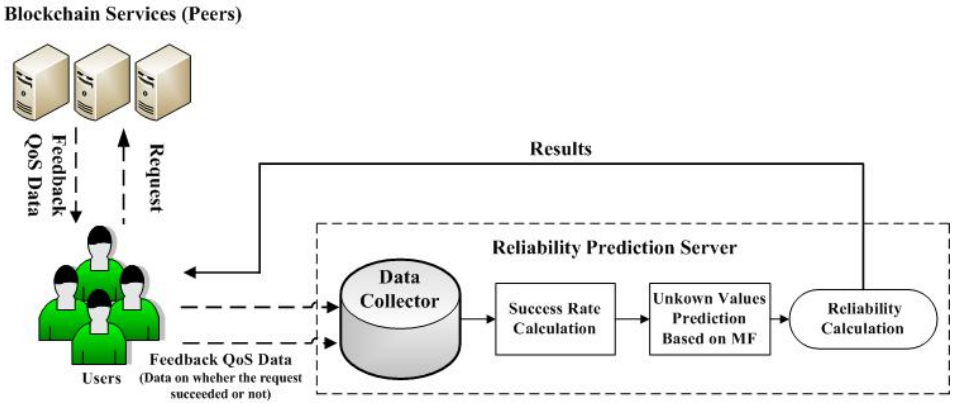

Fig. 2: Framework of Blockchain Services Reliability Prediction (BSRPF) under BaaS

3.2 Reliability Prediction Method for Blockchain Services

We argue that blockchain services are the nodes or blockchain peers that can be composed of the blockchain application, and blockchain users are the developers of blockchain applications that can invoke blockchain services. For a group of users, each user can send a request to each blockchain service (peer), and the result of the request is the success rate. Because numerous blockchain peers exist in the real world, users cannot send a request to each peer; therefore, the success rate of users for blockchain services without requests should be predicted according to the similarity between peers.

Success rate calculation. After collecting data that signify the success or failure of a request, we can use these data to calculate the successful request rate for invoked blockchain services. First, we set a value as MaxBlockBack to denote the extreme value for the block backwardness of the peer in the blockchain. Subsequently, we set a value as MaxRTT to represent the maximum round-trip time for the peer[23]. The successful request rate can be calculated as follows:

For each user U_i and peer P_j, we used a counter for successful requests as $SuccessRequest_{i,j}$, and a failure counter as $FailureRequest_{i,j}$. As per[23], each batch of user sends requests to peer, and the peer will respond successfully if and only if it returns the correct block and the recent block height in time. If peer P_j responds successfully, then it is counted into $SuccessRequest_{i,j}$,

else, it is counted into $FailureRequest_{i,j}$. The successful request rate of U_i and P_j is calculated using Eq.(1).

$$SuccessRate_{i,j} = \frac{SuccessRequest_{i,j}}{SuccessRequest_{i,j} + FailureRequest_{i,j}} \tag{1}$$

After calculating the successful request rate, we used the success rate to predict the unknown entries in the matrix and predicted the reliability of the blockchain services.

Low-Rank Matrix Factorization. Given a set of N users $U = \{u_1, u_2, , u_n\}$ and a set of M peers $P = \{p_1, p_2, , p_m\}$, N and P can form an $N \times M$ matrix R. The entry in R is indicated as r_{ij} which is on the i^{th} row j^{th} column r_{ij}, representing the success rate of $user_i$'s request to $peer_j$. The value of r_{ij} is equal to null when u_i does not request p_j; otherwise it is not null. In this study, we used user success request rate as the QoS data.

According to the success rate matrix, we discovered that the distribution of the data deviated significantly. Therefore, applying the MF model directly to the original data may significantly reduce the prediction accuracy. To solve this problem, we applied the BoxCox transformation, a classical data transformation method, to the success rate matrix. This technique is used to stabilize data variance and yield data that are closer to a normal distribution to adapt to the matrix decomposition hypothesis. The transformation is rank preserving and is performed using a continuous power function defined as follows:

$$boxcox(x) = \begin{cases} (x^{\alpha} - 1)/\alpha & if \alpha \neq 0 \\ \log(x) & if \alpha = 0 \end{cases} \tag{2}$$

where the parameter α controls the extent of the transformation. For simplicity, we denote $\widehat{R_{ij}} = \text{boxcox}(R_{ij})$, $\widehat{R_{max}} = \text{boxcox}(R_{max})$ and $\widehat{R_{min}} = \text{boxcox}(R_{min})$ due to its monotonously nondecreasing property of Box-Cox transformation. R_{max} and R_{min} are the maximal and minimal values respectively. Similarly, $\widehat{R_{max}}$ and $\widehat{R_{min}}$ are the maximal and minimal values after data transformation. Then we map the data into the range [0,1] by linear normalization.

$$r_{ij} = (\widehat{R_{ij}} - \widehat{R_{min}})/(\widehat{R_{max}} - \widehat{R_{min}}) \tag{3}$$

To predict unknown entries in the matrix, it is necessary to fit the matrix into the factorization model and then use the factorization model for subsequent predictions. MF is a typical factor analysis model. In the same feature space, a high-dimensional matrix is decomposed into two low-dimensional feature matrices.

In this study, the success rate matrix $R \in \mathbb{R}^{N \times M}$ is assumed to have a low-rank structure, that is, it has a rank of $K \ll \min\{M, N\}$. R can be decomposed into two rank-K matrices $U \in \mathbb{R}^{K \times N}$ and $P \in \mathbb{R}^{K \times M}$. R can be calculated as $R = U^T P$. The column vectors of U and P have a natural interpretation. The i^{th} column vector U_i of U is the potential factor that determines the behavior of user i and the j^{th} column vector P_j of P is the potential factor that determines the features of ij. The dot product ${U_i}^T P_j$ is the model predicted score of u_i's success rate on ij.

We adopted PMF[24] and a probabilistic linear model with Gaussian observation noise. Our target was to maximize the posterior probability and minimize the following loss function

$$\mathcal{L} = \frac{1}{2} \sum_{i=1} \sum_{j=1} (R_{ij} - U_i^T P_j)^2 + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_P}{2} \|P\|_F^2 \tag{4}$$

In Eq.(4), R_{ij} is the available entry in the matrix. The first term is the squared loss. λ_S and λ_U are both small positive decimal numbers to control the extent of regularization which

can avoid over-fitting problems, and $\|.\|_F^2$ represents the Frobenius norm. To minimize the loss function, we computed the gradients of the loss function with respect to U_i and P_j as follows:

$$\frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j \in \mathcal{L}_i} (U_i^T P_j - r_{ij}) P_j + \lambda_U U_i \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial P_j} = \sum_{j \in \mathcal{U}_i} (U_i^T P_j - r_{ij}) U_i + \lambda_P P_j \tag{6}$$

Then alternative update on U_i and P_j can be done through:

$$U_i \leftarrow U_i - \eta \frac{\partial \mathcal{L}}{\partial U_i} \tag{7}$$

$$P_j \leftarrow Pj - \eta \frac{\partial \mathcal{L}}{\partial P_j} \tag{8}$$

In Equ.(7) and (8), η is the learning rate to control each iteration's change. After training, the prediction of u_i's success request rate of p_j as predicted by the model is the dot product ${U_i}^T P_j$:

$$SuccessRate_{i,j} \approx \widetilde{SuccessRate}_{i,j} = U_i^T P_j \tag{9}$$

Predict Reliability. After completing the steps above, we obtained the predicted success rate from blockchain requester U_i to blockchain peer P_j. To predict the reliability of P_j observed by U_i, we adopted a typically used exponential reliability function[25]:

$$Reliability_{i,j}(t) = e^{-(1-SuccessRate_{i,j}) \times t} \tag{10}$$

4 Experiment and Result

In this section, we describe our experiments to verify BSRPF and then discuss the parameters in the proposed model. By comparing the results with other methods and different parameters, we demonstrate the high accuracy of BSRPF.

4.1 Dataset

In our experiment, we used the real-world dataset proposed in reference [23]. It includes a 100×200 Success Rate matrix of 100 blockchain requesters and 200 blockchain peers. The blockchain peers are from 21 countries and the requesters are from 15 countries. In this dataset, more than 2,000,000 test cases were collected. To make our experiment more realistic, we mapped the success rate values from different requesters to peers into [0,1].

4.2 Evaluation Metrics

In this experiment, we employ the root mean square error (RMSE) to measure the difference between the predicted and the measured values. It is defined as:

$$RMSE = \sqrt{\frac{\sum_{i,j}(R_{i,j} - \widehat{R}_{i,j})^2}{N}} \tag{11}$$

Where $R_{i,j}$ is the known value, which denotes the success rate of requester i to peer j, and $\hat{R_{i,j}}$ is the corresponding predicted value. N is the number of predicted values. The smaller the RMSE value, the higher the prediction accuracy we get.

4.3 Performance Comparison

To evaluate the performance of our methods, we compared our method with three other methods for reliability prediction: user-based approach using PCC (UPCC)[26], item-based approach using PCC (IPCC)[27], and user-item-based approach (UIPCC)[28]. UPCC is a collaborative filtering method based on the request of the similarity between blockchain users to predict unknown values, and the request of similar blockchain users' PCC. The IPCC only employs similar blockchain peers for the prediction. UIPCC is a combination of UPCC and IPCC. In the experiment, these three methods were compared with our methods to predict the same training success rate matrix.

For each round, we randomly deleted the entries in the generated success rate matrix to transfer it to the target density, and the deleted entries were set as the test values. The density was set as density = $\{30\%, 40\%, 50\%, 60\%, 70\%, 80\%\}$. We set the dimensionality K = 5 to decompose the matrix into two rank-K matrices. We set $\langle MaxBlockBack = 12, MaxRTT = 1000\rangle$ to evaluate the accuracy for situations with high requirements for confirming blockchain data. We set $\langle MaxBlockBack = 100, MaxRTT = 5000\rangle$ to evaluate the accuracy for daily usages that have high tolerances for block backwardness and latency. The experimental results are shown in Tables 1 and 2.

Table 1: Accuracy Comparison of RMSE of Blockchain Reliability Prediction Method (MaxBlockBack=12, MaxRTT=1000)

Method	Density = 30%	Density = 40%	Density = 50%	Density = 60%	Density = 70%
UPCC	0.3646	0.3601	0.3623	0.3583	0.3547
IPCC	0.1022	0.1001	0.0963	0.0942	0.0889
UIPCC	0.1069	0.1045	0.1011	0.0985	0.0937
BSRPF	**0.0946**	**0.0895**	**0.0867**	**0.0818**	**0.0790**

Table 2: Accuracy Comparison of RMSE of Blockchain Reliability Prediction Mothod (MaxBlockBack=100, MaxRTT=5000)

Method	Density = 30%	Density = 40%	Density = 50%	Density = 60%	Density = 70%
UPCC	0.4597	0.4566	0.4591	0.4550	0.4536
IPCC	0.0898	0.0858	0.0861	0.0819	0.0801
UIPCC	0.1003	0.0967	0.0969	0.0931	0.0916
BSRPF	**0.0890**	**0.0855**	**0.0851**	**0.0805**	**0.0801**

From Tables 1 and 2, we can infer the following:

1) BSRPF obtained lower RMSE values than the other methods in terms of the success rate, with different matrix densities. This indicates that BSRPF is more accurate than existing methods and further verifies the effectiveness of our method. Concretely, compared with the UIPCC, BSRPF achieved an improvement of 14.6% on average, as shown in Table 1; and 12.2%, as shown in Table 2.

2) Compared with the UPCC, IPCC, and UIPCC, BSRPF yielded more accurate predictions. This occurred because the PCC methods exclusively use the information for prediction, which is similar to the requesters and peers, whereas BSRPF uses all available information in the success rate matrix.

3) With the increase in density, BSRPF is more accurate than the PCC-based methods. For example, it achieved 8.0% higher accuracy than UIPCC when the matrix density was 30%, and 12.5% more accuracy when the matrix density was 70%.

4.4 Impact of Matrix Density

Matrix density is the percentage of unknown entries in the matrix, which indicates the amount of available information for performing predictions. To demonstrate the effect of matrix density, we set the matrix density from 30% to 70% with a step increase of 10%.

As shown in Fig.3, with the increase in the matrix density, the value of RMSE decreased slowly. This means that more accurate prediction results can be achieved by obtaining more blockchain information.

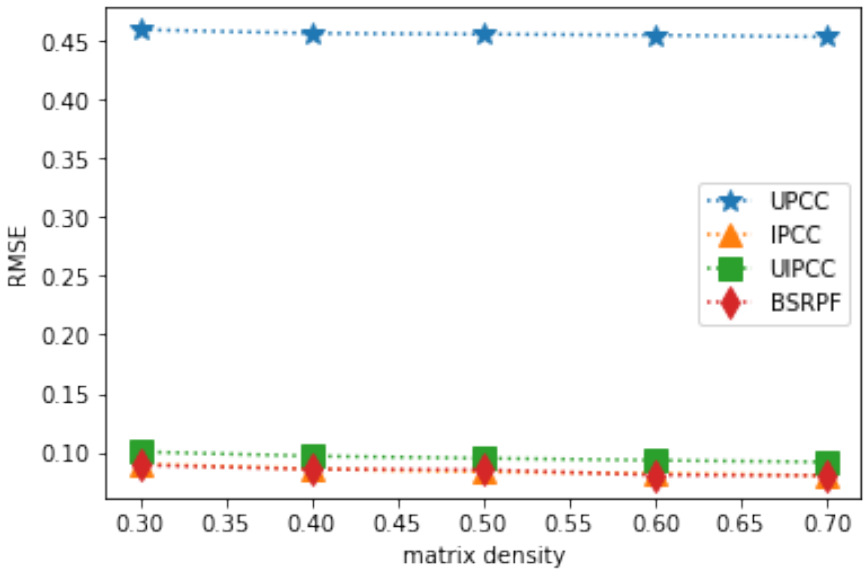

Fig. 3: Impact of Matrix Density($MaxBlockBack = 100, MaxRTT = 5000, \lambda = 1, Dimensionality = 5$)

5 Conclusion and Future Work

Selecting suitable blockchain services to build blockchain-based applications is crucial in BaaS. To obtain highly reliable blockchain services, we present a personalized prediction framework for blockchain services named BSRPF. In this framework, we first calculate the known success rate. Subsequently, based on the success rate matrix, we employed MF to predict the unknown success rate values and finally calculate the reliability of the blockchain service. Extensive experiments were conducted on a real-world dataset and compared with other methods, which indicated that our framework yielded more accurate prediction results.

In future research, to achieve better performances, we will extract factors of the request success rate, such as the round-trip time of the request to the peer and block hash, and then import these factors into our model to improve the accuracy.

6 Acknowledgment

This research was financially supported by the National Natural Science Foundation of China (No.61702318), the Shantou University Scientific Research Start-up Fund Project (No.NTF18024),2018 Provincial and Municipal Vertical Coordination Management Science and Technology Planning Project (No.180917124960518), 2019 Guangdong province special fund for science and technology ("major special projects + task list") project, and in part by 2019 Li Ka-shing foundation crossover research project.

References

1. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services **14**, 352 (2018)
2. Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C.: Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 839–858. IEEE Computer Society, Los Alamitos, CA, USA (2016)
3. Liang, W., Tang, M., Long, J., Peng, X., Xu, J., Li, K.: A Secure FaBric Blockchain-Based Data Transmission Technique for Industrial Internet-of-Things. IEEE Trans. Industrial Informatics **15**, 3582–3592 (2019)
4. Liang, W., Fan, Y., Li, K., Zhang, D., Gaudiot, J.: Secure Data Storage and Recovery in Industrial Blockchain Network Environments. IEEE Transactions on Industrial Informatics , 1–1 (2020) https://doi.org/10.1109/TII.2020.2966069.
5. Lu, Q., Liu, Y., Weber, I., Zhu, L., Zhang, W.: uBaaS: A unified blockchain as a service platform. Future Generation Computer Systems **101**, 564–575 (2019)
6. Guo, L., Mu, D., Cai, X., Tian, G., Hao, F.: Personalized QoS Prediction for Service Recommendation With a Service-Oriented Tensor Model. IEEE Access **7**, 55721–55731 (2019)
7. Wu, J., Chen, L., Feng, Y., Zheng, Z., Zhou, M., Wu, Z.: Predicting Quality of Service for Selection by Neighborhood-Based Collaborative Filtering. Systems, Man, and Cybernetics: Systems, IEEE Transactions on **43**, 428–439 (2013)
8. Yang, Y., Zheng, Z., Niu, X., Tang, M., Lu, Y., Liao, X.: A Location-Based Factorization Machine Model for Web Service QoS Prediction. IEEE Transactions on Services Computing. , 1–1 (2018) https://doi.org/10.1109/TSC.2018.2876532
9. Zheng, Z., Lyu, M.R.: Personalized Reliability Prediction of Web Services. ACM Transactions on Software Engineering and Methodology (TOSEM). **22**, 25–25 (2013)
10. Li, S., Wen, J., Luo, F., Cheng, T., Xiong, Q.: A Location and Reputation Aware Matrix Factorization Approach for Personalized Quality of Service Prediction. In: 2017 IEEE International Conference on Web Services (ICWS), pp. 652–659. IEEE Computer Society, Los Alamitos, CA, USA (2017)
11. Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., Mei, H.: Personalized QoS Prediction forWeb Services via Collaborative Filtering. In: IEEE International Conference on Web Services (ICWS 2007), pp. 439–446. IEEE Computer Society, Los Alamitos, CA, USA (2007)
12. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing. **7**, 76–80 (2003)
13. Zheng, Z., Ma, H., Lyu, M.R., King, I.: WSRec: A Collaborative Filtering Based Web Service Recommender System. In: 2009 IEEE International Conference on Web Services, pp. 437–444. IEEE Computer Society, Los Alamitos, CA, USA (2009)
14. Zhu, J., Kang, Y., Zheng, Z., Lyu, M.R.: A Clustering-Based QoS Prediction Approach for Web Service Recommendation. In: 2012 IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, pp. 93–98. IEEE Computer Society, Los Alamitos, CA, USA (2012)

15. Hoffman, T.: Latent semantic models for collaborative filtering. ACM Transactions on Information Systems (TOIS). **22**, 89–115 (2004)
16. Rennie, J.D.M., Srebro, N.: Fast Maximum Margin Matrix Factorization for Collaborative Prediction. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 713–719. Association for Computing Machinery, New York, NY, USA (2005)
17. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. Proceedings of the 25th International Conference on Machine Learning. **25**, 880–887 (2008)
18. Xiao, J., Lou, J., Jiang, J., Li, X., Yang, X., Huang, Y.: Blockchain Architecture Reliability-Based Measurement for Circuit Unit Importance. IEEE Access. **pp**, 1–1 (2018)
19. Chen, W., Zheng, Z., Cui, J., Ngai, E.C.H., Zheng, P., Zhou, Y.: Detecting Ponzi Schemes on Ethereum: Towards Healthier Blockchain Technology. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, pp. 1409–1418. ACM, Los Alamitos, CA, USA (2018)
20. Lei, K., Zhang, Q., Xu, L., Qi, Z.: Reputation-Based Byzantine Fault-Tolerance for Consortium Blockchain. In: 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), pp. 604–611. IEEE Computer Society, Lyon, France (2018)
21. Kalodner, H., Goldfeder, S., Chator, A., Möser, M., Narayanan, A.: BlockSci: Design and applications of a blockchain analysis platform. arXiv:Cryptography and Security. **pp**, 1–14 (2017)
22. Cai, W., Du, X., Xu, J.: A Personalized QoS Prediction Method for Web Services via Blockchain-Based Matrix Factorization. Sensors. **19**, 2749–2749 (2019)
23. Zheng, P., Zheng, Z., Chen, L.: Selecting Reliable Blockchain Peers via Hybrid Blockchain Reliability Prediction. CoRR. **abs/1910.14614**, 1–11 (2019)
24. Salakhutdinov, R., Mnih, A.: Probabilistic Matrix Factorization. In: Proceedings of the 20th International Conference on Neural Information Processing Systems, pp. 1257–1264. Curran Associates Inc., Red Hook, NY, USA (2007)
25. Lyu, M.R.: Handbook of Software Reliability Engineering. Software IEEE. **18**, 98–98 (1996)
26. Breese, J., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 43–52. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2013)
27. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295. Association for Computing Machinery, New York, NY, USA (2001)
28. Zheng, Z., Lyu, M.R.: Collaborative reliability prediction of service-oriented systems. In: Software Engineering, International Conference on, pp. 35–44. IEEE Computer Society, Los Alamitos, CA, USA (2010)