Double Security Guarantee: Protecting User Privacy and Model Security in QoS Prediction

Jianlong Xu
College of Engineering
Shantou University
Shantou, China
xujianlong@stu.edu.cn

Zhuo Xu
College of Engineering
Shantou University
Shantou, China
20zxu3@stu.edu.cn

Jian Lin
College of Engineering
Shantou University
Shantou, China
20jlin3@stu.edu.cn

Weiwei She
College of Engineering
Shantou University
Shantou, China
17wwshe@stu.edu.cn

***Abstract*—Quality of Service (QoS) prediction has played an important role in selecting the optimal cloud service for users, and how to protect users' privacy with high prediction accuracy has become the focus of attention in service computing. Although federated learning (FL) methods have been widely applied to protect user privacy, when a federated learning model is attacked by malicious users, it may lead to wrong prediction results. In order to protect both user privacy security and prediction model security, we propose a double security guaranteed matrix factorization model named DSGMF. In this model, we design a global gradient allocation method though contribution-based rewards. Meanwhile, to identify and remove potential free-riders, we explore free-rider attack and employ reputation-based detection method. Our proposed model is evaluated on a real-world QoS dataset, and the experimental results validate the effectiveness of our approach.**

***Index Terms*—QoS Prediction; Cloud Services; Privacy Preservation; Federated Learning; Model Security**

I. Introduction

With the development of cloud computing technology, Software as a Service (SaaS) is widely used as a way to deliver and license software directly over the internet [1]. Service-Oriented Architecture (SOA) allows software developers to build new online applications by combining different cloud services [2]. For example, an online travel planning application might be a combination of services such as route recommendations, ticket booking, service ratings, weather forecasts, electronic payments, and so on. These services are often not unique and a very large number of similar services exist that can be selected. Quality of Service (QoS) is often used to describe the non-functional attributes of a service. The values of QoS may vary due to factors such as geographical location and network environment.

Based on the QoS values observed by users, developers can identify high-quality services for use in combinations. However, as a single user can only observe QoS values that have been invoked locally, it is difficult to obtain QoS values for all candidate services.

Collaborative filtering (CF) is a widely used QoS prediction technique for web services and can be classified as memory-based CF, model-based CF, and other hybrids CF methods [3]. By mining the connections between users and users, users and services, and services and services, the QoS values of candidate services can be effectively predicted. Among them, matrix factorization (MF), as a typical model-based CF method, has received much attention from researchers [4].

However, it is important to note that traditional MF methods require participating users to submit their original QoS records, which pose threats to user privacy. As a result, some users may be reluctant to contribute their valuable data to participate in the construction of the prediction model. With user privacy in mind, many approaches have been proposed, such as encryption [5], obfuscation [6], anonymization [7] and other methods of manipulating data. While these methods serve to protect user privacy, they on the one hand increase the overhead of data transfer and put pressure on the central server. On the other hand, the prediction accuracy could be compromised as the data involved in the prediction is ambiguous.

The role of federated learning (FL) in privacy preservation is noticed [8]. FL is a distributed learning approach that does not require users to submit raw data. In a federated matrix factorization model, the user is part of the model (the local model) whose only exchange with the central server is the parameters and no longer the user's QoS values. However, traditional approaches to federated matrix factorization have been too optimistic in trusting all users involved in the computation, ignoring possible threats to model security such as free-rider attacks from adversarial users.

The free-rider attack comes from users who want to benefit from the global model to train a local model, but the may not want to contribute their own gradients to the global due to their privacy or computational resource considerations. Such users are also known as free-riders.

Therefore, we design a double security guaranteed matrix factorization (DSGMF) model to simultaneously protect user privacy and model security. Specifically, using federated learning techniques to predict QoS while protecting user privacy, and identifying latent attackers based on user reputation during central server aggregation to guarantee model security. During server aggregation, we calculated the user's contribution for generating gradients rewards.

In summary, the contributions of this paper are as follows:

- We propose a double security guaranteed matrix factorization model. Our approach can guarantee both user privacy security and prediction model security.

2474-2473/22/$31.00 ©2022 IEEE
DOI 10.1109/SCC55611.2022.00031

- We explore the effectiveness of free-rider attacks on the federated matrix factorization model and propose using user reputation-based detection methods to identify adversarial users and guarantee the security of the model.
- We conduct experiments on a large-scale real-world dataset, and the experimental results show that the proposed DSGMF can abate the free-rider threat to the model compared to the conventional federated matrix factorization approach.

The rest of the paper is organized as follows: Section II presents the background and related work. Section III demonstrates the proposed DSGMF in detail. Section IV describes our experiment setting and results. Finally, Section V concludes the paper and looks at future work.

II. Background and Related Work

Accurate QoS data provides a reference for recommending quality services to users. How to obtain (predict) unknown QoS data for candidate services has received a lot of attention from researchers. Early studies were conducted with a centralized mindset where they assumed that QoS values for services were easily available from third-party organizations or service providers. These approaches can be divided into collaborative and non-collaborative filtering approaches. Matrix factorization has received a great deal of attention in CF-based approaches, with probabilistic matrix factorization (PMF) [9], non-negative matrix factorization (NMF) [10] and, for online scenarios, AMF approaches being proposed [11]. All of these MF approaches predict the QoS of candidate services with QoS data collected from all users on the server-side. They focus on improving the accuracy of the prediction results while ignore the risk of user privacy leakage during the collection process and the impact of adversarial users on the model.

With the increasing emphasis on privacy, QoS prediction methods with privacy protection are included in user requirements. Zhu et al. designed a simple but effective privacy-preserving framework by applying data obfuscation techniques, under this framework they developed P-UIPCC and P-PMF [6]. Qi et al. used a locally sensitive hashing approach to predict QoS values [12]. Zhang et al. used an improved differential privacy approach to dynamically disguise the raw QoS data in an edge environment to protect user data privacy [13]. The above approaches are more suitable for centralized training platforms with powerful computation resources rather than for general users.

Federated Learning, a distributed learning framework which is suitable for individual users with limited resources and has good applications in privacy protection [8]. Zhao et al. proposed a privacy-preserving framework for page recommendations based on federated learning and model-agnostic meta-learning (MAML), called Fed4Rec [14]. Zhang et al. designed a federated learning-based QoS prediction method named FMF [15]. They further improved the efficiency of prediction by reducing the system overhead, and this efficient and privacy-preserving approach is called EFMF.

The security of predictive models is as important as the privacy concerns of users. Ye et al. note that outliers in QoS records may significantly degrade prediction performance. They propose an outlier-resilient QoS prediction method, which uses Cauchy loss to measure the difference between the observed QoS values and the predicted values [16]. Considering the impact of untrustworthy users on the prediction results in a real environment, Xu et al. proposed a highly plausible method called Reputation-based Matrix Factorization (RMF) for predicting unknown Web service QoS values [4].

Although FL is effective in protecting user privacy, the architectures are vulnerable to active attacks by internal participants due to the involvement of multiple local participants with unknown reliability. An attacker can act as an innocent participant in federated learning by uploading poisoned updates to the server so that it can easily affect the performance of the global model. Cao et al. proposed a model of fake client poisoning attack (MPAF) targeting federated learning [17].

Free-rider is a widely studied attack in peer-to-peer systems [18]. Free riders generally refer to individuals who benefit from assets of public nature but do not want to pay for them due to privacy or computational cost concerns. In QoS prediction, we can regard a free-rider as a user who wants to access the global model to train local data but does not want to upload the real local model. These free-riders often upload random or spurious gradient data, which can cause damage to the global model. Therefore, how to defend against this possible threat to the models is an issue that needs to be addressed in QoS prediction.

III. Double Security Guaranteed Matrix Factorization

A. Conventional QoS Prediction Using Matrix Factorization

In general, for a QoS matrix $\in \mathbb{R}^{m\times n}$ with m users and n services, it can be factorized into a user latent matrix $U \in \mathbb{R}^{k\times m}$ and a service latent matrix $S \in \mathbb{R}^{k\times n}$ [11]. Therefore, by the inner product of U and S, it is possible to fit the QoS matrix R well and predict the missing QoS, e.g., $\hat{R}_{ij} = U_i^T S_j$. To achieve $R \approx \hat{R}$, we resolve to minimize the following loss function:

$$\mathcal{L} = \frac{1}{2}\sum_{i,j} I_{ij}(R_{ij} - U_i^T S_j)^2 + \frac{1}{2}(\sum_i \|U_i\|_2^2 + \sum_j \|S_j\|_2^2), \tag{1}$$

where I_{ij} is the flag value and $I_{ij} = 1$ means that R_{ij} is recorded, otherwise it is 0. The first part of the equation calculates the sum of squared errors between the observed QoS (R_{ij}) and the predicted QoS ($\hat{R}_{ij} = U_i^T S_j$). In the remaining part, $\|\boldsymbol{.}\|$denotes the Euclidean norm, which is used to prevent overfitting. λ is the parameter to control the extent of regularization.

To derive the solution of U and S, gradient decent is usually employed by an iterative process until convergence:

$$U_i \leftarrow U_i - \eta\frac{\partial \mathcal{L}}{\partial U_i} \quad S_j \leftarrow S_j - \eta\frac{\partial \mathcal{L}}{\partial S_j}, \tag{2}$$

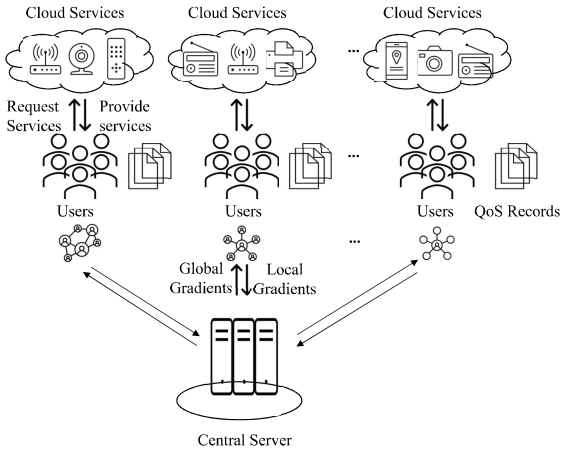

Fig. 1. Privacy-preserving Federated Matrix Factorization Framework.

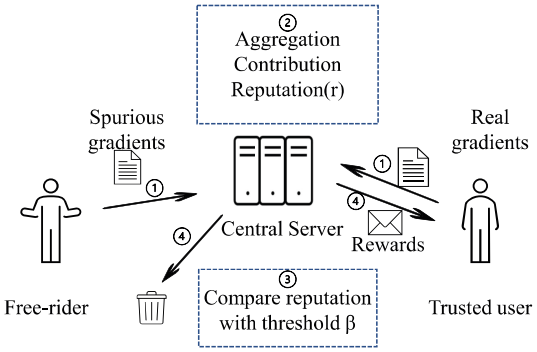

Fig. 2. Reputation-based Model Security Framework.

where η is the learning rate that controls the step size at each iteration.

B. Privacy Security Guaranteed via Federated MF

Conventional matrix factorization is performed after centrally collecting QoS data from users, which increases the risk of user privacy exposure. In our proposed DSGMF, we use federated learning (FL) to guarantee user privacy security. The idea of FL is to decentralize the raw data from the central server to the end devices [15]. Under the federated learning framework, only model parameters are exchanged between local and central, and sensitive data of users are not exposed.

Fig. 1 shows the privacy-preserving federated matrix factorization framework. It can be seen that the decentralized users request to the cloud service providers (CSPs) to invoke various services. The CSPs respond and provide the appropriate services. When the services provided by the CSPs are obtained, the user also observes the corresponding QoS values and forms records. In each round of FL, local users update the user latent factors and service latent factors, and then they upload the gradients of service latent factors to the central server. The main role of the central server is to aggregate the gradients and then assign the new global gradients to all users. Since each user updates only the service latent factors it has invoked, the aggregated gradient is able to express more information.

Formally, for each round the user latent factors $U_i \in \mathbb{R}^{k\times 1}$ of the local participants are updated by the downloaded copy of the global model, where k denotes the dimension. The central server generally uses $Fedavg$ to aggregate the local gradient data uploaded by participants:

$$\Delta S = \frac{1}{m}\sum_{i} \Delta S_i, \tag{3}$$

where m is the number of user, ΔS is the aggregated gradients, ΔS_i is the local gradients submitted by user i.

Unlike the previous global model allocation for federated learning, DSGMF considers all users to be $selfish$ as they want to contribute more high-quality gradients to complete local training better and faster. We allocate the global gradient based on the user's contribution and call it a reward:

$$S_{ir}^{(t)} = sparsity(\Delta S_g^{(t)}, alloct_i) - r_i^{(t-1)}\Delta S_i^{(t)}, \tag{4}$$

where $S_{ir}^{(t)}$ is the reward, $\Delta S_g^{(t)}$ is the aggregated gradient, $\Delta S_i^{(t)}$ is the gradient uploaded by user i, $alloct_i = M \times r_i^{(t)}/r_{i\_max_i}^{(t)}$ represents the number of parameters that can be downloaded by user i, as determined by the relative reputation.

C. Model Security Guaranteed via Reputation

Our proposed reputation-based model security is shown in Fig. 2. There are four main steps involved:(I) Users upload the gradients; (II) Gradient aggregation & calculate the contribution and reputation; (III) Reputation comparison; (IV) Remove the users whose reputation is lower than β and sends contribution-based rewards to the users who are considered trustworthy.

1) Contribution-based Rewards:
Generally, the more services a user invokes the more QoS data is recorded, and the richer the gradient information uploaded by that user when participating in model training. If we provide the same global gradients that each participant can download during FL training, it is unfair to users who have rich QoS data. Because the rich gradient information they uploaded costs more time or computational resources. Thus, a fairer approach is to generate the global gradients assigned to each participant based on its contribution.

In our approach, the contribution-based allocation of global gradients is regarded as a reward. To quantitatively measure whether the rewards assigned by the central server correspond fairly to the contributions of the participants, we use the Pearson correlation coefficient for calculations. Consider a simple example when a group of users contribute $d = [1, 2, 13]$ and the candidate rewards are $p_1 = [2, 4, 6]$, $p_2 = [4, 16, 26]$. Both rewards obey the principle that the larger the contribution, the larger the reward, but p_2 is distributed in a $fair$ way, i.e., the reward is twice the contribution. By calculating the Pearson coefficient: $\rho_1(d, p_1) = 0.9011$, $\rho_2(d, p_2) = 1.0$, $\rho_2 > \rho_1$.

Thus, the actual service latent matrices trained locally by participant i should be:

$$S_i^{(t)} = S_i^{(t-1)} + S_{ir}^{(t)}, \tag{5}$$

where $S_i^{(t)}$ represents the service latent matrix used by participant i for training in round t, and $S_{ir}^{(t)}$ represents the model reward received by participant i before starting the current training round.

2) Calculation of Reputation:

In our proposed DSGMF method, free-riders as participants can obtain global gradients from the server to train the local model, but in the upload phase, they randomly generate spurious gradient data and send them to the central server.

Specifically, a user's reputation is related to the gradients submitted to the global. We use reputation to identify whether participant i is an adversarial user. Participants with high reputations should contribute high-quality gradient values to the global to make the model converge in a faster and better direction, while participants who submit low-quality gradient data have a correspondingly low reputation because it reduces the efficiency of the model. The gradients submitted by free-riders are often random and spurious, and these gradients are not beneficial to the training of the global model. If these free-riders keep submitting low-quality gradients to the central server, the global model fails to hit a satisfactory state or even moves in an unacceptable direction. These low-reputation participants are threats to the security of the overall QoS prediction system.

To measure the reputation of a participant during the submission process, DSGMF determines the reputation value for the current round by calculating the cosine similarity $\hat{r}_i^{(t)} = cos(\Delta S_g^{(t)}, \Delta S_i^{(t)})$ between its submitted gradients $\Delta S_i^{(t)}$ in the current round and the global gradients $\Delta S_g^{(t)}$ after server aggregation. Note that this round of reputation can only reflect the reputation of that user's round of submissions. To prevent the occasional drop-in reputation value, we combine the user's reputation of previous rounds and the user's calculated reputation of the current round to obtain its final reputation value for this round as follows:

$$r_i^{(t)} = \alpha r_i^{(t-1)} + (1-\alpha)\hat{r}_i^{(t)}, \tag{6}$$

where α is a parameter used to control the smooth update of r.

D. Algorithm Description

Algorithm 1 demonstrates our proposed DSGMF method in the form of pseudo-code. It can be seen that the algorithm is divided into two parts: participant i and the central server. The local matrix factorization algorithm is executed on the participant side. After receiving the rewards from the server, participant i performs training updates using the user latent vector U_i and the service latent vector S_i. Honest participants upload the true local training gradients while free-riders upload spurious gradients. The operations performed at the central server-side are as follows:

Algorithm 1 Double Security Guaranteed Matrix Factorization

Input: locally observed QoS records Q, learning rate η, moving average coefficient α, reputation threshold β, regularization parameter λ.

Notations: $r_i^{(t)}$ is i's reputation in round t; $R = \{i | r_i^{(t)} \geq \beta\}$ is the set of trusted user; U_i is the user latent vector of user i; S_i is the service latent matrix kept by user i; S_{ir} is the reward gradients to user i; ΔS_g is the aggregated gradients

Participant i

Download model rewards $S_{ir}^{(t)}$ and update local copy of latent service matrix: $S_i^{(t)} = S_i^{(t-1)} + S_{ir}^{(t)}$
Copy service latent matrix: $S_{i*}^{(t)} = S_i^{(t)}$
for $q_{ij} \in Q$ **do**
    $U_i^{(t)} \leftarrow U_i^{(t-1)} - \eta(q_{ij} - p_{ij})p'_{ij}S_{i*}^{(t)} + \lambda U_i^{(t-1)}$
    $S_{i*}^{(t)} \leftarrow S_{i*}^{(t-1)} - \eta(q_{ij} - p_{ij})p'_{ij}U_i^{(t)} + \lambda S_{i*}^{(t-1)}$
    if i is honest **then**
        Calculate gradients: $\Delta S_i^{(t)} = S_i^{(t)} - S_{i*}^{(t)}$
    else
        Generate spurious gradients.
Upload gradients.

Central Server

Aggregation:
$\Delta S_g^{(t)} = \sum_{i \in R} \Delta S_i^{(t)}$
for $i \in \mathcal{R}$ **do**
    $\hat{r}_i^{(t)} = cos(\Delta S_g^{(t)}, \Delta S_i^{(t)})$
    $r_i^{(t)} = \alpha r_i^{(t-1)} + (1-\alpha)\hat{r}_i^{(t)}$
    if $r_i^{(t)} < \beta$ **then**
        Remove i from $\mathcal{R}$.
Reward:
for $i \in \mathcal{R}$ **do**
    $alloct_i = M \times r_i^{(t)} / r_{i\_max_i}^{(t)}$
    $S_{ir}^{(t)} = sparsity(\Delta S_g^{(t)}, alloct_i) - r_i^{(t-1)}\Delta S_i^{(t)}$

Aggregation. In the aggregation phase, the central server aggregates the gradients of all users from the trusted set R in the form of a sum, generating a global gradient ΔS_g. Then calculate each user's reputation value for this round.

By comparing the magnitude of the reputation $r_i^{(t)}$ with the value of the reputation threshold β, the identified adversarial users are removed from the set of trusted users R after a specified number of rounds. In the setting, we do not perform the removal operation in the first round to prevent the impact of the error on the reputation due to the random initialization of the model. After a certain number of rounds (e.g. 5 rounds) of accumulation, the user reputation can reach a relatively stable state and the server side can perform the identification and removal operations for the adversarial users.

Reward. In the reward stage, we reflect fairness through the contribution-based reward distribution method.

First, the server computes $allocat_i$, then sparsifies the aggregated gradient $\Delta S_g^{(t)}$, and then subtracts the gradient uploaded by user i from the sparse version. The sparsification here is performed by retaining the maximum value to sparse

the gradient through relative reputation, which reduces the gradient quality. That is, users with high contributions are rewarded with less sparsified gradients, i.e., higher quality gradients, which also corresponds to high-quality gradients submitted by users.

IV. Experiment

A. Dataset Description

To verify the validity of the proposed model, we used a publicly and widely used real-world web service QoS dataset [19]. These data come from records of 339 users from more than 30 countries invoking 5825 services in over 80 countries. We use the response time (RT) in this dataset as QoS data, where the RT value measures the time interval between a user making a request and getting a response. Within the dataset, RT (sec) values range from 0 to 20, with a mean value of $0.91sec$.

B. Evaluation Metric

We use the mean absolute error (MAE) to evaluate the prediction accuracy of the proposed DSGMF and the training loss to show the variation of the model performance:

$$MAE = \frac{\sum_{i,j} |R_{i,j} - \hat{R}_{i,j}|}{N}, \tag{7}$$

$$loss = (R_{ij} - \hat{R}_{ij})^2, \tag{8}$$

where $R_{i,j}$ is the observed QoS value of service j invoked by user i, $\hat{R}_{i,j}$ is the predicted QoS value, and N is the number of all predicted values.

C. Result Analysis

1) Accuracy Comparison on Response Time: We compared the proposed DSGMF with FMF [13] (a privacy-preserving matrix factorization approach by applying federated learning), using MAE as a metric to assess the prediction accuracy on RT. We set the learning rate η = 0.001, the regularization parameter λ = 0.1, the reputation threshold β = 0.01, the moving average coefficient α = 0.8, the number of rounds to start removing free-rider = 5, and the QoS data density to 30%. The attack intensity was set to [0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3], where 0 means no attack, 0.05 means 5% of the users are set to free-rider. The experimental results are shown in Fig. 3. It can be seen that the MAE of both models becomes larger as the intensity of the free-rider attack increases. This proves that free-rider does adversely affect the QoS prediction performance of the matrix factorization method under the federated learning framework. When the attack intensity is 0, DSDMF is able to get a lower MAE, which indicates that the contribution-based reward is effective. The MAE of both grows as the attack intensity increases. However, the MAE of DSGMF is always lower than that of FMF under the same attack intensity. Since DSGMF provides reputation-based identification and resistance to free-riders, the change in MAE of its prediction results is smaller than that of FMF.

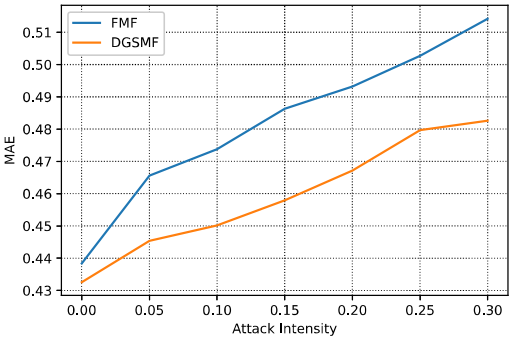

Fig. 3. Accuracy comparison with FMF under different attack intensity.

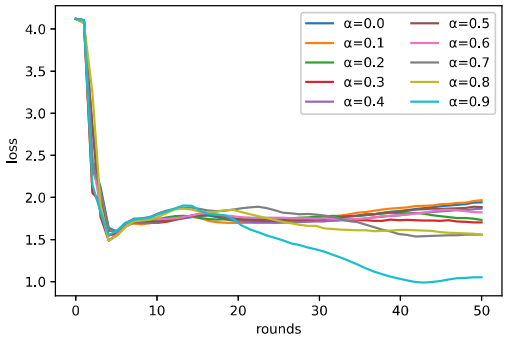

Fig. 4. Effect of the moving average coefficient α

2) Effect of the Moving Average Coefficient α: The moving average coefficient α is used to integrate the current round with the previous reputation. By adjusting the size of α, it is possible to determine the weight of the current round's reputation calculated by cosine similarity in the integrated reputation. Thus, the reputation in each round can be updated in a smooth form. We analyzed the effect of α on the training error (loss). Experiments were conducted at [0, 1] with a step size of 0.1 and an attacking intensity of 0.3, the settings of other parameters are consistent with the experiments of 1). The results of the variation of loss with training rounds are obtained as shown in Fig. 4. As we can see, there is a small upward movement in the loss when round=5. This is due to the removal of the identified free-rider at the beginning of round 5, which causes the loss to fluctuate. As the training rounds increase, the loss tends to flatten out. By comparing the different moving average coefficients α, it can be noticed that the larger the α is in the process of leveling off, the smaller the value of the stabilized loss, which indicates that the training error is also smaller. In particular, the loss is most obviously small at α = 0.9. α is the coefficient used to control the speed of reputation movement. According to Eq. 6, we know that the larger α means the greater the proportion of historical reputation in the current round of reputation, which can better

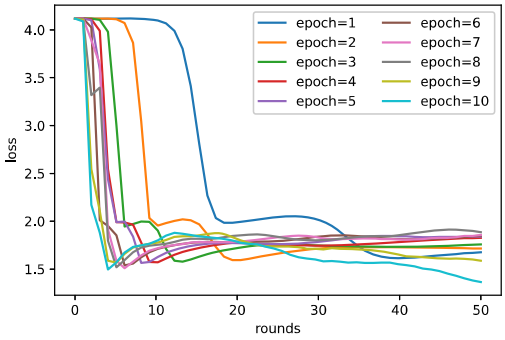

Fig. 5. Effect of Local Training Epoch

avoid the fluctuation of reputation caused by chance.

3) Effect of Local Training Epoch: The local epoch is the number of iterations for training the user's local matrix factorization. Users with different computational resources can choose different epochs. Thus, we evaluate the impact of epoch on model training. We set the local epoch to a total of ten groups from 1 to 10, the attack intensity is 0.3, and the settings of other parameters are consistent with the experiments in 1).

From Fig 5, we can see that the number of local training epochs affects the convergence speed of the model. Specifically, when only one round of local training is performed, the global training loss starts to decline only after greater than 10 rounds. As the epoch increases, the global training loss starts to decrease in smaller and smaller rounds, and when epoch=10, the training loss starts to decrease significantly in the 3rd round. Along with the growth of the local epoch, the training loss reaches a plateau more quickly. But the local training epoch can not infinitely increase either. From Fig. 5, we can see that as the epoch increases from 5 to 10, the state of decreasing loss is getting closer and closer, which indicates that although local provides more and more computational resources, the effect improvement is getting smaller and smaller.

V. Conclusion and Future Work

The work in this paper is dedicated to addressing the privacy security and model security issues in the QoS prediction process. The proposed DSGMF is a double security guaranteed model in the face of free-rider attacks, using a federated learning approach to address the user-side data privacy problem and a global model security using a reputation-based approach. The results show that free-rider attacks can negatively affect the matrix factorization prediction model under the federal learning framework, and our DSGMF can effectively identify free-rider users and reduce their impact on the model.

For future work, to create a more secure service recommendation system, we plan to investigate different QoS prediction models and explore the method to identify and defend against attacks (e.g. inference attacks) by using other advanced methods, such as deep learning, cloud-edge coordination.

Acknowledgement

This research was financially supported by Guangdong province special fund for science and technology ("major special projects + task list") project (No. STKJ2021201), 2020 Li Ka Shing Foundation Cross-Disciplinary Research Grant (No. 2020LKSFG08D) and in part by Guangdong Province Basic and Applied Basic Research Fund (2021A1515012527).

References

[1] S.-W. Chou and C.-H. Chiang, "Understanding the formation of software-as-a-service (saas) satisfaction from the perspective of service quality," *Decision Support Systems*, vol. 56, pp. 148–155, 2013.

[2] M. W. Aziz, N. Ullah, and M. Rashid, "A process model for service-oriented development of embedded software systems," *IT Professional*, vol. 23, no. 5, pp. 44–49, 2021.

[3] Z. Zheng, L. Xiaoli, M. Tang, F. Xie, and M. R. Lyu, "Web service qos prediction via collaborative filtering: A survey," *IEEE Transactions on Services Computing*, pp. 1–18, 2020.

[4] J. Xu, Z. Zheng, and M. R. Lyu, "Web service personalized quality of service prediction via reputation-based matrix factorization," *IEEE transactions on reliability*, vol. 65, no. 1, pp. 28–37, 2015.

[5] S. Bhagavan, M. Gharibi, and P. Rao, "Fedsmarteum: Secure federated matrix factorization using smart contracts for multi-cloud supply chain," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 4054–4063.

[6] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "A privacy-preserving qos prediction framework for web service recommendation," in *2015 IEEE International Conference on Web Services*. IEEE, 2015, pp. 241–248.

[7] F. Martelli, M. E. Renda, and J. Zhao, "The price of privacy control in mobility sharing," *Journal of Urban Technology*, vol. 28, no. 1-2, pp. 237–262, 2021.

[8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[9] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative web service qos prediction via neighborhood integrated matrix factorization," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 289–299, 2012.

[10] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

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

[12] L. Qi, H. Xiang, W. Dou, C. Yang, Y. Qin, and X. Zhang, "Privacy-preserving distributed service recommendation based on locality-sensitive hashing," in *2017 IEEE International conference on web services (ICWS)*. IEEE, 2017, pp. 49–56.

[13] P. Zhang, H. Jin, H. Dong, W. Song, and A. Bouguettaya, "Privacy-preserving qos forecasting in mobile edge environments," *IEEE Transactions on Services Computing*, pp. 1–1, 2020.

[14] S. Zhao, R. Bharati, C. Borcea, and Y. Chen, "Privacy-aware federated learning for page recommendation," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 1071–1080.

[15] Y. Zhang, P. Zhang, Y. Luo, and J. Luo, "Efficient and privacy-preserving federated qos prediction for cloud services," in *2020 IEEE International Conference on Web Services (ICWS)*. IEEE, 2020, pp. 549–553.

[16] F. Ye, Z. Lin, C. Chen, Z. Zheng, and H. Huang, "Outlier-resilient web service qos prediction," in *Proceedings of the Web Conference 2021*, 2021, pp. 3099–3110.

[17] X. Cao and N. Z. Gong, "Mpaf: Model poisoning attacks to federated learning based on fake clients," *arXiv preprint arXiv:2203.08669*, 2022.

[18] X. Xu and L. Lyu, "A reputation mechanism is all you need: Collaborative fairness and adversarial robustness in federated learning," *arXiv preprint arXiv:2011.10464*, 2020.

[19] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating qos of real-world web services," *IEEE transactions on services computing*, vol. 7, no. 1, pp. 32–39, 2012.