

Research Article

Time-Aware Cross-Platform IoT Service Recommendation with Privacy Preservation

Can Zhang , Junhua Wu , Chao Yan, and Guangshun Li

School of Computer Science, Qufu Normal University, Rizhao, Shandong, China

Correspondence should be addressed to Junhua Wu; shdwjh@163.com

Received 3 May 2021; Revised 26 June 2021; Accepted 8 November 2021; Published 6 December 2021

Academic Editor: Dou Wanchun

Copyright © 2021 Can Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IoT service recommendation techniques can help a user select appropriate IoT services efficiently. Aiming at improving the recommendation efficiency and preserving the data privacy, the locality-sensitive hashing (LSH) technique is adopted in service recommendation. However, existing LSH-based service recommendation methods ignore the intrinsic temporal feature of IoT services. In light of this challenge, we integrate the temporal feature into the conventional LSH-based method and present a time-aware approach with the capability of privacy preservation for IoT service recommendation across multiple platforms. Experiments on a real-world dataset are conducted to validate the advantage of our proposed approach in terms of accuracy and efficiency in recommendation.

1. Introduction

The rapidly increasing number of IoT devices and services are continuously producing a vast amount of data. Among those data, the quality of service (QoS) data is generated when a user invoked a service. Service recommendation techniques can employ the historical quality data to reduce the users' service selection burden and help them find out appropriate services efficiently. Typically, taking collaborative filtering recommendation algorithm for example, through predicting the quality value when a target user invokes a service according to its similar users' quality data, a recommendation system can suggest a list of optional services for a target user.

However, in the IoT environment, traditional recommendation algorithms are often nonavailable in practice [1]. One fundamental reason is that users may choose services from different platforms. For example, in the context of a smart home, users may use service provided by Hikvision to capture video data, service from Huawei to monitor the air quality, and service from Xiaomi to control household appliances. Therefore, the historical QoS data of users is not centralized, but stored across different platforms [2, 3]. Moreover, because of conflicts in economic interests and

data privacy concerns, service providers are unwilling to share their data with each other. Furthermore, the volume of the historical QoS data is often massive, thus it is impractical to share such large amount of data with other platforms.

Considering these challenges, the locality-sensitive hashing (LSH) algorithm [4] is adopted in service recommender system. Specifically, through hashing high dimensional historical quality data of users into scalar hash value, user privacy is preserved. Besides, as the hashing process is conducted individually in each platform, there is no need to transfer historical data.

However, most LSH-based service recommender systems view the historical QoS data as static and unique, rarely taking other context factors (e.g., time) into account. This may lead to less reasonable and accurate recommended results.

In light of the abovementioned challenges, we integrate temporal dimension into the conventional LSH-based recommendation method and present a time-aware recommendation approach for IoT services across multiple platforms. The proposed approach achieves higher accuracy than conventional recommendation approaches that have the capability of privacy preservation. The contributions of this paper are summarized as follows.

- (1) We improve the conventional LSH-based recommender system by incorporating it with a time factor, so as to adapt the intrinsic feature of IoT services, and achieve better performance in IoT service recommendation.
- (2) We conduct extensive experiments on a real-world dataset to validate the advantage of our method. Experimental results demonstrate that the proposed method outperforms the other state-of-the-art approaches in both recommendation accuracy and efficiency, while preserving data privacy among multiple platforms.

The rest of this article is structured as follows. Section 2 reviews recent work on recommender system for IoT service. Section 3 formulates the problem of IoT service recommendation and presents our motivation. Section 4 describes a time-aware cross-platform IoT service recommendation approach with privacy-preserving capability. Section 5 demonstrates the implementation and results of experiments. Section 6 summarizes the whole paper and addresses future work.

2. Related Work

In this section, we review the related research work on time-aware IoT services recommendation with privacy preservation from the following three aspects.

2.1. IoT Service Recommendation. Most existing researches on recommender system for IoT services can be classified into three groups: content-based filtering, collaborative filtering, and link-based methods [1]. In [5, 6], Mashal et al. formulated the IoT recommendation as a hypergraph model, which connects users, objects, and services with hyperedges and presents a graph-based recommender system considering the unique feature of IoT services. Yao et al. put forward a unified framework based on probabilistic factor; they calculated user similarity and device similarity and then fused them together to make more accurate recommendation [7, 8]. In [9], Mashal et al. proposed a multiagent approach to establish a distributed recommender system in IoT environment. However, in the above literature, the time factor, which is one of the most common features of IoT service, is not considered in IoT recommendation system. This may decrease the accuracy of recommendation result.

2.2. Time-Aware Service Recommendation. A number of research works have taken the time factor into account to obtain more accurate recommendation results. In [10, 11], Wang and Zhu present a spatial-temporal QoS value prediction approach. Temporal sequences of historical QoS data are employed to build feature models, while the spatial information of web services is exploited to reduce the searching space. In [12], Zhong and Fan built a time-aware recommender system for mashup creation. An extraction method for service pattern based on LDA and time series prediction is presented. In [13], Yu and Huang took both

time and location factors into account; they represent the temporal quality data as a three-dimensional matrix and use CF techniques to make prediction and recommendation.

All of the abovementioned methods employed the time factor to enhance the performance of recommendation system; however, none of those methods take privacy preservation into account, which is necessary when QoS data is collected from different platform to obtain more comprehensive user preference.

2.3. Privacy-Preserving Service Recommendation. As IoT service data from different platform contains sensitive user information, it is crucial to preserve user's privacy while sharing valuable data across platforms [14–19]. In [20], Ma et al. proposed K-anonymity method to protect user privacy through hiding sensitive user identification information. However, this may influence the data availability and decrease the performance of recommendation systems accordingly. In [21], Dou et al. suggest not publishing all the observed QoS data but the optimal data; however, users may still leak some sensitive information. In [4], Qi et al. first introduced LSH technique into service recommendation; by hashing high dimensionality QoS data into low dimensionality indices, data privacy can be protected in an efficient way. However, time factor is still not considered in the approach.

As a conclusion, existing IoT service recommendation methods fail in taking time factor and privacy preservation into account simultaneously [22–26]. In light of this challenge, we improve the conventional LSH method and present a time-aware cross-platform IoT service recommendation algorithm with privacy preservation.

3. Formulation and Motivation

3.1. Problem Formulation. Concretely, our time-aware cross-platform IoT service recommendation model can be formulated as a five-tuple $\text{IoTSerRec}_{t\text{-LSH}}(SP, U, IS, H, u_{\text{target}})$, where

- (1) $SP = \{sp_1, \dots, sp_c\}$: sp_k ($1 \leq k \leq c$) represents the k -th IoT service platform. Each platform provides a part of a user's QoS data.
- (2) $U = \{u_1, \dots, u_m\}$: u_k ($1 \leq k \leq m$) denotes the k -th IoT service user. As a user may invoke IoT service from different platforms, his/her historical QoS data is stored across multiple platforms.
- (3) $IS = \{is_{1,1}, \dots, is_{1,n_1}, \dots, is_{c,1}, \dots, is_{c,n_c}\}$: $is_{i,j}$ ($1 \leq i \leq c, 1 \leq j \leq n_i$) denotes the j -th IoT service on the i -th platform; n_i means the number of IoT services provided by the i -th platform.
- (4) $Q = \{Q_1, \dots, Q_t\}$: Q_k ($1 \leq k \leq t$) denotes the historical user-service quality data at the k -th time slot. $Q_k = \{q_{k,1,1}, \dots, q_{k,1,n_1}, \dots, q_{k,m,1}, \dots, q_{k,m,n_m}\}$: $q_{k,i,j}$ denotes the quality value of the j -th IoT service when the i -th user invokes at the k -th time slot.
- (5) $u_{\text{target}} \in U$: the user who needs recommendation service.

3.2. Motivation. As shown in Figure 1, there are three IoT service platforms, Alibaba (denoted as sp_1), Huawei (denoted as sp_2), and Google (denoted as sp_3). IoT services $is_{1,1}, \dots, is_{1,n_1}$ are deployed in Alibaba, $is_{2,1}, \dots, is_{2,n_2}$ are deployed in Huawei, and $is_{3,1}, \dots, is_{3,n_3}$ are deployed in Google. Users $\{u_1, u_2, u_3\}$ keep on invoking IoT services in the above platforms to finish certain tasks.

For conventional user-based CF recommendation approach, if we want to make recommendation for u_{target} , we need to find similar users of u_{target} according to the historical QoS data first [27–29]. However, there are three challenges in the collaboration process among Alibaba, Huawei, and Google. (1) Considering user privacy, each platform cannot share their own QoS data to each other [30–33]. (2) Since the user-service quality data keeps updating overtime, its volume becomes increasingly massive, which significantly reduces the collaboration efficiency and scalability [19, 34]. (3) As a user often invokes an IoT service constantly, a user-service pair is made up of a series of QoS values, which makes traditional user-based recommendation approach unsuitable for this situation [35, 36].

Considering the abovementioned challenges, we present a novel LSH-based IoT service recommendation method. The method will be elaborated in Section 4.

4. Privacy-Preserving and Time-Aware LSH-Based IoT Service Recommendation

Algorithm: IoTSerRec_t-LSH

In this section, we present a time-aware LSH-based recommendation algorithm for IoT services across multiple platforms, which is denoted as IoTSerRec_t-LSH. In summary, our proposed algorithm is divided into three steps:

Step 1: calculating user indices at each time slots based on LSH. For an IoT service platform, $sp_k (1 \leq k \leq z)$, each user u 's QoS data in sp_k at the t -th time slot is mapped into $H_{t,k}(u)$, which denotes the k -th subindex of user u at the t -th time slot. User u 's complete index at the t -th time slot $H_t(u)$ is merged as $(H_{t,1}(u), \dots, H_{t,c}(u))$ offline. The set of indices of all users at each time slot is regarded as a hash table.

Step 2: finding out top-K users most similar to u_{target} . Compute u_{target} 's complete index at each time slot according to step 1. Then, calculate the similarity between u_{target} and other users according to the user indices at each time slot, and return the top-K users with the highest similarity score with u_{target} .

Step 3: recommending IoT service for u_{target} . Predict the quality value of services never invoked by u_{target} based on u_{target} 's top-K neighbors' quality data. Then, retrieve the quality-optimal one to u_{target} .

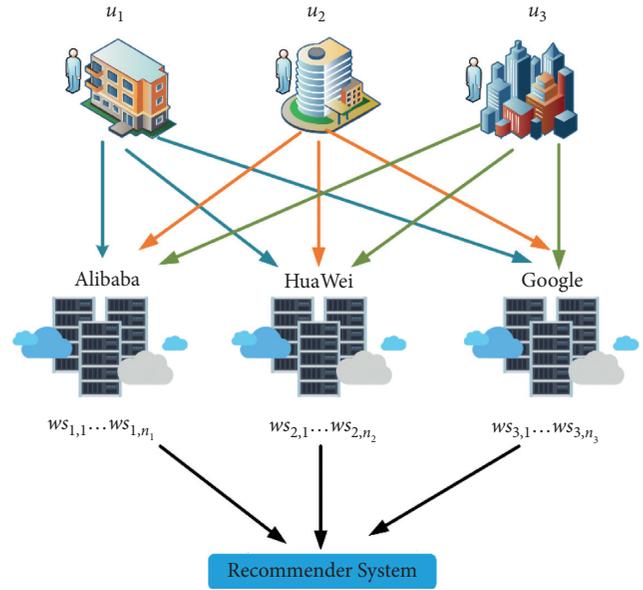


FIGURE 1: Cross-platform IoT service recommendation: an intuitive example.

Next, we demonstrate the implementation of each step in details.

4.1. Step 1: Calculating User Indices at Each Time Slots Based on LSH. In the IoT environment, a user (i.e., u_i) usually invokes a IoT service (i.e., is_j) at intervals and continuously generates a sequence of QoS data (i.e., $(\dots, q_{t-1,i,j}, q_{t,i,j})$) [37]. To reduce the scale of historical QoS data, we only employ the historical QoS data of the latest t time slots (i.e., $[q_{1,i,j}, \dots, q_{t-1,i,j}, q_{t,i,j}]$). Therefore, we denote the QoS data in the p -th platform at the t_k -th time slot as a matrix shown in (1), where the i -th row describes the quality value of all n_p services provided in the p -th platform invoked by the i -th user at the t_k -th time slot. Note that if the i -th user does not employ the j -th IoT service at the t_k -th time slot, $q_{t_k,i,j} = 0$.

$$Q_{t_k,p} = \begin{bmatrix} q_{t_k,1,1} & \cdots & q_{t_k,1,n_p} \\ \vdots & \ddots & \vdots \\ q_{t_k,m,1} & \cdots & q_{t_k,m,n_p} \end{bmatrix}. \quad (1)$$

Next, to preserve the user's privacy, we utilize LSH technique to map a user's QoS data to less-sensitive user index. Since classical CF-based recommendation algorithms often employ Pearson correlation coefficient (PCC) as a measure of user similarity, we adopt the LSH function for the PCC distance to realize the transformation.

For each platform sp_p and each time slot t_k , we randomly generate a n_p -dimensional vector $\vec{v}_{t_k,p}$ from the range $[-1, 1]$. For user u_i , the LSH function for time slot t_k is

defined in (2). Here, $Q_{t_k}(u_i) = [Q_{t_k,1}(u_i), \dots, Q_{t_k,c}(u_i)]$, and $Q_{t_k,p}(u_i) = [q_{t_k,i,1}, \dots, q_{t_k,i,n_p}]$.

$$h(Q_{t_k}(u_i)) = \begin{cases} 1, & \text{if } \sum_{p=1}^c (Q_{t_k,p}(u_i) \circ \overrightarrow{v_{t_k,p}}) > 0, \\ 0, & \text{if } \sum_{p=1}^c (Q_{t_k,p}(u_i) \circ \overrightarrow{v_{t_k,p}}) > 0. \end{cases} \quad (2)$$

Thus, according to (2), a user's observed IoT service data at the p -th platform is firstly mapped into a less-sensitive one-dimensional float value; then, all the float values at each platform are accumulated and mapped into a binary value.

Because LSH is a neighbor search method based on probability, one hash function usually leads to less accurate search results. Thus, we adopt amplified LSH through employing multiple hash functions and hash tables. Concretely, for each time slot t_k , we define r hash functions based on r vectors randomly generated from the range $[-1, 1]$. Afterwards, we obtain a $t * r$ 0-1 matrix, represented as $H(Q(u_i))$ in (3). Indices of all users in U are stored in a hash table.

$$H(Q(u_i)) = \begin{bmatrix} h_{1,1}(Q_1(u_i)) & \cdots & h_{1,r}(Q_1(u_i)) \\ \vdots & \ddots & \vdots \\ h_{t,1}(Q_t(u_i)) & \cdots & h_{t,r}(Q_t(u_i)) \end{bmatrix}. \quad (3)$$

4.2. Step 2: Finding Out Top-K Users Most Similar to u_{target} . In step 1, we have generated a hash table; a user's quality data in recent t time slots is mapped into a $t * r$ 0-1 matrix. For convenience, we denote the matrix (in (3)), as shown in (4), where $\overrightarrow{h_{t_k}(u_i)}$ is a vector referring to the t_k -th row in the matrix. Thus, the hash values of all users' quality data in recent t time slots can be represented by $t * m$ matrix (shown in (5)).

$$HQ(u_i) = \begin{bmatrix} \overrightarrow{h_1(u_i)} \\ \dots \\ \overrightarrow{h_t(u_i)} \end{bmatrix}, \quad (4)$$

$$H(Q) = \begin{bmatrix} \overrightarrow{h_1(u_1)} & \cdots & \overrightarrow{h_1(u_m)} \\ \vdots & \ddots & \vdots \\ \overrightarrow{h_t(u_1)} & \cdots & \overrightarrow{h_t(u_m)} \end{bmatrix}. \quad (5)$$

Because of the probability-based feature of LSH, it is too strict to find neighbors according to single hash table. Thus, we build $LL > 1$ hash tables through repeating step 1 L times, which can reduce the number of possible "false-negative" neighbors.

Next, we compute the similarities between u_i and the other users at the t_k -th time slot by comparing $\overrightarrow{h_{t_k}(u_i)}$ with the other elements in the t_k -th row in (5). The similarity between u_i and u_j at time slot t_k is denoted as $\text{sim}_{t_k}(u_i, u_j)$. We first initialize $\text{sim}_{t_k}(u_i, u_j)$ to 0. If $\overrightarrow{h_{t_k}(u_i)} = \overrightarrow{h_{t_k}(u_j)}$ holds in any hash table T_p ($1 \leq p \leq L$), we increment $\text{sim}_{t_k}(u_i, u_j)$

by one. The similarity between u_i and u_j is calculated with the formula in (6), where ω_{t_k} is a tunable parameter to control the influence of different time slot. The tunable parameters meet the constraint defined in (7).

$$\text{sim}(u_i, u_j) = \sum_{t_k=1}^t \text{sim}_{t_k}(u_i, u_j), \quad (6)$$

$$\sum_{t_k=1}^t \omega_{t_k} = 1. \quad (7)$$

Finally, the K users with the highest $\text{sim}(u_i, u_j)$ values are returned as the similar user set of u_i , denoted as SIM_U_Set .

4.3. Step 3: Recommending IoT Service for u_{target} at Time Slot t . In step 3, we have obtained the top-k most similar users of u_{target} , i.e., $\text{SIM_U_Set}(u_{target})$. Next, we recommend the optimal IoT service for u_{target} at time slot t based on $\text{SIM_U_Set}(u_{target})$. In detail, we first predict the quality value of IoT services which have not been employed by u_{target} at time slot t according to formula (8); then, we make recommendation for u_{target} according to the predicted quality value.

$$q_{t,target,j} = \frac{\sum_{u_i \in \text{SIM_U_Set}(u_{target})} \text{sim}(u_{target}, u_i) * q_{t,i,j}}{\sum_{u_i \in \text{SIM_U_Set}(u_{target})} \text{sim}(u_{target}, u_i)}. \quad (8)$$

5. Experiments

To testify the feasibility of our proposed method, extensive experiments are conducted on a real-world QoS dataset, i.e., WS-DREAM [17]. The dataset consists of quality values (i.e., response time and throughput) of 4532 web services invoked by 142 users at 64 different time slots. In our experiments, we consider only one quality dimension, i.e., response time. Moreover, each country that owns services is used to simulate a geographically distributed IoT service platform.

Next, we generate test dataset by randomly removing a proportion of data in the dataset. If $P\%$ QoS values have been erased, the sparsity of the test dataset is defined as $P\%$.

Because of the intrinsic nature of LSH, the privacy-preservation capability of our approach is not testified here. Specifically, we testify and compare three evaluation measures.

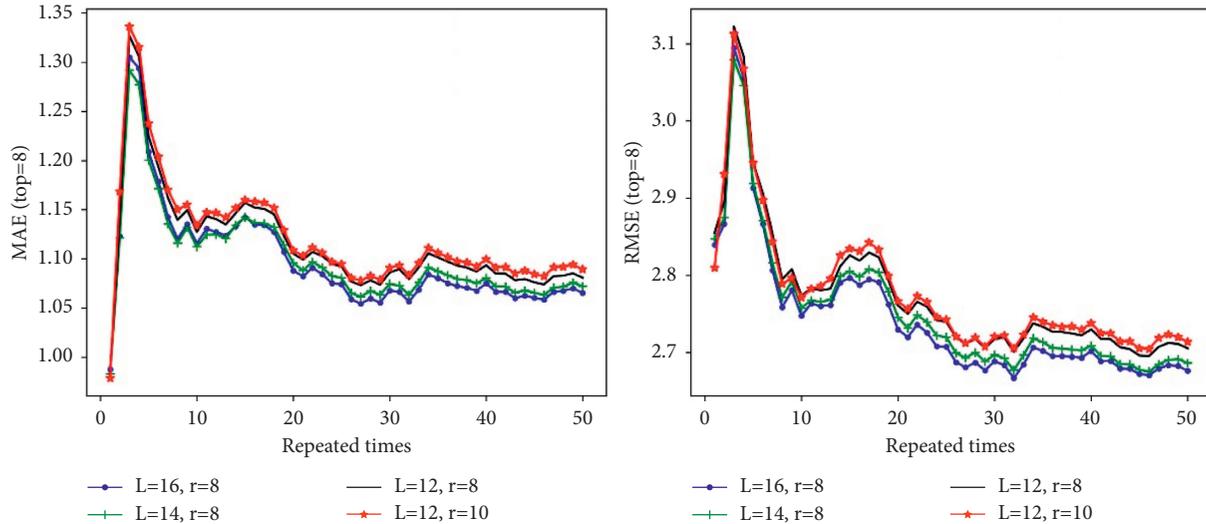


FIGURE 2: MAE and RMSE w.r.t experiment times.

- (1) Time cost: time consumption of our proposed approach, which is employed to measure the efficiency and performance of a recommendation system.
- (2) MAE (mean absolute error): average difference between predicted value and real value, which is employed to measure the accuracy of a recommender system.
- (3) RMSE (root mean square error): square root of the average of squared error between predicted value and real value, which is also employed to measure the accuracy of a recommendation system.

Concretely, in our experiments, we testify and compare five profiles in our experiments.

5.1. Profile 1: Determining the Best Value of Parameters L and r .

In our approach, the number of hash tables (i.e., L) and the number of hash functions in a hash table (i.e., r) can significantly influence the accuracy of a recommender system. Thus, we compare the prediction accuracy (i.e., RMSE and MAE) under different L - r settings and find out the best value of L and r in this profile. For each L - r setting, we repeat our approach on 50 different test datasets and all of the test datasets' sparsity is fixed to 10%. The reason why we choose 50 different test datasets is that we find the average accuracy tends to be convergent after repeated 50 times (as shown in Figure 2).

Table 1 demonstrates the MAE and RMSE values of our algorithm under different combination of L , r , and K when the sparsity of test dataset is set to 10%. Results reveal that our algorithm has the best accuracy when parameters are set as follows: $L = 16$ and $r = 8$. This holds when top varies from 5 to 10.

5.2. Profile 2: Finding the Proper Value for Parameter K .

In our proposed method, we need to find out a target user's K most similar users. To this end, we conduct experiment

with various values of K , where parameters L and r are fixed to 16 and 8, respectively. Therefore, we can find out the proper value of K in our method. Figure 3 shows that, after repeating the experiment on different test datasets 50 times, our method achieves the highest average accuracy (both MAE and RMSE) when $top = 8$.

5.3. Profile 3: Accuracy Comparison of Three Recommendation Algorithms.

To demonstrate the feasibility of our method, we compare $IoTserRec_{t-LSH}$ with two state-of-the-art algorithms UPCC [18] and $SerRec_{distr-LSH}$ [4] within different dataset sparsity. Based on the findings in Profile 1 and Profile 2, parameters in $IoTserRec_{t-LSH}$ are defined as follows: $L = 16$, $r = 8$, and $K = 8$. The following observations are made based on the results presented in Figure 4:

- (1) The MAE and RMSE values of all three approaches decrease as the dataset sparsity increases. When the sparsity is high, less useful QoS values are fetched, which leads to a less precise prediction.
- (2) $SerRec_{distr-LSH}$ performs much worse than our approach since it does not consider the historical temporal information.
- (3) When the dataset sparsity is lower than 50%, the MAE value of our approach is a little lower than UPCC. In spite of this, our algorithm still outperforms other algorithms in most cases. The reason is that the quality values at current time slot are sufficient to find the most similar users for UPCC when the dataset sparsity is low. However, temporal information is necessary within sparser dataset.

5.4. Profile 4: Efficiency Comparison of Three Recommendation Algorithms.

In this profile, we make a comparison among the three recommendation algorithms in terms of the efficiency. All of the experiments are performed on a computer with Intel i5 processor and 16.0GB RAM, which runs

TABLE 1: Accuracy comparisons of $\text{IoTSerRec}_{t\text{-LSH}}$ under different combination of parameters.

Parameter	$K=5$		$K=8$		$K=10$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
$L=16, r=6$	1.0782	2.7286	1.0771	2.6885	1.0787	2.6828
$L=16, r=8$	1.0647	2.717	1.0651	2.6762	1.0702	2.6808
$L=16, r=10$	1.0835	2.7429	1.0726	2.6912	1.0723	2.682
$L=14, r=6$	1.0804	2.7323	1.0727	2.684	1.0783	2.6836
$L=14, r=8$	1.0733	2.7297	1.0719	2.6864	1.0771	2.6906
$L=14, r=10$	1.0892	2.7558	1.082	2.7033	1.0788	2.6943
$L=12, r=6$	1.087	2.7451	1.0816	2.6939	1.0842	2.691
$L=12, r=8$	1.0808	2.7411	1.0807	2.7053	1.0824	2.6979
$L=12, r=10$	1.0908	2.7573	1.0893	2.7138	1.0934	2.709

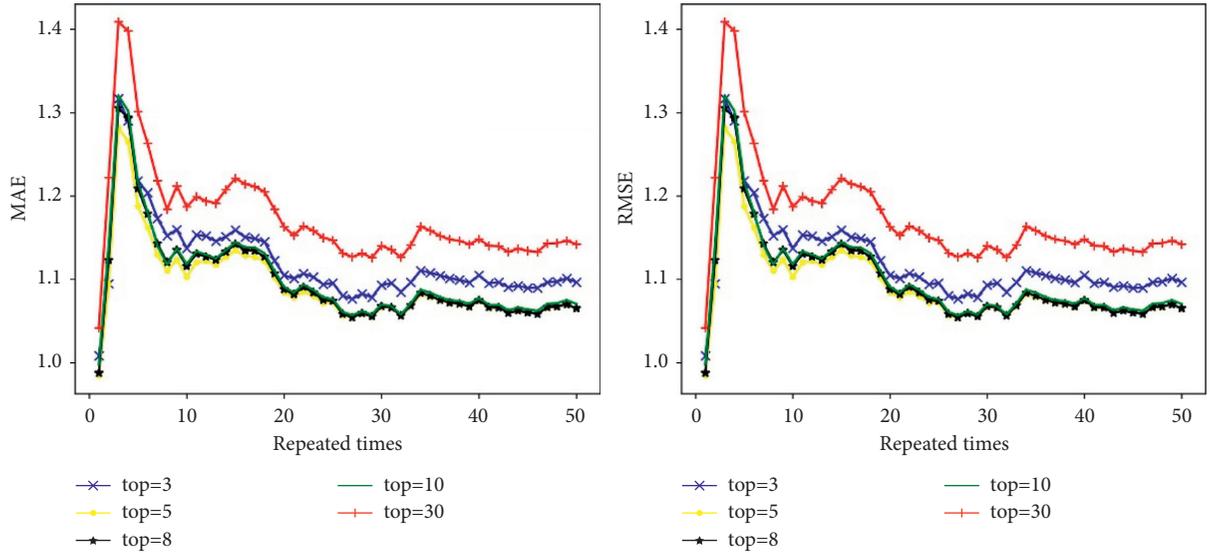
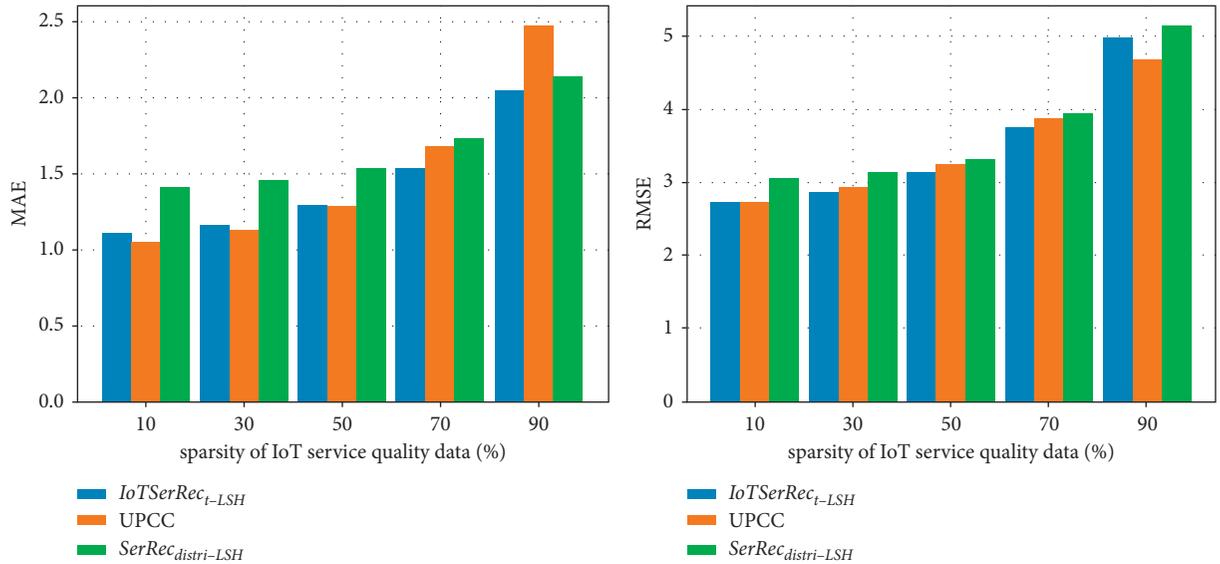
FIGURE 3: MAE and RMSE of $\text{IoTSerRec}_{t\text{-LSH}}$ w.r.t. top.

FIGURE 4: Recommendation accuracy comparison.

Windows 10 and Python 3.6. The experimental results (in Figure 5) show that the time consumption of UPCC is much higher than the other approaches since the UPCC method

calculates correlation coefficients online. Furthermore, the time cost of our method is close to that of $\text{SerRec}_{\text{distr-LSH}}$ since they both adopt the offline recommendation strategy.

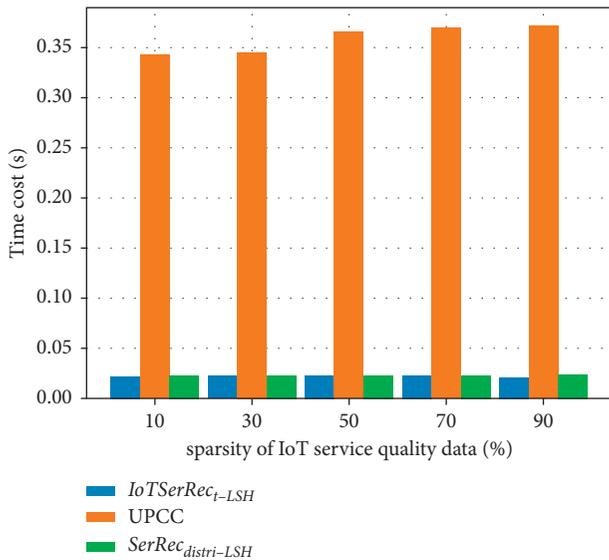


FIGURE 5: Recommendation efficiency comparison.

6. Conclusion

The number of IoT services is rapidly growing. Recommendation technology can significantly relieve the target users' selection burden. However, because of user privacy concerns, it is impractical for different platforms to directly share their data with each other. In this paper, we proposed a novel cross-platform LSH-based IoT service recommendation method with privacy preservation. Through LSH technique, high dimensionality historical quality data is hashed into less-sensitive indices. Moreover, historical data at different time slots are employed to make more accurate recommendation. Finally, a number of experiments are conducted on real-world dataset WS-DREAM. The experimental results demonstrated the advantage of our approach in terms of accuracy, efficiency, and the capability of privacy preservation.

However, some limitations should be addressed. As we did not consider the periodicity of the quality value of IoT service, the number of time slots is fixed in the current method. Furthermore, spatial information of both users and service providers is not employed in our approach. Therefore, we will further enhance our algorithm by taking more context factors such as periodicity [38, 39] and location [40, 41] into account.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] L. Yao, X. Wang, Q. Z. Sheng, S. Dustdar, S. Zhang, and S. Dustdar, "Recommendations on the internet of things: requirements, challenges, and directions," *IEEE Internet Computing*, vol. 23, no. 3, pp. 46–54, 2019.

- [2] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multi-dimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3075849, 8 pages, 2018.
- [3] Q. Yu, Z. Zheng, and H. Wang, "Trace norm regularized matrix factorization for service recommendation," in *Proceedings of the IEEE International Conference on Web Services*, pp. 34–41, Santa Clara, CA, USA, October 2013.
- [4] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2616–2624, 2017.
- [5] I. Mashal, T. Y. Chung, and O. Alsaryrah, "Toward service recommendation in Internet of Things," in *Proceedings of the Seventh International Conference on Ubiquitous and Future Networks*, pp. 328–331, Sapporo, Japan, July 2015.
- [6] L. Yao, Q. Z. Sheng, A. H. H. Ngu, and X. Li, "Things of interest recommendation by leveraging heterogeneous relations in the Internet of Things," *ACM Transactions on Internet Technology*, vol. 16, no. 2, pp. 1–25, 2016.
- [7] L. Yao, Q. Z. Sheng, A. H. H. Ngu, H. Ashman, and X. Li, "Exploring recommendations in internet of things," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 855–858, Gold Coast Queensland, Australia, July 2014.
- [8] A. Forestiero, "Multi-agent recommendation system in internet of things," in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 772–775, Madrid, Spain, May 2017.
- [9] I. Mashal, O. Alsaryrah, and T.-Y. Chung, "Testing and evaluating recommendation algorithms in Internet of Things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 6, pp. 889–900, 2016.
- [10] X. Wang, J. Zhu, Z. Zheng, W. Song, Y. Shen, and M. R. Lyu, "A spatial-temporal qos prediction approach for time-aware web service recommendation," *ACM Transactions on the Web*, vol. 10, no. 1, pp. 1–25, 2016.
- [11] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: a time-aware personalized QoS prediction framework for web services," in *Proceedings of the IEEE 22nd International Symposium on Software Reliability Engineering*, pp. 210–219, Hiroshima, Japan, December 2011.
- [12] Y. Zhong, Y. Fan, K. Huang, W. Tan, and J. Zhang, "Time-aware service recommendation for mashup creation," *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 356–368, 2015.
- [13] C. Yu and L. Huang, "A web service QoS prediction approach based on time- and location-aware collaborative filtering," *Service Oriented Computing and Applications*, vol. 10, no. 2, pp. 135–149, 2016.
- [14] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.
- [15] D. Li, C. Chen, Q. Lv et al., "An algorithm for efficient privacy-preserving item-based collaborative filtering," *Future Generation Computer Systems*, vol. 55, pp. 311–320, 2016.
- [16] B. S. Jena, C. Khan, and R. Sunderraman, "High performance frequent subgraph mining on transaction datasets: a survey

- and performance comparison,” *Big Data Mining and Analytics*, vol. 2, no. 3, pp. 159–180, 2019.
- [17] 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, 2014.
- [18] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” in *Proceedings of the 4th Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, Madison, Wisconsin, July 1998.
- [19] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, “Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1145–1153, Article ID 2969489, 2021.
- [20] T. Ma, Y. Zhang, J. Cao et al., “KDVEM: a k-degree anonymity with vertex and edge modification algorithm,” *Computing*, vol. 70, no. 6, pp. 1336–1344, 2015.
- [21] W. Dou, X. Zhang, J. Liu, and J. Chen, “HireSome-II: towards privacy-aware cross-cloud service composition for big data applications,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 455–466, 2015.
- [22] Y. Yu, M. Li, L. Liu, Y. Li, and J. Wang, “Clinical big data and deep learning: applications, challenges, and future outlooks,” *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 288–305, 2019.
- [23] Y. Hu, Q. Peng, X. Hu, and R. Yang, “Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering,” *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 782–794, 2015.
- [24] G. Li, S. Peng, C. Wang, J. Niu, and Y. Yuan, “An energy-efficient data collection scheme using denoising autoencoder in wireless sensor networks,” *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 86–96, 2019.
- [25] L. Liu, X. Chen, Z. Lu, L. Wang, and X. Wen, “Mobile-Edge computing framework with data compression for wireless network in energy internet,” *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 271–280, 2019.
- [26] X. Zhou, X. Xu, W. Liang et al., “Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1377–1386, 2022.
- [27] Y. Zhang, J. Pan, L. Qi, and Q. He, “Privacy-preserving quality prediction for edge-based IoT services,” *Future Generation Computer Systems*, vol. 114, pp. 336–348, 2020.
- [28] Y. Liu, A. Pei, F. Wang et al., “An attention-based category-aware GRU model for the next POI recommendation,” *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3174–3189, 2021.
- [29] Q. Zhu, X. Ma, and X. Li, “Statistical learning for semantic parsing: a survey,” *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 217–239, 2019.
- [30] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, “Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5790–5798, 2021.
- [31] D. Kim, J. Son, D. Seo, Y. Kim, H. Kim, and J. T. Seo, “A novel transparent and auditable fog-assisted cloud storage with compensation mechanism,” *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 28–43, 2020.
- [32] L. Qi, C. Hu, X. Zhang et al., “Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, p. 1, 2020.
- [33] X. Zhou, Y. Li, and W. Liang, “CNN-RNN based intelligent recommendation for online medical pre-diagnosis support,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2021.
- [34] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, “Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5213–5222, 2021.
- [35] S. Kumar and M. Singh, “A novel clustering technique for efficient clustering of big data in hadoop ecosystem,” *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 240–247, 2019.
- [36] Y. Huang, Y. Chai, Y. Liu, and J. Shen, “Architecture of next-generation E-commerce platform,” *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 18–29, 2019.
- [37] L. Qi, H. Song, X. Zhang, G. Srivastava, X. Xu, and S. Yu, “Compatibility-aware web API recommendation for mashup creation via textual description mining,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 1s, pp. 1–19, 2021.
- [38] X. Xu, B. Shen, S. Ding et al., “Service offloading with deep Q-network for digital twinning-empowered internet of vehicles in edge computing,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1414–1423, 2022.
- [39] X. Xu, R. Mo, X. Yin et al., “PDM: privacy-aware deployment of machine-learning applications for industrial cyber-physical cloud systems,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5819–5828, 2021.
- [40] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, “Variational LSTM enhanced anomaly detection for industrial big data,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3469–3477, 2021.
- [41] X. Xu, Q. Huang, H. Zhu et al., “Secure service offloading for internet of vehicles in SDN-enabled mobile edge computing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.