Security-Driven Hybrid Collaborative Recommendation Method for Cloud-based IoT Services

Shunmei Meng[1], Zijian Gao[1], Qianmu Li[1], Hao Wang[2], Hong-Ning Dai[3], Lianyong Qi*[4,5]
[1]Department of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2]Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjvik, Norway
[3]Faculty of Information Technology, Macau University of Science and Technology, Macau
[4]School of Information Science and Engineering, Qufu Normal University, China
[5]State Key Laboratory for Novel Software Technology, Nanjing University, China.

{mengshunmei@njust.edu.cn, zijiangao_njust@163.com, qianmu@njust.edu.cn, hawa@ntnu.no, hndai@ieee.org, lianyongqi@qfnu.edu.cn*}

Abstract. The rapid development of IoT (Internet of Things) systems and cloud techniques has paved the way for recommender systems to facilitate the daily life of users. However, the accompanying cybersecurity risks, such as environmental attacks and software attacks, must not be ignored. Thus, the security problem in recommender systems becomes a serious challenge for cloud-based IoT services. Moreover, most of existing collaborative recommendation algorithms mainly focus on user-item interaction relationships but seldom consider user-user or item-item co-occurrence relationships, which may affect prediction accuracy. To overcome the above shortcomings, this paper proposes a security-driven hybrid collaborative recommendation method to deal with the large-scale IoT services accessible by clouds in a more scalable and secure manner. Our proposal integrates the factorization-based latent factor model with the neighbor-based collaborative model to mine not only user-service interaction relationships but also user-user and service-service co-occurrence relationships. Moreover, the local sensitive hash (LSH) technique is adopted to speed up the neighbor searching and preserve users' sensitive information for security concerns based on hash mapping. Finally, experiment results demonstrate that the proposed method can improve prediction accuracy while guaranteeing information security.

Keywords: Security, Collaborative recommendation, IoT services, MF, LSH

1. Introduction

Recently, the ubiquity and density of IoT (Internet of Things) systems are increasing rapidly, which can be found in broad range of commercial and industrial applications, such as smart cities and healthcare applications. With the advancement of IoT systems and cloud techniques, large-scale candidate cloud-based IoT services are

provided by increasing IoT nodes [1-2]. Besides, rapid development in big data processing techniques also allows diverse user-related information to be collected form the IoT sensors. How to find appropriate IoT services from massive candidates and handle the humongous sensor data in an efficient, economical, smart and secure manner becomes a critical task [3-4]. Recommender systems have been proved effective in dealing with information overload. More and more users, research institutions, businesses, hospitals and industry companies wish to benefit from recommender systems, since it can be used to improve user experience, engagement, and revenue [5-7].

While IoT systems and cloud-based techniques has powered up the implementation of recommender systems and help users in their daily life, they also may bring cybersecurity risks and threats to users and put users in a complex and insecure network environment [8-9]. The diverse user-related information (such as presence, observed rating values, location information) collected form the IoT sensors may involve users' activity patterns, emotions, behaviors or health condition [10]. It may be stolen for illegal use or even resold to unauthorized parties for profits, putting users at risk. Thus, developing effective recommendation algorithms with security concerns becomes necessary for cloud-based IoT services to provide users with intelligent recommendations tailored to their needs.

A number of recommendation algorithms have been put forward in the literature, among which the collaborative filtering (CF) recommendation technique is one of the most effective ones. There are two effective CF recommendation models: neighbor-based (including user-based CF and item-based CF) CF model and factorization-based CF model [5]. Neighbor-based CF aims to capture the similarity relationships and make predictions according to the previous behaviors of similar neighbors. However, to find the nearest neighbors of the target user and IoT services, the similarity between all pairs should be calculated, which is time-consuming. In addition, neighbor-based CF algorithms are not applicable to dealing with spare rating data. Data sparsity problem is more likely to impact the prediction performance, thus becoming a hot topic in recommendation studies [11-12]. Factorization-based CF algorithms, including matrix factorization (MF) and high-order factorization, are effective in solving the data sparsity problem [13-15]. However, the traditional factorization-based collaborative model usually mines user-item relationships by simple inner product and makes predictions directly from matrix rating patterns. It only captures user-item interaction relationships but ignores user-user and item-item co-occurrence relationships, which may miss some important information. Inspired by the Pennington's algorithm in references [16-17] where word embedding vectors are learned based on the word-word co-occurrence in documents, we use the MF technique to mine user and service embedding vectors in this work. Then similar neighbors of the target users (or the target IoT service) can be obtained based on the extracted embedding vectors.

In addition, the security problem is a serious challenge to be addressed in complex cyber networks including IoT-based recommender systems [18]. Most existing recommendation models seldom consider the security risks related to collected data and the possible inferences from the collected data, which lack secure ways to manage and analyze the sensitive information of IoT users [19]. Hash mapping techniques have been proved to be effective tools to protect sensitive information by mapping the

original real data into fuzzy hash values [20]. Therefore, in this paper, a hash technique named Local sensitive hash (LSH) is adopted to process privacy-related information of IoT users, where user-related information and inference data will be mapped into low-dimensional hash values. Then the hash values, rather than the original user-related information, are input for neighbor searching. In this way, the sensitive information of IoT users is blurred and protected. Besides, the LSH mechanism is effective in similarity searching, which can be employed to accelerate the neighbor searching in traditional collaborative recommendation approach.

Based on the above observations, in this paper, we propose a security-driven hybrid collaborative recommendation method (abbreviated as SHCR) for cloud-based IoT services. It integrates the factorization-based latent factor model with the improved neighbor-based CF model for hybrid collaborative recommendation. Moreover, to improve recommendation efficiency and protect users' privacy information, the LSH mechanism is employed to speed up neighbor searching with security concerns. The main contributions of this paper are presented as follows:

- Firstly, the factorization-based latent factor model is adopted to extract the feature representations of users and IoT services, where user-user and service-service co-occurrence embedding vectors are learned with the factorization technique.
- To reduce security risk and handle massive data, the LSH mechanism is used to realize fast neighbor searching under security concerns. Then the final prediction is performed by the improved neighbor-based CF recommendation algorithm.
- Finally, external experiments are designed and conducted to validate the effectiveness of our method. Experimental results show that our method could improve the recommendation efficiency while protecting users' privacy information.

The remainder of this paper is organized as follows: Related studies are reviewed in Section 2. Section 3 describes the system model of our proposal, as well as the key techniques used. Section 4 presents the security-driven hybrid collaborative recommendation method. Experiments are designed and analyzed in Section 5, followed by the conclusion and future work in Section 6.

2. Related Work

Recently, with the advancement of cloud techniques and IoT technology, the recommendation models for cloud-based IoT services have been researched in many literatures [21-24]. Saleem et al. [22] introduce a scheme for the exploitation of the social IoT for recommendations among large-scale IoT applications. The authors in [23] also propose a hyper-graph-based service recommendation model and study the performance of typical recommendation algorithms on IoT service recommendation. The research [24] provides a health-centric recommendation model, which supports travelers with long-term diseases and followers of strict diet. In addition, Artificial Intelligence (AI) technology has been emerged as a dynamic and fast-growing research

area in recent years, which is also applied in recommender systems to improve prediction accuracy and provide intelligent services [17, 25-29]. The authors in [25] propose a neural network-based CF recommendation model where a multi-layer perceptron is adopted to learn the user-item interaction relations. In reference [26], the deep learning technique and the time-aware CF model are tightly coupled with the consideration of cold start, where the deep neural network is applied to learn the features of items. Fu et al. [17] also propose a deep-learning-based CF recommendation approach where prediction is made based on the factorization-based CF model and a multi-view feedforward neural network. Ebesu et al. [28] present a deep collaborative memory network model to integrate neighbor-based CF model and factorization-based CF model to find user-item specific neighbors so as to improve recommendation performance. However, most of existing deep-learning-based recommendation algorithms and recommendation algorithms for IoT services mainly focus on improving the recommendation accuracy but pay little attention to the security problem.

CF recommendation model is the most widely used recommendation technique, which mainly focuses on mining user-item interaction relationships and making predictions based on the behaviors (such as ratings, tags, comments) of previous users [30]. There are mainly two well-known CF recommendation models, i.e., neighbor-based CF model and factorization-based CF model. Bu et al. [31] propose a Multiclass Co-Clustering model to mine user-item, user-user, item-item relations, and then integrate typical CF recommendation approach with subclusters to improve recommendation accuracy. The authors in [32-34] focus on integrating big data analysis technique into collaborative recommendation algorithm to improve the effectiveness of recommendation performance. Lian et al. [35] provide a scalable implicit-feedback-based CF recommendation model on semantic content, which establishes the relationships of users and items with graph Laplacian regularized matrix factorization for location recommendation.

Factorization-based CF recommendation techniques attract more attentions since the Netflix Prize in 2006. Pan et al. [36] provide a transfer learning model for collaborative recommendation based on generic mixed factorization mechanism to mine heterogeneous explicit feedbacks. He et al. [37] propose a recommendation algorithm based on the fast MF mechanism where the element-wise Alternating Least Squares (ALS) technique is adopted to optimize the MF model. Besides of the MF scheme, high-order tensor factorization scheme is also applied in CF recommendation algorithms to mine deeper relationships among users, items and other factors. In our previous studies [5, 38], CP (Canonical Polyadic) decomposition model is adopted to mine the relationships among users, items, time and location information for context-aware recommendation. Wang et al. [39] introduce a tensor-based big-data-driven routing recommendation model where a tensor matching approach integrates the controlling tensor, seed tensor and orchestration tensor for efficient routing paths recommendation. However, most existing factorization-based recommendation methods adopt the factorization technique to predict the target ratings directly, which only mine the interaction relationships between users and items. While in our proposal, we will apply the factorization-based latent factor model to extract the embedding vectors of users and IoT services (not predicting ratings directly), which reflect the co-

occurrence features of users and IoT services. In our method, we consider not only the interaction relationships but also the co-occurrence relationships between users and IoT services.

To deal with the security problem and protect users' sensitive information in dynamic and complex network environment, security concerns should be considered in recommendation algorithms. The studies [40-41] focus on the security-aware recommendation mechanisms for online social networks. Meng et al. [40] propose a privacy-preserving social recommendation method under personalized privacy settings where previous behaviors of users and social relationships are modeled in a secure manner. Li et al. [41] introduce a user group-based security-aware recommendation algorithm in online social communities. Some researchers also integrate the machine learning techniques into recommendation algorithms under security concerns [42-43]. The literature [42] presents a security-ware factorization-based recommendation method under local differential privacy where a dimensionality reduction mechanism and a novel binary scheme based on sampling are employed to reduce high dimensionality. Shu et al. [43] provide a privacy-preserving recommendation model for crowdsourcing where a key derivation method based on the MF technique is proposed for multi-keyword task-worker matching while protecting both the privacy of users and workers. The hash mapping technique is an effective technique for privacy preservation by hash mapping. The LSH mechanism is a well-known dimensionality reduction mechanism, which can be to realize fast similarity searching and privacy preservation in conventional CF recommendation models [44-45]. Based on LSH, the original user-related information and the inference data will be mapped into low-dimensional hash values. Then the hash values, rather than the original rating data of users, are input for the neighbor searching. Thus the sensitive information of IoT users will be blurred and protected. Therefore, in this paper, we employ the LSH mechanism to accelerate the neighbor searching in collaborative recommendation under security concerns.

3. System Overview

3.1 System model

In this paper, the two CF-based models, i.e., neighbor-based CF model and factorization-based latent factor model are combined to improve the prediction accuracy of recommender systems. First of all, the MF technique is used to learn the embedding vectors of users and IoT services instead of predicting ratings directly. Afterward, the LSH mechanism is adopted to cluster similar users (or IoT services) based on hash mapping, which not only speeds up similarity searching but also preserves users' privacy information by hash mapping. To be more specific, the neighbor searching is based on the hash values, rather than the observed rating data, which protects the sensitive information in users' rating data. Then the nearest neighbors of the target user can be determined as similar users are mapped into the same bucket by hash mapping. Finally, predictions and recommendations are made based on the improved neighbor-

based collaborative recommendation model. Fig. 1 shows the framework of our security-driven recommendation model.

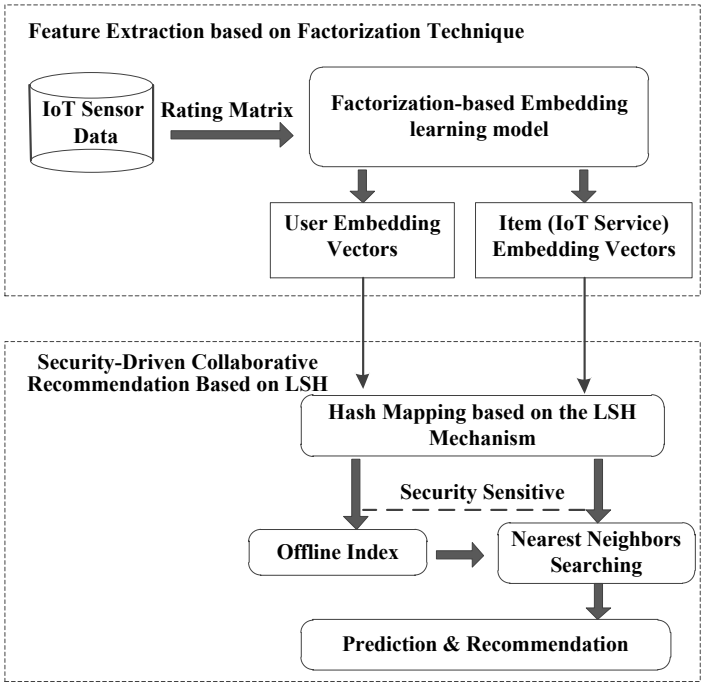


Fig. 1 Framework of security-driven hybrid collaborative recommendation model.

Feature extraction based on factorization: Firstly, the MF technique is used to learn embedding vectors of users and IoT services based on the user-user and service-service co-concurrence matrices mined from the user-service rating matrix. This process transforms original rating information into user-user and service-service co-occurrence embedding vectors.

Security-driven collaborative recommendation based on LSH: Once the feature vectors of users and IoT services are obtained through the above step, user vectors (or service vectors) are grouped by the LSH mechanism under security concerns. Then the nearest neighbors of the target user (or the target service) can be selected by the LSH-based hash mapping technique. At last, recommendations are put forward based on the improved neighbor-based CF recommendation model.

3.2 Factorization techniques

The MF technique is demonstrated effective in mining the latent factors among users and items. In traditional MF, users and items are mapped as vectors associated with a joint latent factor space [13], as defined below:

$$R = P^T \cdot Q = \left(p_{u1},\ p_{u2,\dots,}p_{uk}\right)^T \cdot (q_{i1}, q_{i2}, \dots, q_{ik}) \tag{1}$$

where R is the user-item rating matrix, P and Q respectively represent the feature vectors of users and items, and k is the number of implied factors related to users and items. Moreover, apart from user-item interaction relationships, biases associated with users and items should also be considered into predictions. Then Equation (1) can be rewritten as:

$$\hat{r}_{ui} = b_u + b_i + \sum_k p_{uk} \cdot q_{ik} \tag{2}$$

where b_u and b_i denote the rating biases of user u and item i, respectively. Biases indicate the effects associated with either users or items, which are independent of user-item interaction relations.

In the MF model, parameters can be learned by minimizing the loss function in Equation (3), i.e, the squared error function between the real rating value (r_{ui}) and the predicted rating value ($\hat{r}_{ui}$), where *Train* is the training rating pairs for r_{ui}.

$$min \sum_{(u,i) \in Train} (r_{ui} - \hat{r}_{ui})^2 \tag{3}$$

Minimization can be achieved by many learning algorithms in machine learning, such as Stochastic Gradient Descent (SGD), Alternating Least Squares (ALS), and Coordinate Descent (CD) [46].

MF is usually used to deal with a two-dimensional matrix. Also, some high-order factorization techniques such as the CP decomposition model, Tucker model, high-order SVD, and so on are available.

3.3 Local Sensitive Hash Mechanism

The LSH mechanism is a well-known machine learning technique effective in protecting sensitive information by hash mapping, which is also effective for the fast query and dimensionality reduction when dealing with massive data [47]. It can also be used in similarity searching, in which similar items will be mapped into the same bucket of the hash table. In this paper, it is applied to neighbor determination, which not only accelerates neighbor searching without investigating every pair, but also realizes the preservation of privacy information by hash mapping. The main scheme of LSH is described in the following, and more details can be found in reference [47].

In LSH, a collection of hash functions is called a hash function family. A hash family $H = \{h_1, h_2, ... h_m\}$ is said to be (B_1, B_2, p_1, p_2)-*sensitive* if each h_i in H satisfies:

(1) If $d(x, y) < B_1$, then $P(h_i(x) = h_i(y)) > p_1$

(2) If $d(x, y) > B_2$ then $P(h_i(x) = h_i(y)) < p_2$

where x and y represent the original data points, $d(x, y)$ is the distance between the data points x and y, $B_1 < B_2$, $P(\cdot)$ is a probability function, $h_i(x)$ and $h_i(y)$ respectively indicate the hash value of x and y after hash mapping, and p_1 and $p_2 (0 < p_1 < p_2 < 1)$ are two constants of the probability thresholds. In LSH, the setting of the hash function should satisfy: (1) the probability that $h_i(x)$ equals to $h_i(y)$ is bigger than P_1

when $(x, y) < B_1$; (2) the probability that $h_i(x)$ equals to $h_i(y)$ is less than P_2 when $d(x, y) > B_2$, as illustrated in Fig. 2 [47]. Obviously, it can be observed that the probability $P(h_i(x) = h_i(y))$ decreases with the rising distance between the original points x and y. Thus after hash mapping, users mapped with the same hash value (i.e., located in the same bucket) are mostly originally adjacent (similar).

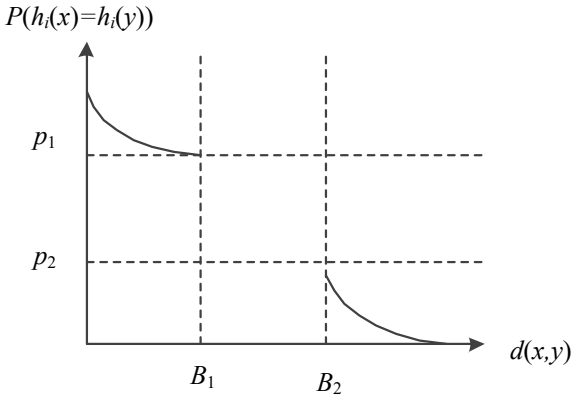


Fig. 2 (B_1, B_2, p_1, p_2)-sensitive hash function.

4. Security-Driven Hybrid Collaborative Recommendation Method

In this work, a security-driven hybrid collaborative method for cloud-based IoT services is proposed. It integrates the factorization-based latent factor model and the LSH mechanism into the neighbor-based CF recommendation model to improve the prediction accuracy while considering the security issue in the meantime. Our proposal consists of two phases, i.e., feature extraction with the factorization technique and security-driven collaborative recommendation with the LSH mechanism, as detailed below.

4.1 Feature Extraction based on the Factorization Technique

Inspired by the idea of references [16] and [17], user and service embedding vectors can be extracted by the MF technique according to the co-occurrence relationships mined from the rating matrix.

Given a user set U and an IoT service set I, assume that the number of users and IoT services are M and N, respectively. $R_{M\times N} = \{r_{ij}\}_{M\times N}$ is the user-service rating matrix, where r_{ij} is the rating of user i on service j. Suppose there are D kinds of ratings. The generation of user and service embedding vectors are respectively described in detail as follows. Firstly, the original rating matrix $R_{M\times N}$ will be transformed into the service co-occurrence matrix (or user co-occurrence matrix). Then embedding vectors can be learned from the co-occurrence matrix based on the MF scheme.

1) Generation of the co-occurrence matrix

Usually, IoT services with more same ratings are more similar. Additionally, services with different ratings may also reflect some implicit information. Therefore, when generating the co-occurrence matrix, we consider both the impact of the same ratings and different ratings. Fig. 3 presents an example of the generation of a user set for each service on each kind of rating. The left part of Fig. 3 shows a rating matrix with 4 users (u_1, u_2, u_3, u_4) and 4 IoT services (t_1, t_2, t_3, t_4). There are 3 kinds of ratings, i.e., {1, 3, 5}, and "0" indicates that the user didn't rate on the service. Firstly, the user set of each service can be obtained according to the ratings from users, as shown in the right part of Fig. 3, where $X_j^d = \{i | r_{ij} = d, \forall\, j \in U\}$ represents the set of users that rated the service i with the rating d.

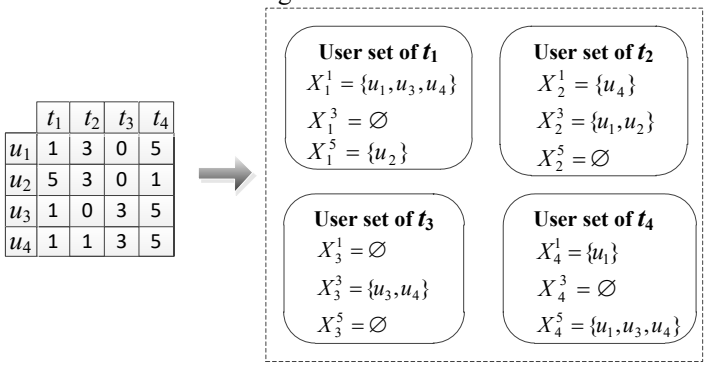
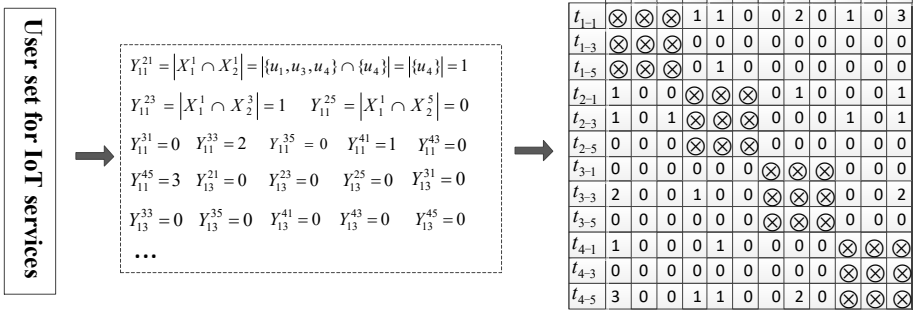


Fig. 3 Example of generation of the user set for services.

Based on the user set for each service generated in the above step, the co-occurrence values for services on different ratings can be calculated to construct the service co-occurrence matrix. Fig. 4 shows the generation of the service co-occurrence matrix for the example in Fig. 3. It is a $W \times W$ matrix, denoted as $IR_{W\times W}$ $(W = D \times M)$, which is constructed with the co-occurrence value $Y_{ig}^{jh} = \left|X_i^g \cap X_j^h\right|$. The co-occurrence values for the same IoT service (the diagonal values of the co-occurrence matrix) are meaningless and not input into the co-occurrence matrix.



	t_{1-1}	t_{1-3}	t_{1-5}	t_{2-1}	t_{2-3}	t_{2-5}	t_{3-1}	t_{3-3}	t_{3-5}	t_{4-1}	t_{4-3}	t_{4-5}
t_{1-1}	⊗	⊗	⊗	1	1	0	0	2	0	1	0	3
t_{1-3}	⊗	⊗	⊗	0	0	0	0	0	0	0	0	0
t_{1-5}	⊗	⊗	⊗	0	1	0	0	0	0	0	0	0
t_{2-1}	1	0	0	⊗	⊗	⊗	0	1	0	0	0	1
t_{2-3}	1	0	1	⊗	⊗	⊗	0	0	0	1	0	1
t_{2-5}	0	0	0	⊗	⊗	⊗	0	0	0	0	0	0
t_{3-1}	0	0	0	0	0	0	⊗	⊗	⊗	0	0	0
t_{3-3}	2	0	0	1	0	0	⊗	⊗	⊗	0	0	2
t_{3-5}	0	0	0	0	0	0	⊗	⊗	⊗	0	0	0
t_{4-1}	1	0	0	0	1	0	0	0	0	⊗	⊗	⊗
t_{4-3}	0	0	0	0	0	0	0	0	0	⊗	⊗	⊗
t_{4-5}	3	0	0	1	1	0	0	2	0	⊗	⊗	⊗

Fig. 4 Generation of the service co-occurrence matrix for the example in Fig. 3.

2) Learning embedding vectors with the factorization technique

After obtaining the service co-occurrence matrix, we use the MF technique to learn the service embedding vectors. In the conventional MF-based CF model, the rating matrix is factorized into inner products of a user-related matrix and an service-related matrix in a joint factor space $R = P^T \mathrm{Q}$. Similarly, in our proposal, the service co-occurrence matrix *IR* is factorized into the inner products of two different service embedding matrices, i.e. $\bar{C}$ and $\tilde{C}$, which denote the latent feature information of services.

$$IR = C^T \tilde{C} = [(\bar{c}_1^1, \bar{c}_1^2, \dots \bar{c}_1^D), (\bar{c}_2^1, \bar{c}_2^2, \dots \bar{c}_2^D), \dots (\bar{c}_M^1, \bar{c}_M^2, \dots \bar{c}_M^D)]^T \cdot [(\tilde{c}_1^1, \tilde{c}_1^2, \dots \tilde{c}_1^D), (\tilde{c}_2^1, \tilde{c}_2^2, \dots \tilde{c}_2^D), \dots (\tilde{c}_M^1, \tilde{c}_M^2, \dots \tilde{c}_M^D)] \quad (4)$$

where $\bar{c}_i^d$ and $\tilde{c}_i^d$ $(1 \le d \le D)$ respectively represent the embedding vectors of service i with d rating in matrix $\bar{C}$ and $\tilde{C}$. Then the overall embedding vector for service i is represented as $c_i = [(\bar{c}_i^1, \bar{c}_i^2, \dots \bar{c}_i^D), (\tilde{c}_i^1, \tilde{c}_i^2, \dots \tilde{c}_i^D)]$.

Y_{ig}^{jh} denotes the co-occurrence value between service i and service j where service i is with rating g and service j is with rating h. It can be obtained from the co-occurrence matrix for IoT services, and $Y_{ig}^{jh} = (\bar{c}_i^g)^T \tilde{c}_j^h$. Then $\bar{c}_i^g$ and $\tilde{c}_j^h$ can be learned by minimizing the loss function between the real occurrence value Y_{ig}^{jh} and the predicted value $(\bar{c}_i^g)^T \tilde{c}_j^h$. Herein, $\log Y_{ig}^{jh}$ is used to replace the original co-occurrence value Y_{ig}^{jh} for smoothing the co-occurrence value and reducing overfitting. Then the factorization function can be rewritten as:

$$\log Y_{ig}^{jh} = (\bar{c}_i^g)^T \tilde{c}_j^h \quad (5)$$

Moreover, biases that indicate the observed deviations of service i and j (denoted as b_i and b_j) are employed to improve the prediction accuracy of the service embedding vectors.

$$\log Y_{ig}^{jh} = b_i + b_j + (\bar{c}_i^g)^T \tilde{c}_j^h \quad (6)$$

Afterward, to learn the embedding vectors $\bar{c}_i^g$ and $\tilde{c}_j^h$, we minimize the squared error between the real co-occurrence value and the predicted co-occurrence value, expressed as:

$$\min_{b,i,j} L_I = \min \sum_{(i,j) \in I,\, Y_{ig}^{jh} > 0} (b_i + b_j + (\bar{c}_i^g)^T \tilde{c}_j^h - \log Y_{ig}^{jh})^2 \quad (7)$$

where I is the set of services, and $(i, j) \in I$ is the service-service pairs. In our method, the SGD technique is adopted to solve Equation (7) and learn the service embedding vectors, which can be calculated as:

$$\frac{\partial L_I}{\partial \bar{c}_i^g} = 2\rho \tilde{c}_j^h, \quad \frac{\partial L_I}{\partial \tilde{c}_j^h} = 2\rho \bar{c}_i^g, \quad \frac{\partial L_I}{\partial b_i} = 2\rho, \quad \frac{\partial L_I}{\partial b_j} = 2\rho \quad (8)$$

where $\rho = b_i + b_j + (\bar{c}_i^g)^T \tilde{c}_j^h - \log Y_{ig}^{jh}$. Then the parameters b_i, b_j, $\bar{c}_i^g$ and $\tilde{c}_j^h$ can be modified based on the following updating equations.

$$b_i \leftarrow b_i - 2\,\lambda\rho,\ \ b_j \leftarrow b_j - 2\,\lambda\rho$$
$$\bar{c}_i^g \leftarrow \bar{c}_i^g - 2\,\lambda\rho\tilde{c}_j^h,\ \ \tilde{c}_j^h \leftarrow \tilde{c}_j^h - 2\,\lambda\rho\bar{c}_i^g \tag{9}$$

where λ is the learning rate. It is employed to update the parameters by a magnitude proportional in the opposite direction of the gradient.

Similarly, the user embedding vectors also can be obtained by the above steps. Firstly, the user co-occurrence matrix $UR_{F\times F}$ $(F = D \times N)$ is got from the user-service rating matrix. Then the user co-occurrence matrix UR is represented as:

$$UR = S^T\tilde{S} = [(\bar{s}_1^1, \bar{s}_1^2, \ldots \bar{s}_1^D), (\bar{s}_2^1, \bar{s}_2^2, \ldots \bar{s}_2^D), \ldots (\bar{s}_N^1, \bar{s}_N^2, \ldots \bar{s}_N^D)]^T$$
$$\cdot [(\tilde{s}_1^1, \tilde{s}_1^2, \ldots \tilde{s}_1^D), (\tilde{s}_2^1, \tilde{s}_2^2, \ldots \tilde{s}_2^D), \ldots (\tilde{s}_N^1, \tilde{s}_N^2, \ldots \tilde{s}_N^D)] \tag{10}$$

where $\bar{s}_i^d$ and $\tilde{s}_i^d$ respectively represent the embedding vectors of user *i* on *d* rating in matrix $\bar{S}$ and $\tilde{S}$. Then the overall embedding vector for user *i* can be represented as s_i = $[(\bar{s}_i^1, \bar{s}_i^2, \ldots \bar{s}_i^D), (\tilde{s}_i^1, \tilde{s}_i^2, \ldots \tilde{s}_i^D)]$.

The embedding vectors $\bar{s}_i^g$ and $\tilde{s}_j^h$ of users are learned by minimizing the following loss function:

$$\min_{b,i,j} L_U = \min \sum_{(i,j)\in U,\, \hat{Y}_{ig}^{jh} > 0} (\hat{b}_i + \hat{b}_j + (\bar{s}_i^g)^T \tilde{s}_j^h - \log \hat{Y}_{ig}^{jh})^2 \tag{11}$$

where U is the set of users, and $(i, j) \in U$ is the user-user pairs. $\hat{Y}_{ig}^{jh}$ denotes the co-occurrence value between user *i* and user *j* where user *i* is with rating *g* and user *j* is with rating *h*. It can be obtained from the co-occurrence matrix for users. The updating equations are listed as:

$$\hat{b}_i \leftarrow \hat{b}_i - 2\,\lambda\varphi,\ \ \hat{b}_j \leftarrow \hat{b}_j - 2\,\lambda\varphi$$
$$\bar{s}_i^g \leftarrow \bar{s}_i^g - 2\,\lambda\varphi\tilde{s}_j^h,\ \ \tilde{s}_j^h \leftarrow \tilde{s}_j^h - 2\,\lambda\varphi\bar{s}_i^g \tag{12}$$

where $\varphi = \hat{b}_i + \hat{b}_j + (\bar{s}_i^g)^T \tilde{s}_j^h - \log \hat{Y}_{ig}^{jh}$.

4.2 Security-Driven Collaborative Recommendation based on LSH

(1) Nearest-Neighbor Searching based on LSH

In our work, the LSH mechanism is used to realize fast neighbor searching with privacy preservation, where users (represented by user embedding vectors) mapped with the same hash values (i.e., located in the same bucket) are mostly originally similar. In LSH, the setting of the hash functions corresponds to the distance measurement, which can be referred to [47]. A collection of hash functions is called a hash family. The original point mapped by a hash family with *m* hash functions will generate *m* hash values, which can be formulated as an m-dimensional vector denoted as the user index. Then users with the same index (hash value) will be mapped into the same bucket and considered more similar.

Section 4.1 has obtained the embedding vectors of users and services. The embedding vector of a user s is represented as $\boldsymbol{s} = (s_1, s_2, \dots s_n), n = D \times N$, where $s_i = [(\bar{s}_i^1, \bar{s}_i^2, \dots \bar{s}_i^D), (\tilde{s}_i^1, \tilde{s}_i^2, \dots \tilde{s}_i^D)]$.

In the neighbor-based CF recommendation model, PCC (Pearson correlation coefficient) is commonly used to find the similarities between users. Here, it is also selected as the similarity measurement in our method. Then a hash function family $h = (h_1, h_2, \dots, h_m)$ corresponding to the PCC distance can be defined as follows:

$$h_i(\boldsymbol{s}) = \begin{cases} 1, & \boldsymbol{s} \cdot \mathbf{A}_i \geq 0 \\ 0, & \boldsymbol{s} \cdot \mathbf{A}_i < 0 \end{cases} \tag{13}$$

where $\boldsymbol{s}$ is the embedding vector of user s, vector $\mathbf{A}_i = (a_1, a_2, \dots a_n)$ ($a_i \in [-1,1]$) is generated randomly and independently, and each $h_i(\boldsymbol{s})$ corresponds to a different random $\mathbf{A}_i$. Based on the hash function family h with m hash functions, the n-dimensional user embedding vector $\boldsymbol{s}$ is mapped into m hash values, which can be formulated as an m-dimensional binary vector $[h_1(\boldsymbol{s}), h_2(\boldsymbol{s}), \dots, h_n(\boldsymbol{s})]$ denoted as the user index where $h_i(\boldsymbol{s}) \in \{0,1\}$. Then the original n-dimensional user embedding vector is transformed into an m-dimensional vector (usually, $m \ll n$). One hash family corresponds to a hash table where users are mapped into the corresponding bucket and users in the same bucket are more similar. To improve the effectiveness of hash mapping, L hash families are used to generate L hash tables. As the hash tables only stores the binary hash values of users, instead of inference feature data or the original rating data, hence, the sensitive information of users is preserved.

Based on the definition of LSH, users with the same hash values (i.e., the same index) are more similar and mapped to the same bucket. L hash families correspond to L hash tables. In each hash table, the target user belongs to one bucket. Thus, for the target user s, its similar neighbor set is the union set of similar users in all L hash tables.

For example, as shown in Fig. 5, the red user u_r is the target IoT user. We intend to find the similar neighbors of user u_r based on the LSH mechanism. Based on the hash function families $H = (H_1, H_2, \dots, H_L)$, L hash tables can be generated (here, L=3). In each hash table, the target user u_r and its similar neighbor users belong to the same bucket. For instance, in Hash Table 1, user u_r and its similar user u_g are mapped into the same bucket (bucket$_{1r}$), i.e., $\{bucket_{1r}\} = \{u_r, u_g\}$. Similarly, $\{bucket_{2r}\} = \{u_r, u_g\}$, and $\{bucket_{3r}\} = \{u_r, u_y, u_b\}$. Then the neighbor user set (similar user set) of the target user u_r is expressed as:

$$\begin{aligned} Neigbor(u_r, H) &= \{bucket_{1r}\} \cup \{bucket_{2r}\} \cup \{bucket_{3r}\} - \{u_r\} \\ &= \{u_r, u_g\} \cup \{u_r, u_g\} \cup \{u_r, u_y, u_b\} - \{u_r\} = \{u_g, u_y, u_b\} \end{aligned}$$

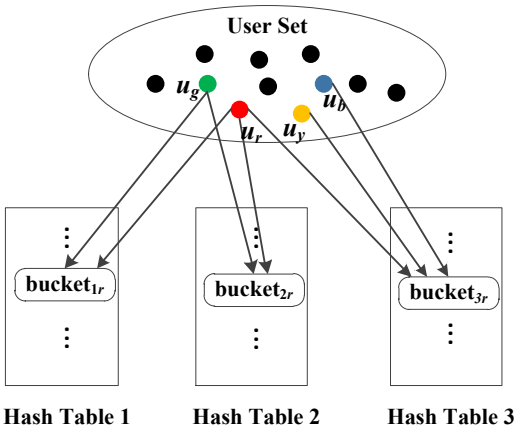


Fig. 5 Generation of hash tables based on LSH

(2) Prediction with LSH-based collaborative method

After determining the neighbor user set of the target user based on the above hash mapping mechanism, we can find the nearest neighbors of the target user s based PCC-based similarity scheme and a threshold δ, as defined below:

$$sim(s,v) = \frac{(\boldsymbol{s} - \overline{\boldsymbol{Nei_s}}) \cdot (\boldsymbol{v} - \overline{\boldsymbol{Nei_v}})}{\|\boldsymbol{s} - \overline{\boldsymbol{Nei_s}}\| \cdot \|\boldsymbol{v} - \overline{\boldsymbol{Nei_v}}\|} \tag{14}$$

where $\boldsymbol{s}$ is the m-dimensional embedding vector of the target user s, $\overline{\boldsymbol{Nei_s}}$ is the average vector of users in $Neigbor(s,H)$, and $Neigbor(s,H)$ is neighbors of user s obtained by LSH.

Then users in $Neigbor(s,H)$ whose similarity $sim(s,v)$ with user s is bigger than δ are included in the nearest neighbor set of the target user s, denoted as $NN(s)$. Once the nearest neighbor set of the target user is determined, the rating prediction of the target user s on service i can be predicted based on the following equation:

$$Pr_{si} = \overline{r_i} + \frac{\sum_{v \in NN(s) \cap NR(i)} (r_{vi} - \overline{r_i}) \cdot sim(s,v)}{\sum_{v \in NN(s) \cap NR(i)} |sim(s,v)|} \tag{15}$$

where $\overline{r_i}$ is the average rating of the rating for service i, $NN(s)$ is the nearest neighbor set of user s, $NR(i)$ is the set of users rated on service i, and r_{vi} is the rating of user v on service i.

The ratings of the target user on other services can also be predicted through the above steps. Finally, the services (or Top-K services) with the highest predicted ratings will be recommended to the target IoT user.

5 Experiments

In this section, experiments are designed and carried out to evaluate the efficiency of our method. Experimental settings are presented firstly and then experimental results are demonstrated with detailed analysis.

5.1 Experimental Settings

1) **Experimental Dataset**

MovieLens dataset is employed to evaluate the performance of our proposal. MovieLens refers to the movie rating datasets collected from the MovieLens website (http://movielens.org) and hosted by the GroupLens website. It is probably one of the most popular datasets for personalized recommendation research and social psychology. MovieLens datasets have various sizes, such as 100K, 1MB and so on. Herein, two kinds of 1M MovieLens datasets are selected. One is the "ml-latest-small" dataset (1 MB) with 100000 rating records from 600 users on 9000 movies; the other is the "ml-1m" dataset (6MB) with 1 million ratings from 6000 users on 4000 movies. In these two datasets, each user rated at least 20 movies and the ratings range from 1 to 5. The five-fold cross validation approach is used to evaluate the recommendation accuracy. The data in MovieLens datasets are divided into two parts, i.e., training data and test data, which account for 80% and 20% respectively.

2) **Comparative Approaches:**

To evaluate the recommendation accuracy of our method (SHCR), we compare it with three other recommendation approaches, i.e., IPCC, MF, and PUIPCC. IPCC is a typical neighbor-based CF recommendation approach, MF is a commonly used factorization-based latent factor model for collaborative recommendation, and PUIPCC is a privacy-aware CF recommendation approach. More details are given below.

IPCC [48]: an item-based CF recommendation method relying on the PCC similarity scheme.

MF [13]: a prediction algorithm based on the BiasSVD technique, which makes predictions by mining the latent factors from the rating matrix.

PUIPCC [49]: a CF-based prediction algorithm using a generic security-aware framework with data obfuscation schemes, which make predictions by integrating the similarity between users and that between items.

3) **Performance Metrics:**

To validate the effectiveness of SHCR in recommendation accuracy, three widely used evaluation metrics, i.e., Mean Absolute Error (MAE), Root-Mean-Square Error (RMSE), Precision and Recall [38] are applied.

MAE and RMSE are to measure the statistical accuracy performance of recommendation algorithms, defined as:

$$MAE = \frac{\sum_{(i,j)\in Train} \left| Pr_{ij} - r_{ij} \right|}{\left| Train \right|}$$

where (i, j) are rating pairs for r_{ij} in the training dataset, r_{ij} is the real rating value and Pr_{ij} is the predicted rating data.

$$RMSE = \sqrt{\frac{1}{|Train|}\sum_{(i,j)\in Train}\left(Pr_{ij} - r_{ij}\right)^2}$$

The lower the MAE and RMSE values are, the more accurate the prediction is. Precision and Recall are defined as.

$$\Pr ecision@K = \frac{\sum_{u\in U}\left|T_K^u \cap R_K^u\right|}{\sum_{u\in U}\left|R_K^u\right|}$$

$$\mathrm{Re}\,call@K = \frac{\sum_{u\in U}\left|T_K^u \cap R_K^u\right|}{\sum_{u\in U}\left|T_K^u\right|}$$

where T_K^u is the total item set in training dataset and R_K^u is the predicted Top-K recommendation item set.

Moreover, to evaluate the impact of LSH on our recommendation algorithm, we analyze the prediction accuracy and the runtime consumption of SHCR under different parameter settings.

5.2 Experimental Evaluation

1) **Recommendation Accuracy**

In this section, to evaluate the prediction accuracy of SHCR, we compare it with the other three methods (IPCC, MF, and PUIPCC) in MAE, RMSE, Precision and Recall. The experimental results are shown in Fig. 6-Fig 8, which are conducted on the "ml-latest-small" dataset.

Comparison in MAE&RMSE: Fig. 6 provides the prediction performance in MAE and RMSE of the four methods. Here, the number of hash functions (m) is set as 4 and the number of hash tables (L) is set as 8 (the optimal setting). In SHCR, to preserve privacy information of IoT users, we use the LSH mechanism to speed up the neighbor searching, which may affect the prediction accuracy since not all the similarity pairs are calculated. To address this shortcoming, the factorization-based latent factor model is integrated with the LSH-based neighbor CF model to improve the prediction accuracy. As the lower MAE and RMSE means the better prediction performance, it can be found that our method outperforms the other three methods and has the lowest MAE and RMSE. Specifically, MAE of SHCR is 13.46% lower than that of IPCC, 5.00 % lower than that of MF, and 7.51% lower than that of PUIPCC. And the RMSE of SHCR is 12.01%, 6.66%, and 7.97% lower than that of IPCC, MF, and PUIPCC respectively. Thus compared with three other comparative methods, the proposed method, i.e., SHCR, is the optimal recommendation policy for IoT services with the consideration of security concerns.

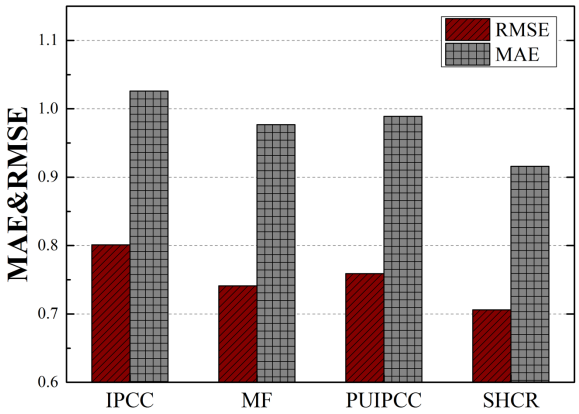


Fig. 6 Recommendation accuracy in MAE&RMSE.

Comparison in Precision and Recall: Fig. 7 and Fig. 8 show the performance of the four methods in Top-K prediction accuracy. Here, the number of hash functions (m) in each hash family and the number of hash tables (L) are fixed as 4 and 8 respectively. Fig. 7 presents Top-K (K=5, 10, 15) recommendation performance in the precision, and Fig. 8 presents Top-K recommendation performance in the recall. Clearly, with the increase of K, the precision of the four methods decreases and the recall increases. Therefore, smaller K means higher Top-K prediction accuracy. Moreover, in terms of the Top-N recommendation, SHCR outperforms the other three comparative methods in both precision and recall, further demonstrating that SHCR has better prediction accuracy under security concerns.

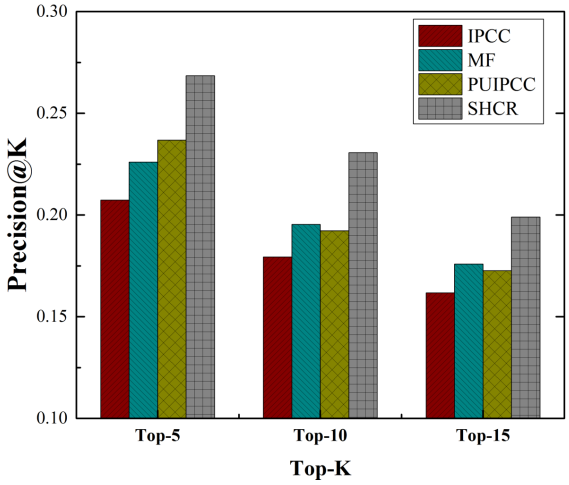


Fig. 7 Top-K recommendation accuracy on precision.

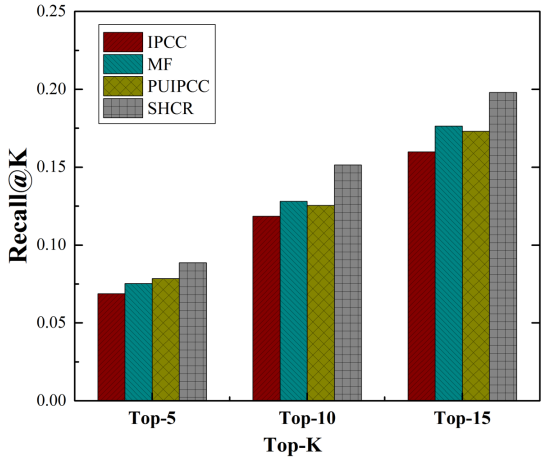


Fig. 8 Top-K recommendation accuracy on recall.

2) Parameter Impact on Recommendation Efficiency

This section is aimed to analyze the impact of parameters m (the number of hash functions) and L (the number of hash labels) on the performance of recommendation accuracy. The simulation is conducted on the "ml-1m" dataset which has a larger scale than the "ml-latest-small" dataset. Fig. 9 presents the MAE value of our method with the change of m and L, where m varies from 2 to 8 with a step of 2 and L varies from 4 to 10 with a step of 2.

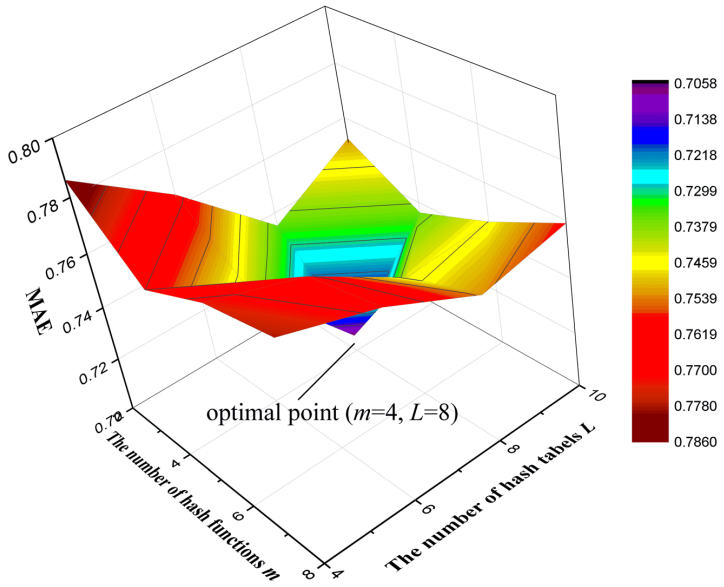


Fig. 9 Impact of the number of hash functions (m) and the number of hash labels (L) on recommendation accuracy (MAE).

From Fig. 9, we can see that as m grows, the MAE value of SHCR decreases first ($2 \leq m < 4$) and then increases ($4 < m \leq 8$). It indicates that the prediction accuracy of our method is promoted with the increase of m ($2 \leq m < 4$) and then degrades with the increase of m ($4 < m \leq 8$). Similarly, with the increase of L, the prediction accuracy of SHCR is improved first ($4 \leq m < 8$) and then degrades ($8 < m \leq 10$). As shown in the 3D map in Fig. 9, the optimal setting of m and L for the best prediction performance is at the point where m is 4 and L is 8. Thus the settings of m and L have an impact on prediction accuracy. Therefore, appropriate settings for the number of hash functions and hash tables should be made to get the best performance.

3) **Parameter Impact on Runtime Efficiency**

This section presents the runtime of SHCR with the change in the numbers of hash functions (m) and the number of hash tables (L). In Fig. 10, when m is small (such as m = 2), the runtime of SHCR almost remains unchanged with the increase of the number of hash tables L. When L is big (such as L=10), our method still keeps stable with the change in the numbers of hash functions m. Overall, the change in runtime consumption of SHCR is not obvious with the change of m and L. It indicates that the change of the numbers of hash functions and hash tables in LSH have a slight influence on the efficiency of SHCR in runtime, guaranteeing the scalability of SHCR. Thus the LSH technique is an effective tool to ensure both recommendation efficiency and security concerns.

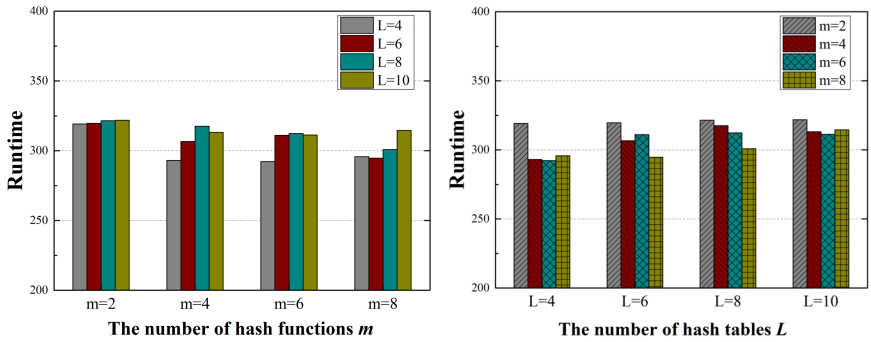


Fig. 10 Impact of the number of hash functions (m) and the number of hash labels (L) on runtime.

4) **Security Analysis**

Privacy preservation is crucial to IoT-service recommender systems, such as IoT-based diagnostic recommender systems and IoT-based food recommender systems, where user-related sensitive information needs to be preserved. In this paper, to protect the privacy information of IoT users, the LSH mechanism is adopted to blur users' specific information through the mapping technique. To be more specific, in neighbor searching, the original user-related information and the inference feature data are mapped into low-dimensional hash values. And the hash values, rather than the original rating data or feature data, are applied to neighbor searching. Thus the sensitive rating information of IoT users in rating data is protected. From the experimental re-

sults in Fig.6-Fig.9, we can see that our proposal can achieve improved prediction accuracy under security concerns.

6 Conclusions

In this paper, we propose a security-driven hybrid collaborative recommendation method for cloud-based IoT services. It integrates the factorization-based latent factor model with the neighbor-based CF model to improve the recommendation performance. Moreover, the LSH mechanism is employed to process privacy-related information of IoT users and speed up the neighbor searching. Firstly, the MF technique is used to learn the embedding vectors of users and IoT services firstly. Afterward, the LSH technique is used for fast similarity searching and the determination of the target user's nearest neighbors under security concerns. Final predictions are made based on the nearest neighbors determined by hash mapping as well as the improved neighbor-based CF recommendation algorithm. Experimental results show that our proposed method achieves improved prediction accuracy while considering the security issues in the meantime. In our further work, we plan to combine the knowledge graph technology and AI techniques with the CF recommendation algorithm to learn the interaction features between users and items, hoping to further improve the recommendation accuracy further.

Acknowledgements

This paper is partially supported by the National Natural Science Foundation of China under Grant No. 61702264, No. 61872219, No. 61761136003, the Fundamental Research Funds for the Central Universities under Grant No. 30918014108, No. 30918012204, the Postdoctoral Science Foundation of China under Grant No. 2019M651835, the Natural Science Foundation of Shandong Province, No. ZR2019MF001, the Open Project of State Key Laboratory for Novel Software Technology under Grant No. KFKT2020B08, the 4th project "Research on the Key Technology of Endogenous Security Switches" (No. 2020YFB1804604) of the National Key R&D Program "New Network Equipment Based on Independent Programmable Chips" (No. 2020YFB1804600), and the 2020 Industrial Internet Innovation and Development Project from Ministry of Industry and Information Technology of China.

References

[1] Xu X, Zhang X, Gao H, et al. BeCome: Blockchain-enabled computation offloading for IoT in mobile edge computing. IEEE Transactions on Industrial Informatics, 2019. DOI: 10.1109/TII.2019.2936869

[2] Gai K, Qiu M, Zhao H. Privacy-preserving data encryption strategy for big data in mobile cloud computing. IEEE Transactions on Big Data, 2017. DOI: 10.1109/TBDATA.2017.2705807.

[3] Xu X, Liu Q, Luo Y, et al. A computation offloading method over big data for IoT-enabled cloud-edge computing. Future Generation Computer Systems, 2019, 95: 522-533.
[4] Zhou X, Liang W, Kevin I, et al. Deep Learning Enhanced Human Activity Recognition for Internet of Healthcare Things. IEEE Transactions on Emerging Topics in Computing, 2018. DOI: 10.1109/TETC.2018.2860051.
[5] Meng S, Qi L, Li Q, et al. Privacy-preserving and sparsity-aware location-based prediction method for collaborative recommender systems. Future Generation Computer Systems, 2019, 96: 324-335.
[6] Quick D, Choo K K R. Dropbox analysis: Data remnants on user machines. Digital Investigation, 2013, 10(1): 3-18.
[7] Li J, Cai T, Deng K, et al. Community-diversified influence maximization in social networks. Information Systems, 2020, 92: 101522.
[8] Gai K, Choo K K R, Qiu M, et al. Privacy-preserving content-oriented wireless communication in internet-of-things. IEEE Internet of Things Journal, 2018, 5(4): 3059-3067.
[9] Xu X, He C, Xu Z, et al. Joint optimization of offloading utility and privacy for edge computing enabled IoT. IEEE Internet of Things Journal, 2019. DOI: 10.1109/JIOT.2019.2944007.
[10] Qi L, Dou W, Wang W, et al. Dynamic mobile crowdsourcing selection for electricity load forecasting. IEEE Access, 2018, 6: 46926-46937.
[11] Pu Z, Li Z, Ash J, et al. Evaluation of spatial heterogeneity in the sensitivity of on-street parking occupancy to price change. Transportation Research Part C: Emerging Technologies, 2017, 77: 67-79.
[12] Liu H, Kou H, Yan C, et al. Link prediction in paper citation network to construct paper correlation graph. EURASIP Journal on Wireless Communications and Networking, 2019, 2019(1): 1-12.
[13] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer, 2009: 42(8).
[14] Luo C, Zhang K, Salinas S, et al. Secfact: Secure large-scale QR and LU factorizations. IEEE Transactions on Big Data, 2017, DOI: 10.1109/TBDATA.2017.2782809.
[15] Zhou X, Liang W, Kevin I, et al. Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data. IEEE Transactions on Emerging Topics in Computing, 2018. DOI: 10.1109/TETC.2018.2860051
[16] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.
[17] Fu M, Qu H, Yi Z, et al. A novel deep learning-based collaborative filtering model for recommendation system. IEEE transactions on cybernetics, 2018, 49(3): 1084-1096.
[18] Chen Y, Zhang N, Zhang Y, et al. Energy efficient dynamic offloading in mobile edge computing for Internet of Things. IEEE Transactions on Cloud Computing, 2019, DOI 10.1109/TCC.2019.2898657.
[19] Pu Z, Li Z, Jiang Y, et al. Full Bayesian Before-After Analysis of Safety Effects of Variable Speed Limit System. IEEE Transactions on Intelligent Transportation Systems, 2020, DOI:10.1109/TITS.2019.2961699.
[20] Chi X, Yan C, Wang H, et al. Amplified LSH-based recommender systems with privacy protection. Concurrency and Computation: Practice and Experience, 2020. DOI: 10.1002/CPE.5681.
[21] Zhou X, Liang W, Huang S, et al. Social Recommendation With Large-Scale Group Decision-Making for Cyber-Enabled Online Service. IEEE Transactions on Computational Social Systems, 2019, 6(5): 1073-1082.

[22] Saleem Y, Crespi N, Rehmani M H, et al. Exploitation of social IoT for recommendation services. 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). IEEE, 2016: 359-364.
[23] Mashal I, Alsaryrah O, Chung T Y. Performance evaluation of recommendation algorithms on Internet of Things services. Physica A: Statistical Mechanics and its Applications, 2016, 451: 646-656.
[24] Subramaniyaswamy V, Manogaran G, Logesh R, et al. An ontology-driven personalized food recommendation in IoT-based healthcare system. The Journal of Supercomputing, 2019, 75(6): 3184-3216.
[25] He X, Liao L, Zhang H, et al. Neural collaborative filtering. Proceedings of the 26th international conference on world wide web. 2017: 173-182.
[26] Wei J, He J, Chen K, et al. Collaborative filtering and deep learning based recommendation system for cold start items. Expert Systems with Applications, 2017, 69: 29-39.
[27] Luo C, Ji J, Wang Q, et al. Channel state information prediction for 5G wireless communications: A deep learning approach. IEEE Transactions on Network Science and Engineering, 7(1): 227-236, 2018.
[28] Ebesu T, Shen B, Fang Y. Collaborative memory network for recommendation systems. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018: 515-524.
[29] Pu, Z, Li, Z, Ke, R, et al. Evaluating the Non-Linear Correlation between Vertical Curve Features and Crash Frequency on Highways using Random Forests. Journal of Transportation Engineering Part A: System, 2020, Doi:10.1061/JTEPBS.0000410.
[30] Zhou X, Liang W, Kevin I, et al. Multi-modality behavioral influence analysis for personalized recommendations in health social media environment. IEEE Transactions on Computational Social Systems, 2019, 6(5): 888-897.
[31] Bu J, Shen X, Xu B, et al. Improving collaborative recommendation via user-item subgroups. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(9): 2363-2375.
[32] Meng S, Dou W, Zhang X, et al. KASR: a keyword-aware service recommendation method on mapreduce for big data applications. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(12): 3221-3231.
[33] Qi L, Dou W, Zhou Y, et al. A context-aware service evaluation approach over big data for cloud applications. IEEE Transactions on Cloud Computing, 2015. DOI: 10.1109/TCC.2015.2511764.
[34] Zhou P, Zhou Y, Wu D, et al. Differentially private online learning for cloud-based video recommendation with multimedia big data in social networks. IEEE transactions on multimedia, 2016, 18(6): 1217-1229.
[35] Lian D, Ge Y, Zhang F, et al. Scalable content-aware collaborative filtering for location recommendation. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(6): 1122-1135.
[36] Pan W, Xia S, Liu Z, et al. Mixed factorization for collaborative recommendation with heterogeneous explicit feedbacks. Information Sciences, 2016, 332: 84-93.
[37] He X, Zhang H, Kan M Y, et al. Fast matrix factorization for online recommendation with implicit feedback. Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. 2016: 549-558.
[38] Meng S, Li Q, Zhang J, et al. Temporal-aware and sparsity-tolerant hybrid collaborative recommendation method with privacy preservation. Concurrency and Computation: Practice and Experience, 2020, 32(2): e5447.
[39] Wang X, Yang L T, Kuang L, et al. A tensor-based big-data-driven routing recommendation approach for heterogeneous networks. IEEE Network, 2019, 33(1): 64-69.
[40] Meng X, Wang S, Shu K, et al. Personalized privacy-preserving social recommendation. Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[41] Li D, Lv Q, Shang L, et al. Efficient privacy-preserving content recommendation for online social communities. Neurocomputing, 2017, 219: 440-454.
[42] Shin H, Kim S, Shin J, et al. Privacy enhanced matrix factorization for recommendation with local differential privacy. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(9): 1770-1782.
[43] Shu J, Jia X, Yang K, et al. Privacy-preserving task recommendation services for crowdsourcing. IEEE Transactions on Services Computing, 2018.
[44] Qi L, Wang X, Xu X, et al. Privacy-Aware Cross-Platform Service Recommendation based on Enhanced Locality-Sensitive Hashing. IEEE Transactions on Network Science and Engineering, 2020, DOI: 10.1109/TNSE.2020.2969489.
[45] Qi L, He Q, Chen F, et al. Data-Driven Web APIs Recommendation for Building Web Applications. IEEE Transactions on Big Data, 2020, DOI: 10.1109/TBDATA.2020.2975587.
[46] Shin K, Sael L, Kang U. Fully scalable methods for distributed tensor factorization. IEEE Transactions on Knowledge and Data Engineering, 2016, 29(1): 100-113.
[47] Rajaraman A, Ullman J D. Mining of massive datasets. Cambridge University Press, 2011.
[48] Linden G, Smith B, York J. Amazon.com Recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 2003 (1): 76-80.
[49] Zhu J, He P, Zheng Z, et al. A Privacy-preserving QoS Prediction Framework for Web Service Recommendation. 2015 IEEE International Conference on Web Services (ICWS), 2015: 241-248.