Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 2401857, 13 pages
https://doi.org/10.1155/2019/2401857

*Research Article*

# Robust and Privacy-Preserving Service Recommendation over Sparse Data in Education

**Xuening Chen,[1] Hanwen Liu,[2] Yanwei Xu,[3] and Chao Yan[2]**

[1] *Student Affairs Office, Qufu Normal University, China*
[2] *School of Information Science and Engineering, Qufu Normal University, China*
[3] *School of Software, Tianjin University, China*

Correspondence should be addressed to Chao Yan; yanchao@qfnu.edu.cn

Received 20 November 2018; Revised 28 December 2018; Accepted 28 May 2019; Published 20 June 2019

Guest Editor: Qingchen Zhang

Service recommendation has become one of the most effective approaches to quickly extract insightful information from big educational data. However, the sparsity of educational service quality data (from multiple platforms or parties) used to make service recommendations often leads to few even null recommended results. Moreover, to protect sensitive business information and obey laws, preserving user privacy during the abovementioned multisource data integration process is a very important but challenging requirement. Considering the above challenges, this paper integrates Locality-Sensitive Hashing (LSH) with hybrid Collaborative Filtering (HCF) techniques for robust and privacy-aware data sharing between different platforms involved in the cross-platform service recommendation process. Furthermore, to minimize the "False negative" recommended results incurred by LSH and enhance the success of recommended results, we propose two optimization strategies to reduce the probability that similar neighbours of a target user or similar services of a target service are overlooked by mistake. Finally, we conduct a set of experiments based on a real distributed service quality dataset, i.e., *WS-DREAM*, to validate the feasibility and advantages of our proposed recommendation approach. The extensive experimental results show that our proposal performs better than three competitive methods in terms of efficiency, accuracy, and successful rate while guaranteeing privacy-preservation.

## 1. Introduction

With the advent of the Web of Things (WoT), tremendous computing resources or services (e.g., web APIs) are emerging rapidly on the Web [1–4], imposing a heavy burden on the service selection decisions of target users in education domain. In this situation, various lightweight service recommendation techniques, e.g., Collaborative Filtering (CF), are proposed to alleviate the abovementioned service selection burdens. Typically, by analysing the historical service usage data (e.g., the quality data of services invoked by users), a recommender system can capture the personalized preferences of a user and output appropriate services to him/her; this way, complex requirements from the user could be satisfied [5–7].

However, in the big data environment, the recommendation bases for educational decisions are sometimes not centralized but are distributed across multiple platforms [8, 9]. Considering the example in Figure 1, user $u_1$ invoked web service $ws_1$ from platform $P_1$ and user $u_2$ invoked web service $ws_2$ from platform $P_2$. Thus previous service quality values of $ws_1$ and $ws_2$ are recorded in platforms $P_1$ and $P_2$, respectively. In this situation, to make comprehensive and accurate service recommendations to the target user, it is necessary for the recommender system to integrate or fuse the distributed educational data across platforms $P_1$ and $P_2$ properly.

However, there are still several challenges in the abovementioned data integration process. First, to protect sensitive business information [10, 11] and obey laws, platform $P_1$ is often reluctant to share its data with $P_2$ and vice versa [12]. Such a cross-platform data sharing failure severely impedes the subsequent service recommendations. Besides, the possible sparsity of service quality data [13, 14] stored in platforms $P_1$ and $P_2$ often leads to few (even null) recommended

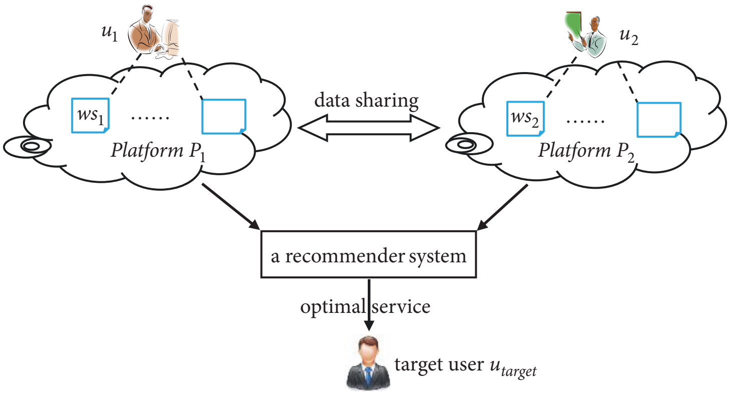


Figure 1: Cross-platform service recommendation example.

results, which decreases the target user's satisfaction degree significantly. Namely, the robustness of the recommender system is not as high as expected.

Considering the drawbacks, a time-efficient and privacy-preserving neighbour search technique, Locality-Sensitive Hashing (LSH), is employed for cross-platform service recommendations, so that the multiple platforms involved in the distributed recommendation process can share their data with each other efficiently and securely. Furthermore, we combine the *LSH* technique with hybrid CF (i.e., *HCF*, including user-based CF and item-based CF), to propose a novel privacy-preserving cross-platform *ser*vice *rec*ommendation approach, named *SerRec*$_{LSH+HCF}$. Benefiting from the advantages of LSH in terms of search efficiency and privacy-preservation, our proposal can achieve a good trade-off among recommendation efficiency, accuracy, successful rate, and privacy-preservation.

In summary, our contributions are three-fold.

(1) We integrate the LSH technique with hybrid CF to guarantee efficient and secure data sharing between different platforms involved in the cross-platform service recommendations in a big educational data environment.

(2) Two solutions are suggested to reduce the probability of "False negative" (i.e., high-quality recommended results are overlooked by mistake) incurred by the inherent shortcoming of LSH and thereby increase the success ratio of recommended list.

(3) A wide range of experiments are conducted on a real distributed service quality dataset, i.e., *WS-DREAM* to validate the feasibility of our proposal. Experiment results show that our proposed *SerRec*$_{LSH+HCF}$ approach outperforms the other state-of-the-art approaches.

The remainder of this paper is structured as follows. Related work is presented in Section 2. In Section 3, we introduce the preliminary knowledge of the LSH technique to be used in our approach. In Section 4, we introduce the details of our proposed privacy-preserving cross-platform service recommendation approach, i.e., *SerRec*$_{LSH+HCF}$. In Section 5, a set of experiments are conducted on *WS-DREAM* dataset to validate the feasibility of our proposal. Finally, in Section 6, we conclude the paper and discuss the future research directions.

## 2. Related Work

To the best of our knowledge, the existing privacy-preservation techniques adopted in the field of service recommendations can be divided into the following four categories: *K-anonymity*, *data obfuscation*, *data decomposition*, and *Locality-Sensitive Hashing*. Next, we introduce the related work from these four perspectives, respectively.

### 2.1. K-Anonymity.

As an effective privacy-preservation technique, $K$-anonymity is successfully applied in [15] to protect the sensitive data of users. The authors in [16] recruit $K$-anonymity technique to generalize the location information that users left in the past so as to protect the users' location privacy when making a recommendation decision. Generally, a larger $K$ value often means better privacy-preservation performance. However, when the $K$ value becomes larger, the availability of anonymous data would be reduced significantly, thereby decreasing the accuracy of recommended results.

### 2.2. Data Obfuscation.

Random data obfuscation technique is proposed in various applications where the real service quality data are replaced by the obfuscated data so that the private information hidden in real service quality data can be protected. However, as the data used to make recommendation decisions have already been obfuscated beforehand, the service recommendation accuracy is reduced accordingly. Differential Privacy (DP) technique is recruited in [17] to obfuscate the sensitive service quality data by noise injection so as to hide the real service quality data when making service recommendation decisions. However, the time complexity of Differential Privacy technique is often high. Besides, when the service quality data for recommendation decisions are updated frequently, the accumulated noise amount will become increasingly larger; in this situation, the data availability is reduced, which influences the accuracy of returned results to some extent.

### 2.3. Data Decomposition.

In [18], the authors propose a data decomposition mechanism to achieve the privacy-preservation goal in service recommendations. Concretely, each sensitive quality data is transformed to be multiple segments with less privacy information; afterwards, these service quality segments with little privacy are sent to different user clients for storage. Thus when a user requests service recommendations, the multiple service quality segments kept by each user client are integrated together for subsequent recommendation decision-making process. As each user only possesses multiple service quality segments from different quality data, instead of the whole service quality data, the sensitive information from users is secured, while this approach still fails to secure certain privacy, for example, *the intersection of services executed by different users*.

*2.4. Locality-Sensitive Hashing (LSH).* As an effective technique for quick neighbour search from massive and high-dimensional data, LSH has recently been introduced into service recommendation for privacy-preservation. In our previous work [19, 20], the LSH technique is combined with user-based CF to protect the sensitive service quality data engaged in recommendation process. In [21], LSH is recruited to build service indices in the distributed environment, so as to reduce the cross-platform data communication cost and improve the recommendation efficiency. However, these LSH-based service recommendation approaches do not consider the low successful rate incurred by the possible sparsity of recommendation data. Moreover, they seldom study the "False negative" recommended results as well as the corresponding resolutions.

With the above analyses, we can conclude that existing privacy-preserving service recommendation approaches either fall short in the efficiency and the capability of privacy-preservation, or they probably overlook high-quality recommended results so that the users' satisfaction degree is decreased. Considering these drawbacks, we integrate the LSH technique and hybrid CF in this paper to propose a novel privacy-preserving service recommendation approach named $SerRec_{LSH+HCF}$. The details of our proposal will be introduced in Section 4.

## 3. Preliminary Knowledge

In Section 3.1, we first formulate the privacy-preserving service recommendation problems to be addressed in this paper. Afterwards, in Section 3.2, we briefly introduce the rationale of the LSH technique to be used in our service recommendation approach.

*3.1. Problem Formulation.* To facilitate the following discussions, we introduce the symbols used in this paper below. $U = \{u_1, \ldots, u_m\}$ and $WS = \{ws_1, \ldots, ws_n\}$ mean user set and service set, respectively; $u_{target}$ and $ws_{target}$ denote a target user and a target service (i.e., a service preferred by the target user), respectively; $q$ is a quality dimension of web services, e.g., *response time* or *throughput* (for simplicity, only one quality dimension is considered in this paper); $q_{i,j}$ denotes the quality of $q$ of service $ws_j$ ($\in WS$) ever-invoked by user $u_i$ ($\in U$) and the $q_{i,j}$ data are often distributed across different platforms in the big data environment.

With the above formulation, our focused privacy-preserving service recommendation problems can be specified more formally as follows: recommend appropriate services from set $WS$ to target user $u_{target}$ based on the historical $q_{i,j}$ data across different platforms and meanwhile protect the real value of $q_{i,j}$ so that the users' private information hidden in $q_{i,j}$ data is still secure.

*3.2. Locality-Sensitive Hashing.* Locality-Sensitive Hashing has been considered as one of the most effective techniques for similar neighbour search due to the following two properties [22]. Here, $A$ and $B$ are two points in original data space, and $h(.)$ denotes a LSH function that is responsible for transforming points $A$ and $B$ into corresponding hash values $h(A)$ and $h(B)$, respectively.

*Property 1.* If $A$ and $B$ are close in original data space, then they will be projected into the same bucket (i.e., $h(A) = h(B)$) after hashing with high probability.

*Property 2.* If $A$ and $B$ are not close in original data space, then they will be projected into different buckets (i.e., $h(A) \neq h(B)$) after hashing with high probability.

Thus, inspired by these two properties, we can utilize the hash values $h(A)$ and $h(B)$ (with little or no privacy) to evaluate the approximation degree of original points $A$ and $B$, without revealing the details of $A$ and $B$. This way, the private information of points $A$ and $B$ can be protected.

## 4. $SerRec_{LSH+HCF}$: Service Recommendation Based on LSH and Hybrid CF

In this section, we introduce our proposed privacy-preserving service recommendation approach, i.e., $SerRec_{LSH+HCF}$. Concretely, in Section 4.1, we utilize the LSH technique and user-based CF to make service quality prediction; in Section 4.2, we utilize the LSH technique and item-based CF to make service quality prediction. Finally, in Section 4.3, we integrate the predicted results of Sections 4.1 and 4.2 and then make service recommendations accordingly.

*4.1. Service Quality Prediction Based on LSH and User-Based CF.* In this subsection, we utilize user-based CF and LSH to look for a target user's similar neighbours (denoted by set $Neighbour\_set(u_{target})$) in a privacy-aware and scalable manner, and then the method makes service quality prediction based on the derived similar neighbours in $Neighbour\_set(u_{target})$.

First, for any $u \in U$, the quality data over dimension $q$ are simply converted into an $n$-dimensional quality vector $\overrightarrow{u} = (q_{u,1}, \ldots, q_{u,n})$. Here, $q_{u,j}$ denotes the quality value of $q$ of $ws_j$ invoked by user $u$ (typically, $q_{u,j} = 0$ if user $u$ did not rate $ws_j$ in the past) and $n$ is the number of candidate web services. Next, we introduce how to utilize the LSH technique to transform vector $\overrightarrow{u}$ with much private information into corresponding user index $h(u)$ with little privacy, based on a pre-selected LSH function $h(.)$.

Concretely, the concrete forms of LSH function $h(.)$ heavily rely on the "distance" for user similarity measurement; in other words, different types of similarity "distance" correspond to different kinds of LSH functions. The Pearson Correlation Coefficient (PCC) is often utilized to calculate user similarity in existing recommender systems, so we choose the LSH function corresponding to the PCC distance in this paper. More concretely, the LSH function $h(.)$ in (1) is adopted [23]. Here, $\overrightarrow{v}$ is an $n$-dimensional vector $(v_1, \ldots, v_n)$, where $v_j$ $(1 \le j \le n)$ is a random value in the range $[-1, 1]$; symbol "$\circ$" represents the dot product between two vectors. This way, through (1), we can transform $\overrightarrow{u}$ with much privacy into a Boolean value $h(u)$ with little privacy.

$$h(u) = \begin{cases} 1 & \text{if } \overrightarrow{u} \circ \overrightarrow{v} > 0 \\ 0 & \text{if } \overrightarrow{u} \circ \overrightarrow{v} \le 0 \end{cases} \quad (1)$$

As LSH is essentially a probability-based similar neighbour search technique, one hash function $h(.)$ is often not enough for finding the similar neighbours of a target user accurately. In view of this observation, we amplify the performance of LSH by adopting $r$ hash functions $\{h_1(.), \ldots, h_r(.)\}$ and $L$ hash tables $\{Table_1, \ldots, Table_L\}$ into the similar neighbour search processes. Concretely, in each hash table, we can build an index for user $u$, denoted by $H(u) = (h_1(u), \ldots, h_r(u))$. Furthermore, two users $u_1$ and $u_2$ are regarded as similar iff condition in (2) holds, where $H_x(u_1)$ and $H_x(u_2)$ denote the indices of $u_1$ and $u_2$ in the $x$-th hash table (i.e., $Table_x$), respectively.

$$\exists x \in \{1, \ldots, L\}, \quad \text{satisfy } H_x(u_1) = H_x(u_2) \quad (2)$$

Likewise, for the target user, i.e., $u_{target}$, we can calculate his/her user index value $H_x(u_{target})$ in $Table_x$ ($x = 1, \ldots, L$), according to the same LSH functions and LSH tables. Then, through the condition in (2), we can determine the similar neighbours of $u_{target}$ and put them into set $Neighbour\_set(u_{target})$. The pseudocode of the above neighbour search process is presented in Algorithms 1 and 2, where Algorithm 1 is used to build the $L$ hash tables for users offline and Algorithm 2 is used to search for the similar neighbours of the target user.

However, as LSH is a probability-based neighbour finding technique, the "False negative" search results are inevitable. In other words, some similar neighbours of a target user may be overlooked by mistake according to the abovementioned LSH-based neighbour search process. In view of this drawback, we propose two optimization strategies to reduce the "False negative" probability and improve the successful rate of neighbour search. Next, we introduce these two strategies, respectively.

*Strategy 1* (neighbour propagation (for users)). The neighbour relationship between different users is essentially depicted by the similarity of user preferences, while the latter (i.e., user preference similarity) obeys a kind of propagation rule. Let us consider the example in Figure 2 where three users $\{u_1, u_2, u_3\}$ and four web services $\{ws_1, ws_2, ws_3, ws_4\}$ are present. The user-service ratings (1∗~5∗) are shown in Figure 2(a), according to which we can determine the neighbour relationship between $u_1$ and $u_2$ as well as the neighbour relationship between $u_1$ and $u_3$. In this situation, we can infer that $u_2$ and $u_3$ are possible neighbours (marked with dotted line in Figure 2(b)) as both of them hold the same or similar preferences with $u_1$, although $u_2$ and $u_3$ are not direct neighbours based on the user-service rating data in Figure 2(a). This way, through the neighbour propagation rule illustrated in Figure 2, we can find more possible neighbours of a target user (in an indirect manner) so as to reduce the "False negative" probability.

Next, we introduce how to integrate the neighbour propagation strategy (for users) into the abovementioned LSH-based neighbouring user search process. Concretely, if users $u_a$ and $u_{target}$ are projected into an identical bucket in any of the $L$ hash tables $\{Table_1, \ldots, Table_L\}$ and users $u_a$ and $u_b$ are projected into an identical bucket in any of the $L$ hash tables, then according to the neighbour propagation strategy (for users), we can infer that $u_b$ is a possible neighbour of $u_{target}$ and put $u_b$ into $Neighbour\_set(u_{target})$. The pseudocode of Strategy 1 is presented in Algorithm 3.

*Strategy 2* (condition relaxation for neighbour search (for users)). According to the inherent characteristic of the LSH technique, the number of hash functions (i.e., $r$) plays an important role in the neighbour search process. Generally, a larger $r$ value often means stricter filtering condition for neighbour search and thereby leads to higher probability of "False negative" search results. Considering this, we relax the search condition for neighbours of the target user to reduce the "False negative" probability. Next, we introduce the concrete condition relaxation process.

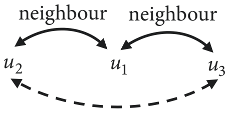
According to the neighbour search condition in (2), $u_i$ is regarded as a neighbour of $u_{target}$ iff $H(u_i) = H(u_{target})$ holds in any hash table, where $H(u_i) = (h_1(u_i), \ldots, h_r(u_i))$ and $H(u_{target}) = (h_1(u_{target}), \ldots, h_r(u_{target}))$. Namely, all the $r$ bit values in $H(u_i)$ are required to be equal to the $r$ bit values in $H(u_{target})$, respectively. Hence, to relax the search condition for neighbours of $u_{target}$ and guarantee high similarity between $u_{target}$ and his/her neighbours $u_i$, one bit difference between the indices of $u_{target}$ and $u_i$ is permitted.

For example, if $H_x(u_{target}) = (1, 1, 1)$ holds in hash table $Table_x$, then the neighbour $u_i$'s index in $Table_x$ is permitted to be (0, 1, 1) or (1, 0, 1) or (1, 1, 0). In other words, any user whose index is equal to (0, 1, 1) or (1, 0, 1) or (1, 1, 0) is a possible neighbour of $u_{target}$. This is the main idea of our proposed search condition relaxation strategy (for users). The pseudocode of Strategy 2 is presented in Algorithm 4, where $H_x(u_{target})_k$ denotes the relaxed search condition (i.e., $k$-th bit is different from that of $H_x(u_{target})$) for $u_{target}$ in $Table_x$; for example, if $H_x(u_{target}) = (1, 1, 1)$, then $H_x(u_{target})_2 = (1, 0, 1)$ holds.

Through Strategies 1 and 2, we can obtain an enlarged set of neighbours of the target user, i.e., $Neighbour\_set(u_{target})$. Next, we make service quality prediction based on the

$$\begin{array}{c} & ws_1 & ws_2 & ws_3 & ws_4 \\ u_1 & 5* & 5* & 1* & 1* \\ u_2 & 5* & 5* & & \\ u_3 & & & 1* & 1* \end{array}$$

(a) User-service rating matrix



(b) Neighbouring user propagation

Figure 2: Neighbouring user propagation example.

```
Inputs: $U = \{u_1, \ldots, u_m\}$
        $WS = \{ws_1, \ldots, ws_n\}$
        $L$: number of LSH tables
        $r$: number of LSH functions
Output: $Table_1, \ldots, Table_L$
For $x = 1, \ldots, L$ do // build hash tables $\{Table_1, \ldots, Table_L\}$ offline
  For $k = 1, \ldots, r$ do
    For $i = 1, \ldots, m$ do
      Build user sub-index $h_k(u_i)$ based on random LSH function $h_k(.)$
  For $i = 1, \ldots, m$ do
    Build user index $H_x(u_i) = (h_1(u_i), \ldots, h_r(u_i))$
  Return hash table $Table_x$ constituted by all the "$u_i \longrightarrow H_x(u_i)$" mappings
```

Algorithm 1: User-hash-table-building ($U$, $WS$, $L$, $r$).

```
Inputs: $u_{target}$ // a target user
        $TB = \{Table_1, \ldots, Table_L\}$
Output: $Neighbour\_set(u_{target})$
For $x = 1, \ldots, L$ do
  Find the bucket $bt$ corresponding to $H_x(u_{target})$ in $Table_x$
  If $u_i \in bt$ and $u_i \neq u_{target}$
  Then put $u_i$ into $Neighbour\_set(u_{target})$
Return $Neighbour\_set(u_{target})$
```

Algorithm 2: Neighbouring-user-search ($u_{target}$, $TB$).

elements in $Neighbour\_set(u_{target})$. Concretely, for web service $ws_j$ never invoked by $u_{target}$ before, its predicted quality over dimension $q$ by $u_{target}$, denoted by $q_{target,j}$, can be calculated by

$$q_{target,j} = \frac{1}{\left|Neighbour\_set\left(u_{target}\right)\right|} * \sum_{u_i \in Neighbour\_set(u_{target})} q_{i,j} \tag{3}$$

*4.2. Service Quality Prediction Based on LSH and Item-Based CF.* Similar to Section 4.1, in this subsection, we first utilize item-based CF and LSH techniques to look for the similar services (named "neighbouring services") of target service $ws_{target}$ (denoted by set $Neighbour\_set(ws_{target})$) in a privacy-aware and scalable manner, and the techniques then make service quality predictions based on the elements in $Neighbour\_set(ws_{target})$.

First, for any web service $ws_j \in WS$, its historical quality data over dimension $q$ ever-invoked by users can be specified by an $m$-dimensional vector $\overrightarrow{ws_j} = (q_{1,j}, \ldots, q_{m,j})$, where $q_{i,j}$ ($1 \leq i \leq m$) denotes the quality value of $q$ of service $ws_j$ invoked by user $u_i$ and $m$ is the number of users. Next, we utilize the LSH technique to transform $\overrightarrow{ws_j}$ with private information into a corresponding service index $h(ws_j)$ with little privacy, based on the random LSH function $h(.)$ in (1). Here, $\overrightarrow{v}$ is an $m$-dimensional real vector $(v_1, \ldots, v_m)$, where $v_i$ ($1 \leq i \leq m$) is a random value in the range $[-1, 1]$.

This way, we can transform $\overrightarrow{ws_j}$ with much privacy into a Boolean value $h(ws_j)$ with little privacy.

Likewise, we amplify LSH through integrating $r$ hash functions $\{h_1(.), \ldots, h_r(.)\}$ and $L$ hash tables $\{Table_1, \ldots, Table_L\}$. Then, in each hash table, we build an index for service $ws_j$, denoted by $H(ws_j) = (h_1(ws_j), \ldots, h_r(ws_j))$. Furthermore, two services $ws_1$ and $ws_2$ are regarded as neighbouring services if the condition in (4) holds where $H_x(ws_1)$ and $H_x(ws_2)$ denote the indices of services $ws_1$ and $ws_2$ in the $x$-th hash table (i.e., $Table_x$), respectively.

$$\exists x \in \{1, \ldots, L\}, \quad \text{satisfy } H_x\left(ws_1\right) = H_x\left(ws_2\right) \tag{4}$$

Then, through (4), we can find out the neighbouring services of $ws_{target}$ and put them into $Neighbour\_set(ws_{target})$. Note that if multiple target services are present, then it is necessary to repeat the above process for each target service to discover all the qualified neighbours. The pseudocode is presented in Algorithms 5 and 6, where Algorithm 5 is used to build the $L$ hash tables for services offline and Algorithm 6 is used to search for the neighbouring services of the target service (repeat Algorithm 6 if multiple target services are present).

```
Inputs: $U = \{u_1, \ldots, u_m\}$
        $u_{target}$ // a target user
        $Neighbour\_set(u_i)$ // before neighbour propagation (for users)
Output: $Neighbour\_set(u_{target})$ // after neighbour propagation (for users)
For each $u_a \in Neighbour\_set(u_{target})$ do
    For each $u_b \in Neighbour\_set(u_a)$ do
        If $u_b \notin Neighbour\_set(u_{target})$
        Then put $u_b$ into $Neighbour\_set(u_{target})$
Return $Neighbour\_set(u_{target})$
```

Algorithm 3: Neighbouring user search based on Strategy 1.

```
Inputs: $U = \{u_1, \ldots, u_m\}$
        $u_{target}$ // a target user
        $TB = \{Table_1, \ldots, Table_L\}$
Output: $Neighbour\_set(u_{target})$ // after condition relaxation for neighbour search (for users)
For $x = 1, \ldots, L$ do
    For $k = 1, \ldots, r$ do
        $H_x(u_{target}) = H_x(u_{target})_k$
        Neighbouring-user-search ($u_{target}$, $TB$) // Algorithm 2
Return $Neighbour\_set(u_{target})$
```

Algorithm 4: Neighbouring user search based on Strategy 2.

However, similar to Section 4.1, “False negative” search results are also inevitable; in other words, certain real neighbours of a target user are probably deemed as non-neighbors. Considering the drawback, Strategies 3 and 4 (actually the variants of Strategies 1 and 2 in Section 4.1) are proposed to reduce the “False negative” probability.

*Strategy 3* (neighbour propagation (for services)). Let’s consider the example in Figure 3 where four users $\{u_1, u_2, u_3, u_4\}$ and three web services $\{ws_1, ws_2, ws_3\}$ are present. The user-service ratings ($1* \sim 5*$) are shown in Figure 3(a), according to which we can determine that $ws_1$ and $ws_2$ are neighbouring services and $ws_1$ and $ws_3$ are neighbouring services. In this situation, we can infer that $ws_2$ and $ws_3$ are possible neighbouring services (marked with dotted line in Figure 3(b)). Thus through the propagation rule illustrated in Figure 3, we can obtain more neighbouring services of a target service so that the “False negative” probability is reduced.

Next, we introduce how to integrate the neighbour propagation strategy (for services) into the abovementioned LSH-based neighbouring service search process. Concretely, if services $ws_a$ and $ws_{target}$ are projected into an identical bucket in any of the $L$ hash tables $\{Table_1, \ldots, Table_L\}$ and services $ws_a$ and $ws_b$ are projected into an identical bucket in any of the $L$ hash tables, then according to the neighbour propagation strategy (for services), we can infer that $ws_b$ is probably a neighbouring service of $ws_{target}$ and hence put $ws_b$ into $Neighbour\_set(ws_{target})$. The pseudocode of Strategy 3 is presented in Algorithm 7.

*Strategy 4* (condition relaxation for neighbour search (for services)). Similar to Strategy 2, in Strategy 4, we relax the search condition for neighbouring services of the target service to reduce the “False negative” probability of search results. Concretely, according to the neighbouring service search condition in (4), service $ws_j$ is regarded as a neighbouring service of $ws_{target}$ iff all the $r$ bit values in $H(ws_j)$ are equal to the $r$ bit values in $H(ws_{target})$, respectively. Therefore, to relax the search condition for neighbouring services of $ws_{target}$ and meanwhile guarantee the high similarity between $ws_{target}$ and its neighbouring services $ws_j$, one bit difference between the indices of $ws_{target}$ and $ws_j$ is permitted.

For example, if condition $H_x(ws_{target}) = (1, 1, 1)$ holds in hash table $Table_x$, then any service whose index is equal to (0, 1, 1) or (1, 0, 1) or (1, 1, 0) is a possible neighbouring service of $ws_{target}$. This is the main idea of our proposed search condition relaxation strategy (for services). The pseudocode of Strategy 4 is presented in Algorithm 8, where $H_x(ws_{target})_k$ denotes the relaxed search condition (i.e., the $k$-th bit is different from that of $H_x(ws_{target})$) for $ws_{target}$ in $Table_x$; e.g., $H_x(ws_{target})_2 = (1, 0, 1)$ holds if $H_x(ws_{target}) = (1, 1, 1)$.

Through Strategies 3 and 4, we can obtain an enlarged set of neighbouring services of the target services, i.e., $Neighbour\_set(ws_{target})$. Next, for each service $ws_j$ never invoked by $u_{target}$, its predicted quality over dimension $q$ rated by $u_{target}$, denoted by $q_{target,j}$, is calculated by equation (5) where $ws_j \in Neighbour\_set(ws_{target})$. Here, $q_{target,target}$ denotes the real service quality of $ws_{target}$ observed by $u_{target}$. Furthermore, if service $ws_j$ appears multiple times in

```
Inputs: $U = \{u_1, \dots, u_m\}$
        $WS = \{ws_1, \dots, ws_n\}$
        $L$: number of LSH tables
        $r$: number of LSH functions
Output: $Table_1, \dots, Table_L$
For $x = 1, \dots, L$ do // build hash tables $\{Table_1, \dots, Table_L\}$ offline
  For $k = 1, \dots, r$ do
    For $j = 1, \dots, n$ do
      Build service sub-index $h_k(ws_j)$ based on random LSH function $h_k(.)$
  For $j = 1, \dots, n$ do
    Build service index $H_x(ws_j) = (h_1(ws_j), \dots, h_r(ws_j))$
  Return hash table $Table_x$ constituted by all "$ws_j \longrightarrow H_x(ws_j)$" mappings
```

Algorithm 5: Service-hash-table-building ($U$, $WS$, $L$, $r$).

```
Inputs: $ws_{target}$ // a target service
        $TB = \{Table_1, \dots, Table_L\}$
Output: $Neighbour\_set(ws_{target})$
For $x = 1, \dots, L$ do
  Find the bucket $bt$ corresponding to $H_x(ws_{target})$ in $Table_x$
  If $ws_j \in bt$ and $ws_j \neq ws_{target}$
  Then put $ws_j$ into $Neighbour\_set(ws_{target})$
Return $Neighbour\_set(ws_{target})$
```

Algorithm 6: Neighbouring-service-search ($ws_{target}$, $TB$).

$Neighbour\_set(ws_{target})$, then the average predicted quality is adopted.

$$q_{target,j} = q_{target,target} \tag{5}$$

*4.3. Aggregation of Predicted Service Quality and Service Recommendation.* We aggregate the two pieces of quality data predicted by (3) and (5) into a comprehensive quality in (6). Here, $q_{user}$ and $q_{item}$ denote the $q_{target,j}$ values predicted in (3) and (5), respectively; $\alpha$ and $\beta$ ($0 \le \alpha, \beta \le 1$ and $\alpha + \beta = 1$) are the aggregation coefficients. At last, we choose the service $ws_j$ whose predicted value (i.e., $q_{target,j}$ in (6)) is the best and return it to $u_{target}$.

$$q_{target,j} = \alpha * q_{user} + \beta * q_{item} \tag{6}$$

## 5. Experiments

In this section, we deploy a group of experiments to validate the feasibility of our proposed $SerRec_{LSH+HCF}$ approach in terms of service recommendation efficiency, accuracy, and successful rate. Concretely, in Section 5.1, we introduce the experiment dataset and configurations that we adopted for experiments; in Section 5.2, experiment comparison results are presented; in Section 5.3, further discussions are given.

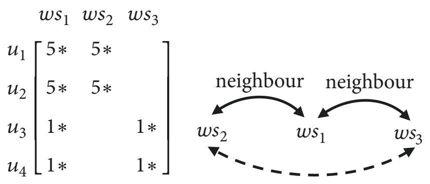
*5.1. Experiment Configurations.* Our experiments are based on a real distributed web service quality dataset *WS-DREAM* [24] that collects real-world service quality data from 339 users on 5825 web services (hosted in different countries). Each country that hosts a group of services is considered to be an individual platform for recommendation scenario simulation. Additionally, partial real values in the dataset are dropped for prediction needs. Moreover, only one quality dimension of services, i.e., *response time*, is considered in our experiments for simplicity. The target user is selected randomly from the user set in *WS-DREAM*, whose invoked services are regarded as the target services recruited in Section 4.2.

In order to validate the feasibility of our proposed $SerRec_{LSH+HCF}$ approach, we test the time cost and *MAE* of our proposal and compare them with three other state-of-the-art recommendation approaches including *UPCC* [25], *P-UIPCC* [17], and *PPICF* [18]. Concretely, *UPCC* is the benchmark service recommendation approach that is based on user-based CF; *P-UIPCC* utilizes the "divide-merge" operations over sensitive service quality data; while in *PPICF*, the real service quality data is transformed into the obfuscated data and then the obfuscated data are used to make service quality prediction and service recommendations.

The experiments were conducted on a Dell laptop with 2.80 GHz processors and 2.0 GB RAM. The machine runs Windows XP and JAVA 1.5. Each experiment was carried out ten times, and the average experimental results were adopted finally.

*5.2. Experiment Results and Analyses.* In the experiments, five profiles are tested and compared to validate the feasibility of our proposal. Here, $\omega$ denotes the density of the user-service quality matrix recruited to make service recommendations; $L$ and $r$ denote the number of hash tables and the number of hash functions, respectively; $\alpha = \beta = 0.5$ holds in (6).

*Profile 1* (computational time of four approaches w.r.t. $\omega$). Next, we measure the service computational time for recommendation process and scalability of four approaches with respect to matrix density $\omega$. Concrete experimental parameters are set as follows: $\omega$ is varied from 5% to 25%, $L = 10$ and $r = 14$ hold. Experimental results are shown in Figure 4.


$$\begin{array}{c|ccc} & ws_1 & ws_2 & ws_3 \\ \hline u_1 & 5* & 5* & \\ u_2 & 5* & 5* & \\ u_3 & 1* & & 1* \\ u_4 & 1* & & 1* \end{array}$$
neighbour
neighbour
$ws_2$
$ws_1$
$ws_3$

(a) User-service rating matrix

(b) Neighbouring service propagation

Figure 3: Neighbouring service propagation example.

```
Inputs: $WS = \{ws_1, \ldots, ws_n\}$
        $ws_{target}$ // a target service
        $Neighbour\_set(ws_j)$ // before neighbour propagation (for services)
Output: $Neighbour\_set(ws_{target})$ // after neighbour propagation (for services)
For each $ws_a \in Neighbour\_set(ws_{target})$ do
  For each $ws_b \in Neighbour\_set(ws_a)$ do
    If $ws_b \notin Neighbour\_set(ws_{target})$
    Then put $ws_b$ into $Neighbour\_set(ws_{target})$
Return $Neighbour\_set(ws_{target})$
```

Algorithm 7: Neighbouring service search based on Strategy 3.

As the experimental results in Figure 4 indicate, the computational time of the four different approaches all increase with the growth of service quality matrix density, i.e., $\omega$, because all the user-service quality data need to be considered in the four approaches and, therefore, more computational time is often required when the quality matrix becomes denser (i.e., when $\omega$ grows). However, our proposed $SerRec_{LSH+HCF}$ approach outperforms the other three approaches in terms of recommendation efficiency and scalability because most jobs in our approach (e.g., user indices building) can be done offline before a service recommendation request arrives, while the time complexity of the remaining jobs (e.g., online neighbour search) is rather small. So generally, our proposal can satisfy the quick response requirements of target users.

*Profile 2* (accuracy of returned results by four approaches w.r.t. $\omega$). We test and compare the recommendation accuracy (i.e., *MAE*, the smaller the better) of four approaches. The following are the experiment parameter settings: $\omega$ is varied from 5% to 25%, $L = 10$, and $r = 14$. Concrete comparison results are presented in Figure 5.

Figure 5 indicates that the accuracy of returned results by *P-UIPCC* and *PPICF* are not high; the reason is that, in order to secure the sensitive user privacy, the service quality data engaged in recommendation process have already been obfuscated in *UIPCC* and *PPICF*, while our $SerRec_{LSH+HCF}$ approach performs better than the other three approaches in terms of recommendation accuracy; this is because only the "most similar" neighbouring users and neighbouring services can be returned by LSH and recruited to make service recommendations. Therefore, the recommendation accuracy is improved considerably.

*Profile 3* (recommendation efficiency of $SerRec_{LSH+HCF}$ w.r.t. $L$ and $r$). In this profile, we test the recommendation efficiency of our $SerRec_{LSH+HCF}$ approach with respect to $L$ and $r$. The parameters are set as follows: $\omega = 25\%$, $L$ is varied from 6 to 14, and $r$ is varied from 8 to 14. Experimental results are shown in Figure 6.

As shown in Figure 6(a), the time cost of our proposal increases approximately with the growth of $L$, as all the $L$ hash tables need to be traversed in order to find the similar neighbours of the target user by (2) and find the similar neighbouring services of the target services by (4), respectively, while Figure 6(b) shows that the time cost decreases when $r$ grows. This is because a larger $r$ value often means stricter search condition for neighbouring users or neighbouring services; and therefore, few search results are obtained when $r$ is large; in this situation, less time is needed to evaluate and rank the few search results.

*Profile 4* (recommendation accuracy of $SerRec_{LSH+HCF}$ w.r.t. $L$ and $r$). We test the recommendation accuracy of our proposed $SerRec_{LSH+HCF}$ approach with respect to $L$ and $r$. The following are the experimental parameter settings: $\omega = 25\%$, $L$ is varied from 6 to 14, and $r$ is varied from 8 to 14. The experiment results are offered in Figure 7.

As Figure 7 shows, the recommendation accuracy of $SerRec_{LSH+HCF}$ increases (i.e., *MAE* drops) with the decrease of $L$ and the growth of $r$. This is because a smaller $L$ value or a larger $r$ value often means stricter search condition for neighbouring users and services; in this situation, only the "most similar" neighbouring users or neighbouring services are returned to make service recommendations. Therefore, the recommendation accuracy is improved accordingly.

```
Inputs: $WS = \{ws_1, \ldots, ws_n\}$
        $ws_{target}$ //a target service
        $TB = \{Table_1, \ldots, Table_L\}$
Output: $Neighbour\_set(ws_{target})$//after condition relaxation for neighbour search (for services)
For $x = 1, \ldots, L$ do
    For $k = 1, \ldots, r$ do
        $H_x(ws_{target}) = H_x(ws_{target})_k$
        Neighbouring-service-search ($ws_{target}$, $TB$) //Algorithm 6
Return $Neighbour\_set(ws_{target})$
```

Algorithm 8: Neighbouring service search based on Strategy 4.

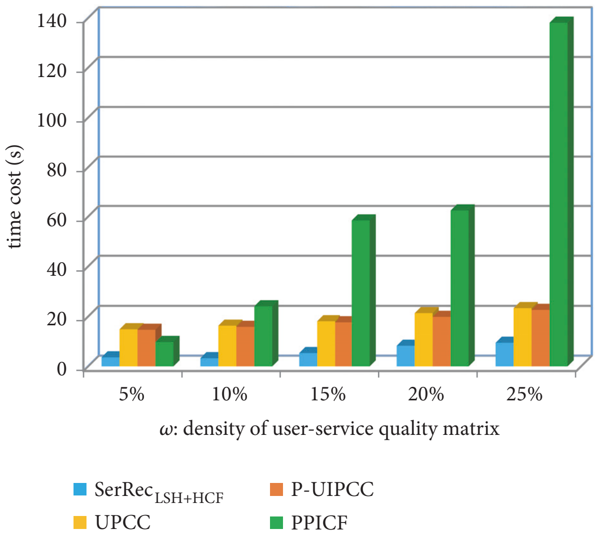


Figure 4: Computational time of different approaches.

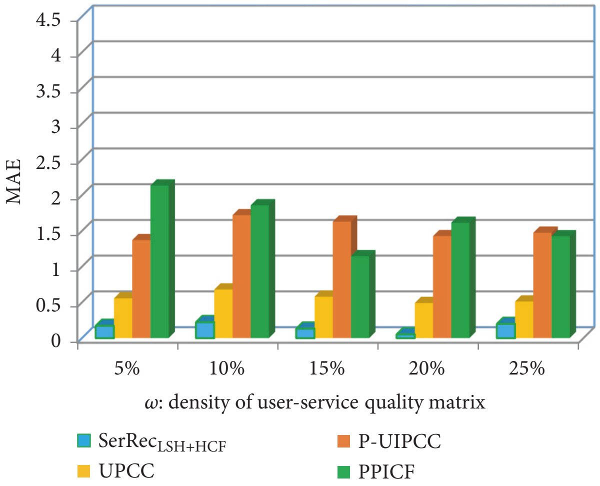


Figure 5: Recommendation accuracy of four approaches.

*Profile 5* (recommendation successful rate comparison). LSH is essentially a probability-based similar neighbour search technique; therefore, our proposed LSH-based service recommendation approach $SerRec_{LSH+HCF}$ cannot always guarantee to return a satisfying recommended result to the target user. In other words, recommendation failure is inevitable. However, as discussed in Section 2, the hybrid CF method can reduce the failure rate to some extent. Therefore, in this profile, we test the recommendation successful rate of our proposal and compare it with the following two benchmark approaches: $SerRec_{LSH+UCF}$ (i.e., the $DistSR_{LSH}$ approach in [26]) and $SerRec_{LSH+ICF}$.

(1) $SerRec_{LSH+UCF}$: integrate LSH with user-based CF

(2) $SerRec_{LSH+ICF}$: integrate LSH with item-based CF

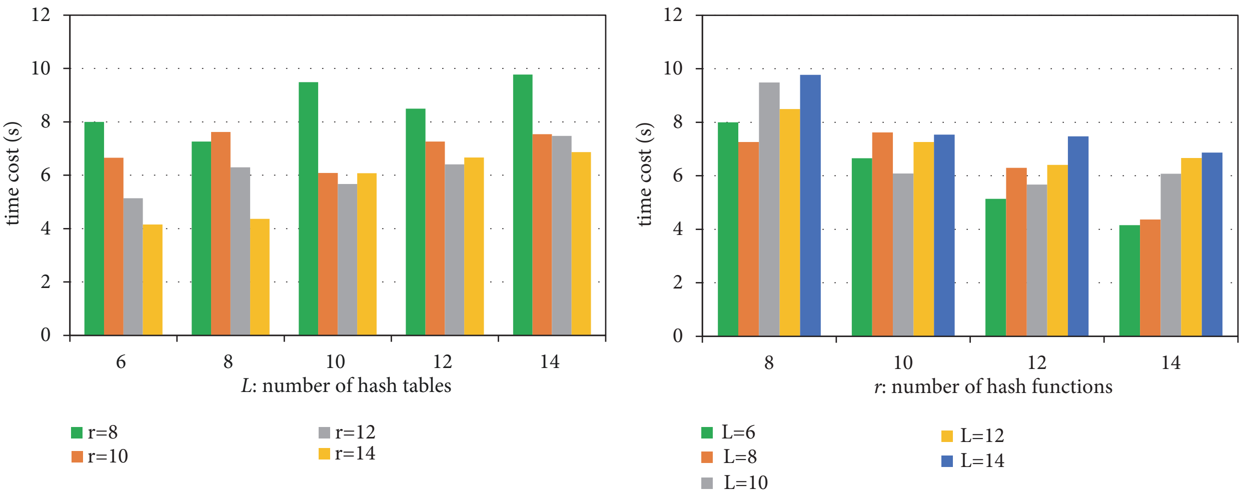
Here, we define the successful rate of a recommendation approach as the ratio between the successful recommendation times and the total recommendation times ($\in[0, 100\%]$). Parameters settings are $\omega = 25\%$, $L = 1$, and $r$ is varied from 8 to 14. Concrete experimental results are presented in Figure 8.

As Figure 8 shows, the successful rates of three recommendation approaches all decrease with the growth of $r$. This is because a larger $r$ value often means stricter filtering condition for the search of neighbouring users and neighbouring services; and, therefore, the successful rate of recommendations is reduced accordingly. Namely, there is a trade-off between successful rate and $r$; specifically, when $r$ is large enough (e.g., when $r = 14, 15\ldots$), the successful rate approaches 0. However, as Figure 6 shows, our approach still outperforms the other two approaches in terms of successful rate as our approach recruits hybrid CF for recommendation, integrating the advantages of both user-based CF and item-based CF.

*5.3. Further Discussions.* Our experiments only adopt one service quality dimension, i.e., *response time*, without considering the probably existed multiple dimensions [27–37] and their respective weight significance values [38–44]. In the future research, we will integrate the dimension and weight information into $SerRec_{LSH+HCF}$ to make the approach more comprehensive. Besides, only one type of service quality data is considered in the experiments. So in the future, we will further extend our proposal by considering the possible data diversity in the big data environment [45–50].

## 6. Conclusions

Collaborative service recommendation has become an effective technique to quickly extract insightful information from



(a)

(b)

FIGURE 6: Efficiency of $SerRec_{LSH+HCF}$ w.r.t. $L$ and $r$.

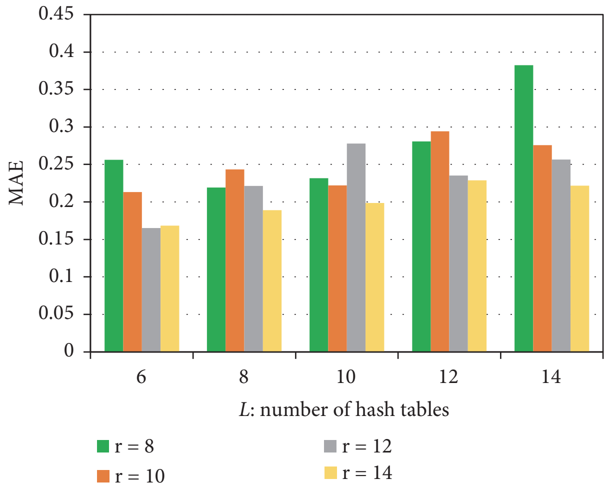


FIGURE 7: Accuracy of $SerRec_{LSH+HCF}$ w.r.t. $L$ and $r$.

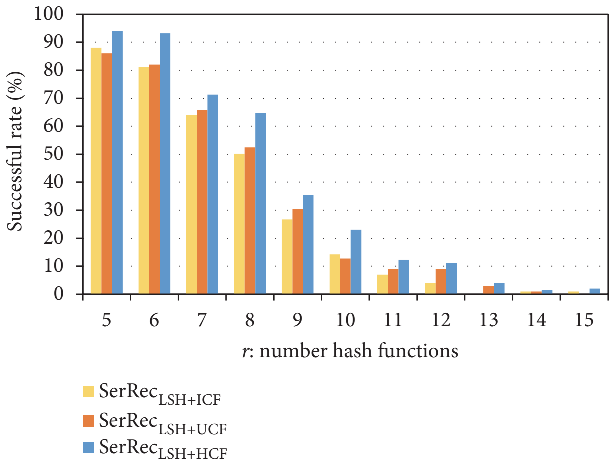


FIGURE 8: Successful rate of three approaches (w.r.t. $r$).

big educational data. However, traditional service recommendation approaches often assume that the service usage data used to make recommendations are centralized, without considering the multisource property of service usage data as well as the privacy leakage risks during the multisource educational data integration. Besides, existing service recommendation approaches often suffer from low robustness due to the possible data sparsity. In view of these drawbacks, we combine the LSH technique and hybrid Collaborative Filtering (HCF) for distributed service recommendations in the big data environment. Furthermore, to minimize the "False negative" recommended results incurred by the inherent shortcoming of LSH, two solutions are introduced in this paper, to reduce the probability that similar users and similar services are overlooked by mistake and thereby enhance the success rate. A wide range of experiments deployed on real-world dataset shows the performances of $SerRec_{LSH+HCF}$ in terms of efficiency, accuracy, and successful rate while securing the sensitive user information.

However, only one quality dimension of web services is considered in the recommendation model, which is often not enough for the practical recommendation requirements. In the future, we will further refine our work by considering multiple quality dimensions as well as their linear correlations [51–53] and nonlinear correlations [54–58]. Besides, data type diversity is another challenge in the big data environment. Therefore, in the future research, we will continue to extend our proposal by integrating the multisource data with diverse data types, e.g., discrete data [59–63], binary data [64], and fuzzy data [65–67].

## Data Availability

The [web service quality] data used to support the findings of this study have been deposited in the [WS-DREAM] repository (http://inpluslab.com/wsdream/)

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This paper is partially supported by the Natural Science Foundation of China (No. 61872219).

## References

[1] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. Jamal Deen, "A cloud-edge computing framework for cyber-physical-social services," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 80–85, 2017.

[2] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, "A double deep q-learning model for energy-efficient edge scheduling," *IEEE Transactions on Services Computing*, 2018.

[3] L. Qi, P. Dai, J. Yu, Z. Zhou, and Y. Xu, ""Time-location-frequency"-aware internet of things service selection based on historical records," *International Journal of Distributed Sensor Networks*, vol. 13, no. 1, pp. 1–9, 2017.

[4] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and F. Bu, "An adaptive droupout deep computation model for industrial IoT big data learning with crowdsourcing to cloud computing," *IEEE Transactions on Industrial Informatics*, 2018.

[5] X. Wang, W. Wang, L. T. Yang, S. Liao, D. Yin, and M. J. Deen, "A distributed HOSVD method with its incremental computation for big data in cyber-physical-social systems," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 481–492, 2018.

[6] K. Dou, B. Guo, and L. Kuang, "A privacy-preserving multimedia recommendation in the context of social network based on weighted noise injection," *Multimedia Tools and Applications*, pp. 1–20, 2017.

[7] L. Qi, W. Dou, and J. Chen, "Weighted principal component analysis-based service selection method for multimedia services in cloud," *Computing*, vol. 98, no. 1, pp. 195–214, 2016.

[8] L. T. Yang, X. Wang, X. Chen et al., "A multi-order distributed HOSVD with its incremental computing for big services in cyber-physical-social systems," *IEEE Transactions on Big Data*, 2018.

[9] X. Wang, L. T. Yang, H. Liu, and M. J. Deen, "A big data-as-a-service framework: state-of-the-art and perspectives," *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 325–340, 2018.

[10] Q. Zhang, L. T. Yang, A. Castiglione, Z. Chen, and P. Li, "Secure weighted possibilistic c-means algorithm on cloud for clustering big data," *Information Sciences*, vol. 479, pp. 515–525, 2018.

[11] S. Zhang, G. Wang, M. Z. Bhuiyan, and Q. Liu, "A dual privacy preserving scheme in continuous location-based services," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4191–4200, 2018.

[12] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," *Future Generation Computer Systems*, vol. 88, pp. 636–643, 2018.

[13] L. Qi, X. Xu, W. Dou et al., "Time-aware IoE service recommendation on sparse data," *Mobile Information Systems*, vol. 2016, Article ID 4397061, 12 pages, 2016.

[14] L. Qi, Z. Zhou, J. Yu, and Q. Liu, "Data-sparsity tolerantweb service recommendation approach based on improved collaborative filtering," *IEICE Transaction on Information and Systems*, vol. E100D, no. 9, pp. 2092–2099, 2017.

[15] S. Zhang, X. Li, Z. Tan, T. Peng, and G. Wang, "A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services," *Future Generation Computer Systems*, vol. 94, pp. 40–50, 2019.

[16] S. Zhang, X. Mao, K. R. Choo, T. Peng, and G. Wang, "A trajectory privacy-preserving scheme based on a dual-K mechanism for continuous location-based services," *Information Sciences*, 2019.

[17] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "A privacy-preserving qos prediction framework for web service recommendation," in *Proceedings of the IEEE International Conference on Web Services (ICWS '15)*, pp. 241–248, New York, NY, USA, July 2015.

[18] 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.

[19] Y. Xu, L. Qi, W. Dou, and J. Yu, "Privacy-preserving and scalable service recommendation based on simhash in a distributed cloud environment," *Complexity*, vol. 2017, Article ID 3437854, 9 pages, 2017.

[20] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3075849, 8 pages, 2018.

[21] C. Yan, X. Cui, L. Qi, X. Xu, and X. Zhang, "Privacy-aware data publishing and integration for collaborative service recommendation," *IEEE Access*, vol. 6, pp. 43021–43028, 2018.

[22] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," *The VLDB Journal*, vol. 99, no. 6, pp. 518–529, 1999.

[23] *Data Mining and Query Log Analysis for Scalable Temporal and Continuous Query Answering*, 2015, http://www.optique-project.eu/.

[24] 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.

[25] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, 1998.

[26] L. Qi, H. Xiang, W. Dou, C. Yang, Y. Qin, and X. Zhang, "Privacy-preserving distributed service recommendation based on locality-sensitive hashing," in *Proceedings of the 24th IEEE International Conference on Web Services, ICWS 2017*, pp. 49–56, USA, June 2017.

[27] X. Wang, L. T. Yang, L. Kuang, X. Liu, Q. Zhang, and M. J. Deen, "A tensor-based big-data-driven routing recommendation approach for heterogeneous networks," *IEEE Network Magazine*, vol. 33, no. 1, pp. 64–69, 2019.

[28] M. Wang and G.-L. Tian, "Robust group non-convex estimations for high-dimensional partially linear models," *Journal of Nonparametric Statistics*, vol. 28, no. 1, pp. 49–67, 2016.

[29] X. Wang and M. Wang, "Variable selection for high-dimensional generalized linear models with the weighted elastic-net procedure," *Journal of Applied Statistics*, vol. 43, no. 5, pp. 796–809, 2016.

[30] P. Wang and L. Zhao, "Some geometrical properties of convex level sets of minimal graph on 2-dimensional Riemannian manifolds," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 130, pp. 1–17, 2016.

[31] X. Wang and M. Wang, "Adaptive group bridge estimation for high-dimensional partially linear models," *Journal of Inequalities and Applications*, vol. 2017, article no. 158, pp. 1–18, 2017.

[32] X. Wang, S. Zhao, and M. Wang, "Restricted profile estimation for partially linear models with large-dimensional covariates," *Statistics & Probability Letters*, vol. 128, pp. 71–76, 2017.

[33] H. Tian and M. Han, "Bifurcation of periodic orbits by perturbing high-dimensional piecewise smooth integrable systems," *Journal of Differential Equations*, vol. 263, no. 11, pp. 7448–7474, 2017.

[34] P. Wang and X. Wang, "The geometric properties of harmonic function on 2-dimensional Riemannian manifolds," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 103, pp. 2–8, 2014.

[35] M. Wang and X. Wang, "Adaptive Lasso estimators for ultrahigh dimensional generalized linear models," *Statistics & Probability Letters*, vol. 89, no. 1, pp. 41–50, 2014.

[36] X. Wang, M. Wang, and X. Wang, "A note on the one-step estimator for ultrahigh dimensionality," *Journal of Computational and Applied Mathematics*, vol. 260, pp. 91–98, 2014.

[37] G.-L. Tian, M. Wang, and L. Song, "Variable selection in the high-dimensional continuous generalized linear model with current status data," *Journal of Applied Statistics*, vol. 41, no. 3, pp. 467–483, 2014.

[38] S. Yang, Z.-A. Yao, and C.-A. Zhao, "The weight distributions of two classes of p-ary cyclic codes with few weights," *Finite Fields and Their Applications*, vol. 44, pp. 76–91, 2017.

[39] Y.-F. Wang, C.-C. Yin, and X.-S. Zhang, "Uniform estimate for the tail probabilities of randomly weighted sums," *Acta Mathematicae Applicatae Sinica*, vol. 30, no. 4, pp. 1063–1072, 2014.

[40] S. Yang, Z.-A. Yao, and C.-A. Zhao, "A class of three-weight linear codes and their complete weight enumerators," *Cryptography and Communications*, vol. 9, no. 1, pp. 133–149, 2017.

[41] J. Cai, "An implicit sigma (3) type condition for heavy cycles in weighted graphs," *Ars Combinatoria*, vol. 115, pp. 211–218, 2014.

[42] S. Yang and Z.-A. Yao, "Complete weight enumerators of a family of three-weight linear codes," *Designs, Codes and Cryptography*, vol. 82, no. 3, pp. 663–674, 2017.

[43] S. Yang and Z.-A. Yao, "Complete weight enumerators of a class of linear codes," *Discrete Mathematics*, vol. 340, no. 4, pp. 729–739, 2017.

[44] S. Yang, X. Kong, and C. Tang, "A construction of linear codes and their complete weight enumerators," *Finite Fields and Their Applications*, vol. 48, pp. 196–226, 2017.

[45] H. Liu and F. Meng, "Some new generalized Volterra-Fredholm type discrete fractional sum inequalities and their applications," *Journal of Inequalities and Applications*, vol. 2016, no. 1, article no. 213, 2016.

[46] P. Li and G. Ren, "Some classes of equations of discrete type with harmonic singular operator and convolution," *Applied Mathematics and Computation*, vol. 284, pp. 185–194, 2016.

[47] P. Li, "Singular integral equations of convolution type with Hilbert kernel and a discrete jump problem," *Advances in Difference Equations*, vol. 2017, no. 1, article no. 360, 2017.

[48] Y. Bai and L. Liu, "New oscillation criteria for second-order delay differential equations with mixed nonlinearities," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 796256, 9 pages, 2010.

[49] Y. Wang and C. Yin, "Approximation for the ruin probabilities in a discrete time risk model with dependent risks," *Statistics & Probability Letters*, vol. 80, no. 17-18, pp. 1335–1342, 2010.

[50] Q. Feng, F. Meng, and Y. Zhang, "Generalized gronwall-bellman-type discrete inequalities and their applications," *Journal of Inequalities and Applications*, vol. 2011, article no. 47, 2011.

[51] G. Guo, W. Shao, L. Lin, and X. Zhu, "Parallel tempering for dynamic generalized linear models," *Communications in Statistics—Theory and Methods*, vol. 45, no. 21, pp. 6299–6310, 2016.

[52] L. L. Liu and Y. Li, "Recurrence relations for linear transformations preserving the strong q-log-convexity," *The Electronic Journal of Combinatorics*, vol. 23, no. 3, pp. 1–11, 2016.

[53] H. Li and S. Wang, "Partial condition number for the equality constrained linear least squares problem," *Calcolo. A Quarterly on Numerical Analysis and Theory of Computation*, vol. 54, no. 4, pp. 1121–1146, 2017.

[54] H. Liu and F. Meng, "Some new nonlinear integral inequalities with weakly singular kernel and their applications to FDEs," *Journal of Inequalities and Applications*, vol. 2015, no. 209, pp. 1–17, 2015.

[55] X. Zhang, L. Liu, Y. Wu, and L. Caccetta, "Entire large solutions for a class of Schrödinger systems with a nonlinear random operator," *Journal of Mathematical Analysis and Applications*, vol. 423, no. 2, pp. 1650–1659, 2015.

[56] Z. Zong, F. Hu, C. Yin, and H. Wu, “On Jensen’s inequality, Hölder’s inequality, and Minkowski’s inequality for dynamically consistent nonlinear evaluations,” *Journal of Inequalities and Applications*, vol. 2015, no. 1, pp. 1–18, 2015.

[57] X. Hao, L. Liu, and Y. Wu, “Positive solutions for nonlinear fractional semipositone differential equation with nonlocal boundary conditions,” *Journal of Nonlinear Science and Applications*, vol. 9, no. 6, pp. 3992–4002, 2016.

[58] X. Hao, L. Liu, and Y. Wu, “Iterative solution for nonlinear impulsive advection- reaction-diffusion equations,” *Journal of Nonlinear Science and Applications*, vol. 9, no. 6, pp. 4070–4077, 2016.

[59] P. Li, “Two classes of linear equations of discrete convolution type with harmonic singular operators,” *Complex Variables and Elliptic Equations*, vol. 61, no. 1, pp. 67–75, 2016.

[60] Z. Zheng, “Invariance of deficiency indices under perturbation for discrete Hamiltonian systems,” *Journal of Difference Equations and Applications*, vol. 19, no. 8, pp. 1243–1250, 2013.

[61] M. Han, X. Hou, L. Sheng, and C. Wang, “Theory of rotated equations and applications to a population model,” *Discrete and Continuous Dynamical Systems- Series A*, vol. 38, no. 4, pp. 2171–2185, 2018.

[62] J. Cai and H. Li, “A new sufficient condition for pancyclability of graphs,” *Discrete Applied Mathematics*, vol. 162, pp. 142–148, 2014.

[63] L. L. Liu and B.-X. Zhu, “Strong q-log-convexity of the Eulerian polynomials of Coxeter groups,” *Discrete Mathematics*, vol. 338, no. 12, pp. 2332–2340, 2015.

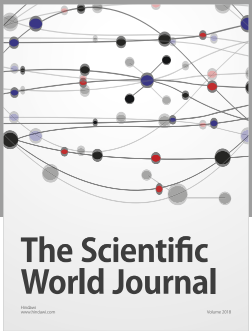
[64] B. Zhang, “Remarks on the maximum gap in binary cyclotomic polynomials,” *Bulletin Mathematique De La Societe Des Sciences Mathematiques De Roumanie*, vol. 59, no. 1, pp. 109–115, 2016.

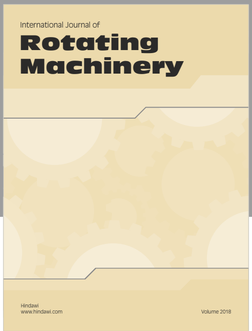
[65] L. Wang, “Intuitionistic fuzzy stability of a quadratic functional equation,” *Fixed Point Theory and Applications*, vol. 2010, Article ID 107182, 7 pages, 2010.

[66] X. Du and Z. Zhao, “On fixed point theorems of mixed monotone operators,” *Fixed Point Theory and Applications*, vol. 2011, Article ID 563136, 8 pages, 2011.

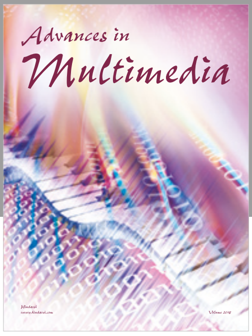
[67] B. Zhu, L. Liu, and Y. Wu, “Local and global existence of mild solutions for a class of nonlinear fractional reaction-diffusion equations with delay,” *Applied Mathematics Letters*, vol. 61, pp. 73–79, 2016.

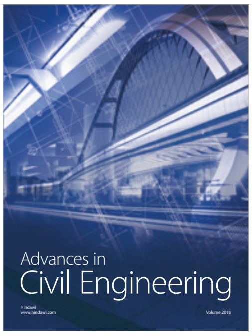
Journal of
Engineering

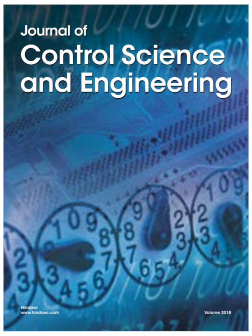

The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

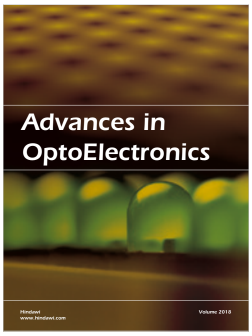

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

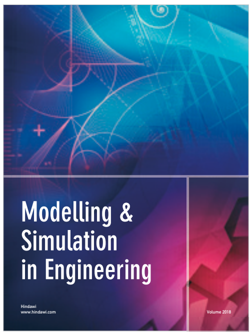
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

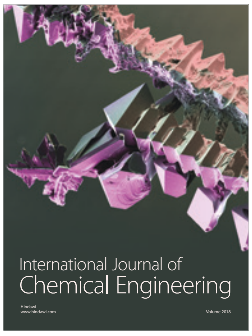

Advances in
OptoElectronics

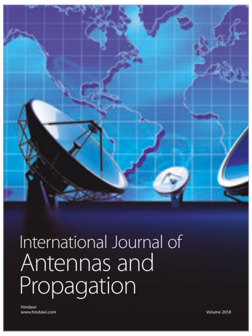
VLSI Design

International Journal of
Navigation and
Observation

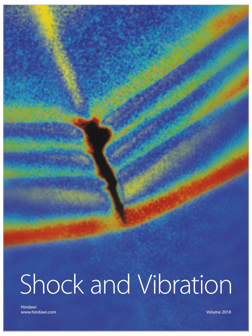

Modelling &
Simulation
in Engineering

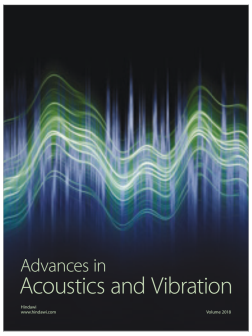
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration