A High-Reliability Multi-Faceted Reputation Evaluation Mechanism for Online Services

Miao Wang, Guiling Wang, Yujun Zhang, and Zhongcheng Li

Abstract—In today's society, there are plenty of services available, and customers are facing bigger challenge in choosing them than ever before. Therefore, it is important to build a reliable reputation mechanism for selecting a credible service. To address the challenges of reputation evaluation, including the diverse and dynamic natures of services, incompleteness of user feedback, and intricacy of malicious ratings, a High-reliability Multi-faceted Reputation evaluation mechanism for online services (HMRep) is proposed. First, HMRep starts with addressing the incomplete feedback and estimates missing ratings based on both the service quality and a user's rating behavior. Second, HMRep identifies and removes malicious collusive raters and irresponsible raters to improve the accuracy of reputation calculation. Further, the reputation calculation is based on the user credibility and incorporates historical information to reflect the change of the services. Finally, we provide a multi-faceted evaluation method to satisfy some specific needs of customers who are only concerned about a subset of a services features. Experimental results verify the design of HMRep, and reveal HMRep can effectively defend against malicious ratings, and accurately calculate the reputation values of services. HMRep can be applied in lots of sectors for different kinds of services, especially those complex ones.

Index Terms—Reputation evaluation, incomplete user feedback, malicious ratings, index weights

1 Introduction

In today's society, with the development of internet, the rise of online services such as e-commerce services, web services, cloud services, travel, and restaurant services has revolutionized the way we choose services and products. In the meanwhile, being surrounded by much more choices than ever on internet can also confuse and challenge us [1]. For example, when we book a hotel, there are so many choices on internet. Even though we can use some booking websites, such as hotels.com [2], we still face the problem of searching the satisfactory one from a large number of hotels. If we want to buy a Lenovo laptop, there are thousands of sellers in Taobao [3] in China. Inexperienced customers are overwhelmed by so many choices. This problem can be mitigated through the use of service reputation. Reputation evaluation mechanism can help customers choose suitable service providers, help good service providers have long-term development and banish bad service providers.

Recently, researchers have presented various reputation evaluation mechanisms in different scenarios. However, four key limitations exist in many previous studies:

- Many previous reputation systems evaluate a service only based on a single rating index and assign a unique overall reputation value for all of its attributes [4], [5], [6], [7]. However, many services are of diverse dimensions and complex patterns. Reputation evaluation for such kind of services should involve multiple rating indexes for multiple attributes, e.g., security, reliability, and performance, etc.
- Current studies neglect the problem of incomplete user feedback. Some users of a service may not have used every of its features, so they cannot evaluate those features. Some other users may miss or not be willing to rate certain indexes of the service. The incompleteness of user feedback affects the accuracy of reputation evaluation and may lead to deviation from objective evaluation [8], [9], [10].
- Existing reputation calculation models fail to detect random and malicious ratings, especially for a service of multiple features and multiple evaluation indexes, which may lead to biased or even wrong reputation calculation. This partially is because most of previous reputation calculation models process users' evaluations for each single index and calculate each index's reputation value independently [11], [12], [13], [14]. Our work is different from previous research and we examine users' ratings on all the indexes to identify irresponsible users who give ratings randomly and malicious users who give biased ratings intentionally.
- Current studies lack adaptability in determining the weight of each reputation index when calculating the overall reputation value. Most of previous studies rely on expert or user opinion to weight the indexes, which is subjective and does not reflect reputation adaptability [11], [12], [13], [14]. We explore the correlation of customer ratings on the multiple indexes of a service and deduce the weights accordingly.

- *M. Wang is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, P.R.China. E-mail: wangm@ict.ac.cn.*
- *G. Wang is with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102. E-mail: gwang@njit.edu.*
- *Y. Zhang and Z. Li are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, P.R.China. E-mail: {zhmj, zcli}@ict.ac.cn.*

Manuscript received 21 Dec. 2015; revised 21 Nov. 2016; accepted 4 Dec. 2016. Date of publication 13 Dec. 2016; date of current version 11 Dec. 2019. Digital Object Identifier no. 10.1109/TSC.2016.2638812

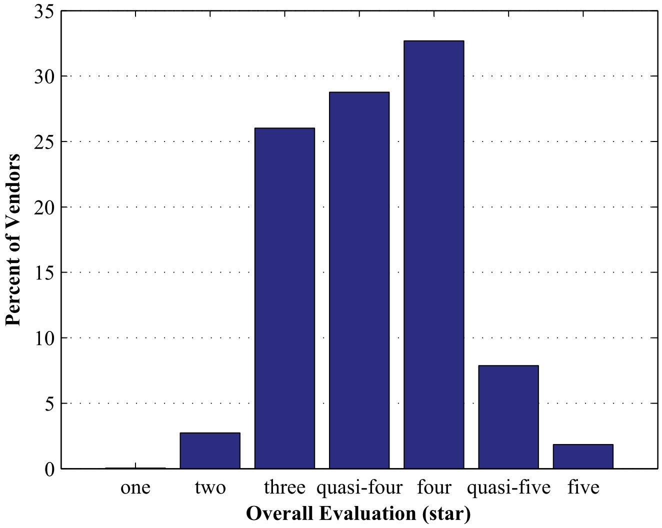


Fig. 1. Distribution of “overall evaluation”.

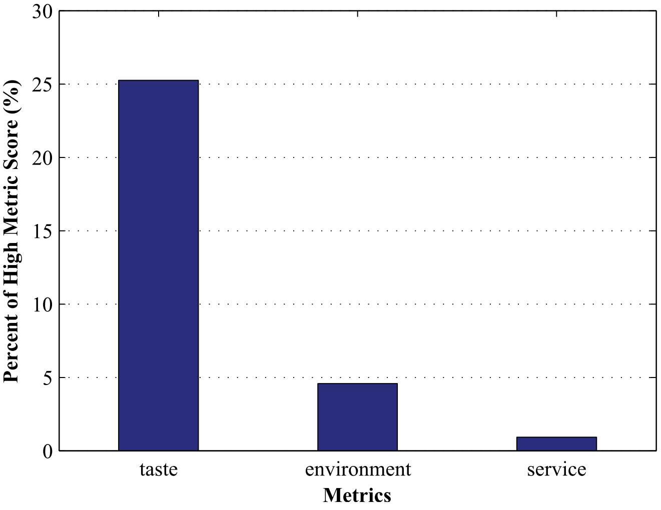


Fig. 2. Percent of high metric score out of low “overall evaluation” vendors.

In order to deal with the challenges of service evaluation, such as the diverse and dynamic nature of the services, incompleteness of user feedback and intricacy of malicious ratings, a High-reliability Multi-faceted Reputation evaluation mechanism for online services (HMRep) is proposed. HMRep employs multiple rating indexes to investigate the services from various perspectives. To address the incomplete user feedbacks, HMRep estimates and fills the incomplete data according to not only service quality but also user characteristics. HMRep also considers users' ratings on multiple indexes simultaneously and establishes a high-reliability reputation calculation model accordingly to resist malicious ratings on reputation calculation. Additionally, HMRep infers the index weights adaptively and evaluates the service with multi-faceted attributes.

The main contributions of this work are listed as follows:

- In order to better understand users' rating behavior, we have collected real trace from a China's popular online rating platform dianping.com (Dianping), which had more than 200 million monthly active users, over 100 million user-generated reviews, and more than 20 million local businesses as of Q3 of 2015 [15]. The trace is analyzed and we discover the inconsistency between rating metrics. The findings provide insight in designing reputation evaluation mechanisms.
- For the first time we bring up the missing feedback problem in the reputation calculation and develop an effective missing rating estimation method. Based on both service quality and a user's rating behavior, we estimate the missing feedback of the user if there is any. Compared with existing incomplete data filling methods, our missing rating estimation algorithm is computationally simple, adaptive to the dynamic nature of services, and does not rely on the potentially misleading expert opinions.
- We have designed and built HMRep, a high-reliability multi-faceted reputation evaluation framework for online services. HMRep can process all index ratings in an interdependent way and thus can effectively identify malicious users and irresponsible users, and improve the accuracy of the reputation calculation. It can also deduce the weight of each index adaptively, and then combine relevant index reputation values to calculate the attribute reputation values of a service from various angles.

Experimental results validate the design of HMRep, and show that our mechanism can accurately estimate the missing feedback, effectively reject malicious ratings, and accurately calculate the reputation values of services.

The remainder of this paper is organized as follows: Section 2 analyzes our crawled trace data. Section 3 details the design of HMRep. In Section 4, performance evaluations are conducted. Section 5 gives an overview of the related work. Finally, Section 6 concludes the paper and presents future work.

2 Preliminary Study

In this section, we introduce our preliminary study on the relationship between various specific ratings and the overall rating.

We choose China's leading online rating platform Dianping [15] as the study object. The platform provides information on dining, shopping, entertainment and many other services. Customers can rate the services and publish their comments. The ratings are open to public and can be collected through web crawler. Our study focuses on Dianping's dining category, where customers can rate on “taste”, “environment” and “service” in addition to “overall evaluation” of vendors. The rating of “overall evaluation” can be one of the seven values: one-star, two-star, three-star, quasi-four-star, four-star, quasi-five-star and five-star. The ratings of “taste”, “environment” and “service” subcategories range from 0 to 40. We collected all the user evaluations in the dinning category by April 2013 in the region of north and northeast, east, central west, and south of China. In total, we get the evaluations of 65,480 vendors.

First, we present the distribution of “overall evaluation”, and show the inconsistency between the “overall evaluation” and the individual rating metrics for certain vendors. The distribution of “overall evaluation” from the entire dataset is shown in Fig. 1. The figure shows that 28.81 percent of the vendors receive low reputation (one-star, two-star and three-star), 28.77 percent of the vendors have median

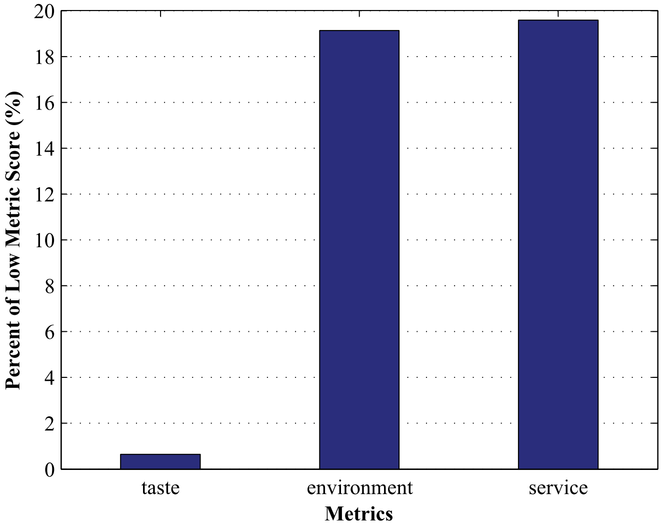


Fig. 3. Percent of low metric score out of high "overall evaluation" vendors.

reputation (quasi-four-star), and the rest achieves high reputation (four-star, quasi-five-star and five-star). Fig. 2 shows the percentage of vendors whose individual ratings are higher than 20, but "overall evaluation" remains low reputation. There are 25.25 percent of low "overall evaluation" vendors receiving high scores in "taste", and 4.58 percent of low "overall evaluation" vendors get endorsed in "environment". The result implies that vendors with low "overall evaluation" may have high individual rating scores. Fig. 3 plots the percentage of vendors whose individual ratings are lower than 20 but "overall evaluation" is high. As shown in the figure, 19.13 percent of high "overall evaluation" vendors fail in "environment", and 19.59 percent of high "overall evaluation" vendors are faulty in "service". The finding shows that the vendors with high "overall evaluation" may have low individual rating scores.

Next we present the inconsistency between the evaluation metrics. Score difference is the difference between the maximal score and the minimal score of the three metrics of a vendor. For example, if the "service" score of a vendor is 1, the "taste" score is 5, and the "environment" score is 10, then the score difference of this vendor is 9. Table 1 shows the score difference distribution for all the vendors. More than half of the vendors get the score difference less than 4. However, some of the vendors get high score difference. The result implies that vendors with the identical "overall evaluation" may behave very differently in distinct metrics.

In summary, even low "overall evaluation" vendors may still have relatively high scores in individual metrics and high "overall evaluation" vendors may have relatively low metric scores as well. Same "overall evaluation" vendors may have large difference in individual metrics. As we see, a vendor's overall reputation cannot dominate its every quality aspect, which suggests the subtle and complicated connection between the overall impression and the individual metrics. Thus the reputation evaluation must be based on multiple indexes, and be evaluated across the overall quality and multiple attributes.

TABLE 1
Score Difference Distribution of "Taste", "Environment" and "Service" for All Vendors

Score difference	[0, 1]	[2, 4]	[5, 8]	[9, 12]	[13, 21]
Percent of vendors	11.56%	56.73%	26.33%	4.55%	0.83%

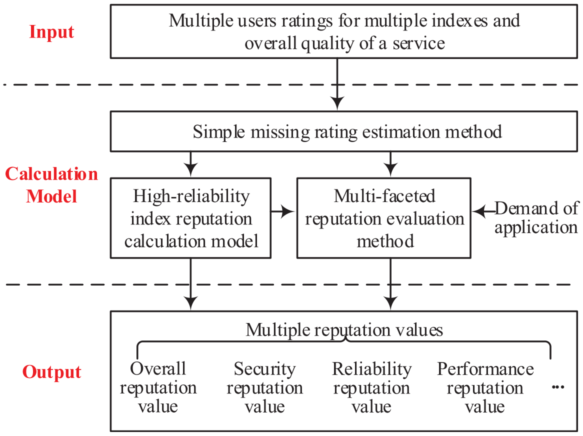

Input
Multiple users ratings for multiple indexes and overall quality of a service
Calculation Model
Simple missing rating estimation method
High-reliability index reputation calculation model
Multi-faceted reputation evaluation method
Demand of application
Output
Multiple reputation values
Overall reputation value
Security reputation value
Reliability reputation value
Performance reputation value
...

Fig. 4. The basic framework of HMRep.

3 HMRep Mechanism

3.1 The Basic Framework

We envision a marketplace for various services from different vendors, in which all the services are evaluated. By employing our High-reliability Multi-faceted Reputation evaluation mechanism for online services (HMRep), the marketplace collects customers' feedback, processes the evaluation data, computes and publishes reputation values about service providers. The reputation values are then used by a consumer for selecting services. For example, a user searches for a cloud storage sharing service for family. He or she needs to know the price, capacity, security, and other aspects for each provider. If a user loves to edit pictures online, he or she may pay attention to the image processing, usability and support. Other users' detailed ratings would be very helpful for him/her to select from vendors.

The basic framework of HMRep is illustrated in Fig. 4. In terms of input, different from the prior literatures, multiple rating indexes are used to investigate services with fine granularity from various perspectives. Fig. 5 gives an example: the security of a service is evaluated from access control, encryption algorithm, key management, and data security etc; the reliability of a service is assessed from the

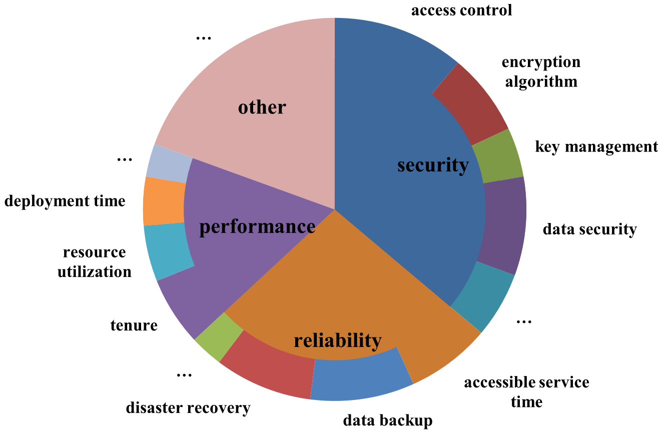


Fig. 5. An evaluation example.

accessible service time, data backup, and disaster recovery etc; and the performance of a service includes the tenure, resource utilization, and deployment time etc. The more detailed the rating indexes are, the more accurately and comprehensively the quality of the service is reflected.

The raw input from customers will be pre-processed in HMRep, considering when multiple rating indexes are introduced, some customers are unable or reluctant to provide evaluations for all indexes, resulting in incomplete data. Several missing data treatment methods have been proposed in machine learning, data mining and knowledge discovery areas [8]:

- Fixed Value Filling method: Vacancy ratings are set to a fixed default value, usually 0 or the median, e.g., 3 point in 5 scales. All the missing items are replaced by the same value.
- Mean Value Filling method: The missing ratings for a given index are substituted by the average score of all known ratings for that index. This method does not distinguish ratings between different users for an index.
- Artificial Filling method: Fill the missing ratings by experts' estimation in this filed. The quality of the method depends on experience of the experts. And the amount of missing items cannot be too large.
- Classifier Filling method: Construct a complete classification by rough set theory [16], decision tree [17] or any other data classification techniques, and infer missing ratings according to the classification rules. This method requires training samples and is computationally expensive for massive data.

However, the incomplete rating problem is neglected in current reputation calculation. In our framework, we introduce the missing rating estimation for the first time. Generally, any above missing data treatment methods are applicable in HMRep framework. But in terms of the limitations of accuracy, objectivity or efficiency, HMRep proposes a simple missing rating estimation method, which estimates and fills the missing data based on both the service quality inferred from other users' evaluation and the evaluation behavior of users who miss some evaluations. Compared with existing incomplete data filling methods, our missing rating estimation algorithm has low computational overhead, adaptive to the dynamic nature of services, and does not rely on experts' opinions.

Given the pre-processed input, HMRep designs a high-reliability index reputation calculation model, which first identifies and filters collusive and irresponsible raters based on the correlation of their multiple index ratings. For example, irresponsible users give ratings randomly. If a user gives quite a number of random ratings in terms of the indexes of a service, existing reputation evaluation mechanisms usually compute each index reputation value independently, consequently this user may be given low or high weight depending on rating deviation. Likewise for collusive users who conspire to submit highly similar feedback to promote their employers or compromise competitors, general reputation evaluation mechanisms identify collusive customers for each index reputation value only based on customers' ratings on this sole index. But HMRep inspects all indexes of a service altogether, clusters users based on evaluation similarity and then finds out malicious users. Once identified, all ratings from the irresponsible users and collusive malicious users will be removed from the reputation calculation. Then HMRep introduces a number of factors, like incentive factor and adaptive reputation learning factor, to improve the reputation calculation accuracy and adaptivity.

Finally HMRep proposes a multi-faceted reputation evaluation method. This is inspired by the cognitive development of human being that learning is always from concrete to abstract knowledge which eventually constitutes guide to action. HMRep digs the abstract attribute reputation values out of the concrete index reputation values, that help customer select services without the need of specific technical knowledge. For example, a user only cares about security, but nothing else. The reputation value presented to this user is the weighted average of all the indexes related to security. Here the critical problem is to assess the relative importance or weight of each index. Recently many weight methods have been proposed for solving multiple attribute decision making (MADM) problem. These methods can be divided into three categories [18]:

- Subjective weighting method obtains the weights based on subjective judgments of the experts, such as Delphi method, expert investigation method, and analytic hierarchy process (AHP) [19].
- Objective weighting method assigns the weights based on the objective information of evaluation matrix by applying mathematic models, such as principal component analysis, standard deviation, and Shannon Entropy [20].
- Combination weighting method is a compound method integrating the subjective and objective weighting method, such as multiplication synthetic normalization method and linear weighted combination method [21].

Generally, any above weight methods can be used in our framework. Most of existing reputation mechanisms determine index weights based on past experiences by subjective weighting methods [11], [12], [13], [14], [22]. However, they are not adaptive to reputation change, and may lead to bias. In this paper, based on the relationship among index ratings and overall rating, we deduce the weight of each index adaptively, and then combine relevant index reputation values to calculate the attribute reputation values of a service from multiple perspectives.

Below, we present the design of HMRep in detail. Without loss of generality, we focus on a single service. The notations used in HMRep are listed in Table 2.

3.2 Simple Missing Rating Estimation Method

3.2.1 Philosophy

HMRep starts dealing with the incompleteness of user feedback. The estimation of a missing rating of a particular user is largely determined by his/her rating behavior, respecting ratings varying with personalities and experiences [23], [24]. For example, demanding users tend to give low ratings while easy users tend to give high ratings; experts may give more objective ratings while the ratings given by first-time users can be more random. Once we understand the rating behavior

TABLE 2
Main Notations Used in HMRep

Symbol	Description
OQ	overall quality of the service
I_q	index q of the service
RM	rating matrix of the service
RU	rating user set of the service
r_k^{t+1}	rating given by user k on OQ in the $(t+1)^{th}$ cycle
r_{kq}^{t+1}	rating given by user k on I_q in the $(t+1)^{th}$ cycle
ERM	eligible rating matrix of the service
EU	eligible user set of the service
b_k	rating behavior of user k
$\widehat{r_{kq}^{t+1}}$	estimated rating on I_q by user k in the $(t+1)^{th}$ cycle
CRM	completed rating matrix of the service
ζ	suspicious user detection threshold
SU_q	suspicious user set of I_q
$P_{sr}\%$	suspicious rating item percentage
SU	suspicious user set of the service
s_{ef}^{t+1}	evaluation similarity of user e and user f in the $(t+1)^{th}$ cycle
γ	collusive user detection threshold
CU	collusive user set of the service
IU_q	irresponsible user set of I_q
$P_{ir}\%$	irresponsible rating item percentage
IU	irresponsible user set of the service
FU	filtered user set of the service
cr_k^{t+1}	credibility of user k in the $(t+1)^{th}$ cycle
η	maximum tolerable rating deviation
IR_q^{t+1}	index reputation value of I_q in the $(t+1)^{th}$ cycle
R^{t+1}	overall reputation value of the service in the $(t+1)^{th}$ cycle
CIR_q^{t+1}	cumulative index reputation value of I_q after $t+1$ cycles
CR^{t+1}	cumulative overall reputation value of the service after $t+1$ cycles
λ	reputation learning factor
α	reputation increasing learning factor
β	reputation reducing learning factor
A_l	attribute l of the service
AR_l^{t+1}	reputation value of A_l in the $(t+1)^{th}$ cycle
CAR_l^{t+1}	cumulative reputation value of A_l after $t+1$ cycles

of a user, we can estimate his/her missing rating based on his/her behavior and the rating given by the public.

We use b_k to characterize the rating behavior of the user k. A positive value of b_k indicates that user k tends to give above-average ratings, while a negative value of b_k, on the contrary, means user k often gives below-average ratings. To calculate b_k, we identify all the items (indexes/overall quality) rated by user k. Mathematically, b_k is computed as the average difference between the ratings given by user k on these items and the average ratings given by all users on these items. For a particular index/overall quality that user k does not rate, the rating can be calculated by adding b_k to the average of ratings given by all the other users.

3.2.2 Data Structure

Before we present the steps to process raw data, we introduce data structure first. Suppose users need to rate on the overall quality (denoted by OQ) and m indexes (denoted by I_q $(1 \le q \le m)$) of a service. Let RU be the rating user set of the service, $\|RU\| = g$. We employ a $g \times (m+1)$ matrix to represent the ratings given by users. The overall rating given by the user $ru_k \in RU$ $(1 \le k \le g)$ on OQ in the $(t+1)^{th}$ evaluation cycle is denoted by r_k^{t+1}. The rating on I_q by ru_k in the $(t+1)^{th}$ evaluation cycle is denoted by r_{kq}^{t+1}. The rating matrix RM in the $(t+1)^{th}$ evaluation round is illustrated as follows:

$$\begin{array}{c} \\ ru_1 \\ ru_2 \\ \vdots \\ ru_g \end{array}\begin{array}{c} \begin{array}{ccccc} OQ & I_1 & I_2 & \cdots & I_m \end{array} \\ \begin{pmatrix} r_1^{t+1} & r_{11}^{t+1} & \square & \cdots & r_{1m}^{t+1} \\ r_2^{t+1} & \square & r_{22}^{t+1} & \cdots & r_{2m}^{t+1} \\ \vdots & \vdots & \vdots & & \vdots \\ \square & r_{g1}^{t+1} & r_{g2}^{t+1} & \cdots & \square \end{pmatrix} \end{array},$$

Note that a square, denoted by $\square$, in row i and column 1 indicates ru_i has not rated OQ, and a $\square$ in row i and column $l+1$ $(1 \le l \le m)$ indicates ru_i has not rated I_l.

3.2.3 Data Processing

The first step to process the raw data is to remove users who have rated less than half of the items (indexes/overall quality). When a user rates inadequate items, we do not have enough information to speculate his/her rating behavior and thus we cannot estimate his/her missing ratings. By removing rows whose number of $\square$ is greater than $\lceil \frac{m}{2} \rceil$ in RM, we obtain an eligible $h \times (m+1)$ rating matrix ERM, which is a sub-matrix of matrix RM:

$$\begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_h \end{array}\begin{array}{c} \begin{array}{ccccc} OQ & I_1 & I_2 & \cdots & I_m \end{array} \\ \begin{pmatrix} r_1^{t+1} & r_{11}^{t+1} & r_{12}^{t+1} & \cdots & \square \\ r_2^{t+1} & \square & \square & \cdots & r_{2m}^{t+1} \\ \vdots & \vdots & \vdots & & \vdots \\ \square & r_{h1}^{t+1} & \square & \cdots & r_{hm}^{t+1} \end{pmatrix} \end{array},$$

where u_l $(1 \le l \le h)$ is an eligible user. The eligible user set is denoted by EU, $\|EU\| = h$ $(h \le g)$.

The second step is to find out the users who do not rate on all indexes/overall quality to complement the missing ratings. Take user $u_k \in EU$ as an example. u_k rates on OQ and a indexes, but misses the other $m-a$ indexes. The indexes rated by u_k are denoted by a set RI_k. To estimate the missing rating on I_q, we first calculate the rating behavior b_k as explained previously.

$$\begin{aligned} b_k &= \frac{(r_k^{t+1} - \mu^{t+1}) + \sum_{p \in RI_k} (r_{kp}^{t+1} - \mu_p^{t+1})}{\|RI_k\| + 1}, \\ \mu^{t+1} &= \sum_{u_i \in U_{OQ}} r_i^{t+1} \Big/ \|U_{OQ}\|, \\ \mu_p^{t+1} &= \sum_{u_i \in U_{Ip}} r_{ip}^{t+1} \Big/ \|U_{Ip}\|, \end{aligned} \tag{1}$$

where $U_{OQ} \subset EU$ and $U_{Ip} \subset EU$ are the user sets who have given ratings on OQ and Ip respectively; μ^{t+1} and μ_p^{t+1} are the average rating on OQ and I_p by all users in U_{OQ} and U_{Ip} respectively. Then we calculate the average rating on I_q by all the users who rate I_q, which is denoted by μ_q^{t+1}. Here we assume that we have enough user ratings to calculate the average rating of them. The missing rating $\widehat{r_{kq}^{t+1}}$ can be estimated by adding b_k and μ_q^{t+1}:

$$\widehat{r_{kq}^{t+1}} = b_k + \mu_q^{t+1}. \tag{2}$$

The algorithm not only exhibits the difference of indexes of the service, but also takes into account the difference of customers.

By substituting all squares in ERM by the estimated ratings, we obtain the complete rating matrix CRM of the service:

$$\begin{array}{c|ccccc} & OQ & I_1 & I_2 & \cdots & I_m \\ \hline u_1 & r_1^{t+1} & r_{11}^{t+1} & r_{12}^{t+1} & \cdots & r_{1m}^{t+1} \\ u_2 & r_2^{t+1} & r_{21}^{t+1} & r_{22}^{t+1} & \cdots & r_{2m}^{t+1} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ u_h & r_h^{t+1} & r_{h1}^{t+1} & r_{h2}^{t+1} & \cdots & r_{hm}^{t+1} \end{array}.$$

3.3 High-Reliability Index Reputation Calculation Model

As shown in Fig. 6, previous reputation calculation models process users' evaluations for each index separately. However, customer rates on multiple indexes simultaneously, which should be examined together to detect malicious users. HMRep inspects all ratings of a service altogether to delete malicious users and calculates the reputation values of the server. First, HMRep investigates the deviation of a user's ratings from those of the public to identify suspicious users. Second, HMRep identifies collusive users and irresponsible users. After removing the ratings of these users, HMRep further calculates the credibility of each remaining user and the reputation values of the service based on the ratings given by these users and their corresponding credibility. The process keeps running with both historical and recent data to aggregate the final reputation values and also reflect the change of service quality.

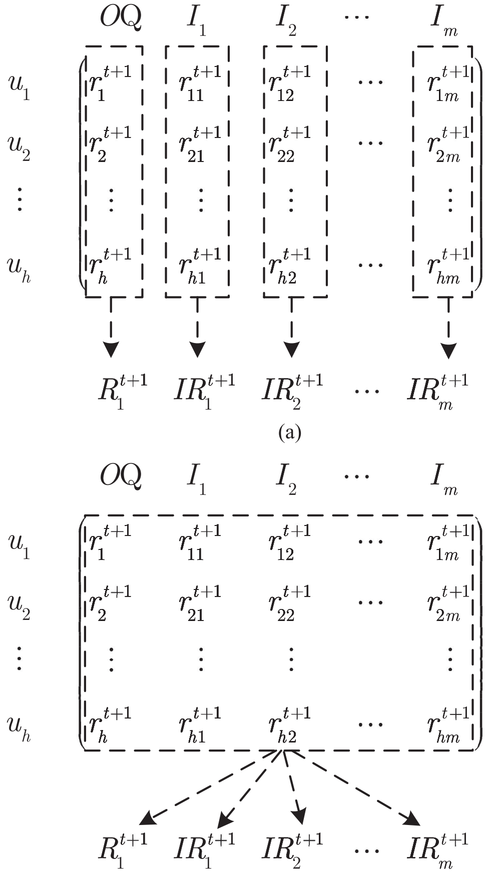
3.3.1 Identify Suspicious Users

Assuming that most users give relatively honest ratings, we first identify suspicious users whose ratings deviate from others significantly by Inequality (3).

For $\forall u_k \in EU$, if

$$\sqrt{\frac{\sum_{u_i \in EU, i \neq k} \left(r_{kq}^{t+1} - r_{iq}^{t+1}\right)^2}{h-1}} > \zeta, \tag{3}$$

u_k's rating on I_q is suspicious, where ζ $(0 \leq \zeta \leq 1)$ denotes the threshold of suspicious user detection. All suspicious users of I_q constitute the suspicious user set of I_q, denoted by SU_q.


OQ
I_1
I_2
$\cdots$
I_m
u_1
u_2
u_h
r_1^{t+1} r_{11}^{t+1} r_{12}^{t+1} r_{1m}^{t+1}
r_2^{t+1} r_{21}^{t+1} r_{22}^{t+1} r_{2m}^{t+1}
r_h^{t+1} r_{h1}^{t+1} r_{h2}^{t+1} r_{hm}^{t+1}
R_1^{t+1} IR_1^{t+1} IR_2^{t+1} $\cdots$ IR_m^{t+1}

(a)

(b)

Fig. 6. Reputation calculation model.

A user who belongs to P_{sr} percent $(0 < P_{sr} < 100)$ or greater of suspicious user sets in terms of indexes/overall quality is called a suspicious user of the service, where P_{sr} percent is called the suspicious rating item percentage. All of such users are denoted by SU.

3.3.2 Identify Collusive Users

In SU, we identify collusive users through evaluation similarity clustering. Those who are employed by service providers to overstate indexes of employers or understate those of competitors are referred to as collusive users. We make the following two assumptions on a collusive group: 1) the target of each member in collusive group is consistent. If the collusive group attacks a target, all members of the group are involved in the attack on the target; 2) the action of each member in collusive group is the same. If the intention of the group is to undermine or exaggerate the reputation value of the target, all members of the group will follow the same pattern.

The collusion detection process is as follows:

1) Calculate the evaluation similarity of each pair of users: for any two users in SU, calculate their similarity in the current evaluation cycle. For $\forall u_e, u_f \in SU$, the evaluation similarity of them in the $(t+1)^{th}$ evaluation cycle is defined as

$$s_{ef}^{t+1} = 1 - \sqrt{\frac{\left(r_e^{t+1} - r_f^{t+1}\right)^2 + \sum_{p=1}^{m} \left(r_{ep}^{t+1} - r_{fp}^{t+1}\right)^2}{m+1}}. \tag{4}$$

2) Construct the maximum spanning tree of fuzzy graph: first construct fuzzy graph $\mathrm{G} = (V, E)$ from elements in SU, where V denotes the set of vertices and E denotes the set of undirected edges. The weight of an edge is the evaluation similarity of the two connected vertices calculated by Equation (4). We then construct maximum spanning tree of graph G [25]. Different edges of the maximum spanning tree may have equal weights, so the maximum spanning tree of fuzzy graph is not unique. But this does not affect the final result of collusion detection.
3) Cut the edges of the maximum spanning tree to perform clustering: cut the edges with the weight below γ $(0 \leq \gamma \leq 1)$ in the maximum spanning tree (γ is called the threshold of collusive user detection). Each resultant connected branch constitutes a cluster, vertices in which are possible collusive users—denoted by CU.

3.3.3 Identify Irresponsible Users

If a user consistently is an outlier when rating on different indexes, we determine the user is an irresponsible user that gives random ratings and should not be included in the reputation calculation of indexes. Removing collusive users CU from index q's suspicious user set SU_q, we get irresponsible user set of I_q: $IU_q = SU_q - CU$.

We employ an irresponsible rating item percentage P_{ir} percent $(0 < P_{ir} < 100)$. If $u_k \in EU$ belongs to the irresponsible user set of more than P_{ir} percent rating items (indexes/overall quality), we determine that u_k is an irresponsible user of the service. The irresponsible user set is denoted by IU.

3.3.4 Calculate Reputation Values

Removing both collusive users and irresponsible users from the eligible user set, we obtain a filtered user set $FU = EU - CU - IU$. The calculation of the reputation value of an index/overall quality is based on both the ratings given by users in FU and the credibility of these users.

The credibility of u_k, denoted by cr_k^{t+1} , is defined by the following formula:

$$\begin{aligned} cr_k^{t+1} &= s_k^{t+1} + \delta_k^{t+1}, \\ s_k^{t+1} &= 1 - \sqrt{\frac{\left(r_k^{t+1} - \bar{r}_{OQ}^{t+1}\right)^2 + \sum_{p=1}^{m} \left(r_{kp}^{t+1} - \bar{r}_{Ip}^{t+1}\right)^2}{m+1}}, \\ \bar{r}_{OQ}^{t+1} &= \sum_{u_l \in FU} r_l^{t+1} \Big/ \|FU\|, \\ \bar{r}_{Ip}^{t+1} &= \sum_{u_l \in FU} r_{lp}^{t+1} \Big/ \|FU\|, \\ \delta_k^{t+1} &= \begin{cases} \frac{1-s_k^{t+1}}{2} \times \left(1 - \frac{1-s_k^{t+1}}{\eta}\right), & 1 - s_k^{t+1} < \eta \\ -\frac{s_k^{t+1}}{2} \times \left(1 - \frac{\eta}{1-s_k^{t+1}}\right), & else, \end{cases} \end{aligned} \tag{5}$$

where s_k^{t+1} is the similarity of user k, comparing with the overall preference of the population; $\bar{r}_{OQ}^{t+1}$ and $\bar{r}_{Ip}^{t+1}$ indicate the average rating on OQ and I_p by all users in FU respectively; δ_k^{t+1} is the incentive factor and η $(0 \leq \eta \leq 1)$ denotes the maximum tolerable rating deviation. When the evaluation given by u_k lies within an acceptance range determined by the average evaluation of all users and the maximum tolerable rating deviation, u_k is treated as trustworthy, and cr_k^{t+1} is gained by a small margin; on the contrary, u_k is considered unlikely credible, cr_k^{t+1} declines rapidly. This is consistent with the sociology understanding of the credibility [26].

The reputation value for index I_q in cycle $t+1$, denoted by IR_q^{t+1}, is calculated as the weighted average of ratings on I_q given by users in FU.

$$IR_q^{t+1} = \sum_{u_k \in FU} cr_k^{t+1} \times r_{kq}^{t+1} \Big/ \sum_{u_k \in FU} cr_k^{t+1}. \tag{6}$$

Similarly, the overall reputation value of the service in cycle $t+1$, denoted by R^{t+1}, is calculated as the weighted average of ratings on OQ given by users in FU.

$$R^{t+1} = \sum_{u_k \in FU} cr_k^{t+1} \times r_k^{t+1} \Big/ \sum_{u_k \in FU} cr_k^{t+1}. \tag{7}$$

3.3.5 Calculate Cumulative Reputation Values

Consider the service quality may fluctuate, improve or deteriorate, we calculate a cumulative reputation value incorporating both historical performance and most current performance.

Without loss of generality, after $t+1$ evaluation cycles, the cumulative reputation value for I_q, denoted by CIR_q^{t+1}, is calculated as the weighted average of past cumulative reputation value CIR_q^t and the reputation value of $t+1$ evaluation cycle.

$$CIR_q^{t+1} = (1-\lambda) \times CIR_q^t + \lambda \times IR_q^{t+1}. \tag{8}$$

where the reputation learning factor λ $(0 < \lambda \leq 1)$ is the weight given to the most current reputation value. A greater λ gives more recent ratings higher weight. Note we set $CIR_q^0 = 0.5$ as the initial value.

CR^{t+1}, the cumulative overall reputation value of the service after $t+1$ evaluation cycles, is given by

$$CR^{t+1} = (1-\lambda) \times CR^t + \lambda \times R^{t+1}. \tag{9}$$

Also, we set $CR^0 = 0.5$ as the initial value.

In order to reflect the change of service quality reasonably, we employ an adaptive reputation learning factor λ to calculate the cumulative reputation value. We can observe that in human society, good reputation is built up gradually and slowly, but it can be destroyed very fast if something bad happens [26]. Mathematically it means the increasing and decreasing of the reputation value is asymmetric. Therefore, we use a different λ when the newly calculated reputation value of the current evaluation cycle is better than the previous one and when it is worse.

$$\lambda = \begin{cases} \alpha, & \mathrm{IR}_q^{t+1} \geq CIR_q^t \\ \beta, & else \end{cases}, \tag{10}$$

TABLE 3
Multiple Reputation Values of the Services

Service	IR_1	IR_2	IR_3	IR_4	R	AR_1	AR_2
S1	0.25	0.25	0.25	0.25	0.30	0.25	0.25
S2	0.80	0.25	0.80	0.80	0.60	0.57	0.80
S3	0.80	0.80	0.25	0.80	0.63	0.80	0.43
S4	0.70	0.70	0.70	0.70	0.66	0.70	0.70

the reputation increasing learning factor α is the weight when the reputation becomes better while the reputation reducing learning factor β is the weight when the reputation becomes worse. $0 < \alpha < \beta \leq 1$.

3.4 Multi-faceted Reputation Evaluation Method

Instead of an overall reputation, customers have different needs and focus. Some may only care about the security but nothing else, while others care about everything. Thus, it is important to provide individualized multi-faceted evaluation. To satisfy a specific evaluation need, we first figure out the related indexes. For example, security is characterized by the indexes of access control, encryption algorithm, key management, and data security. Then the weight of each index has to be obtained. The individualized reputation value is the weighted average of the reputation values of its all relevant indexes.

The determination of the weights is important and challenging. First, the weights reflect the correlation between overall rating and indexes ratings. Overall rating is definitely not a simple average of index ratings. Second, the weights reveal the underlying relationship between indexes which is intangible but sensible to customers. Finally, using the weights, we can assess the service from higher and broader angles, with more accuracy. Different from most of existing works using fixed weights, HMRep derives the weights from customers' ratings dynamically. This section analyzes the relationship between users' ratings on indexes and overall quality, and then infers each index's weight relative to the overall reputation. The idea behind it is that the rating of the overall quality is built up from all specific relevant ratings on indexes. Given the ratings of the overall quality and individual indexes by all the legitimate users, the objective weights can be deduced.

We select d customers $\{ u_{f_1} \quad u_{f_2} \quad \cdots \quad u_{f_d} \}$ from FU in all evaluation cycles to compute the weights of indexes. Let

$$R_{OQ} = \begin{pmatrix} r_{f_1} & r_{f_2} & \cdots & r_{f_d} \end{pmatrix}^T$$

be the ratings given by these users on overall quality of the service,

$$R_I = \begin{pmatrix} r_{f_1 1} & r_{f_1 2} & \cdots & r_{f_1 m} \\ r_{f_2 1} & r_{f_2 2} & \cdots & r_{f_2 m} \\ \vdots & \vdots & & \vdots \\ r_{f_d 1} & r_{f_d 2} & \cdots & r_{f_d m} \end{pmatrix}$$

be the ratings given by these users on m indexes of the service, we have the following equation:

$$R_{OQ}{}^T = R_I \times W^T + C^T, \tag{11}$$

where $W = \begin{pmatrix} w_1 & w_2 & \cdots & w_m \end{pmatrix}^T$ is the weight vector, $C = \begin{pmatrix} c & c & \cdots & c \end{pmatrix}^T$ is the constant vector and $\|C\| = d$. Employing the n-dimensional linear fitting method [27], we can get W and C.

TABLE 4
Service Selection Strategy and Decision

Selection strategy	Decision
maximum overall quality	S4
maximum attribute 1	S3
maximum attribute 2	S2

For a particular attribute A_l $(1 \leq l \leq n)$, its reputation value can be calculated as the weighted average of the reputation values of the relevant indexes $\begin{pmatrix} I_{l1} & I_{l2} & \cdots & I_{lz} \end{pmatrix}$. Suppose the corresponding weight vector is denoted by $W_l = \begin{pmatrix} w_{l1} & w_{l2} & \cdots & w_{lz} \end{pmatrix}$. AR_l^{t+1}, the reputation value of attribute A_l in the $t+1$ evaluation cycle, is given by

$$AR_l^{t+1} = \sum_{i=1}^{z} IR_{li}^{t+1} \times w_{li} \Bigg/ \sum_{k=1}^{z} w_{lk}. \tag{12}$$

CAR_l^{t+1}, the cumulative reputation value of attribute A_l after $t+1$ evaluation cycles, is calculated as

$$CAR_l^{t+1} = \sum_{i=1}^{z} CIR_{li}^{t+1} \times w_{li} \Bigg/ \sum_{k=1}^{z} w_{lk}. \tag{13}$$

For better understanding of our algorithm, a multi-faceted reputation evaluation example is given below. There are 4 services, each of which has 4 rating indexes and 2 attributes. Attribute 1 relates to index 1 and index 2, and index 3 and index 4 belong to attribute 2. The index and overall reputation values of the services are listed in column 2 to column 6 in Table 3. Suppose the weight vector $W = \begin{pmatrix} 0.35 & 0.25 & 0.2 & 0.1 \end{pmatrix}^T$ and constant $c = 0.1$. The attribute reputation values of the services are calculated and listed in the last two columns in Table 3. Customers can choose a service with different preferences. As shown in Table 4, there are 3 service selection strategies: maximum overall quality, maximum attribute 1 and maximum attribute 2. Different strategy will lead to different decision. Users can pick services at higher level that is depicted by computing attribute reputation values.

4 Evaluations

In this paper, we design a comprehensive reputation evaluation system for service selection which has multiple components, such as missing rating estimation, index weight calculation, malicious user detection, and reputation value calculation. Some components can also be utilized by other reputation evaluation mechanisms. For example, index weight calculation is an essential component of many evaluation mechanisms.

In the following section, we first evaluate our missing rating estimation method and index weight calculation algorithm. The parameters and their default values of HMRep are listed in Table 5. The evaluation is based on the data set collected from Dianping [15] and the performance is verified against the real data. For example, when to evaluate the missing rating estimation method, we randomly

TABLE 5
Parameters and Their Default Values of HMRep

Parameter	Description	Values
ζ	suspicious user detection threshold	0.55
$P_{sr}\%$	suspicious rating item percentage	30%
γ	collusive user detection threshold	0.85
$P_{ir}\%$	irresponsible rating item percentage	50%
η	maximum tolerable rating deviation	0.1
α	reputation increasing learning factor	0.2
β	reputation reducing learning factor	0.6

remove some user ratings from the true data set, estimate the values using our algorithm, and compare with the removed original data.

Then we evaluate the overall effectiveness of our reputation evaluation system by using simulation. We focus on three metrics: 1) the capability of detecting malicious users; 2) the accuracy of the calculated service quality with the existence of malicious nodes compared to the true service quality; 3) sensitiveness of the calculated reputation value to the change of service quality, which is to measure when the service quality changes, whether the calculated reputation value can quickly reflect the change.

4.1 Missing Rating Estimation Method

We establish a 180×30 matrix based on Dianping dataset, in which 180 customers rate on the "taste" for 30 vendors. For each customer, we randomly delete 5 ratings while ensuring that every vendor is rated by at least 30 customers. We then estimate the deleted ratings using HMRep, Mean Value Filling method (MVF) and decision tree C4.5 [8] respectively. MVT and C4.5 are methods broadly used to treat missing values in machine learning.

Let RU be the rating user set and DV_i be the deleted vendor set of u_i. We use rr_{iq} and er_{iq} to denote the normalized real rating and the estimated rating on "taste" given by u_i for vendor q respectively. We compute the missing rating estimation error ($MREE$) as

$$\text{MREE} = \frac{\sum_{u_i \in RU, q \in DV_i} \left|er_{iq} - rr_{iq}\right| \Big/ \|DV_i\|}{\|RU\|}. \tag{14}$$

Fig. 7 shows the MREE of HMRep, MVF and C4.5. As shown in the figure, compared to MVF, the MREE of HMRep reduced by about 15 percent. HMRep has a lower missing rating estimation error. This is because unlike MVF which fills in a vendor's all missing ratings using the same average value calculated from all known ratings, HMRep takes into account users' rating characteristics. It estimates the unknown rating from a customer to a vendor by including the average value of the vendor and the rating behavior of the user. Although the MREE of HMRep is about 8 percent larger than C4.5. The execution time of HMRep is significantly shorter than C4.5.

4.2 Index Weight Calculation Algorithm

We randomly select a part of vendors from our Dianping dataset as a training set. The rest of the vendors shall be a validation set. The constant and weight vector of "taste", "environment" and "service" are computed based on the training set. Using the constant and weight vector, the ratings on "overall evaluation" for the validation set are computed. Note that for the consistent range of values, all ratings on "taste", "environment" and "service" are normalized by dividing them by 40, and all ratings on "overall evaluation" are normalized by dividing them by 5.

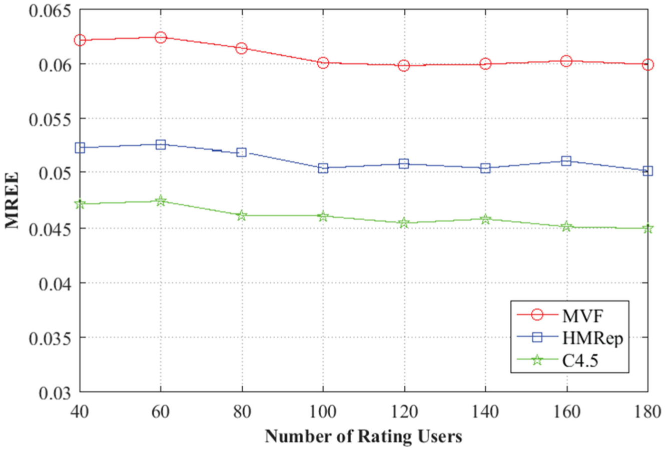


Fig. 7. Missing rating estimation error.

Let VV be the validation vendor set, RR_i be the normalized real reputation value of "overall evaluation" for vendor i, CR_i be the computed reputation value of "overall evaluation" for vendor i. We define the index weight calculation error ($IWCE$) as

$$\text{IWCE} = \sum_{i \in VV} |RR_i - CR_i| \Big/ \|VV\|. \tag{15}$$

Table 6 shows the derived constant, weights and IWCE of HMRep. The weight of "taste" is almost twice of "environment" and "service", that suggests customers care more about "taste".

For comparison, we recalculate the overall reputation by Shannon Entropy [20] and simple average approach (denoted as Ave). In Ave, we set the weight $w = [0.3333, 0.3333, 0.3333]^T$ and an additional constant $c = 0.2$, which is the systematical bias between the real reputation and the weighted reputation. In Fig. 8, the results of HMRep and Shannon Entropy are quite close. Compared to Ave, the ICWE of HMRep reduced by about 14 percent, HMRep outperforms Ave completely.

Fig. 9 further compares the real reputation values and computed reputation values for 130 vendors (the size of training set is 1,200). We sort the 130 vendors by the true reputation value. The real ratings are clustered around 0.6, 0.7, 0.8 and 0.9 due to the fixed scoring rule (three-star, quasi-four-star, four-star, and quasi-five-star). The data points calculated by HMRep, plotted as dots, are closer to the Dianping's actual scores than the data points calculated by Ave, plotted as asterisks. The result shows 80.77 percent of the rating difference between HMRep and Dianping is within 5 percent, 90.76 percent of it is less than 10 percent. In comparison, only 18.46 percent of the ratings between Ave and Dianping have 5 percent difference, 68.46 percent of them have 10 percent difference. This is because our mechanism takes advantage of the hidden relationship among indexes and the overall quality and deduces the weight of each index adaptively.

TABLE 6
Derived Constant, Weights and IWCE of HMRep

Training set size	Constant	Weight of "taste"	Weight of "environment"	Weight of "service"	*IWCE*
100	0.0115	0.6194	0.3367	0.3868	0.0370
200	0.0109	0.7196	0.3572	0.2565	0.0362
300	0.0164	0.7578	0.3188	0.2426	0.0359
400	0.0089	0.7273	0.2009	0.4014	0.0364
500	0.0432	0.6790	0.2076	0.3914	0.0360
600	0.0275	0.7330	0.2237	0.3481	0.0359
700	0.0525	0.7198	0.2693	0.2653	0.0360
800	0.0070	0.8095	0.2602	0.2658	0.0359
900	0.0303	0.7291	0.3325	0.2340	0.0359
1000	0.0125	0.7573	0.2605	0.3023	0.0361
1100	0.0138	0.7546	0.2368	0.3330	0.0359
1200	0.0124	0.7998	0.2744	0.2517	0.0359
1300	0.0080	0.7962	0.2426	0.2914	0.0359
1400	0.0173	0.7604	0.2641	0.2935	0.0358

4.3 Reputation Calculation System

4.3.1 Methodology

We evaluate the effectiveness of HMRep through simulation. Our simulated service vendor has 50 indexes. Q_i $(1 \leq i \leq 50)$, the true quality of index i, is modeled as a random number between 0 and 1. Customers rate on these indexes of the service on a 0-1 scale. Three categories of customers are simulated:

- Creditable users give honest ratings, i.e., they rate the service from index i as Q_i.
- Collusive users give complementary ratings, i.e., they rate the service from index i as $1 - Q_i$.
- Irresponsible users give random ratings on P_{ir} percent indexes and give honest ratings on other indexes. Here the random rating to index i means a random score between 0 and 1 regardless of Q_i.

Collusive and irresponsible users together are referred to as malicious users. In our evaluation, we simulate 1,000 users with two user compositions: 1) 700 creditable users and 300 malicious users; 2) 900 creditable users and 100 malicious users. The two user compositions represent two scenarios: the percentage of malicious users is high (30 percent) and the percentage of malicious users is modest (10 percent).

One of the most common reputation evaluation mechanisms, Weighted Majority Algorithm (WMA), is implemented and served as a benchmark [28], [29], [30], [31]. WMA assigns weights to advisors, makes a prediction based on the weighted sum of the ratings provided by them, and furthermore tunes the weights dynamically during interactions.

The simulation works on a continuous basis. In each simulation cycle, 10 customers rate on the 50 indexes of the service. Using these users' ratings, new reputation values of the service are calculated by HMRep and WMA respectively. Then the simulation shifts to the next cycle. We repeat 5 simulation experiments and take the average as final experiment results.

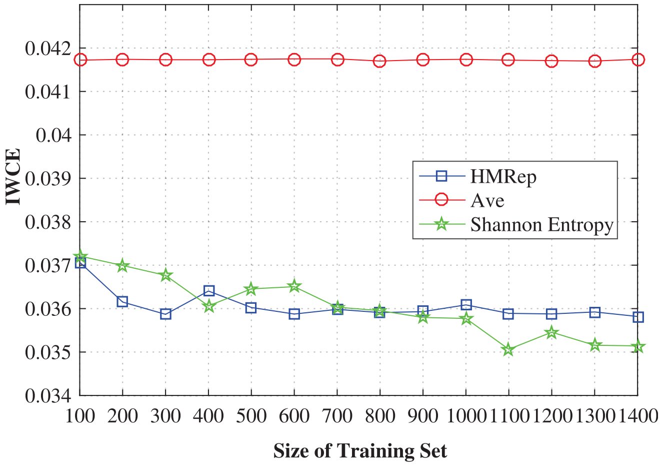


Fig. 8. Index weight calculation error.

4.3.2 Malicious User Detection

This section demonstrates HMRep's ability to detect malicious users. We define two *evaluation metrics*: false positive rate (FPR) and false negative rate (FNR). Let GS, CS and IS be the set of creditable, collusive and irresponsible users respectively. Let DGS, DCS and DIS be the set of detected creditable, collusive and irresponsible users respectively.

- FPR, the percentage of creditable users who are mistakenly categorized as malicious users, is defined as

$$FPR = \|GS \cap (DCS \cup DIS)\| / \|GS\|. \qquad (16)$$

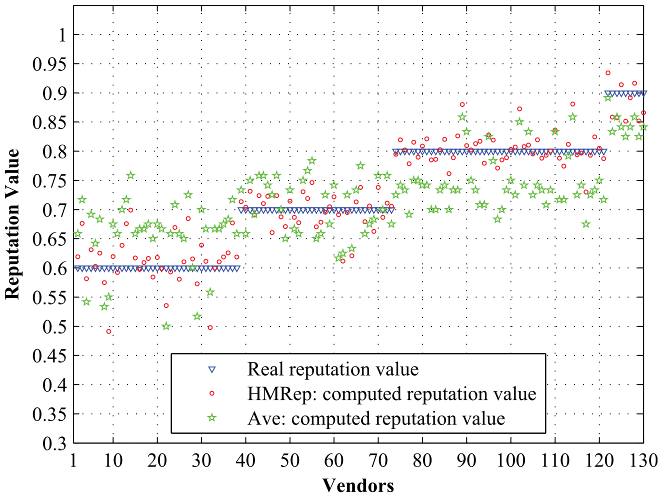
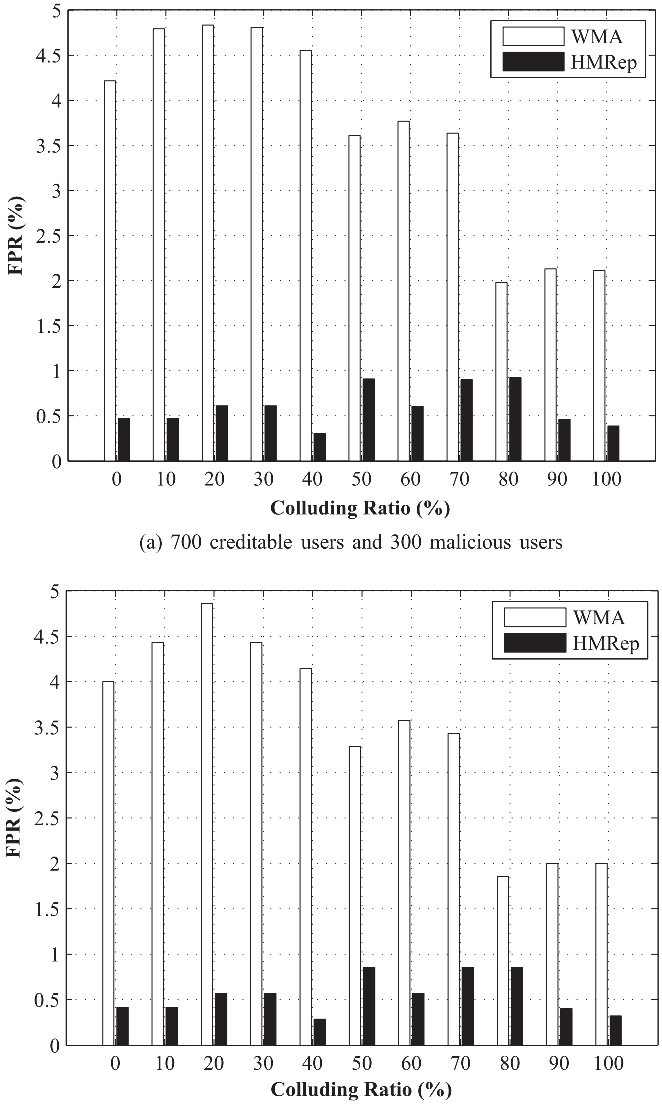


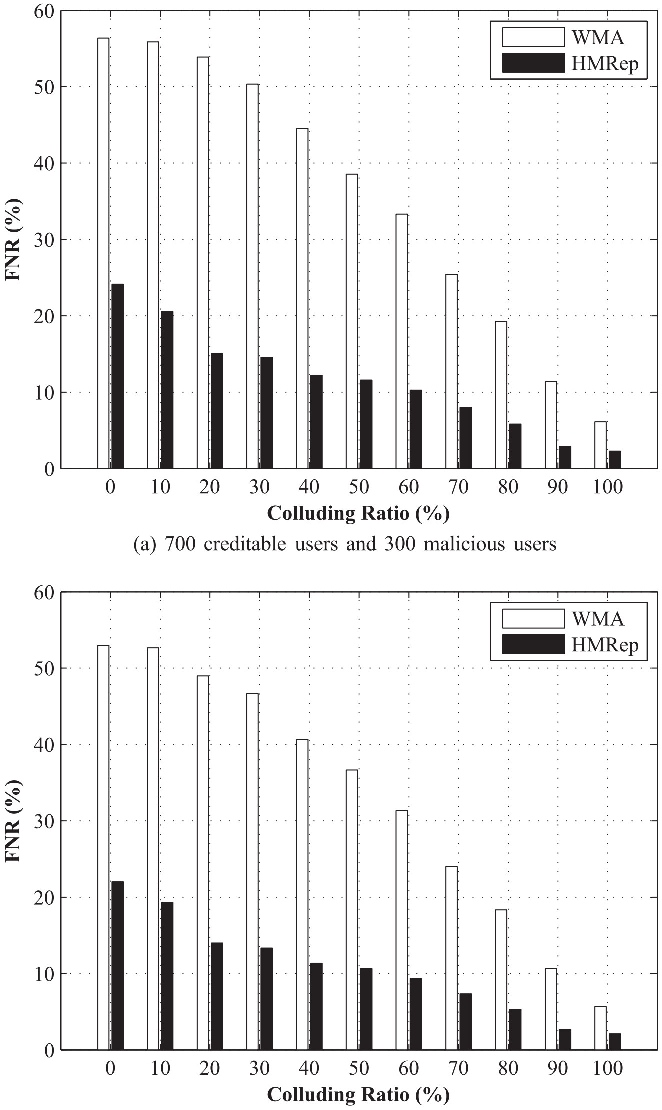
Fig. 9. Computed reputation value of 130 vendors.



(a) 700 creditable users and 300 malicious users

(b) 900 creditable users and 100 malicious users

Fig. 10. False positive rate.



(a) 700 creditable users and 300 malicious users

(b) 900 creditable users and 100 malicious users

Fig. 11. False negative rate.

- *FNR*, the percentage of malicious users who are not detected and are categorized as creditable users, is defined as

$$FNR = \|(CS \cup IS) \cap DGS\| / \|CS \cup IS\|. \qquad (17)$$

To evaluate our scheme's capability of detecting malicious users, the collusive user ratio varies from 0 percent to 100 percent. For example, if the collusive user ratio is 50 percent, that means half of the malicious users are irresponsible and give random ratings and half of the malicious users are colluding.

Fig. 10 shows the *FPR* of HMRep and WMA under different collusive user ratios. As shown in Fig. 10, FPR of HMRep keeps at a low state no matter how much the collusive node ratio increases. Fig. 11 shows the *FNR* of HMRep and WMA under different collusive user ratios. With the increasing of collusive users and decreasing of irresponsible users in malicious users, FNR of HMRep slides gradually.

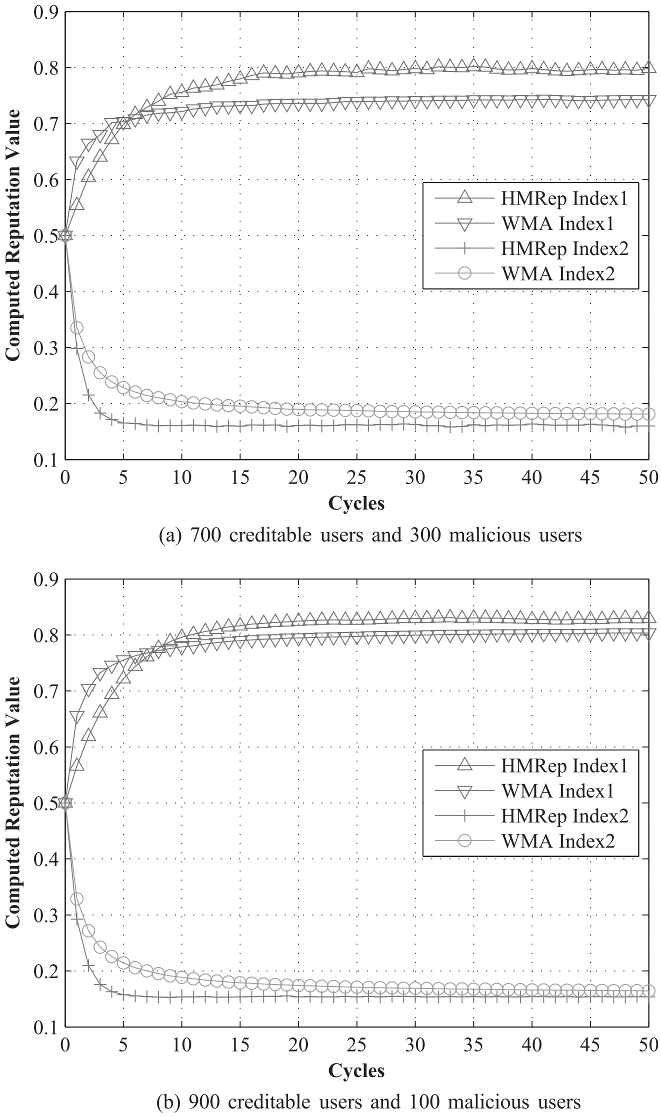
Figs. 10 and 11 indicate HMRep has a consistently excellent ability to identify malicious users in various circumstances. In addition, the figures demonstrate HMRep beats WMA. This is because HMRep takes into account all index ratings of users simultaneously to accurately identify and filter unreliable ratings.

4.3.3 Reputation Calculation and Rating Difference

The following experiments examine reputation calculation of HMRep. Without loss of generality, we focus on index 1 and index 2. The quality of index 1 and index 2 is set to 0.85 and 0.15 respectively. In this section, the malicious users are composed by 50 percent collusive users and 50 percent irresponsible users. We introduce one new *evaluation metric*: rating difference. Rating difference for index i (RD_i) is the difference between the reputation value and real quality for index i, which is defined as

$$RD_i = \left|CIR_i - Q_i\right|. \qquad (18)$$

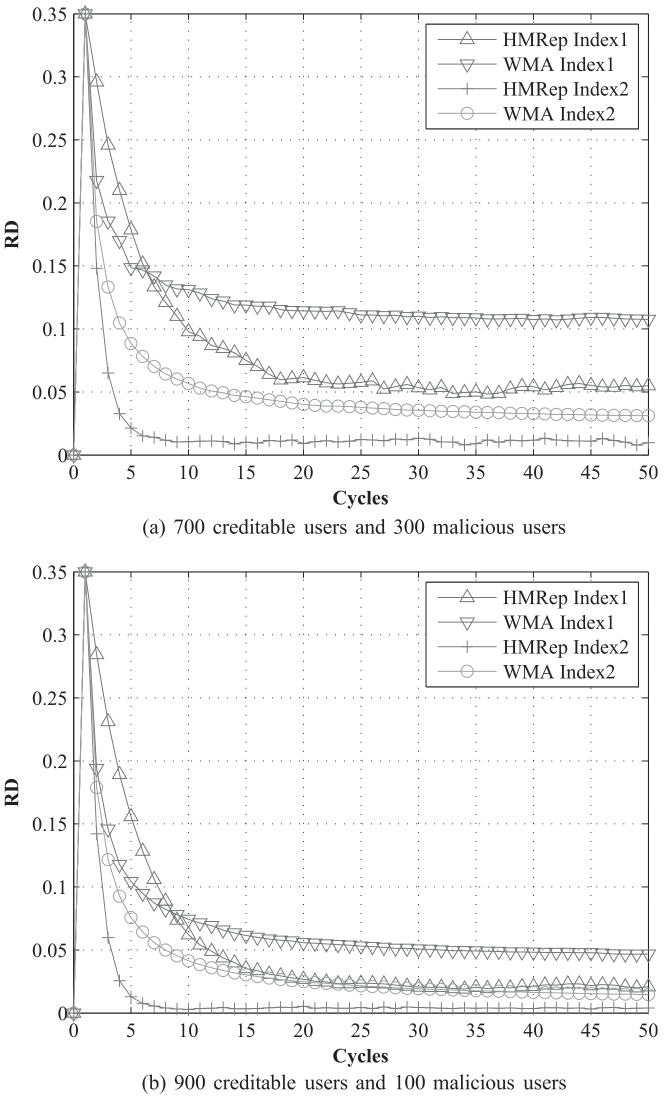
Fig. 12 shows the reputation values of index 1 and index 2 calculated by HMRep and WMA. In Fig. 12b, the reputation value of index 1 calculated by HMRep reaches 0.83 while WMA reaches 0.80 after 25 cycles. Meanwhile, the



(a) 700 creditable users and 300 malicious users

(b) 900 creditable users and 100 malicious users

Fig. 12. Computed reputation values of index 1 and index 2.



(a) 700 creditable users and 300 malicious users

(b) 900 creditable users and 100 malicious users

Fig. 13. Rating differences of index 1 and index 2.

reputation value of index 2 calculated by HMRep reaches 0.15 while WMA reaches 0.17. In conclusion, the index reputation values calculated by HMRep are close to the real quality values more accurately than WMA. Note that the differences between results given by HMRep and WMA in Fig. 12a are greater than those in Fig. 12b because of less creditable users.

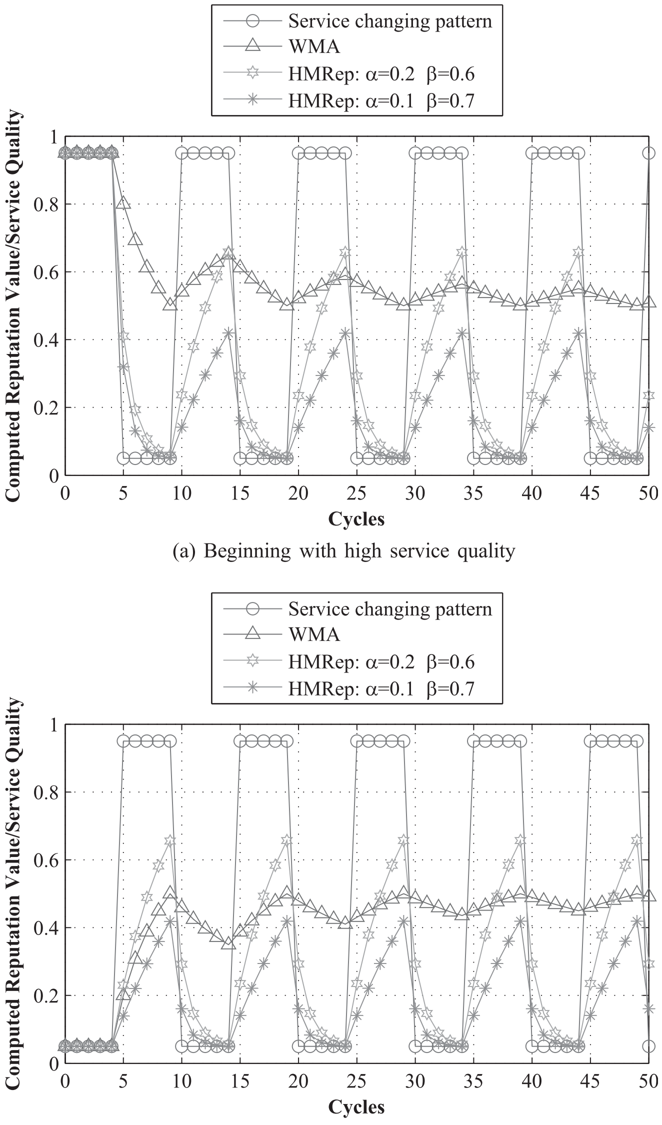
Fig. 13 further shows the rating difference of index 1 and index 2. The simulations show that RD_1 and RD_2 decrease over time, and the RD_1 and RD_2 of HMRep are smaller than those of WMA. This indicates that our approach can more effectively purge the negative interference of malicious evaluations, and hence more accurately calculate the index reputation values. Compared to Fig. 13b, notably, there are more dishonest users in Fig. 13a, so Fig. 13a has a larger rating difference.

4.3.4 Response to Service Quality Change

This section inspects HMRep's response to the change of service quality. We simulate a community with all creditable users and a service provider whose service oscillates between high quality (0.95) and low quality (0.05). Specifically, the service quality is 0.95 for five cycles, and then drops to 0.05 for five cycles and the pattern repeats. We consider two scenarios: 1) the pattern starts with high service quality; 2) the pattern starts with low service quality. We aim to assess how fast that HMRep can catch up with the service quality change.

Fig. 14 shows the calculated reputation value by HMRep and WMA, compared to the true service quality. We can see from the figure, that HMRep exhibits a more sensitive reaction to the change than WMA and thus can better reflect the true service quality when the service quality changes, thanks to the adaptive reputation learning factor. Moreover, HMRep can reflect the change of service quality more quickly with bigger reputation learning factor.

When inspecting the adaptation to the increase of service quality and the decrease of service quality, the reputation of the high quality service node declines rapidly to an extremely low value due to providing low quality service during some cycles. On the contrary its reputation cannot return to the original high value by providing high quality service during the following several cycles. A slow reaction to the upward quality change may exclude those service providers who frequently change their quality and encourage consistently reliable service providers.



(a) Beginning with high service quality

(b) Beginning with low service quality

Fig. 14. Response to service quality change.

5 Related Work

Many reputation evaluation mechanisms have been proposed for e-commerce [32], P2P [33], and web services [11] etc. to assist customers in choosing service providers. However, the diverse and dynamic natures of services, incompleteness of customer feedback and intricacy of malicious ratings pose new challenges for reputation evaluation.

In term of input of reputation evaluation, some reputation mechanisms obtain them via monitoring data only, which requires deploying extra components on client-side machines or service providers [22], [34], [35]. Some reputation mechanisms obtain reputation evidences via users' ratings only [32], [36], [37], [38], [39]. And other reputation mechanisms collect evidences from both monitoring data and users' ratings [11], [12], [13], [14]. In this work, we focus on reputation evidences provided by users' feedback only.

Since common customers use only a part of functions provided by a service and some of them are unwilling to provide feedback, some ratings can be missed. In most cases, ratings are not independent from each other. Thus, we can infer the missing ratings by identifying relations among ratings. Although researchers have proposed many missing data treatment methods in machine learning and data mining areas, for instance, Fixed Value Filling, Mean Value Filling, Artificial Filling, and Classifier Filling [8], [9], [10], however, the incompleteness of customer feedback is not considered in current reputation evaluation mechanisms.

Moreover, a key limitation of some existing reputation models is that they do not pay enough attention to malicious ratings [12], [32], [37], [40], [41]. Because of the openness of service environment, honest and cheating users coexist. For some illegal benefits, ratings submitted by the malicious users are either much higher or much lower than the actual service. Although some reputation models [11], [13], [14], [22], [38], [39] provide methods to detect malicious ratings, they individually calculate each index reputation value, assuming that ratings for each index are independent. Hence, malicious ratings cannot be detected effectively and may lead to imprecise result.

In reputation calculation models presented in the references [4], [5], [6], [7], [32], [37], [38], [39], only an overall reputation value is computed based entirely on successful invocation rate instead of multi-criteria assessment. Single reputation value systems are easy to use in reputation-oriented service comparison and selection. However, a single reputation value based on a single index cannot depict the real reputation level very well under certain circumstances [42]. The models in the references [11], [12], [13], [14], [22] introduce multiple indexes such as response time, invocation fee, and accessibility etc. to evaluate a service from different perspectives, but they use a subjective approach dependent on expert or user opinions to assign index weights. Thus, these mechanisms lack adaptability to weight the reputation indexes.

TABLE 7
Comparison of HMRep and Existing Mechanisms

Mechanism	Input	Missing rating estimation	Malicious rating detection	Index weight calculation	Output reputation values
HMRep	users' ratings	√	interdependent	objective	multiple
DSS07 [32]	users' ratings	×	×	N/A	single
T-ASE13 [37]	users' ratings	×	×	N/A	single
TSC15 [38]	users' ratings	×	independent	N/A	single
Two-dimensional [39]	users' ratings	×	independent	N/A	single
IETDL13 [22]	monitoring data	×	independent	subjective	multiple
Rateweb [11]	monitoring data and users' ratings	×	independent	subjective	multiple
Bayesian network [12]	monitoring data and users' ratings	×	×	subjective	multiple
PHAT [13], [14]	monitoring data and users' ratings	×	independent	subjective	multiple

The detailed comparison between HMRep and some existing reputation evaluation mechanisms is shown in Table 7.

6 Conclusion

Reputation is an important factor for users to select the most desired service when they are given a huge number of options. This paper proposes the HMRep mechanism to calculate service reputation to cope with the diverse and dynamic natures of services, incompleteness of user feedback, and intricacy of malicious ratings. The main features of this mechanism lie in: 1) first introduce the missing rating estimation in the reputation calculation and propose a simple missing rating estimation method, which forecasts and substitutes the missing ratings based on service quality as well as the characteristics of user feedback; 2) present a high-reliability index reputation calculation model, which deals with customers' ratings simultaneously and introduces a variety of methods to resist malicious ratings for accuracy and adaptability improvement in the reputation calculation; 3) propose a multi-faceted reputation evaluation method, which derives the index weights from user ratings adaptively and evaluates the service reputation from several angles with multiple attributes. Evaluation on Dianping indicates the validity of our missing rating estimation method and index weight calculation algorithm. Simulation results show that HMRep can effectively detect malicious rating users, accurately calculate the reputation value of services, and reasonably and sensitively response to the change of service quality.

In our future work, we will investigate more subtle malicious ratings to enhance the precision of reputation calculation model, such as camouflaged collusive users who give honest ratings on some indexes to fool the reputation evaluation mechanism. In addition, we will also investigate the challenges of deploying HMRep in the real world, such as cloud computing environment.

Acknowledgments

This work was supported by the 973 Program of China (2012CB315804), National Natural Science Foundation of China (61402446, 61572474, 61672500, 61133015), Instrument Developing Project of CAS (YZ201426), and National S&T Support Project of China (2012BAH45B01). Preliminary version of this paper was accepted by MobiSPC 2014.

References

[1] M. B. Blake and M.F. Nowlan, "Taming web services from the wild," *IEEE Internet Computing*, vol. 12, no. 5, pp. 62–69, Oct. 2008.

[2] "Hotels." [Online]. Available: http://www.hotels.com/, Accessed on: May 2016.

[3] "Taobao." [Online]. Available: http://www.taobao.com/, Accessed on: May 2016.

[4] Z. Liang and W. Shi, "Analysis of ratings on trust inference in open environments," *Performance Eval.*, vol. 65, no. 2, pp. 99–128, 2008.

[5] Y. Wang, K. J. Lin, D. S. Wong, and V. Varadharajan, "Trust management towards service-oriented applications," *Service Oriented Comput. Appl.*, vol. 3, no. 2, pp. 129–146, 2009.

[6] D. H. Kong and Y. Q. Zhai, "Trust based recommendation system in service-oriented cloud computing," in *Proc. Int. Conf. Cloud Service Comput.*, 2012, pp. 176–179.

[7] K. Chen, H. Y. Shen, K. Sapra, and G. Liu, "A social network based reputation system for cooperative P2P file sharing," *World Wide Web*, vol. 26, no. 8, pp. 2140–2153, 2015.

[8] G. E. A. P. A. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Appl. Artif. Intell.*, vol. 17, no. 5–6, pp. 519–533, 2003.

[9] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 33, no. 3, pp. 249–268, 2007.

[10] M. J. Cracknell and A. M. Reading, "Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information," *Comput. Geosciences*, vol. 63, no. 1, pp. 22–33, 2014.

[11] Z. Malik and A. Bouguettaya, "Rateweb: Reputation assessment for trust establishment among web services," *Very Large Data Bases J.*, vol. 18, no. 4, pp. 885–911, 2009.

[12] H. T. Nguyen, W. Zhao, and J. Yang, "A trust and reputation model based on Bayesian network for web services," in *Proc. IEEE Int. Conf. Web Services*, 2010, pp. 251–258.

[13] B. Li, R. Song, L. Liao, and C. Liu, "A user-oriented trust model for web services," in *Proc. IEEE 7th Int. Symp. Service Oriented Syst. Eng.*, 2013, pp. 224–232.

[14] B. X. Li, L. Liao, H. Leung, et al., "PHAT: A preference and honesty aware trust model for web services," *IEEE Trans. Netw. Service Manage.*, vol. 11, no. 3, pp. 363–375, Sep. 2014.

[15] "Dianping," [Online]. Available: http://www. dianping.com/, Accessed on: May 2016.

[16] F. V. Nelwamondo and T. Marwala, "Rough set theory for the treatment of incomplete data," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2007, pp. 1–6.

[17] Y. F. Qiu, X.Y. Zhang, X. Li, and L.-S. Shao, "Research on the missing attribute value data-oriented for decision tree," in *Proc. Int. Conf. Signal Process. Syst.*, 2010, vol. 2, pp. 637–639.

[18] W. Tang, A. Z. Chen, D. M. Li, and Y. Yang, "The application of combination weighting approach in multiple attribute decision making," in *Proc. Int. Conf. Mach. Learning Cybern.*, 2009, pp. 2724–2728.

[19] R. Biloslavo and S. Dolinsek, "Scenario planning for climate strategies development by integrating group Delphi, AHP and dynamic fuzzy cognitive maps," in *Proc. Portland Int. Conf. Manage. Eng. Technol.*, 2008, pp. 1103–1111.

[20] L. C. Wen, X. F. Zhang, and H. Wang, "Method of synthetic evaluation based on the principal component analysis and entropy weight," in *Proc. Int. Conf. Comput. Appl. Syst. Model.*, 2010, pp. 312–315.

[21] X. Li and X. P. Xiao, "Study on the combination weighting method of hybrid multiple attribute decision-making," in *Proc. IEEE Int. Conf. Grey Syst. Intell. Serv.*, 2011, pp. 561–565.

[22] X. Y. Li and J. P. Du, "Adaptive and attribute-based trust model for service-level agreement guarantee in cloud computing," *IET Inf. Secur.*, vol. 7, no. 1, pp. 39–50, 2013.

[23] L. Xiong, L. Liu, and M. Ahamad, "Countering feedback sparsity and manipulation in reputation systems," in *Proc. Int. Conf. Collaborative Comput.: Netw. Appl. Worksharing*, 2007, pp. 203–212.

[24] M. Wang, Z. J. Xu, Y. J. Zhang, and H. Zhang, "Modeling and analysis of PeerTrust-like trust mechanisms in P2P Networks," in *Proc. IEEE Global Commun. Conf.*, 2012, pp. 2689–2694.

[25] J. Clark and D. A. Holton, *A First Look at Graph Theory. Singapore: World Scientific*, 1991, pp. 1–330.

[26] Y. L. Sun, Z. Han, W. Yu, and K. J. R. Liu, "A trust evaluation framework in distributed networks: Vulnerabilityanalysis and defense against attacks," in *Proc. 25th IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1–13.

[27] M. Fisz, *Probability Theory and Mathematical Statistics*. Malabar, FL, USA: Krieger Pub. Co., pp. 1–677, 1980.

[28] B. Yu, M. P. Singh, and K. Sycara, "Developing trust in large-scale peer-to-peer systems," in *Proc. IEEE Symp. Multi-Agent Secur. Survivability*, 2004, pp. 1–10.

[29] Z. Q. Liang and W. S. Shi, "Performance evaluation of rating aggregation algorithms in reputation systems," in *Proc. Int. Conf. Collaborative Comput.: Netw. Appl. Worksharing*, 2005, pp. 1–10.

[30] L. Xie and Z. G. Han, "Trust and reputation modeling of entry and exit in evolutionary peer-to-peer systems," in *Proc. IEEE Int. Conf. Inf. Manage. Eng.*, 2010, pp. 688–693.

[31] C. Jia, L. Xie, X. C. Gan, W. Liu, and Z. Han, "A trust and reputation model considering overall peer consulting distribution," *IEEE Trans. Syst. Man Cybern. - Part A: Syst. Humans*, vol. 42, no. 1, pp. 164–177, Jan. 2012.

[32] A. Josang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Syst.*, vol. 43, no. 2, pp. 618–644, 2007.

[33] L. Xiong and L. Liu, “PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.

[34] E. M. Maximilien and M. P. Singh, “Multiagent system for dynamic web services selection,” in *Proc. Workshop Service-Oriented Comput. Agent-Based Eng.*, 2005, pp. 25–29.

[35] J. Wu, “A trust evaluation model for web service with domain distinction,” in *Proc. IEEE Int. Conf. Granular Comput.*, 2010, pp. 525–529.

[36] Z. Liu, A. An, S. Liu, and J. Li, “A prediction QOS approach reputation-based in web services,” in *Proc. 5th Int. Conf. Wirel. Commun., Netw. Mobile Comput.*, 2009, pp. 1–4.

[37] Y. Wu, C.G. Yan, and Z. J. Ding, “A novel method for calculating service reputation,” *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 634–642, Jul. 2013.

[38] S. G. Wang, Z. B. Zheng, Z. P. Wu, M. R. Lyu, and F. Yang, “Reputation measurement and malicious feedback rating prevention in web service recommendation systems,” *IEEE Trans. Services Comput.*, vol. 8, no. 5, pp. 755–767, Sep./Oct. 2015.

[39] Y. Wang and L. Li, “Two-dimensional trust rating aggregations in service-oriented applications,” *IEEE Trans. Services Comput.*, vol. 4, no. 4, pp. 257–271, Oct.-Dec. 2011.

[40] W. Conner, A. Iyengar, T. Mikalsen, . I. Rouvellou, and K. Nahrstedt, “A trust management framework for service-oriented environments,” in *Proc. Int. Conf. World Wide Web*, 2009, pp. 891–900.

[41] E. S. Shamila and V. Ramachandran, “Evaluating trust for web services access,” in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res.*, 2010, pp. 1–8.

[42] H. Y. Shen and G. X. Liu, “An efficient and trustworthy resource sharing platform for collaborative cloud computing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 862–875, Apr. 2014.

Miao Wang received the BS degree in electronic engineering and the MS degree in instructional technology from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, and the PhD degree in computer science from the Chinese Academy of Sciences, Beijing, China, in 2010. She is currently an associate professor in the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include Internet architecture and network security.

Guiling Wang received the BS degree in software from Nankai University, Tianjin, China, in 1999 and the PhD degree in computer science and engineering from Pennsylvania State University, Pennsylvania, in 2006. She is currently a professor in the New Jersey Institute of Technology, Newark, New Jersey. Her research interests include mobile computing, network and systems security.

Yujun Zhang received the BS degree in software from Nankai University, Tianjin, China, in 1999 and the PhD degree in computer science from the Chinese Academy of Sciences, Beijing, China, in 2004. He is currently a professor in the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include future Internet architecture and network security.

Zhongcheng Li received the BS degree in computer science from Peking University in 1983, and the MS and PhD degrees in computer science from Chinese Academy of Sciences, in 1986 and 1991, respectively. He is currently a professor in the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include mobile computing, and network security.