EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

A mechanism for detecting dishonest recommendation in indirect trust computation

Naima Iltaf[1], Abdul Ghafoor[2*] and Uzman Zia[3]

Abstract

Indirect trust computation based on recommendations form an important component in trust-based access control models for pervasive environment. It can provide the service provider the confidence to interact with unknown service requesters. However, recommendation-based indirect trust computation is vulnerable to various types of attacks. This paper proposes a defense mechanism for filtering out dishonest recommendations based on a measure of dissimilarity function between the two subsets. A subset of recommendations with the highest measure of dissimilarity is considered as a set of dishonest recommendations. To analyze the effectiveness of the proposed approach, we have simulated three inherent attack scenarios for recommendation models (bad mouthing, ballot stuffing, and random opinion attack). The simulation results show that the proposed approach can effectively filter out the dishonest recommendations based on the majority rule. A comparison between the exiting schemes and our proposed approach is also given.

Introduction

The rapid development of collaborative, dynamic, and open environments has increased awareness on security issues. It is becoming widely acknowledged that traditional security measures fail to provide the necessary flexibility for interactions between known and unknown entities in an uncertain environment due to statically defined security policies and capabilities [1]. Much research on trust-based access control models for pervasive environment has been carried out [2-6], which use trust as an elementary criterion for authorizing known, partially known, and unknown entities to interact with each other. Indirect trust computation holds key importance in trust- based access control models. When the service provider has no personal experience with the requesting entity to compute direct trust, indirect trust computation is used as a way to evaluate and distribute trust [1]. The basis for indirect trust computation is seeking recommendation for further information to define the trustworthiness of the unfamiliar service requester. It requests recommendation, with respect to the entity in question, from peer services. If peer services provide honest recommendations, a service provider can accurately determine the trustworthiness of an unknown service requester. This gives the service provider the confidence to interact with an unknown service requester [2].

However, reliance on peer services to seek the recommendation of an unfamiliar service requester can lead to erroneous decisions if the recommender provides recommendations that deviate from their experience. The recommenders can falsely provide dishonest recommendation either to elevate trust values of malicious entities or to lessen the trust values of honest entities. If these recommendations are aggregated blindly without filtering false recommendations, they can skew the evaluation of an entity's trustworthiness. Therefore, a mechanism to avoid the influence of dishonest recommendations from malicious recommenders is a fundamental problem for trust models.

Consider the following scenarios that show the importance of recommendation models in a pervasive environment as well as a mechanism to filter dishonest recommendations in such models:

Scenario 1. Bob is an employee of a Paris-based multinational company and, due to official commitments, travels frequently between France and the USA. Bob just arrived at Los Angeles Airport on a

*Correspondence: abdulghafoor-mcs@nust.edu.pk
[2]Department of Electrical Engineering, National University of Sciences and Technology, Rawalpindi 46000, Pakistan
Full list of author information is available at the end of the article

business trip and was consuming his lunch at a cafe when he received a call from his employer to reach Ohio by night to attend an important meeting the next day. Incidentally Bob's travel agent was not available. However, Prime Travels had a seat available to Ohio in their next flight. Using his smart phone, Bob registers himself with the booking service of Prime Travels and generates a request for reservation. Since Bob has never made any reservation with the registration service of Prime Travels before, it broadcasts a recommendation request to the services being offered at the airport to ascertain Bob's trustworthiness. It received recommendations from different services Bob used during his transits at Los Angeles Airport including payment to a cafe, internet access, money transfer to a money exchange service, and reservations made through the registration service of some other travel agents. A few travel agents, competitor to Prime Travels, intentionally responded with bad recommendations. Prime Travels requires a mechanism to filter these dishonest recommendations from the honest one to ascertain the trustworthiness of Bob for his decision making.

Scenario 2. Alice is a frequent visitor of H&M shopping mall near her work place. After office hours, while Alice was visiting the shopping mall, she received a call from her colleague that she has forgotten to mail an important document to one of her customers. Alice needed internet access on her smart phone to mail the document. Alice searches for available internet service providers in the mall and forwards a request to an available wireless hotspot identified as MegaIT in the mall to allow internet access on her device. Since she had never used the service before, MegaIT broadcasts a message to different services available in H&M to ask for recommendations. Since Alice had been a frequent visitor with a history of interactions with other shopping, saloon, and dining services in the mall, these service providers give recommendations to MegaIT. In order to grant access, MegaIT requires some mechanism to determine which recommendations it should use to determine the trustworthiness of Alice.

There can be three possible types of malicious recommendation [7]. Bad mouthing recommendations (BM) are those malicious recommendations that cause the evaluated trustworthiness of an entity to decrease, ballot stuffing (BS) recommendations cause the evaluated trustworthiness of the entity to increase, and random opinion (RO) recommendations are those in which a recommender gives the recommendations randomly opposite the true behavior of the entity in question. In this paper, we propose a new mechanism to filter out dishonest nodes from influencing the indirect trust computation. The proposed mechanism (an extension of [8]) is based on the assumption that a dishonest recommendation is one that is inconsistent with other recommendations and has a low probability of occurrence in the recommendation set. Based on this assumption, a new dissimilarity function for detecting deviations in a recommendation set is defined. An extensive comparison between the proposed and existing techniques is also provided to demonstrate the effectiveness of the proposed mechanism.

Related work

The dynamism of pervasive computing environment allows *ad hoc* interaction of known and unknown autonomous entities that are unfamiliar and possibly hostile. In such environment where the service providers have no personal experience with unknown service requesters, trust and recommendation models are used to evaluate the trustworthiness of unfamiliar entities. Recently, research in designing defense mechanisms to detect dishonest recommendation in these open distributed environments has been carried out [9-26]. The defense mechanisms against dishonest recommendations has been grouped into two broad categories, namely exogenous method and endogenous method [9]. The approaches that fall under endogenous method use other external factors along with the recommendations (reputation of recommender and credibility of recommender) to decide the trustworthiness of the given recommendation. However, these approaches assume that only highly reputed recommenders can give honest recommendations and vice versa. In [10], Xiong and Liu presented an approach (PeerTrust) that avoids aggregation of the individual interactions. Their model computes the trustworthiness of a given peer based on the community feedback about the participant's past behavior. The credibility factor of the feedback source is computed using a function of trust value as its credibility value. The model also incorporates personalized similarity between the experience with other partners for reputation on ranking discrepancy. Chen et al. [11] distinguishes between recommendations by computing the reputation for each recommender. The reputation is measured on the basis of the quality and quantity of recommendation it provides. The recommender's reputation is used as a weight when aggregating the recommendations of all the recommenders. However, the model does not consider the service type of the recommender on which its recommendation is based. Malik and Bouguettaya [12] also proposed using rater credibility for its recommendation assessment. It believes that only highly reputed recommenders can give honest recommendations. These models use other external information

sources to gather the reputations of the recommender. Ganeriwal et al. [13] believe that the weight of its recommendations about others is dependent on its own reputation for service providing. In other words, if it provides a reliable service, then the recommendations it provides is also reliable. In [14], the global reputation of a node is aggregated from local trust scores weighted by the global reputation scores of all senders. Since these models are based on the assumption that entities with high reputation provide honest recommendations, that makes it vulnerable to attack. A smart attacker may behave well for a while to get a high reputation and then provide all dishonest recommendations that cannot be detected by schemes using reputation [15], that is, a recommender can build reputation with different expectations and intentions, and the recommendation they provide can be different from their experience. Recently, models for online communities have proposed using the social element of the recommender as an additional source of information in the recommender system. They believe that people trust their peers with whom they are socially connected and use their recommendations. The main idea behind this approach is that users tend to connect to users with similar preferences. Trusting the opinion of others is based on the social link between the two entities. In [16], the authors presented the correlation between trust and social networks by establishing a rating system for movies based on community system. They demonstrated in their experiments that social trust is able to evaluate similarity in a more distinctive way when the ratings are extreme and with large differences. In [17], the authors have modeled a social network as a directed graph and have evaluated recommendations based on the position and interconnections of the user represented as actors in the graph. The model employs social network analysis metric including centrality and rank prestige to identify the influence of actors in the social network. In [18], a framework to build a recommendation system by identifying a group of experts in a social network is proposed. The model recommends experts with appropriate knowledge based on the information desired by the user. The authors elaborate the efficiency of the proposed approach by applying the model in a research community. In [19], the authors present a probabilistic matrix factorization approach for the recommender system. The model applies trusted friends' opinion in a social network to gather recommendations. The research believes that the user's friend recommendation has an impact on user preferences in a social network. All these models [16-19] believe that there exist social relationships between users in the system that affect the evaluation of recommendation trustworthiness. However, in open spaces comprised of multiple devices (pervasive environment), these devices in close physical proximity form an *ad hoc* network for spontaneous service access [20]. In such an open, dynamic environment where devices are continuously leaving/joining the network, it is difficult to rely on a formal social relationship.

In endogenous method, the recommendation seeker has no personal experience with the entity in question. It relies only on the recommendations provided by the recommender to detect dishonest recommendation. The method believes that dishonest recommendations have different statistical patterns from honest recommendations. Therefore, in this method, filtering of dishonest recommendation is based on analyzing and comparing the recommendations themselves. In trust models where indirect trust based on recommendations is used only once to allow a stranger entity to interact, endogenous method based on the majority rule is commonly used. Dellarocas [21] has proposed an approach based on controlled anonymity to separate unfairly high ratings and fair ratings. This approach is unable to handle unfairly low ratings [22]. In [23], a filtering algorithm based on the beta distribution is proposed to determine whether each recommendation R_i falls between q quartile (lower) and $(1-q)$ quartile (upper). Whenever a recommendation does not lie between the lower and upper quartile, it is considered malicious and its recommendation is excluded. The technique assumes that recommendations follow beta distribution and is effective only if there are effectively a large number of recommendations. Weng et al. in [24] proposed a filtering mechanism based on entropy.The basic idea is that if a recommendation is too different from majority opinion, then it could be unfair. The approach is similar to other reputation-based models except that it uses entropy to differentiate between different recommendations. A context-specific and reputation-based trust model for pervasive computing environment was proposed [25] to detect malicious recommendation based on control chart method. The control chart method uses mean and standard deviation to calculate the lower confidence limit (LCL) and upper confidence limit (UCL). It is assumed that the recommendation values that lie outside the interval defined by LCL and UCL are malicious, therefore discarded from the set of valid recommendations. It considers that a metrical distance exists between valid and invalid recommendations. As a result, the rate of filtering out the false positive and false negative recommendation is really high. Deno et al. [26] proposed an iterative filtering method for the process of detecting malicious recommendations. In this model [26], an average trust value (T_{avg}) of all the recommendations received (T_{R}) is calculated.

The inequality $\mid T_{\text{avg}}(B) - T_{\text{R}}(B) \mid > S$, where B is the entity for which recommendations are collected from i recommenders (R) and S is a predefined threshold in the interval [0 1], is evaluated. If that inequality holds, then the recommendation is false and is filtered out. The

method is repeated until all false recommendations are filtered out. The effectiveness of this approach depends on choosing a suitable value for S. These detection mechanisms can be easily bypassed if a relatively small bias is introduced in dishonest recommendations.

Proposed approach

The objective of indirect trust computation is to determine the trustworthiness of an unfamiliar service requester from the set of recommendations that narrow the gap between the derived recommendation and the actual trustworthiness of the target service. In our approach, a dishonest recommendation is defined as an outlier that appears to be inconsistent with other recommendations and has a low probability that it originated from the same statistical distribution as the other recommendation in the data set. The importance of detecting outliers in data has been recognized in the fields of database and data mining for a long time. The outlier deviation-based approach was first proposed in [27], in which an exact exception problem was discussed. In [8], the author presented a new method for deviation-based outlier detection in a large database. The algorithm locates the outlier by a dynamic programming method. In this paper, we have extended this outlier detection technique to filter out dishonest recommendations. Our approach (Algorithm 1) is based on the fact that if a recommendation is far from the median value of a given recommendation set and has a lower frequency of occurrence, it is filtered out as a dishonest recommendation. Suppose that an entity X requests to access service A. If service A has no previous interaction history with X, it will broadcast the request for recommendations, with respect to X. Let R denote the set of recommendations collected from recommenders.

$$R = \{r_1, r_2, r_3, \ldots\ldots, r_n\}$$

where n is the total number of recommendations. Since smart attackers can give recommendations with little bias to go undetected, we divide the range of possible recommendation values into b intervals (or bins). These bins define which recommendations we consider to be similar to each other such that all recommendations that lie in the same bin are considered alike. b has an impact on the detection rate. If the bins are too wide, honest recommendations might get filtered out as dishonest. On the other hand, if the bins are too narrow, some dishonest recommendations may appear to be honest and vise versa. In this paper, we have tuned b = 10 such that Rc_1 comprises all recommendations that lie between interval [0 0.1], Rc_2 comprises all recommendations between interval [0.1 0.2], and so on for (Rc_3, …, Rc_{10}). After grouping the recommendations in their respective bins, we compute a histogram that shows count f_i of the recommendations falling in each bin. Let H be a histogram of a set of recommendation classes where

$$H(R) = \big\{\langle Rc_1, f_1\rangle, \langle Rc_2, f_2\rangle, \langle Rc_3, f_3\rangle, \langle Rc_4, f_4\rangle, \langle Rc_5, f_5\rangle, \langle Rc_6, f_6\rangle, \langle Rc_7, f_{71}\rangle, \langle Rc_8, f_8\rangle, \langle Rc_9, f_9\rangle, \langle Rc_{10}, f_{10}\rangle\big\}$$

where f_i is the total number of recommendations falling in Rc_i. From this histogram $H(R)$, we remove all the recommendation classes with zero frequencies and get the domain set (Rdomain) and frequency set (f)

$$R\text{domain} = \{Rc_1, Rc_2, Rc_3, \ldots\ldots, Rc_{10}\}$$
$$f = \{f_1, f_2, f_3, \ldots\ldots, f_{10}\}.$$

Definition 1. The dissimilarity function $\mathrm{DF}(x_i)$ is defined as

$$\mathrm{DF}(x_i) = \frac{|x_i - \mathrm{median}(x)|^2}{f_i} \quad (1)$$

Algorithm 1 Recommendation

Require: *Set of Recommendations*
Ensure: $R\text{domain}_{\text{dishonest}}$

for $i = 1 \rightarrow 10$ **do**
  $Rc_i = i/10$
  f_i = number of recommendations in interval $[i/10 - 0.1, i/10]$
end for
for $i = 1 \rightarrow 10$ **do**
  if $f_i <> 0$ **then**
    $R\text{domain}[k] = Rc_i$
    $H[k++] = \{Rc_i, f_i\}$
  end if
end for
$\bar{x} = \mathrm{Median}(R\text{domain})$
for each k in Rdomain **do**

$$\mathrm{DF}[k] = \frac{|R\text{domain}[k] - \bar{x}|^2}{f_k} \quad \text{//calc deviation}$$

end for
$\mathrm{SRdomain} = \mathrm{SortDesc}(R\text{domain}, \mathrm{DF})$
$D_0 = \emptyset$
for j = 1 to size of (SRdomain) - 1 **do**
  $D_j \bigcup (\mathrm{SRdomain}_j)$
  $\mathrm{SF}_k = \mathrm{SmoothingFactor}(D_j)$
end for
$\mathrm{SF}_{\max}$ = max ($\mathrm{SF}(D_k)$)
$f_{\min}$ = min freq of k in SRdomain with $\mathrm{SF} = \mathrm{SF}_{\max}$
$R\text{domain}_{\text{dishonest}}$ = all k in SRdomain with $\mathrm{SF}_k = \mathrm{SF}_{\max}$ and $f_k = f_{\min}$
return $R\text{domain}_{\text{dishonest}}$

where x_i is a recommendation class from a recommendation set x.

Under the proposed approach, the dissimilarity value of x_i is dependent on the square of absolute deviation from the median, i.e., $|x_i - \text{median}(x)|^2$. The median is used to detect deviation because it is resistant to outliers. The presence of outliers does not change the value of the median. In Equation 1, the square of absolute deviation from the median is taken to signify the impact of extremes, i.e., the farther the recommendation value x_i is from the median, the larger the squared deviation is. Moreover, the dissimilarity value of x_i is inversely proportional to its frequency. In Equation 1, $|x_i - \text{median}(x)|^2$ is divided by frequency f_i. In this way, if a recommendation is very far from the rest of the recommendations and its frequency of occurrence is also low, Equation 1 will return a high value. Similarly, if a recommendation is close to the rest of the recommendations (i.e., similar to each other) and its frequency of occurrence is also high, Equation 1 will return a low value.

For each Rc_i, a dissimilarity value is computed using Equation 1 to represent its dissimilarity from the rest of the recommendations with regard to their frequency of occurrence. All the recommendation classes in Rdomain are then sorted with respect to their dissimilarity value $\text{DF}(Rc_i)$ in descending order. The recommendation class at the top of the sorted Rdomain with respect to its $\text{DF}(x_j)$ is considered to be the most suspicious one to be filtered out as dishonest recommendation. Once the Rdomain is sorted, the next step is to determine the set of dishonest recommendation classes from Rdomain set. To help find the set of dishonest recommendation classes from the set of recommendations in Rdomain, Arning et al. [27] defined a measure called smoothing factor (SF).

Definition 2. A SF for each SRdomain is computed as

$$\text{SF}(\text{SRdomain}_j) = C(R\text{domain} - \text{SRdomain}_j) \times (\text{DF}(R\text{domain}) - \text{DF}(\text{SRdomain}_j)) \quad (2)$$

where $j = 1, 2, 3 \ldots, m$, and m is the total number of distinct elements in SRdomain. C is the cardinality function and is taken as the frequency of elements in a set $\{R\text{domain} - \text{SRdomain}_j\}$. The SF indicates how much the dissimilarity can be reduced by removing a suspicious set of recommendation (SRdomain) from the Rdomain.

Definition 3. The dishonest recommendation domain ($R\text{domain}_{\text{dishonest}}$) is a subset of Rdomain that contributes most to the dissimilarity of Rdomain and with the least number of recommendations, i.e., $R\text{domain}_{\text{dishonest}} \subseteq R\text{domain}$. We say that SRdomain_x is a set of dishonest recommendation classes with respect to SRdomain, C, and $\text{DF}(\text{SRdomain}_j)$ if

$$\text{SF}(\text{SRdomain}_x) \geq \text{SF}(\text{SRdomain}_j) \quad x, j \in m$$

for all Rdomain, C, and SRdomain_j.

In order to find out the set of dishonest recommendation $R\text{domain}_{\text{dishonest}}$ from Rdomain, the mechanism defined by the proposed approach is as follows:

- Let Rc_k be the k^{th} recommendation class of Rdomain and SRdomain be the set of suspicious recommendation classes from Rdomain, i.e., SRdomain $\subseteq$ Rdomain.
- Initially, SRdomain is an empty set, $\text{SRdomain}_0 = \{\}$
- Compute $\text{SF}(\text{SRdomain}_k)$ for each SRdomain_k formed by taking the union of SRdomain_{k-1} and Rc_k .

$$\text{SRdomain}_k = \text{SRdomain}_{k-1} \cup Rc_k \quad (3)$$

  where $k = 1, 2, 3 \ldots, m-1$, and m is the distinct recommendation class value number in sorted Rdomain.
- The subset SRdomain_k with the largest $\text{SF}(\text{SRdomain}_k)$ is considered as a set containing dishonest recommendation classes.
- If two or more subsets in SRdomain_k have the largest SF, the one with minimum frequency is detected as the set containing dishonest recommendation classes.

After detecting the set $R\text{domain}_{\text{dishonest}}$, we remove all recommendations that fall under the dishonest recommendation classes.

An illustrative example

To illustrate how our deviation detection mechanism filters out unfair recommendations, this section provides an example that goes through each step of our proposed approach. Let X be a service requester who has no prior experience with service provider A. In order to determine the trustworthiness of X, A will request recommendations from its peer services who have previous interaction with X. Let $R = \{r_1, r_2, r_3, \ldots\ldots, r_n\}$ be a set of recommendations received by $n = 122$ recommenders for service requester R. After receiving the recommendations, they are grouped in their respective bins. Table 1 shows how the received recommendations are grouped in their respective classes. After arranging the recommendations in their respective recommendation class Rc_i, we remove the recommendation classes with zero frequencies and calculate $\text{DF}(Rc_i)$ for each recommendation class using Equation 1. Table 2 shows the sorted list of recommendation classes with respect to their dissimilarity value.

Table 1 Frequency distribution of recommendations

Rc_i	Recommendation value rc_i	Frequency f_i
Rc_1	0.1	41
Rc_2	0.2	23
Rc_3	0.3	37
Rc_4	0.4	0
Rc_5	0.5	0
Rc_6	0.6	0
Rc_7	0.7	0
Rc_8	0.8	13
Rc_9	0.9	8
Rc_{10}	1.0	0

In Table 2 the recommendation class Rc_5 has the highest deviation value, so it is taken as a suspicious recommendation class and is added to the suspicious recommendation domain (SRdomain), and its SF is calculated. Next we take the union of the suspicious recommendation domain SRdomain_1 and the next recommendation class in the sorted list, i.e., Rc_4 and calculate its SF using Equation 2. This process is repeated for each Rc_i of Rdomain until $\text{SRdomain} = R\text{domain} - Rc_m$, where $m = 5$.

Table 3 shows that the SF of SRdomain_2 has the highest value. Therefore, the recommendation classes {0.8, 0.9} in SRdomain_3 are considered as dishonest recommendation classes, and these recommendation classes are removed from the Rdomain.

Performance evaluation of the proposed approach

In this section, we evaluate our model in a simulated multi-agent environment. We carry out different sets of experiments to demonstrate the effectiveness of the proposed model against different attack scenarios (BM attack, BS attack, and RO attack). Results indicate that the model is able to respond to all three types of attack when the percentage of malicious recommenders is varied from 10% to 40%. We have also studied the performance of the model by varying the offset introduced by the malicious recommender in their recommended trust value. It was observed that the performance of the models decreases only when the percentage of malicious recommenders is above 30% and the mean offset between the honest and dishonest recommendation is minimum (0.2).

Table 2 Recommendation classes sorted with respect to their DF

Rc_i	Recommendation value rc_i	Frequency f_i	DF(Rc_i)
Rc_5	0.9	8	0.061249
Rc_4	0.8	13	0.02769
Rc_3	0.3	37	2.7027E−4
Rc_1	0.1	41	2.4390E−4
Rc_2	0.2	23	0.0

Table 3 Smoothing factor computation

SRdomain	Rdomain SRdomain	DF(Rdomain)	SF
{0.9}	{0.8, 0.3, 0.1, 0.2}	0.061	6.9825
{0.9, 0.8}	{0.3, 0.1, 0.2}	0.0889	8.9317
{0.9, 0.8, 0.3}	{0.1, 0.2}	0.0890	5.967
{0.9, 0.8.0.3, 0.1}	{0.2}	0.0894	2.0574

Experimental setup

We simulate a multi-agent environment using AnyLogic 6.4, where agents (offering and requesting services) are continuously joining and leaving the environment. The agents are categorized into two groups, i.e., agents offering services as service provider agents (SPA) and agents consuming services as service requesting agents (SRA). We conduct a series of experiments for a new SPA to evaluate the trustworthiness of an unknown SRA by requesting recommendation from other SPAs in the environment. All SPAs can also act as recommending agents (RA) for other SPAs . The RA gives recommendations, in a continuous range [0 1], for a given SRA on the request of a SPA. The RA can either be honest or dishonest depending on the trustworthiness of its recommendation. An honest RA truthfully provides recommendation based on its personal experience, whereas a dishonest RA insinuates a true experience to a high, low, or erratic recommendation with a malicious intent. The environment is initialized with set numbers of honest and dishonest recommenders ($N = 100$). The simulation is run in steps, the total number of which is defined by *NSTEPS*.

Experiment 1 : validation against attacks

To analyze the effectiveness of the proposed approach, three inherent attack scenarios (bad mouthing, ballot stuffing, and random opinion attack) for recommendation models have been implemented in the above defined simulation environment.

Bad mouthing attack

BM is one in which the intention of the attacker is to send malicious recommendations that will cause the evaluated trustworthiness of an entity to decrease. Let us suppose that the service provider asks for recommendations regarding an unknown service requester A. In this experiment we assume that a certain percentage of the recommenders are dishonest and launch a BM attack against (A) by giving dishonest recommendations. It is assumed that the actual trust value of A is 0.7. At the

initial step of the simulation, the environment has 10% dishonest RA who attempt to launch a bad mouthing attack against A by providing low recommended trust values (between the range [0 0.3]). To elaborate the efficacy of the proposed approach, we vary the percentage of dishonest recommenders from 10% to 40%. Figure 1a,b,c,d shows the SF calculated for each SRdomain. It is shown that in each case the proposed approach is able to detect the set of bad mouthers giving low recommendation between 0.1 and 0.3. For example, in Figure 1a when the percentage of dishonest recommenders is 10%, the SRdomains and respective SF values are as follows:

SRdomain 1 {0.1},	8.64
SRdomain 2 {0.1, 0.2, 0.3, 0.8},	13.62
SRdomain 3 {0.1, 0.2},	16.12
SRdomain 4 {0.1, 0.2, 0.3, 0.8, 0.6},	6.82
SRdomain 5 {0.1, 0.2, 0.3},	20.4

Since the SF of SRdomain_5 has the highest value, the recommendation classes {0.1, 0.2, 0.3} are considered as dishonest recommendation classes, and the recommendations that belong to these recommendation classes are considered as dishonest recommendations.

Ballot stuffing attack

BS is one in which the intention of the attacker is to send malicious recommendations that will cause the evaluated trustworthiness of an entity to increase. Let us suppose that the service provider asks for recommendations regarding an unknown service requester B. It is assumed that the actual trust value of B is 0.3. A certain percentage of recommenders providing the recommendation to the service provider are dishonest and gives a high recommendation value between 0.8 to 1.0, thus launching a BS attack. We evaluate the proposed approach by varying the percentage of dishonest recommenders from 10% to 40%. Figure 1e,f,g,h shows the SF values for SRdomains in each case. It is evident from the results that the model is able to detect dishonest recommendations even when the percentage of dishonest recommendations is 40%. From Figure 1h (when the percentage of dishonest recommendations is 40%), the SF values of each SRdomain are as follows:

SRdomain 1 {1.0, 0.9},	5.038
SRdomain 2 {1.0, 0.9, 0.8, 0.1, 0.2},	1.843
SRdomain 3 {1.0, 0.9, 0.8},	5.47
SRdomain 4 {1.0},	4.41
SRdomain 5 {1.0, 0.9, 0.8, 0.1},	3.667

The proposed approach is able to detect the dishonest recommendations as SRdomain 3 with the highest SF value of 5.47.

Random opinion attack

RO attack is one in which the malicious recommender gives the recommendations randomly opposite the true behavior of the entity in question. Let us suppose that the recommenders launch a RO attack while providing recommendations for a service requester C. The dishonest recommenders provide either very low recommendations (0.1 to 0.2) or very high recommendations (0.8 to 1.0). We vary the percentage of dishonest recommenders from 10% to 40% for the experiment. The SF values for the respective SRdomains in each case are shown in Figure 1i,j,k,l. The proposed approach successfully detects random opinion attack and is able to filter out the dishonest set of recommenders in each case.

Experiment 2: validation against deviation

The detection rate of unfair recommendations by varying the number of malicious recommenders cannot fully describe the performance of the model as the damage caused by different malicious recommenders can be very different depending on the disparity between the true recommendation and unfair recommendation(i.e., offset). The offset introduced by the attackers in the recommended trust value is a key factor in instilling deviation in the evaluated trust value of SRA. We have carried out a set of experiments to observe the impact of different offset values introduced by different malicious recommenders on the final trust value. We define mean offset (MO) as the difference between the mean of honest recommendations and the mean of dishonest recommendations. For the experiment, we have divided MO into four different levels $L_1 = 0.2$, $L_2 = 0.4$, $L_3 = 0.6$, and $L_4 = 0.8$. It is assumed that the actual trust value of SRA is 0.2, and the dishonest recommender's goal is to boost the recommended trust value of SRA (BS attack). The experiment was conducted in four different rounds by varying the MO level from L_4 to L_1 (i.e., from maximum to minimum). In each round, the recommended trust value is computed with different percentages of dishonest recommenders (10%, 20%, 30%, and 40%).

Figure 2 shows the performance of the proposed approach during each round of the experiment. The results in Figure 2a,b show that when the MO level is high (L_3 and L_4), the proposed approach computes the actual recommended trust value accurately for all percentages of dishonest recommenders. However, in Figure 2c,d, when the MO level (L_1 and L_2) is low, the detection rate of the proposed approach deteriorates slightly because the dishonest recommendations are very close to the honest recommendations. However, it is also observed that even though the detection rate is low due to less MO between honest and dishonest recommendations, the damage caused by undetected dishonest recommendations is very low. The largest damage was observed when

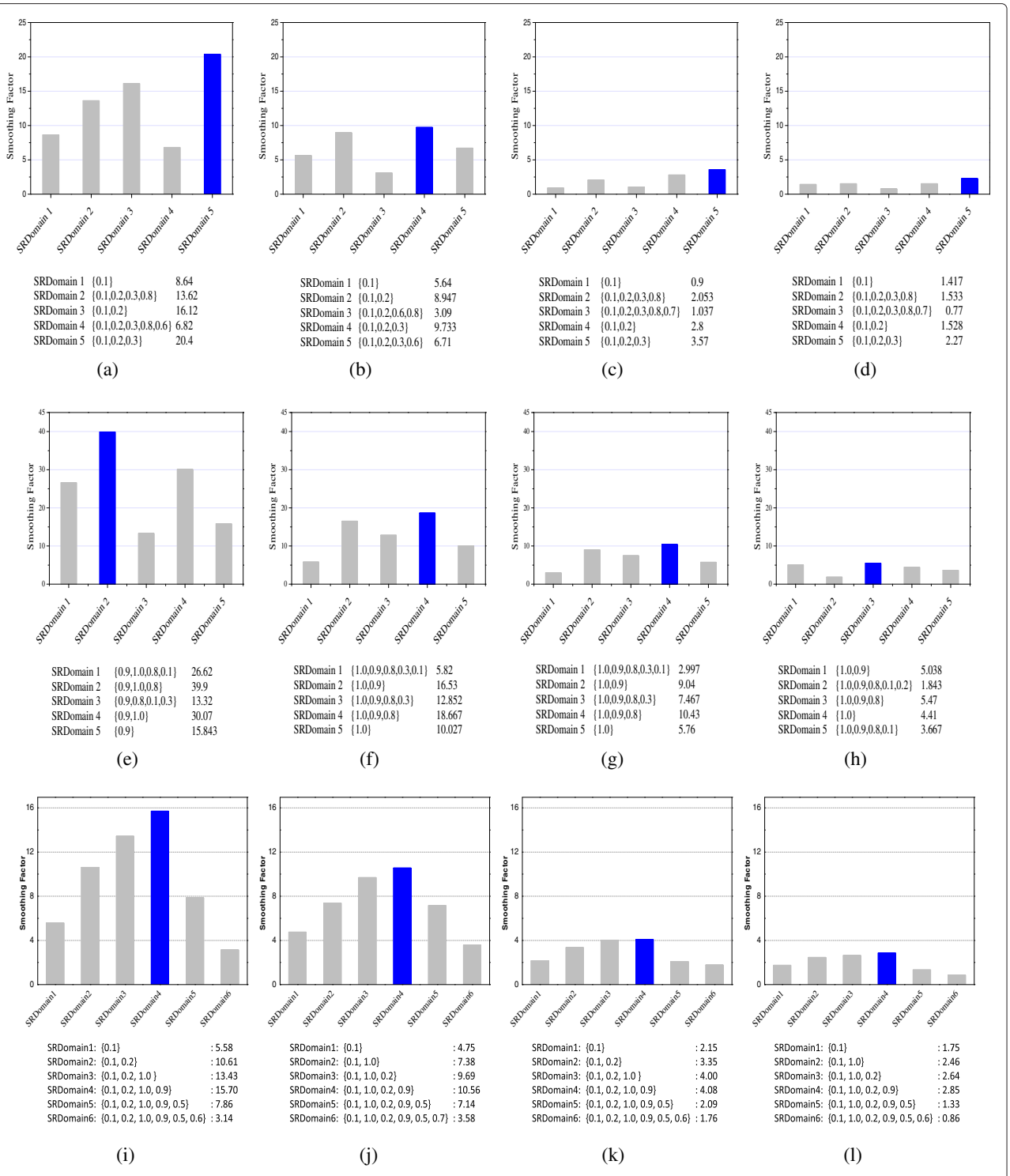

Figure 1 Detecting attack. (a) BM, 10% dishonest recommender. **(b)** BM, 20% dishonest recommender. **(c)** BM, 30% dishonest recommender. **(d)** BM, 40% dishonest recommender. **(e)** BS, 10% dishonest recommender. **(f)** BS, 20% dishonest recommender. **(g)** BS, 30% dishonest recommender. **(h)** BS, 40% dishonest recommender. **(i)** RO, 10% dishonest recommender. **(j)** RO, 20% dishonest recommender. **(k)** RO, 30% dishonest recommender. **(l)** RO, 40% dishonest recommender.

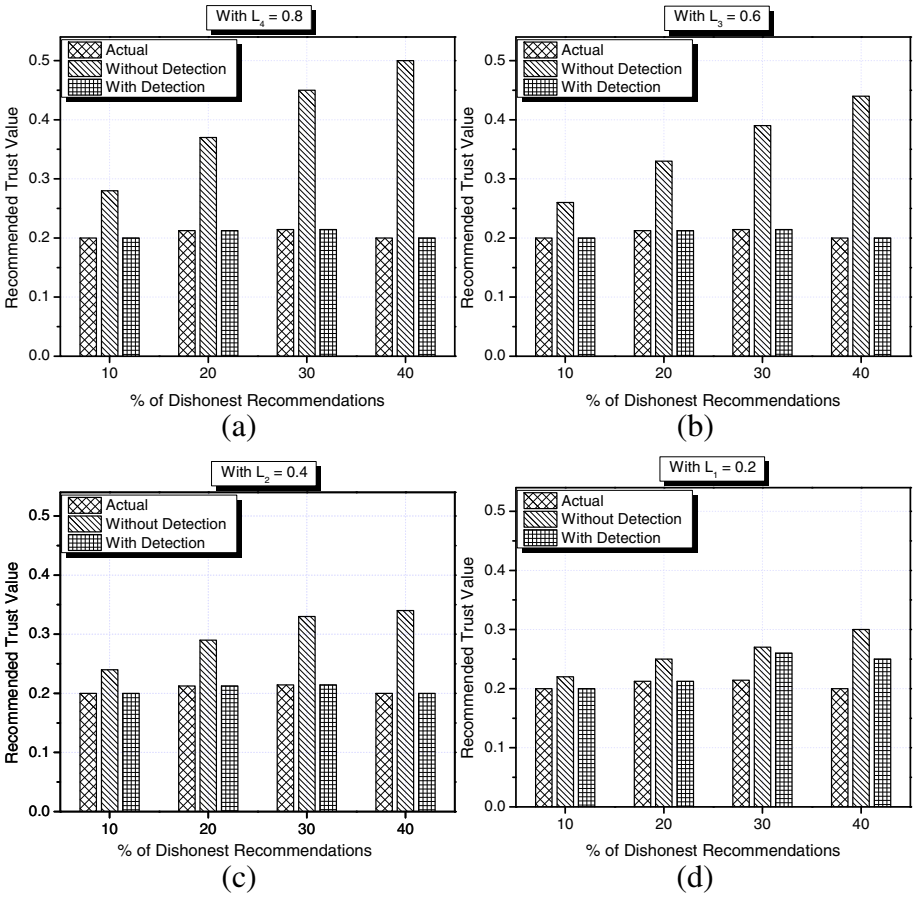

Figure 2 Accuracy by varying offset. Mean offset at **(a)** L_4, **(b)** L_3, **(c)** L_2, and **(d)** L_1.

the percentage of dishonest recommenders is 40% and the mean offset is 0.2 (Figure 2d). In this case, the bias introduced by the undetected dishonest recommendation in recommended trust value ($0.25 - 0.2 = 0.05$) is a very low value and does not have much impact on the final recommended trust value.

Comparative experiments

In this section, we focus on the comparative analysis on our proposed approach with other competing approaches. Since the proposed approach is an extension of [8], we have carried out a set of experiments to demonstrate the improved performance of the proposed approach as compared to [8] termed as the base model. The experimental results substantiate the enhanced capability of the proposed approach to detect dishonest recommendations by varying MO and the percentage of dishonest recommendations. On the contrary, the performance of the base model degrades considerably as compared to the proposed approach. In the literature, many approaches have been proposed to evaluate accurate recommended trust value in the presence of dishonest recommendations. We compare the performance of our proposed approach with those of Ahamed et al. [25], Whitby et al. [23], and Deno et al. [26]. These three models utilize endogenous approach based on majority rule to evaluate the recommended trust value and are, therefore, comparable in their capability and performance with the proposed approach.

Comparison with the base model

In the last section, it was observed that MO and the number of dishonest recommenders play a vital role in introducing deviation in the recommended trust value. The efficiency of the proposed approach has been established through a series of experimental results. In order to further elucidate the performance of the proposed approach, a comparative analysis between the proposed approach and the base model [8] was carried out. It has already been established that dishonest recommendations are difficult to detect when either the percentage of the dishonest recommenders is high or the MO level is very low. Therefore, in this experiment we have simulated two

scenarios: (1) the MO level is set very low (L_1 = 0.2), and the percentage of dishonest recommenders is varied from 10%to 48%; (2) the MO level is kept very high (L_4 = 0.8) while varying the percentage of dishonest recommenders from 10%to 48%. The experiment was conducted for 50 simulation runs, each time with different randomly generated data set of honest and dishonest recommendations. Figure 3 shows the average detection rate of the proposed approach and base model observed during each round of the experiment. Figure 3a shows that the proposed approach can accurately detect dishonest recommendations when their percentage is less than 36%. Even when the percentage of dishonest recommenders is 48%, the detection rate is higher than 70%, whereas the base model is unable to detect all dishonest recommenders even when the percentage of dishonest recommendations is as low as 10%. Moreover, Figure 3b shows that at high MO level (L_4), the detection rate of the base model drastically falls as the percentage of dishonest recommendations exceeds 28%. On the contrary, the performance of the proposed approach remains 100% when the percentage of dishonest recommenders is less than 50%.

Comparison with existing approaches

To illustrate the effectiveness of the proposed deviation-based approach in detecting dishonest recommendations, we have compared our approach with other approaches proposed in the literature based on quartile [23], control limit chart [25], and iterative filtering [26] to detect dishonest recommendations in indirect trust computation. A set of experiments has been carried out by applying the approaches to detect dishonest recommendations in two different scenarios. For the first set of experiments, we assume that a certain percentage of the recommenders are dishonest and launch bad mouthing attack by giving recommendations between 0.1 to 0.3. For the second set of experiments, the dishonest recommenders are assumed to give a high recommendation value between 0.8 to 1.0, thus launching a ballot stuffing attack. In both set of experiments, the percentage of dishonest recommenders is varied from 10% to 45%. For comparison, we have used Matthews correlation coefficient (MCC) to measure the accuracy of all four approaches in detecting dishonest recommendations [28]. MCC is defined as a measure of the quality of binary (two-class) classifications. It takes into account true and false positives and negatives. The formula used for MCC calculation is

$$\mathrm{MCC} = \frac{(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FP} \times \mathrm{FN})}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. MCC returns a value between −1 and 1 (1 means perfect filtering, 0 indicates no better than random filtering, and −1 represents total inverse filtering). To avoid infinite results while calculating MCC, it is assumed that if any of the four sums (TP, FP, TN, and FN) in the denominator is zero, the denominator is arbitrarily set to one.

The Figure 4 shows the comparison of MCC values of the proposed approach with different models with varying percentage of dishonest recommendations (from 10% to 4%). According to the results, the proposed approach can effectively detect dishonest recommendations evident from a constant MCC of +1 for both sets of experiments. On the other hand, in [25], in the case of bad mouthing attack (Figure 4a), MCC increases slowly as the percentage of dishonest recommenders increases from 10% to 30% but then decreases promptly to 0 as the percentage of dishonest recommender increases from 30% to 45%. The same behavior was observed in the case of ballot stuffing attack (Figure 4b). In [26], when the percentage of dishonest recommender increases to 40%, the MCC rate starts to decrease as well. Thus, all three approaches ([25,26], and

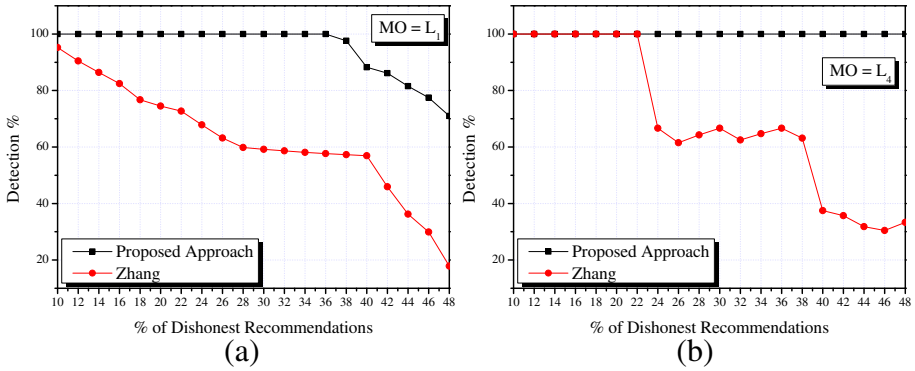

Figure 3 Comparison between the base model and proposed model. (a) Mean offset is low. **(b)** Mean offset is high.

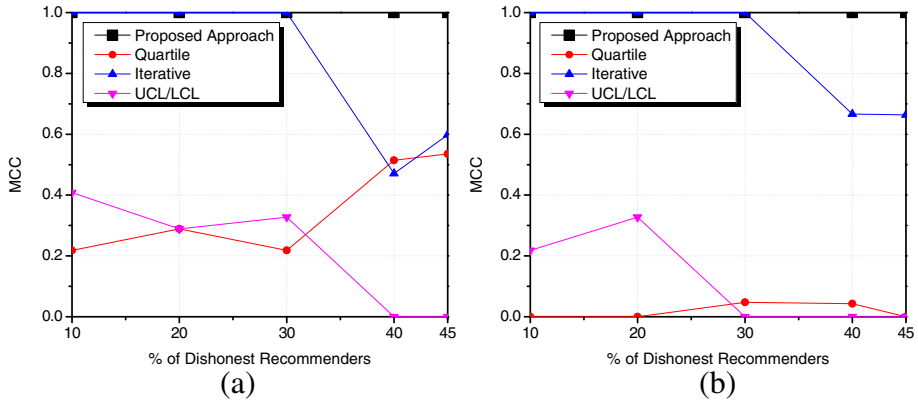

Figure 4 Filtering accuracy in terms of MCC. (a) Bad mouthing attack and **(b)** ballot stuffing.

[23]) fail to achieve perfect filtering of dishonest recommendation as the percentage of dishonest recommenders increases.

For an in-depth analysis of [25,26], and [23], false positive rate (FPR) and false negative rate (FNR) are computed for using the following equations:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

and

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}}$$

The value of FPR and FNR lies between [0 1]. The lower value of FPR and FNR indicates better performance. Figure 5a shows the comparison of FNR and FPR of [25] with the proposed approach based on the results accumulated after the experiments for BM attack. Although the FPR of [25] remains consistent at zero when the percentage of dishonest recommendations is increased from 10% to 40%, at the same time, its FNR progressively increases and reaches its maximum value at 40%. Similarly, Figure 5b shows that [26] maintains zero FPR and FNR until dishonest recommenders are less than 30% of the total recommenders. However, as the number of dishonest recommenders increases above 30%, the model behaves poorly by showing a rapid increase in FPR and FNR. Figure 5c shows that although the FPR of [23] improves as the percentage of dishonest recommenders increases, simultaneously, the FNR starts to grow rapidly for percentages greater than 20%. On the contrary, the proposed approach maintains zero FNR and FPR even when the percentage of dishonest recommenders reaches 40%. Figure 6a explicates the results observed from the performance of [25] under ballot stuffing attack. The approach maintains zero FPR throughout the experiment; however, it filtered out a high number of honest recommenders as dishonest, evident from the high FNR. Similarly, the performance of [26] remains stable until the percentage of dishonest recommenders remains below 30% (Figure 6b). However, the approach also shows a rapid growth in FNR as the percentage of dishonest

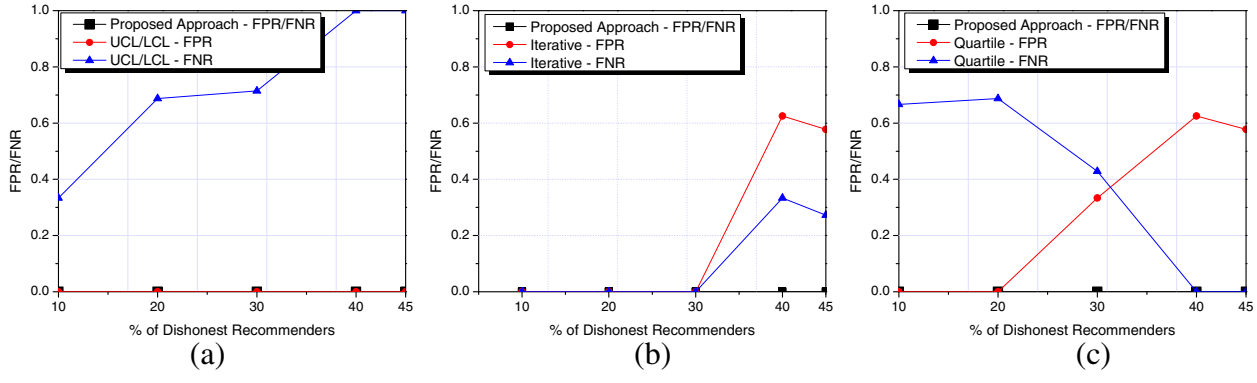

Figure 5 FPR vs FNR (in bad mouthing attack). (a) Control limit charts. **(b)** Iterative filtering. **(c)** Quartile.

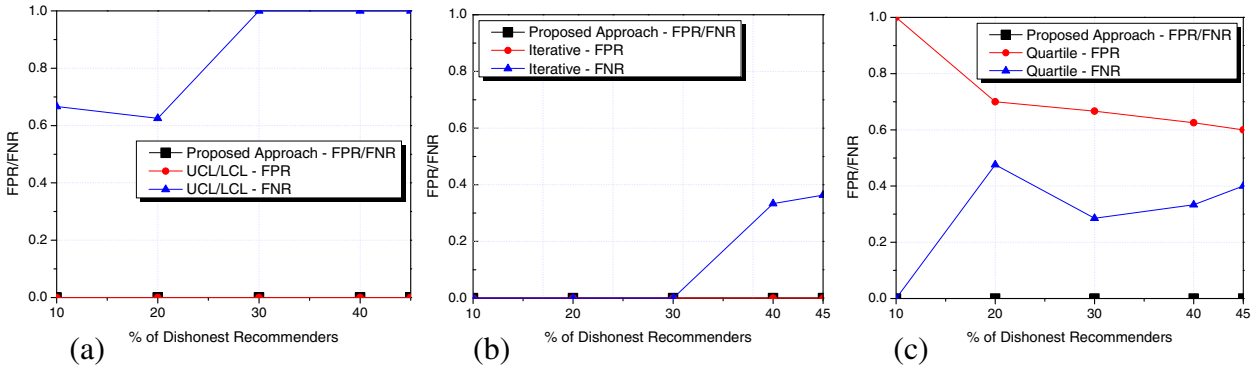

Figure 6 FPR vs FNR (in ballot stuffing attack). (a) Control limit charts. **(b)** Iterative filtering. **(c)** Quartile.

recommenders increases above 30%. Figure 6c shows that [23] is completely unable to detect ballot stuffing. The approach shows a high FPR even at low percentages of dishonest recommenders. It can be seen from the results of Figures 5 and 6 that the proposed approach remains resistant to the attack under both experiments (as the FBR and FNR remains zero), thus outperforming other approaches.

From the above discussion, we can conclude that both [25] and [23] perform poorly in the presence of increasing percentage of dishonest recommenders. It is also observed that [26] performs well provided that the recommendation threshold is selected appropriately. On the contrary, the proposed approach is not reliant on any external parameter and is able to detect 100% dishonest recommenders provided that they are in the minority (<50%).

Conclusions

A mechanism for detecting dishonest recommendation in indirect trust computation is proposed. The main focus in the present work was to detect dishonest recommendations based on their dissimilarity value from the complete recommendation set. Since median is resistent to outlier, we have proposed a dissimilarity function that captures how dissimilar a recommendation class is from the median of the recommendation set. The algorithm uses a smoothing factor which detects malicious recommendations by evaluating the impact on the dissimilarity metric by removing a subset of recommendation classes from the set of recommendations.

Experimental evaluation shows the effectiveness of our proposed method in filtering dishonest recommendations in comparison with the base model. Results show that the proposed method is successfully able to detect dishonest recommendations by utilizing absolute deviation from the median as compared to the base technique which tends to fail as the percentage of dishonest recommendations increases. We have carried out a detailed comparative analysis with the base approach by varying the percentage and the offset introduced by the dishonest recommendations. Results that indicate improved performance of the proposed approach, which is able to produce 70% detection rate at a minimum offset of 0.2, have been shown. On the contrary, the base approach is unable to detect any dishonest recommendations at all. It is also shown that for different attacks (bad mouthing, ballot stuffing, and random opinion attack), the proposed method successfully filters out dishonest recommendations. A comparison between existing approaches and the proposed approach is also presented, which clearly shows the better performance of the proposed approach. In our future work, we will study the possibility of incorporating the proposed approach to existing reputation models that make decision on the basis of recommendations.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Computer Software Engineering, National University of Sciences and Technology, Rawalpindi 46000, Pakistan. [2]Department of Electrical Engineering, National University of Sciences and Technology, Rawalpindi 46000, Pakistan. [3]Department of Computer Software Engineering, Center of Advanced Studies in Engineering, Islamabad 46000, Pakistan.

Received: 16 December 2012 Accepted: 28 June 2013
Published: 13 July 2013

References

1. W Wagealla, M Carbone, C English, S Terzis, P Nixon, A formal model on trust lifecycle management, in *Workshop on Formal Aspects of Security and Trust*, (Pisa, 9–12 September 2003), pp. 184–195
2. C English, W Wagealla, P Nixon, S Terzis, A McGettrick, H Lowe, Trusting collaboration in global computing, in *1st International Conference on Trust Management* (Springer, Hiedelberg, 2003), pp. 136–149
3. B Shand, N Dimmock, J Bacon, Trust for ubiquitous,transparent collaboration, in *Proceedings of the First IEEE International Conference on*

Pervasive Computing Communications (IEEE, Los Alamitos, 2003), pp. 153–160
4. MK Deno, T Sun, Probabilistic trust management in pervasive computing. IEEE/IFIP Int. Conf. Embedded Ubiquitous Comput. **2**, 610–615 (2008)
5. N Iltaf, A Ghafoor, M Hussain, Modeling interaction using trust and recommendation in ubiquitous computing environment. EURASIP J. Wireless Commun. Netw. **2012**, 119 (2012)
6. F Almenarez, A Marin, D Diaz, A Cortes, C Campo, C Garcia, Trust management for multimedia P2P applications in autonomic networking. Adhoc Netw. **9**, 687–690 (2011)
7. K Hoffman, D Zage, C Nita-Rotaru, A survey of attack and defense techniques for reputation systems. ACM Comput. Surv. **42**(1), 1–31 (2009)
8. Z Zhang, X Feng, New methods for deviation-based outlier detection in large database, in *Sixth International Conference on Fuzzy Systems and Knowledge Discovery* (IEEE, Los Alamitos, 2009), pp. 495–499
9. A Josang, R Ismail, C Boyd, A survey of trust and reputation systems for online service provision. Decis. Support Syst. **43**(2), 618–644 (2007)
10. L Xiong, L Liu, Peertrust: supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans. Knowl. Data Engr. **16**(7), 843–857 (2004)
11. M Chen, JP Singh, Computing and using reputations for internet ratings, in *3rd ACM Conference on Electronic Commerce* (ACM, New York, 2001), pp. 154–162
12. Z Malik, A Bouguettaya, Evaluating rater credibility for reputation assessment of web services, in *8th International Conference on Web Information Systems Engineering* (Springer, Heidelberg, 2007), pp. 38–49
13. S Ganeriwal, LK Balzano, MB Srivastava, Reputation-based framework for high integrity sensor networks. ACM Trans. Sensor Netw. **4**, 1–37 (2008)
14. R Zhou, K Hwang, Powertrust: a robust and scalable reputation system for trusted peer-to-peer computing. IEEE Trans. Parallel Distributed Syst. **18**(4), 460–473 (2007)
15. X Liu, A Datta, H Fang, J Zhang, Detecting imprudence of reliable sellers in online auction sites, in *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications* (IEEE, Los Alamitos, 2012), pp. 246–253
16. C Ziegler, J Golbeck, Investigating interactions of trust and interest similarity, Decision Support Systems **43**(2), 460–475 (2007). doi:10.1016/j.dss.2006.11.003
17. I Varlamis, M Eirinaki, M Louta, A study on social network metrics and their application in trust networks, in *2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (IEEE, Los Alamitos, 2010), pp. 168–175
18. E Davoodi, M Afsharchi, K Kianmehr. 7th International Conference on Hybrid Artificial Intelligent Systems, Salamanca, March 2012, A social network-based approach to expert recommendation system, in *Hybrid Artificial Systems* (Springer, Heidelberg, 2012), pp. 91–102
19. H Ma, I King, M Lyu, Learning to recommend with social trust ensemble, in *32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM, New York, 2006), pp. 203–210
20. F Almenarez, A Marin, D Diaz, A Cortes, C Campo, C Garcia, Managing ad-hoc trust relationships in pervasive computing environments, in *Proceedings of the Workshop on Security and Privacy in Pervasive Computing, SPPC'04*, (Vienna, 20 April 2004)
21. C Dellarocas, Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior, in *2nd ACM Conference on Electronic Commerce* (ACM, New York, 2000), pp. 150–157
22. S Liu, J Zhang, C Miao, Y Theng, A Kot, An integrated clustering-based approach to filtering unfair multi-nominal testimonies. Comput. Intell. (2012). http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8640.2012.00464.x/full
23. A Whitby, A Josang, J Indulska, Filtering out unfair ratings in Bayesian reputation systems, in *3rd International Joint Conference on Autonomous Agents and Multi Agent Systems* (IEEE, Washington, 2005), pp. 106– 117
24. J Weng, C Miao, A Goh, An entropy-based approach to protecting rating systems from unfair testimonies. IEICE Trans. Inf. Syst. **89**(9), 2502–2511 (2006)
25. SI Ahamed, M Haque, M Endadul, F Rahman, N Talukder, Design, analysis, and deployment of omnipresent formal trust model (FTM) with trust bootstrapping for pervasive environments. J. Syst. Software. **83**(2), 253–270 (2010)
26. MK Deno, T Sun, I Woungang, Trust management in ubiquitous computing: a Bayesian approach. Comput. Commun. **34**(3), 398–406 (2011)
27. A Arning, R Agrawal, P Raghavan, A linear method for deviation detection in large databases, in *2nd International Conference on Data Mining and Knowledge Discovery* (AAAI, Portland, 1996), pp. 164–169
28. BW Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta. **405**, 442–451 (1975)

doi:10.1186/1687-1499-2013-189
Cite this article as: Iltaf *et al.*: **A mechanism for detecting dishonest recommendation in indirect trust computation.** *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:189.