Information Sciences 306 (2015) 150–165

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A novel item anomaly detection approach against shilling attacks in collaborative recommendation systems using the dynamic time interval segmentation technique

Hui Xia [a,*], Bin Fang [a], Min Gao [b], Hui Ma [a], Yuanyan Tang [c], Jing Wen [a]

[a] School of Computer Science, Chongqing University, Chongqing 400044, China
[b] School of Software Engineering, Chongqing University, Chongqing 400044, China
[c] Department of Computer and Information Science, Macao University, Macao

ARTICLE INFO

Article history:
Received 17 April 2014
Received in revised form 2 February 2015
Accepted 9 February 2015
Available online 16 February 2015

Keywords:
Anomaly detection
Skewness
Time interval
Personalized recommendation
Stability

ABSTRACT

Various types of web applications have gained both higher customer satisfaction and more benefits since being successfully armed with personalized recommendation. However, the increasingly rampant shilling attackers apply biased rating profiles to systems to manipulate item recommendations, which not just lower the recommending precision and user satisfaction but also damage the trustworthiness of intermediated transaction platforms and participants. Many studies have offered methods against shilling attacks, especially user profile based-detection. However, this detection suffers from the extraction of the universal feature of attackers, which directly results in poor performance when facing the improved shilling attack types. This paper presents a novel dynamic time interval segmentation technique based item anomaly detection approach to address these problems. In particular, this study is inspired by the common attack features from the standpoint of the item profile, and can detect attacks regardless of the specific attack types. The proposed segmentation technique could confirm the size of the time interval dynamically to group as many consecutive attack ratings together as possible. In addition, apart from effectiveness metrics, little attention has been paid to the robustness of detection methods, which includes measuring both the accuracy and the stability of results. Hence, we introduced a stability metric as a complement for estimating the robustness. Thorough experiments on the MovieLens dataset illustrate the performance of the proposed approach, and justify the value of the proposed approach for online applications.

1. Introduction

The explosive growth of online resources (including information and products) has resulted in an excessive number of irrelevant or unnecessary options for people [21,40], although they are processed by accessing and retrieving techniques. Personalized recommendation, especially the collaborative filtering (CF)-based mechanism, has been successfully introduced to filter out irrelevant resources [1,18,22,32,34,40,46,48,54,56] and has been widely accepted in many different domains, such as auxiliary teaching [14], online learning [15,19,37], movie and TV programs [6,36], tourism [9,24], online social networks (including communities) [3,30,55,58], digital libraries [49,52], and technology transfer offices [45].

* Corresponding author. Tel.: +86 18512387460.
E-mail address: summertulip@126.com (H. Xia).

http://dx.doi.org/10.1016/j.ins.2015.02.019
0020-0255/

The majority of CF-based recommendation systems rely on opinions from user to item, which are expressed in the form of rating [1,22,32,40,54] and are totally vulnerable to shilling attacks [39,42,43] designed to increase/reduce the probability of the target item being recommended by inputting certain amounts of fake rating profiles, so that the attackers can benefit [25,39]. Typically, some profit-driven raters (i.e., item providers) may inject a great deal of positive ratings to promote the reputation of their own items and negative ratings to undermine their competitors. It appears that shilling attacks are emerging as a great threat for the recommendation system because they can generate large volumes of useless information, mislead review comments, and finally successfully change recommendation results.

There are various types of solutions against shilling attacks for the CF algorithm, and the most common way is to detect the fake user profile, that is, finding out the malicious user directly through the features of attack types [8,10–13,17,28,33,38,39,61]. However, many models are limited to certain attack types, the features of which have been extracted explicitly or scrutinized by researchers [8,25,39]. In addition, the majority of approaches belong to "anomaly user detection" rather than "attacker detection" because the generated anomaly user could be genuine. For example, a captious but authentic user may be classified as an anomaly user by the detection method if he/she usually gives low scores to dissatisfied items, the qualities of which are actually high and could satisfy other people. Actually, many studies neglect the difference between "anomaly user" and "attacker", although it may influence the misclassification rate or false alarm rate, as some genuine users are misclassified as attackers [38].

To solve these problems, we proposed to detect anomaly items directly, which is equal to finding out items attacked by fake profiles directly. This is because the basic assumption of an item is that its intrinsic quality follows the uniform distribution [27]; the resulting rating distribution of this item remains stable without attack ratings. Once it changes greatly, the item is definitely considered under attack. In addition, this approach is generally effective for nearly all attack types, as all effective attacks must change the statistical characteristics of the target item along with the underlying intention of the attackers. For instance, to improve the recommending possibility of one item, large numbers of extremely high ratings must be injected for that item, and the following mean and mode rating values of that item definitely increase. Hence, we could detect any attack regardless of the specific attack type through indications of the changes in rating distribution.

An additional point is many attacks are short periods so that the attackers could maximum their profits, which means that attack ratings in nearly all time-ordered rating sequences of target items must be close to each other or even neighbors. We then proposed a dynamic time segmentation technique to divide the whole rating series into several time intervals and gather together as many attack ratings as possible, which lowers the computational cost and can be applied online effectively.

The key point of any detection method is the performance, and the most popular aspect is the accuracy [7,10–12,17,23,28,38,39,62]. However, we believe that there are other important aspects of detection algorithm performance aside from the accuracy that have been largely overlooked in the current literature. In particular, we introduce a new stability metric for a complementary assessment of the robustness of the detection algorithm inspired by [2,41] because the robustness of the detection algorithm should include two aspects: the first is the accuracy, and the second is the stability.

The rest of this paper is organized as follows. Section 2 briefly discusses the related work on shilling attacks and commonly used detection methods. In Section 3, we elaborate the intrinsic features and the categories of shilling attacks through the perspective of the item profile, and we also offer a comprehensive description of the stability metric. Section 4 lists our anomaly detection method. Next, in Section 5, we experiment and analyze the performance of the proposed algorithm in three aspects: the effectiveness, the robustness and the timeliness. Finally, we present our paper's conclusions and note directions for future work in Section 6.

2. Related work

2.1. Shilling attack types

There are two categories in shilling attacks concerning attack intention: push attacks and nuke attacks [25,26,39,64]. Attacks that intend to increase the reputation of some targeted items are referred to as push attacks, while others aiming to decrease the popularity of the targeted items are known as nuke attacks. Gunes et al. [25] indicated several widely used shilling attack types based on the research of Mobasher et al. [39], as displayed in Table 1. We make a light modification by combining bandwagon and reverse bandwagon attack types together, as they are close to each other in terms of attack strategies, and extending love/hate attacks to push items [39]. I_S is a set of selected items that have some relationships with target items. I_F is a set of randomly selected filler items, and i_t is the target item given with fake rating, usually $r_{\max}$ for push attacks and $r_{\min}$ for nuke attacks. The remaining items are left unrated and indicated as I_Φ.

In addition to these attack types, other attacks that could avoid detection schemes have been proposed [16], such as obfuscated attacks [59], average over popular items (AoP) [28], and mixed attack types [8].

2.2. Techniques against shilling attacks

One effective way against shilling attacks is to decrease the negative influence of attack profiles by improving the robustness of recommendation algorithms. Some CF algorithms are enhanced by semantic analysis [39], and some alleviate

Table 1
Attack profile summary [25,39], r_{max} is the maximum rating, r_{min} is the minimum rating.

Attack type	Random	Average	Probe (informed)	Bandwagon/reverse bandwagon
I_S	Not used (NU)	NU	RC; system response	Popular/unpopular items; r_{max}
I_F	Randomly chose (RC); system mean (SM)	RC; item mean (IM)	Seed items; true preferences	RC; SM
I_φ	$I - I_F$	$I - I_F$	$I - \{I_F \cup I_S\}$	$I - \{I_F \cup I_S\}$
i_t	r_{max}/r_{min}	r_{max}/r_{min}	r_{max}/r_{min}	r_{max}/r_{min}
	Segment	Love/hate	Hybrid	Consistency
I_S	Segmented items; r_{max}	NU	Popular/known average items; r_{max}/item average	Favorite items of a user; r_{max}
I_F	RC; r_{min}	RC; r_{max}	RC; SM	$I - I_S$; Random
I_φ	$I - \{I_F \cup I_S\}$	$I - \{I_F \cup I_S\}$	$I - \{I_F \cup I_S\}$	φ
i_t	r_{max}	r_{max}/r_{min}	r_{max}/r_{min}	r_{max}/r_{min}

negative impact by the trust and reputation [29,31,63] mechanism, which is available through Bayesian inference [35,50,51,53], belief propagation, and matrix factorization [4,5]. However, little can be learned about the fake rating profile through this solution. Hence, many more researchers pay attention to detecting malicious users and attacked items.

The former user-based solution tries to separate the shilling attack profiles from genuine ones with intelligent learning algorithms, including supervised classification, semi-supervised classification, and unsupervised clustering. Many studies employed kNN, C4.5, and SVM classifiers in supervised classification to detect attacks using popular attributes, such as rating deviation from mean agreement (RDMA), weighted degree of agreement (WDA), filler mean target difference (FMTD), and target model focus (TMF) [11,12,39,60]. Then, the useful attributes are extended to three types, generic attributes, model attributes and intraprofile attributes [25,39], to detect more attack types. All existing attributes are contained in one semi-supervised way to train unlabeled profiles and detect fake profiles [13,61]. Many unsupervised algorithms have been used in detection such as the k-means clustering approach [8] and the principal component analysis (PCA)-based variable selection clustering method [16,28,38], which has been proven to be preferable to probabilistic latent semantic analysis (PLSA)-based clustering method [38]. Recently, Lee et al. [33] proposed a new detector using a clustering method and the Group RDMA (GRDMA) metric. Actually, shilling attack models in disguise [8,33,38] are still difficult for user anomaly detection because, for existing attack attributes, it is difficult to separate the attack profiles from the normal ones.

The latter item-based detection type manages the problem by separating the attacked items, as well as their attacked time intervals. Statistical anomaly detection is one such approach and relies on two statistical control techniques, the X-Bar control limit and confidence interval control limit [7]. An item is considered suspicious if its average value falls outside of the confidence level. Similarly, Zhang et al. [62], quantified the changes in sample average and sample entropy caused by an attack event to detect shilling attacks. Recently, an attack model-free detection algorithm using chi-square distribution (χ^2) was introduced to compare the distributions of ratings in different time intervals and determine anomalous time intervals [23]. Many item anomaly detection approaches suffer from confirming the size of the time interval. In [7,23], the size of the time interval is empirically identified although not convincingly, while in [62], the proposed heuristic approach designed to fix the size of the time interval requires the information of the length of an attack event.

2.3. Evaluation criterion

The majority of the literature in shilling attack detection has focused on enhancing the detection rate or precision. Apart from that, the performance could be assessed through the false alarm rate [8,17,23,28,62], which measures the percentage of genuine users who are detected as attack users; the recall rate [7,38,39], quantified by the percentage of truly detected attack profiles divided by all attack profiles; and the F1 measure [7,10–12], integrating the precision and recall rate together. However, these metrics are utilized for the evaluation of the effectiveness of algorithms, and little research has studied the robustness of the detection algorithms, although the robustness of a CF system has been explored [2,41]; it requires high recommendation accuracy, as well as stability, so that the recommendations remain unaffected after being attacked.

3. Preliminaries

In this part, some basic notations are first presented for clarity. Then, several common features and three types of shilling attacks are analyzed. Finally, the definition and analysis of the new stability metric are introduced.

3.1. Basic definitions

Definition 1 (*Item profile*). The profile of an item k, r'_k, refers to its rating series received from users in user set U, and r_k is another time-ordered version of r'_k (ascending), as seen in (1). $r^i_{u_jk}$ indicates that the rating of item k from user u_j ranks ith in r_k.

$$r_k = \left\{ r^1_{u_ak}, r^2_{u_bk}, ..r^i_{u_jk}.., r^n_{u_mk}, i \right\}, \quad i = 1, 2, 3, \ldots, n; \quad u_a, u_b, u_j, u_m \in U \tag{1}$$

Definition 2 (*Rating matrix*). The profile of all items P forms the rating matrix R, for which the rating space is defined as $U \times P$. Each cell in R represents the score from user u to item k, that is, $r_{uk} \in R$, which is used for representing a rating from u to k. Hence, each column in R represents the profile of that item, while each row in R indicates the user profile. Let D_R be the detection space, detecting ratings in R and classifying them into two groups, the normal group and the abnormal one.

Definition 3 (*Time matrix*). Similarly, we can arrange the rating time from all users in U to all items in P together as a time matrix with the space of $U \times P$, in which each column is a rating time set of the specific item, while each row collects the history rating time of the user.

Definition 4. Checkpoint. In the online system, a checkpoint refers to a timestamp when the detection procedure is triggered to carry a single execution. Many detection methods check the system regularly; hence, the system can collect different scales of rating series or the portion of R, R_{sub}, among different checkpoints. Suppose we have a consecutively checkpoint series C, $C = \{c_1, c_2, c_3, c_4, c_5\}$; then, the relationship of the corresponding R_{sub} is $R^{c_1}_{\text{sub}} \subseteq R^{c_2}_{\text{sub}} \subseteq R^{c_3}_{\text{sub}} \subseteq R^{c_4}_{\text{sub}} \subseteq R^{c_5}_{\text{sub}}$.

Definition 5 (*Life Cycle (LC) [47,23]*). In each submatrix, the LC of an item can only be defined as the time span from the start of rating time S to the last rating time E before the checkpoint c_i. Hence, in the latter checkpoint, the LC of the item is not changed until the new rating comes again.

Definition 6 (*Time interval [23]*). A period of LC means a time interval of an item.

Definition 7 (*Detection model*). The detection mechanism should construct a classification model $f : R \to \{0, 1\}$; where 0 means a normal group, and 1 implies an abnormal group. Let $D^{c_i}_{\text{sub}}$ be the detection space of $R^{c_i}_{\text{sub}}$ in checkpoint c_i; hence, $D^{c_i}_{\text{sub}} = \{(r^i_{uk}, \text{cid}) | i \leqslant c_i, r^i_{uk} \in R, \text{cid} \in \{0, 1\}\}$. The detection model is then constructed by the detection system S; thus, it can be denoted as $\overline{\text{cid}} = f_{S, D^{c_i}_{\text{sub}}}(r^i_{uk})$, where $\overline{\text{cid}}$ is the results generated from S.

3.2. Identifying common attack features under item profile

According to the aforementioned shilling attack types in Section 2.1, we can note that only when the rating distribution of the target item has great changes after certain sizes of shilling attacks can the results of recommendation systems be changed. At the same time, malicious users have to assess costs and benefits about the attacks, so that they could maximize their economic benefits but cost little. Hence, the common attack features are:

- Attackers prefer to mark the target item with ratings that are largely different from its profile characteristics (i.e., the mean value). Usually, $r_{\max}/r_{\min}$ is the first choice for push/nuke attacks.
- The attack scales should be sufficiently large so that the statistical characteristics of the target item could vary along with the intention of the attackers.
- Attack schedules should be short-term plans considering leveraging the costs and benefits.

The proposed detection algorithm is totally based on the above common features, which means the method can be used in any attack type regardless of the specific shilling attack type.

3.3. Classifying shilling attacks under item profile

From Section 2.1, shilling attacks are classified into many types under the user profile. This is because the preferences and rating behavior of a user are fickle, and the resulting user profile is various, as well as the fake user profile. In this research, we simplified attack types into three classes under the item profile, which could contain any shilling attack types (we only take push attacks into consideration because the basic principle of the nuke attack is the same as the shilling attack, and the detection approaches for the push attack can be easily changed into those of the nuke attack):

- Maximum Injection (MI). Target item is assigned with r_{max} from all attack profiles. Actually, the attack types listed in Table 1 can be classified into MI.
- Target Shifting (TS). Target shifting involves setting the rating of the target item in a certain percentage of attack profiles with $r_{max} - 1$ for push attacks ($r_{min} + 1$ for nuke attacks). The existing user shifting and target shifting attack strategies are included in TS.
- Target Noise (TN). Target noise means a percentage of attack profiles are rated with randomly selected noise ratings. It can be viewed as MI if the noise percentage is close to zero. It contains the obfuscated strategies, such as noise injection.

The classification of shilling attacks under item perspective is simple and universal. It also facilitates the experiment procedure in our research because we need to simulate only three types of attack models to validate the universality of the proposed algorithm, instead of the various types of attacks mentioned in Section 2.1.

3.4. Stability of the detection algorithm

As opposed to the stability notion mentioned in recommendation model [2,41], we define the detection model as stable if the detection results of mutual ratings in different rating scales collected at different checkpoints are consistent. In other words, the stability measures the extent to which the detection algorithm provides detection results that are consistent in different rating scales of R_{sub}. Specifically, for a robust detection algorithm, the detection results in one checkpoint should not change greatly compared with that in the prior checkpoint under a high detection rate. Here, we illustrate a simple example for a better understanding of the stability.

We collected two scales of rating series, as well as the corresponding rating time series of an item i in two different checkpoints c_1 and $c_2 (c_2 > c_1)$, as illustrated in Fig. 1(a). It can be observed that c_1 locates at t3 (randomly selected) and c_2 at t2. The start time of item i's LC is t1. Hence, the first time scale (the blue[1] empty circle) is from t1 to t3 (c1), and the second scale (the red-filled circle) ranges from t1 to t2 (c2), which contains the first time scale totally.

Using the X-Bar based detection algorithm [7], we generated two types of detection results corresponding to the two checkpoints, as Fig. 1(b) indicates (0 means the rating is normal, and 1 indicates that the rating is abnormal). It is quite clear that ratings at the 11th, 13th, 19th, 22th and 23th have different detection results in two scales, that is, the detection results are inconsistent in different time scales using the same X-Bar-based detection method.

This inconsistency indicates the instability of the detection algorithm, which potentially lowers user satisfaction with the system and further misleads the decision of the system, such as adapting the trust/reputation value of each user. Hence, it is valuable for designers to be aware of the stability of the detection method.

There are two inconsistent cases in the majority of detection methods. One case is the detection result of rating r_{uk} being normal in checkpoint c_i, but abnormal in c_j. On the contrary, the other case is that the detection result is abnormal in the earlier checkpoint c_i, but turns into normal in the later checkpoint c_j. The former case often occurs in an item's early LC state, in which the information is not sufficient for the system filtering out the abnormal rating; hence, we consider it reasonable and only pay attention to the latter case. To calculate the stability metric, we should record the detect shift of each rating, which means the shift in detection results at different checkpoints, $\Delta r^i_{uk} = \left|\overline{\text{cid}}_{c_i} - \overline{\text{cid}}_{c_j}\right|$. After that, we measure the stability by computing the mean absolute difference (MAS, mean absolute shift) or root mean squared difference (RMSS, root mean squared shift), which are among the most popular and widely used metrics. Then, the aforementioned stability can be measured using (2) and (3).

$$\text{MAS} = (1/|D^{c_i}_{\text{sub}}|) \sum_{r^i_{uk} \in R^{c_i}_{\text{sub}}} \Delta r^i_{uk} \tag{2}$$

$$\text{RMSS} = \sqrt{(1/|D^{c_i}_{\text{sub}}|) \sum_{r^i_{uk} \in R^{c_i}_{\text{sub}}} (\Delta r^i_{uk})^2} \tag{3}$$

4. Dynamic time interval segmentation and hypothesis test detection-based framework (SDF)

In this paper, we quantified the characteristics of the time-ordered rating sequence of an item with a skewness metric, which describes the asymmetry of the probability distribution of a real-valued random variable about its mean [20]. Then, the changes of the skewness quantities between the neighboring ratings imply the influence of the latter coming rating to the whole rating distribution. Moreover, the rate of the change or the first order difference of skewness at each rating represents the contribution of the corresponding rating to the whole existing distribution at its rating time, from both the normal ratings and the attack ratings. Hence, we could perceive shilling attacks through the changes in the skewness of rating series. Based on this, we proposed a novel item anomaly detection framework, as Fig. 2 indicates.

[1] For interpretation of color in Fig. 1, the reader is referred to the web version of this article.

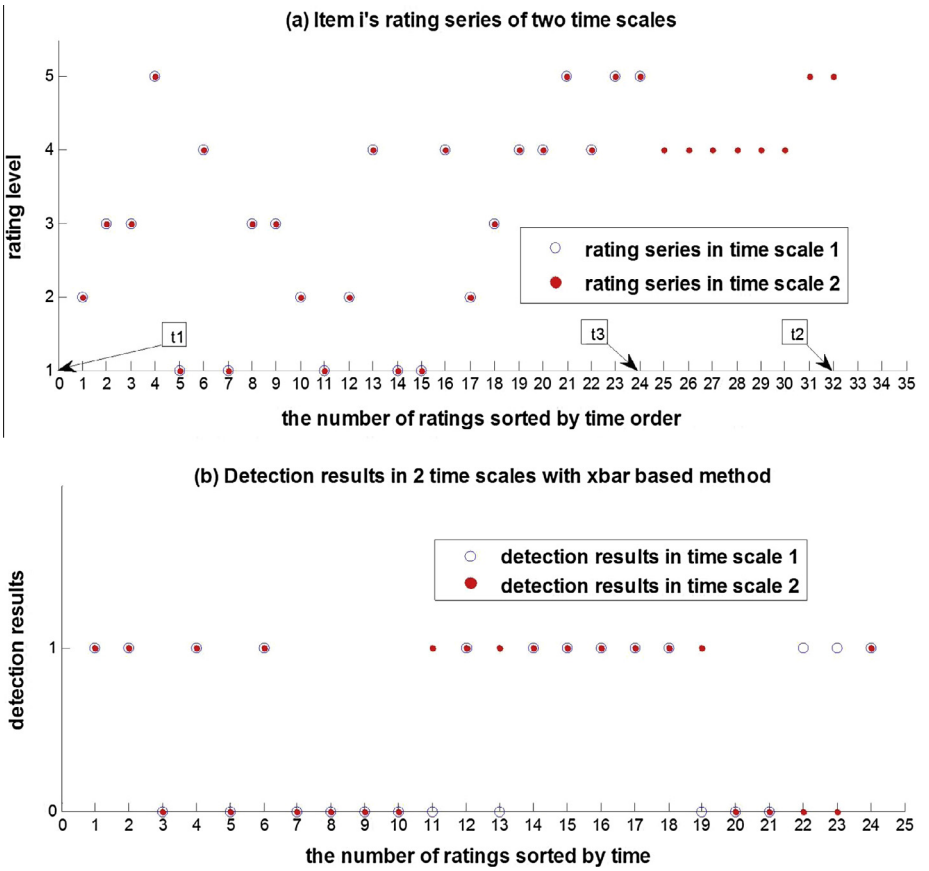

Fig. 1. (a) Item *i*'s rating series of two time scales and (b) detection in two time scales with X-Bar-based method.

The whole framework includes two parts: time interval segmentation and abnormal interval detection. The former requires the representation of the changes of the rating distribution and segmentation process, and we could identify the intervals with suspicious ratings in the latter process.

4.1. Segmenting the LC of each item into several time intervals dynamically at a checkpoint

First, assume that, at checkpoint c_i, we gathered a rating series r_k of an item k order by time, in which each rating has its corresponding skewness value, and the skewness collection can be denoted in the form below:

$$\gamma_3^k = \{\gamma_3^k(1), \gamma_3^k(2), \ldots, \gamma_3^k(i), \ldots, \gamma_3^k(n)\} \tag{4}$$

where $\gamma_3^k(i)$ is the corresponding skewness value, $\gamma_3^k(i) = E(r_{[1\ldots i]k} - Er_{ik})^3/(Dr_{ik})^{3/2}$, Er_{ik} is the expectation value of ratings from the first to the ith one, and Dr_{ik} is the variance [57].

The first-order difference of the skewness quantity at each rating time or the contribution of each rating to the whole rating distribution can be computed through the shifts in the skewness quantities with its neighbor, as indicated in (5), which represents the shift between the ith and $(i+1)$th ratings:

$$\nabla\gamma_3^k(i+1) = \partial\gamma_3^k(i+1)/\partial(i+1) = \gamma_3^k(i+1) - \gamma_3^k(i) \tag{5}$$

Then, we need to cluster the ratings by analyzing if the contributions of neighbor ratings are of the same type. One effective way is the multiplication of them, for example, $\nabla\gamma_3^k(i)$ is the contribution of the ith rating, while $\nabla\gamma_3^k(i+1)$ is from the $(i+1)$th rating; the multiplication between them will have three types of results:

$$D(i,i+1) = \nabla\gamma_3^k(i) * \nabla\gamma_3^k(i+1)\begin{cases} >0 \\ <0 \\ =0 \end{cases} \tag{6}$$

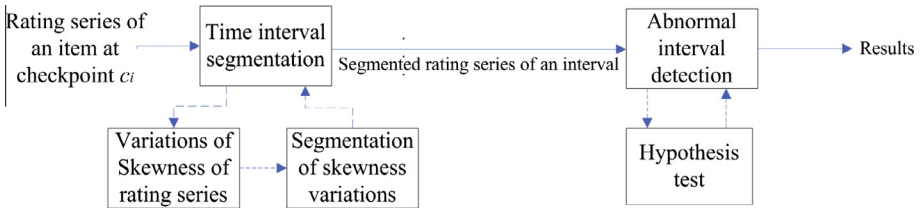

Fig. 2. Dynamic time interval segmentation and hypothesis test detection-based framework (SDF).

The positive result of $D(i, i+1)$ indicates both ith and $(i+1)$th ratings have the same type of contribution. That is, they both lead the whole rating distribution toward the same right/left leaning; hence, the ith and $(i+1)$th rating should be combined into the same interval group. On the other hand, the negative result indicates the diverse effects of the ith and $(i+1)$th rating, that they should be separated into different groups. As for the zero condition, we put them into the same group for the sake of simplicity.

This type of segmentation actually assembles consecutive attack ratings of the same type into the same interval groups. In addition, the scale of the segmented time interval is flexible depending on the changes in rating distributions. In particular, we designated this type of segmentation algorithm as the variations of skewness-based dynamic time interval segmentation algorithm. The whole segmentation algorithm for each item can be seen in Table 2.

4.2. Detecting an anomalous item through hypothesis testing

After the segmentation procedure, the whole rating series of item k has been divided into n groups, $g_k = \left\{g_k^1, g_k^2, \ldots, g_k^{i'}, \ldots, g_k^n\right\}$, where $g_k^{i'}$ is the (i')th time interval group with average value $\bar{x}_{ki'}$ and the number of ratings $n_g^{i'}$. In addition, the distribution of the whole LC ratings could be described as a normal distribution, according to the central limit theorem [44], and $\bar{x}_k$ and δ_k are defined as the sample mean and the average squared deviation of the rating series of the whole LC of the item k.

The detection problem can be solved through the statistical hypotheses. The detection result depends on testing if the statistical characteristics of the subinterval ratings are the same as that of the whole rating series during the entire LC of item k. There are two hypotheses:

$$\mathrm{H}_0: \quad \bar{x}_{ki'} \approx \bar{x}_k \qquad \mathrm{H}_1: \quad \bar{x}_{ki'} \neq \bar{x}_k$$

Table 2
Dynamic time interval segmentation algorithm for each item.

Input: Time ordered rating series of item k, r_k, at checkpoint c;
Output: Segmented time interval collection g_k;

Procedures:
1: Get the serial number of each rating in r_k, and the last number n.
2: For $i = 1, 2, 3, \ldots, n$
3: Identify mean and standard derivation values of rating series of item k before the ith rating using (7) and (8) iteratively.

$$\bar{x}_k^i = ((i-1) * \overline{x_k^{i-1}} + r_{u_j k}^i)/i \tag{7}$$

$$\delta_k^i = \sqrt{\left(\left(\delta_k^{i-1}\right)\text{^}2 * (i-1) + \left(r_{u_j k}^i - \bar{x}_k^i\right)\text{^}2\right)/i} \tag{8}$$

4: Apply (5) for computing the first order differential of skewness quantity, and $\gamma_3^k(i)$ the contribution value, $\nabla\gamma_3^k(i)$, then the set $\{\nabla\gamma_3^k(1), \nabla\gamma_3^k, \ldots, (2)\nabla\gamma_3^k(i)\}$ collects all the contributions up to the ith rating.
5: If $i = 1$, then the first rating, as well as its corresponding rating time rank number are assigned to group $g_k^{i'}$ $(g_k^{i'} = 1)$; $i = i + 1$;
6: If $\nabla\gamma_3^k(i) = 0$, then i and its rating are assigned to group $g_k^{i'}$;
7: Otherwise $\nabla\gamma_3^k(i)$!= 0, compute J according to (9);

$$J = \nabla\gamma_3^k(i-1) * \nabla\gamma_3^k(i) \tag{9}$$

8: If J >= 0, then i with its rating are assigned to the group containing $(i-1)$ and the $(i-1)$th rating;
9: Otherwise $J < 0$, then $g_k^{i'} = g_k^{i'} + 1$, and the ith rating are assigned to the new group $g_k^{i'}$;
10: End of for i;

The critical region to accept H_0 or reject H_0 is determined by (10) with a significance level ∂, which utilizes the bilateral testing of standard normal distribution.

$$\left\{\frac{|\bar{x}_{ki'} - \bar{x}_k|}{\delta_k}\sqrt{n_g^{i'}} > u_{1-\partial/2}\right\} \tag{10}$$

In this formula, the significance level ∂ (type I error) is the predetermined threshold probability (and usually is a low value) of wrongly rejecting the null hypothesis given that the H_0 is true. It helps researchers to decide if the null hypotheses can be rejected [44]. It is selected as required, and in the next part, we will present the procedure for selecting the best suited significance level. In addition, $1 - \partial$ is the confidence level, which measures the reliability of the result.

If (10) is true, we should reject H_0, which means the corresponding interval contains suspicious rating(s). This, in turn, means that if (10) is false, we should accept H_0 and regard ratings in the corresponding interval as genuine.

After the hypotheses testing process, both anomaly item and its attacked intervals are confirmed; hence, the system could further detect the malicious users in those intervals or treat them as evidence or prior knowledge of the necessity of updating the trust and reputation level for user and item.

5. Experiment evaluation

We selected a Movielens dataset [65], which has 1682 items with 100,000 ratings from 942 users. Ratings are discrete-valued between 1 and 5. We sorted ratings for each item by their time stamp. The generation of attack event (here, we only focus on the push attack) is in line with three types of strategies as indicated in Section 3.3.

As opposed to the categorization in [7], we only considered low average rating for the push attack (the experiment can easily be changed and applied to a nuke attack as Section 3.3 indicates); meanwhile, we integrated the high density rating item category into the medium density category because the length of the former is too small to yield reliable results. Thus, we only have two categories: the rating length of the first category is from 40 to 80 per item, and the second is larger than 80. We measure "size of attack" as the ratio of the attack rating count to the pre-attack rating count of an item, that is, if the ratio of attack size varies from 10% to 100%, the item in the above two groups has at least 4 attack profiles.

To analyze the performance of the proposed algorithm, especially in online contexts, we simulated an online detection procedure by allocating several checkpoints to collecting different scales of rating series. Hence, each item has different sample mean values and mean square deviations with the ever increasing ratings at those checkpoints. In addition, we explored the performance of the detection algorithm in several aspects:

- The effectiveness. The detection rate and the false alarm rate are selected to validate the efficacy of the detection algorithms. The detection rate (DR) is actually defined as the number of detected attack events divided by the number of attack events. An attack event is considered to be detected if an interval containing attacks is marked as an anomaly. The false alarm rate (FAR) is defined as the number of normal intervals that are predicted as anomalies divided by the number of normal intervals.
- The robustness. With a certain high accuracy, the robustness of the detection algorithm requires the stability of the detection results in different time scales. According to Section 3.4, different detection subspaces can be constructed through different checkpoints, and the shifts are from any two neighbor checkpoints. Hence, the measurement in (2) and (3) can be denoted as (11) and (12) in practice, in which C denotes the collection of all checkpoints.

$$\mathrm{MAS} = (1/(|C| - 1)) \sum_{c_i \in C, c_{(i+1)} \in C} |\overline{\mathrm{cid}}_{c_i} - \overline{\mathrm{cid}}_{c_{(i+1)}}| \tag{11}$$

$$\mathrm{RMSS} = \sqrt{(1/(|C| - 1)) \sum_{ci \in C, c(i+1) \in C} \left(\overline{\mathrm{cid}}_{c_i} - \overline{\mathrm{cid}}_{c_{(i+1)}}\right)^2} \tag{12}$$

- The timeliness. Because we simulate the whole experiment in online circumstances, it is necessary to analyze the timeliness of the proposed method. If the algorithm has a lower time complexity, it will have a better timeliness performance.

Section 5 is organized as follows: Section 5.1 discusses how to identify the necessary parameters in SDF. In Section 5.2, we analyze the detection results of the proposed approach. The effectiveness, robustness and the timeliness of SDF are discussed in Section 5.3–5.5, respectively. Additionally, in Section 5.6, we check the robustness of the introduced stability metric and confirm the introduced metric could be used to represent the stability.

5.1. Locating the parameters used in SDF

5.1.1. Identifying a best significance level for detection

The significance level should be a small value to protect the null hypothesis and to prevent, as far as possible, the investigator from inadvertently making false claims. We listed six significance level values, varying from 0.005 to 0.05, and their

boundaries in Table 3. Fig. 3 depicts the ROC curves between FAR and DR of the two groups in three attack types. This figure indicates that the changes in DR are not as obvious compared to FAR, and a best significance level could be 0.03.

5.1.2. Confirming the boundary of the average rating of each segmented time interval

In online circumstances, usually the existing rating count of an item is not sufficient for computing an unbiased sample average, or the rating series are inserted with attack ratings, so that the sample average based on those ratings is definitely biased. For example, we generated large numbers of push attacks (ratio is 100%) for hundreds of items for which the average rating values are lower than the medium rating 3 in the Movielens dataset. After shilling attacks, the sample average values of all items are larger than 3, as Fig. 4 illustrates, which is far from the reality.

To reduce these biases, we made an additional boundary for the average rating of each segmented time interval. First, we classified the average value in each segmented interval by comparing it with $\bar{r}$ ($\bar{r} = 3$, the reason for this is that in the Movielens dataset, there are five rating levels from 1 to 5; if $\bar{r} \leqslant 2$, the low ratings are only 1 and 2, which is too small, while the intervals of the high rating are bigger, from 3 to 5, and if $\bar{r} \geqslant 4$, the low rating intervals are larger than that of the high rating. Hence, we had 3 for trading off the low rating interval and high rating interval). Then, if more than half of the average values are larger than $\bar{r}$, the sample average is limited to not being smaller than $\bar{r}$; otherwise, if more than half of the intervals are smaller than $\bar{r}$, the boundary of the sample average is not larger than $\bar{r}$. The statistical information of the attacked items in Fig. 4 is displayed in Fig. 5. Although the computed sample average values are larger than 3, far less than half of the interval average values for nearly all items are larger than 3. Therefore, we could bound those item sample average values as not larger than 3. This is definitely effective in reducing the difference between the sample average and the reputation of an item.

5.1.3. Identifying the range of the attack size that SDF could handle

As Fig. 6 indicates, the attack size varies from 10% to 100% on items with at least 20 ratings. As the attack size increases, the DR remains larger than 90%, while being slightly worse as the FAR increases. Hence, the proposed SDF can handle the attack size ranges from 10% to 100% and is particularly effective in sizes smaller than 80%, with a DR larger than 90% and FAR smaller than 4.2%.

5.2. Analyzing the intrinsic features of SDF

Fig. 7 displays the DRs and FARs of two groups in various attack sizes on all three attack types. In Fig. 7(a), we could note that the DR of MI grows slowly and maintains a high value as the attack size ratio increases because MI is more easily detected with a larger attack size ratio. As for the TS attack, its trend indicates a continuous decline for the increasing attack size ratio. This is because it can obfuscate itself very well, especially in larger attack size ratio conditions. With regard to a TN attack, we can find that it remains stationary in nearly all attack size ratios but with lower accuracy, as the TN attack can

Table 3
Significance level value ∂ and related boundary value $u_{(1-\partial/2)}$.

∂	0.05	0.04	0.03	0.02	0.01	0.005
$u_{(1-\partial/2)}$	1.96	2.05	2.17	2.33	2.57	2.81

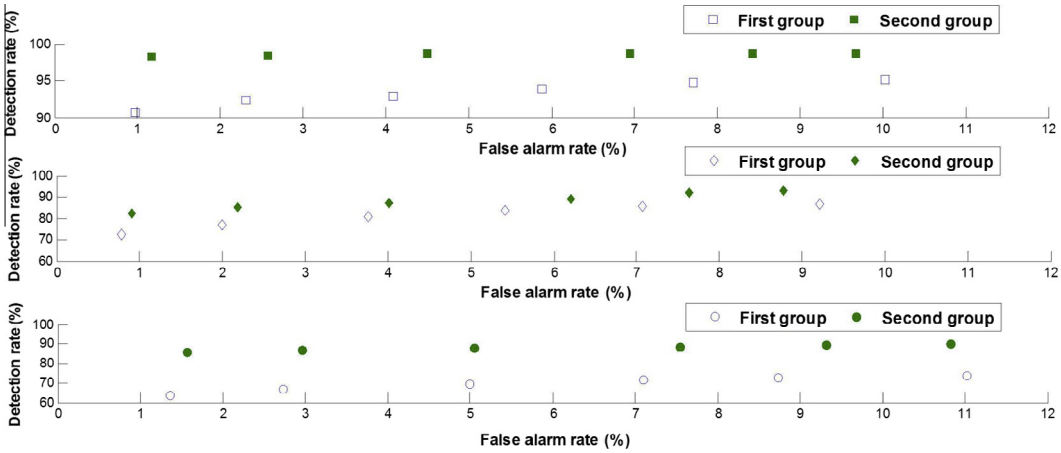

Fig. 3. ROC curves for MI (upper), TS (middle) and TN (lower), with six significance levels for both groups.

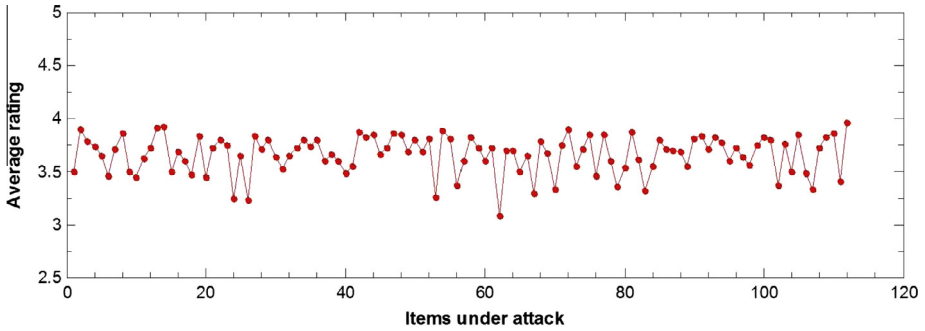

Fig. 4. Sample average rating values of items after 100% push attack.

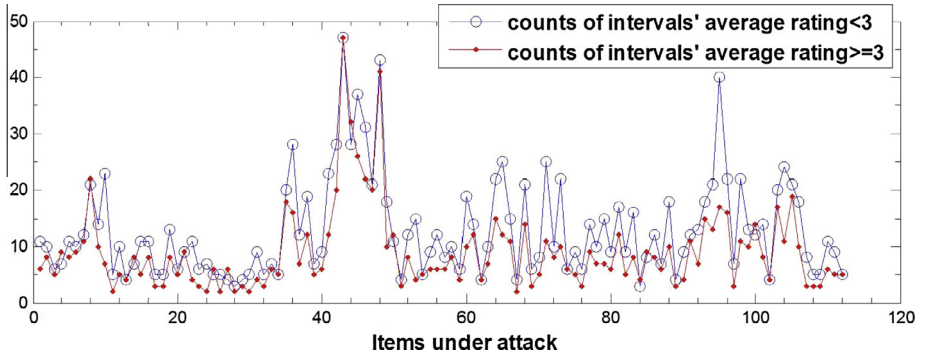

Fig. 5. Numbers of interval average rating values larger than $\bar{r}$ versus smaller than $\bar{r}$, 100% push attacks, $\bar{r} = 3$.

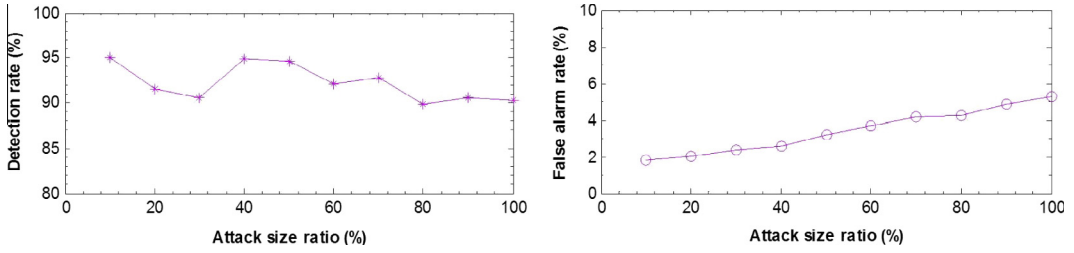

Fig. 6. DR (left) and FAR (right) in different attack size ratios.

be decomposed into two parts: the random selected ratings for the noise part and the MI attack for the left part, in which the noise part is more confusing and the MI part is more easily detected, respectively, with the ever increasing attack size ratio. The same trend can be seen in Fig. 7(b) for the DR of the second group. However, the whole precision in (b) is higher than that of (a), and this is in line with the common knowledge that more ratings in each item lead to a more accurate hypothesis testing in SDF.

In Fig. 7(c), we would like to determine that the TS and MI false alarm trend grows with the increasing attack size ratio, while the trend in TN stays the same. The reason is the same as that in Fig. 7(a) and (b), that is, the attack size of TS and MI grow larger with the increasing attack size ratio, while the attack size of TN remains the same because the noise ratio remains stable in our experiment. Fig. 7(d) displays nearly the same characters for the second group in the FAR as in Fig. 7(c). The trends of FAR between (c) and (d) are almost the same because the rating ratio is the same.

In conclusion, we could obtain $DR_{MI} \geqslant DR_{TN} > DR_{TS}, FAR_{TN} \leqslant FAR_{MI} < FAR_{TS}$, which conforms to the truth that TS is hard to detect with the smallest DR and largest FAR, MI is easy and apparent to detect with the largest DR and smaller FAR, and TN, combined with MI and noise, is somewhat difficult to detect with a larger DR and smallest FAR.

5.3. Comparing effectiveness of the SDF with X-Bar based approach

The attack profiles were injected into the second half LC of each target item, for the purpose that the X-Bar-based approach could learn the normal rating features in both ideal and non-ideal cases. The first ideal case involves taking the

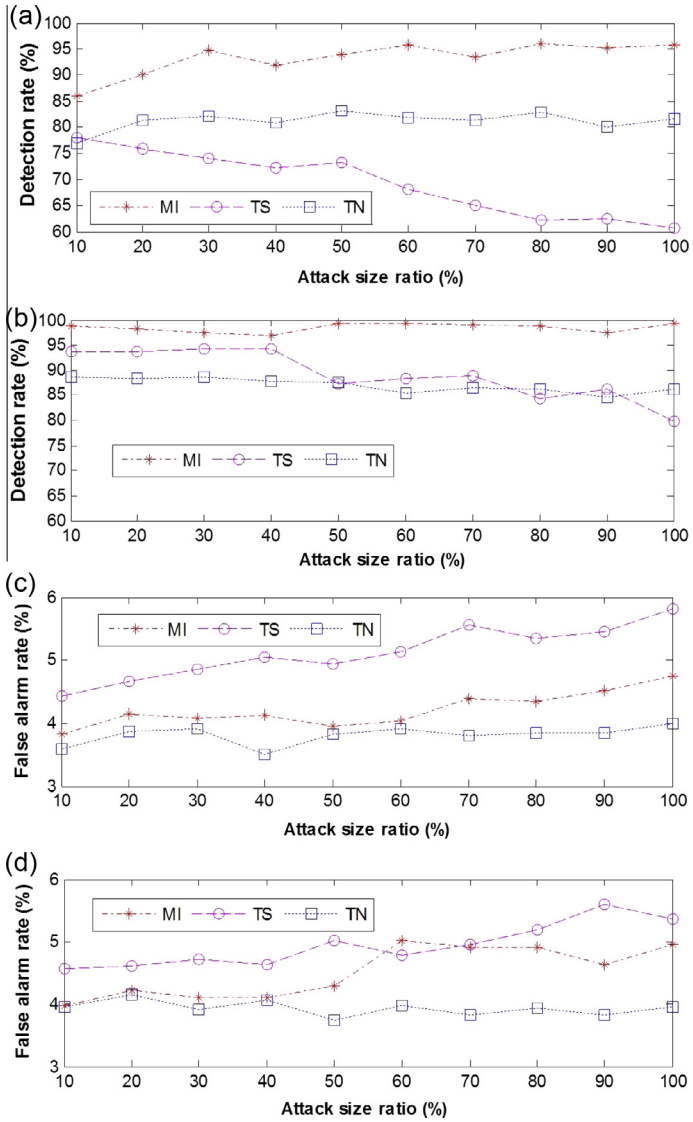

Fig. 7. (a) and (c) are the DR and FAR of the first group in all attack size ratio against all three attack strategies, respectively; (b) and (d) are the DR and FAR of the second group in all attack size ratios against all three attack strategies, respectively.

rating series in the first normal half LC as the learning data, which is actually hard to collect in an online system. The second non-ideal case considers all the existing rating series of the target items at the checkpoint as training data, which could be applied online more easily. The SDF, however, identifies the boundary value of the sample average using the data collected only in the non-ideal case. We repeated the whole procedure on all three attack types 10 times for each target item, and the results are displayed in Fig. 8.

From Fig. 8, we can observe that, both the DR and FAR of SDF in each attack type remain, on average, for the most part unchanged in both the ideal and non-ideal cases, while the X-Bar-based approach performs better in the ideal case. The reason is that the parameter confirmation procedure of SDF is effective in both conditions according to its statistical mechanism, while the learning process in X-Bar-based approach only performs better in ideal condition. That is, the quality of data used for that procedure has little influence on SDF, while it has a great impact on the X-Bar-based approach.

In addition, the DR of SDF is close to that of the X-Bar-based approach in the ideal condition, especially in the MI attack types (as seen in Fig. 8(a)) and, at the same time, much better than that of the X-Bar-based approach in the non-ideal condition (as seen in Fig. 8(a), (c), (e)). The FAR of SDF, however, remains far lower than that of the X-Bar-based approach in any condition regardless of the specific attack type (as seen in Fig. 8(b), (d), (e)). That is to say, in non-ideal online circumstances, SDF can still yield high detection accuracy, as well as low FAR, while the X-Bar-based approach drops significantly. This is in accordance with the understanding that the X-Bar-based approach requires a great deal of correct evidence (difficult to obtain in reality) to obtain higher DR, while SDF considers the actual detection situations at the very beginning. Specifically,

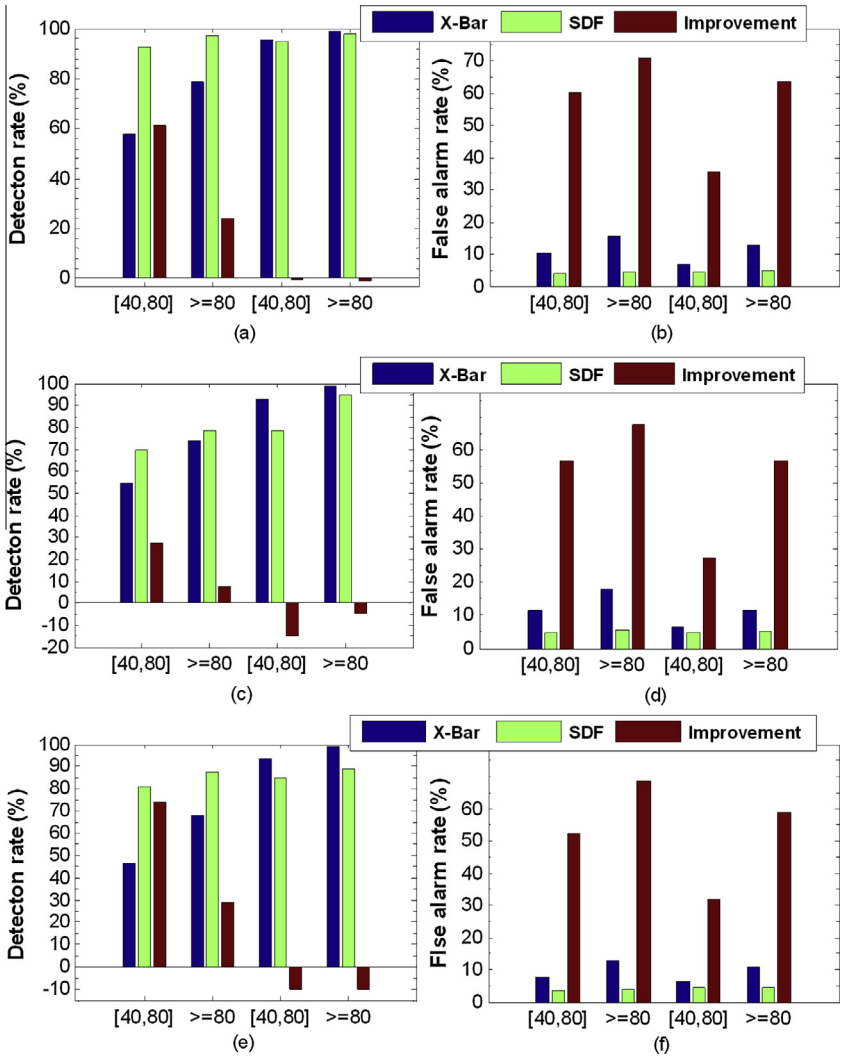

Fig. 8. Detection results in non-ideally scenario versus in ideally one (a and b) Comparisons of DR and FAR in maximum injection (MI), with the first two groups from non-ideally scenario and the second two groups from ideally scenario. (c and d) Comparisons of DR and FAR in target shifting (TS), with the first two groups from non-ideally scenario and the second two groups from ideally scenario. (e and f) Comparisons of DR and FAR in target noise (TN), with the first two groups from non-ideally scenario and the second two groups from ideally scenario.

we can also note that, in an ideal condition, SDF does not perform well in DR, especially in Fig. 8(c). This is because the changes in skewness quantities vary little if new income ratings are shifted. Meanwhile, the DR of SDF grows with increases in the rating series length. This is in line with the fact that the SDF requires little prior knowledge.

5.4. Comparing the robustness of SDF with other detection approaches

The robustness of the detection algorithm includes two aspects: the accuracy and the stability. The accuracy of the SDF has been explored in Section 5.3, and the accuracy of SDF is normally higher than 80% (in the TS attack type). In this section, the stability of SDF and several other detection algorithms will be discussed. We applied four typical anomaly detection methods: SDF, the X-Bar-based method in the ideal case, the X-Bar-based method in the non-ideal case, and PCA-based detection.

We first injected attack profiles into the rating series. The first three methods were attacked with MI attacks, and we particularly designed average attack profiles for PCA because it directly works on users. Suppose there is a checkpoint series with five elements, which divide the detection space D^I of each item into five subspaces $\{D^{I1}, D^{I2}, D^{I3}, D^{I4}, D^{I5}\}, D^{I1} \subseteq D^{I2} \subseteq D^{I3} \subseteq D^{I4} \subseteq D^{I5}$. Then, let four methods run on all five subspaces; we counted the detect shift between adjacent subspaces, which can be denoted as $\{\langle D^{I1}, D^{I2}\rangle, \langle D^{I2}, D^{I3}\rangle, \langle D^{I3}, D^{I4}\rangle, \langle D^{I4}, D^{I5}\rangle, \langle D^{I5}, D^{I1}\rangle\}$. We finally computed MAS and RMSS for each method and listed them in Table 4.

Table 4
Average MAS and RMSS on four methods, 60% attack size for the first three, 20 attacks for PCA. The bold values signify the effectiveness of the proposed method (SDF) comparing to other three methods.

Metric	SDF	X-Bar ideal	X-Bar non-ideal	PCA
MAS (%)	**0.18**	2.53	1.29	0.89
RMSS (%)	**0.56**	20.48	18.21	9.23

Table 5
Average MAS and RMSS between nonadjacent checkpoints of the four methods, 60% attack size for the first three, 20 attacks for PCA. The bold values signify the effectiveness of the proposed method (SDF) comparing to other three methods.

Metric	SDF	X-Bar ideal	X-Bar non-ideal	PCA
MAS (%)	**0.18**	3.14	1.48	1.20
RMSS (%)	**0.56**	25.29	20.71	10.93

It can be observed that both the MAS and RMSS of the proposed SDF are the smallest values compared with the other three methods, which means the results produced by SDF shifted a minimum distance at those five continuous checkpoints. That is, the proposed SDF retains a relatively high consistency and has a strong stability when working online. The PCA-based method ranks second, and the X-Bar-based approach is the least stable.

For a better understanding of the whole ranks, we should analyze the mechanisms in them. Although the ratings are collected at different checkpoints, SDF can determine how to segment rating series as long as the time-ordered rating sequence of each item is fixed and confirm the average boundary of each interval before testing. The PCA-based method ranks second and performs well because it detects the anomaly user by projecting a high dimensional data to a low dimensional space and then extracts users with high similarity together as attackers. The last X-Bar-based approach distinguishes malicious ratings and their time intervals from the existing rating data, the results of which may be different if the learning process is in different rating scales at different checkpoints. Hence, we could admit that the proposed SDF is relatively robust because it preserves both high accuracy and strong stability.

5.5. Comparing the timeliness of SDF and other detection approaches through time complexity

The supervised classification-based detection methods require a learning process, while the complexity of those unsupervised clustering-based detection methods are usually at least $O(L^2)$. The statistical-based methods are different, for example, the time complexity in [62] is $O(L \log L)$, and X-Bar-based method is $O(L)$.

The boundary confirmation and hypothesis testing process in SDF can be directly executed after a new interval is segmented according to Sections 4 and 5.1.2. Hence, the process can generate the detection result of an interval once it was separated, the time complexity only has positive correlation with the rating count of each item, $O(L)$, and has no concern with the learning process. When applying online, the SDF will cost the least amount of time in result generation.

5.6. Checking the robustness of the stability metric

When calculating stability according to (11) and (12), we only pay attention to the stability between neighbor checkpoints with the smallest number of new added ratings. This will definitely inspire further exploration on the stability between nonadjacent checkpoints with more newly added ratings. Naturally, one would expect that at least the same or even higher level of instability will be observed, but the more important question is whether this will result in the same relative stability rankings of different detection algorithms or the robustness of the stability metric.

Similar to the process in Section 5.4, we performed an experiment in a nonadjacent checkpoint group $\{\langle D^{I1}, D^{I3}\rangle, \langle D^{I2}, D^{I4}\rangle, \langle D^{I3}, D^{I5}\rangle, \langle D^{I4}, D^{I5}\rangle, \langle D^{I5}, D^{I1}\rangle\}$. The results are summarized in Table 5. From Table 5, we find that the stability patterns of different detection algorithms are consistent with the patterns observed in Table 4. In particular, because of the new added ratings, the instability increase can be observed in both the X-Bar-based approach and the PCA method. However, the proposed SDF is unchanged, which is in line with the mechanism of SDF. In addition, the inherent stability differences among the different techniques remain, as evidenced by the same relative stability rankings of the different techniques: SDF demonstrates the highest stability, followed by the PCA method, while the X-Bar-based approach exhibits the lowest levels of stability among all the techniques.

In summary, the proposed stability metric demonstrates robustness when comparing the shifts between nonadjacent checkpoints. Hence, (11) and (12) can be considered the standard approaches for stability computation, as defined in Section 3.4.

6. Conclusions and future work

6.1. Conclusions

The proposed detection framework offers an effective solution to the very real problem of detecting the attacked item with its malicious rating intervals and, further, the malicious users, regardless of the attack type. The detection algorithm utilizes the rate of change of skewness quantities for time interval segmentation. This dynamic interval segmentation technique is, to the best of our knowledge, the only one that can successfully cluster the consecutive attack ratings of the same type together, form the intervals with different scales, and effectively identify the attacked intervals given the conditions that the previous studies are all based on the fixed time intervals. In the online simulation, the results empirically indicate that SDF is effective in a wide range of attack sizes with great robustness and can be applied in any shilling attack type online with linear time complexity.

In addition, we introduced a type of performance metric, the stability metric, which is of great value especially in the online system because it can represent the robustness of the detection algorithm together with the detection accuracy and will definitely help the system make further decisions and enhance user satisfaction. The subsequent experimental work proved that the metric is robust in indicating the stability levels of different detection methods.

6.2. Future work

Keeping these promising results as our starting point, we intend first to ensure the identification of malicious users based on the detected suspicious interval of each item through SDF. Actually, users who have rated the item in suspicious intervals exceeding normal thresholds will be considered attackers, so we could count the frequency of user anomalous ratings and record those that exceed it; if the approach is incorporated into other typical detection algorithms, it will decrease their computational cost and increase the detection precision.

Second, the question remains as to how to lower the impact of attacks after identifying each item's suspicious intervals. One effective way is to update the reputation and trust value of each user based on the detection results from SDF. After that, the system could make better decisions considering user reputation and trust values.

Third, there is the issue of improving the detection performance against TS attack, which is hard for nearly all algorithms to detect. One possible way is to gather other attributes of users (i.e., login times, the scope of favoring items and reviews) together to determine suspicious ratings. This is also the main point of our future research.

Finally, it is remarkable of describing the relationship between stability and accuracy of an algorithm. Questions such as exploring the trade off or the orthogonality between these two metrics are left for future research.

Acknowledgments

This work was supported, in part, by the National Key Basic Research Program of China (973 Program 2013CB329103 of 2013CB329100), the National Natural Science Foundations of China (NSFC-61173129, 71102065, 61103116, 91420102, 61472053), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20120191110026), the Fundamental Research Funds for the Central Universities under Grant (106112014 CDJZR 095502).

References

[1] G. Adomavicius, T. Alexander, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749, http://dx.doi.org/10.1109/TKDE.2005.99.
[2] G. Adomavicius, J.J. Zhang, Stability of recommendation algorithms, ACM Trans. Inform. Syst. (TOIS) 30 (4) (2012) 23, http://dx.doi.org/10.1145/2382438.2382442.
[3] V. Agarwal, K.K. Bharadwaj, A collaborative filtering framework for friends recommendation in social networks based on interaction intensity and adaptive user similarity, Social Network Anal. Min. 3 (3) (2013) 359–379, http://dx.doi.org/10.1007/s13278-012-0083-7.
[4] E. Ayday, F. Fekri, Application of belief propagation to trust and reputation management, in: Proc. Information Theory Proceedings (ISIT), 2011 IEEE Int. Symp., IEEE, St. Petersburg, Russia, 2011, pp. 2173–2177, http://dx.doi.org/10.1109/ISIT.2011.6033943.
[5] E. Ayday, F. Fekri, Iterative trust and reputation management using belief propagation, IEEE Trans. Dependable Secure Comput. 9 (3) (2012) 375–386, http://dx.doi.org/10.1109/TDSC.2011.64.
[6] A.B. Barragáns Martínez, E. Costa Montenegro, J.C. Burguillo, M. Rey-López, F.A. Mikic Fontea, A. Peleteiroa, A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition, Inform. Sci. 180 (22) (2010) 4290–4311, http://dx.doi.org/10.1016/j.ins.2010.07.024.
[7] R. Bhaumik, C. Williams, B. Mobasher, R. Burke, Securing collaborative filtering against malicious attacks through anomaly detection, in: Proc. 4th Workshop on Intelligent Techniques for Web Personalization (ITWP'06), Boston, MA, 2006.
[8] R. Bhaumik, B. Mobasher, R. Burke, A clustering approach to unsupervised attack detection in collaborative recommender systems, in: Proc. 7th IEEE Int. Conf. on Data Mining, Omaha, USA, 2011, pp. 181–187.
[9] J. Borràs, A. Morento, A. Valls, Intelligent tourism recommender systems: a survey, Expert Syst. Appl. 41 (16) (2014) 7370–7389, http://dx.doi.org/10.1016/j.eswa.2014.06.007.
[10] K. Bryan, M. O'Mahony, P. Cunningham, Unsupervised retrieval of attack profiles in collaborative recommender systems, in: RecSys '08 Proc. ACM Conf. Recommender Systems, Lousanne, Switzerland, 2008, pp. 155–162, http://dx.doi.org/10.1145/1454008.1454034.
[11] R. Burke, B. Mobasher, C. Williams, R. Bhaumik, Classification features for attack detection in collaborative recommender systems, in: Proc 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, Philadelphia, USA, 2006, pp. 542–547, http://dx.doi.org/10.1145/1150402.1150465.

[12] R. Burke, B. Mobasher, C. Williams, R. Bhaumik, Detecting profile injection attacks in collaborative recommender systems, in: 8th IEEE Int. Conf. E-Commerce Technology, 3rd IEEE Int. Conf. Enterprise Computing, E-Commerce, and E-Services, San Francisco, USA, 2006, p. 23, http://dx.doi.org/10.1109/CEC-EEE.2006.34.
[13] J. Cao, Z. Wu, B. Mao, Y.C. Zhang, Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system, World Wide Web 16 (5–6) (2013) 729–748, http://dx.doi.org/10.1007/s11280-012-0164-6.
[14] C. Carlos, O. Rodriguez, J. Rivera, J. Betancourt, M. Mendoza, E. León, E. Herrera-Viedma, A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes, Inform. Process. Manage. 49 (3) (2013) 607–625, http://dx.doi.org/10.1016/j.ipm.2012.12.002.
[15] W. Chen, Z.D. Niu, X.Y. Zhao, Y. Li, A hybrid recommendation algorithm adapted in e-learning environments, World Wide Web 17 (2) (2014) 271–284, http://dx.doi.org/10.1007/s11280-012-0187-z.
[16] Z.P. Cheng, N. Hurley, Effective diverse and obfuscated attacks on model-based recommender systems, in: Proc. 3rd ACM Conf. Recommender systems (RecSys'09). New York, USA, 2009, pp. 141–148, http://dx.doi.org/10.1145/1639714.1639739.
[17] P.A. Chirita, W. Nejdl, C. Zamfir, Preventing shilling attacks in online recommender systems, in. Proc. 7th Annual ACM International Workshop on Web Information and Data Management, WIDM '05, Bremen, Germany, 2006, pp. 67–74, http://dx.doi.org/10.1145/1097047.1097061.
[18] Y.H. Cho, J.K. Kim, Application of Web usage mining and product taxonomy to collaborative recommendations in e-commerce, Expert Syst. Appl. 26 (2) (2004) 233–246, http://dx.doi.org/10.1016/S0957-4174(03)00138-6.
[19] C. Cristian, M.Á. Sicilia, S. SáNchez-Alonso, E. GarcíA-Barriocanal, Evaluating collaborative filtering recommendations inside large learning object repositories, Inform. Process. Manage. 49 (1) (2013) 34–50, http://dx.doi.org/10.1016/j.ipm.2012.07.004.
[20] J.L. Devore, Probability and Statistics for Engineering and the Sciences, eighth ed., Cengage Learning, Boston, MA, 2011. pp. 300–344.
[21] A. Edmunds, A. Morris, The problem of information overload in business organisations: a review of the literature, Int. J. Inform. Manage. 20 (1) (2000) 17–28, http://dx.doi.org/10.1016/S0268-4012(99)00051-1.
[22] M. Gao, K.C. Liu, Z.F. Wu, Personalisation in web computing and informatics: theories, techniques, applications, and future research, Inform. Syst. Front. 12 (5) (2010) 607–629, http://dx.doi.org/10.1007/s10796-009-9199-3.
[23] M. Gao, Q. Yuan, B. Ling, Q.Y. Xiong, Detection of abnormal item based on time intervals for recommender systems, Sci. World J. 2014 (2014), http://dx.doi.org/10.1155/2014/845897.
[24] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, Mobile recommender systems in tourism, J. Network Comput. Appl. 39 (99) (2014) 319–333, http://dx.doi.org/10.1016/j.jnca.2013.04.006.
[25] I. Gunes, C. Kaleli, A. Bilge, H. Polat, Shilling attacks against recommender systems: a comprehensive survey, Artif. Intell. Rev. (2012) 1–33, http://dx.doi.org/10.1007/s10462-012-9364-9.
[26] F.M. He, X.R. Wang, B.X. Liu, Attack detection by rough set theory in recommendation system, in: IEEE Int. Conf. Granular Computing (GrC), San Jose, CA, USA, 2010, pp. 692–695, http://dx.doi.org/10.1109/GrC.2010.130.
[27] N. Hu, P.A. Pavlou, J. Zhang, Why do online product reviews have a J-shaped distribution? Overcoming biases in online word-of-mouth communication, Market. Sci. 198 (7) (2007).
[28] N.J. Hurley, Z.P. Cheng, M. Zhang, Statistical attack detection, in: RecSys'09, Proc. 3rd ACM Int. Conf. Recommender Systems, New York, USA, 2009, pp 149–156, http://dx.doi.org/10.1145/1639714.1639740.
[29] A. Jøsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decis. Support Syst. 43 (2) (2007) 618–644, http://dx.doi.org/10.1016/j.dss.2005.05.019.
[30] J.K. Kim, H.K. Kim, H.Y. Oh, Y.U. Ryu, A group recommendation system for online communities, Int. J. Inform. Manage. 30 (3) (2010) 212–219, http://dx.doi.org/10.1016/j.ijinfomgt.2009.09.006.
[31] Y. Kim, R. Phalak, A trust prediction framework in rating-based experience sharing social networks without a web of trust, Inform. Sci. 191 (2012) 128–145, http://dx.doi.org/10.1016/j.ins.2011.12.021.
[32] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, J. Riedl, GroupLens: applying collaborative filtering to Usenet news, Commun. ACM 40 (1997) 77–87, http://dx.doi.org/10.1145/245108.245126.
[33] J.S. Lee, D. Zhu, Shilling attack detection – a new approach for a trustworthy recommender system, INFORMS J. Comput. 24 (1) (2012) 117–131, http://dx.doi.org/10.1287/ijoc.1100.0440.
[34] Y. Li, L. Lu, X.F. Li, A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce, Expert Syst. Appl. 28 (1) (2005) 67–77, http://dx.doi.org/10.1016/j.eswa.2004.08.013.
[35] C. Li, Z.G. Luo, Detection of shilling attacks in collaborative filtering recommender systems, in: 2011 Int. Conf. IEEE. Soft Computing and Pattern Recognition (SoCPaR), Dalian, China, 2011, pp. 190–193, http://dx.doi.org/10.1109/SoCPaR.2011.6089138.
[36] N.N. Liu, L.H. He, M. Zhao, Social temporal collaborative ranking for context aware movie recommendation, ACM Trans. Intell. Syst. Technol. (TIST) 4 (1) (2013) 15, http://dx.doi.org/10.1145/2414425.2414440 (special section on twitter and microblogging services, social recommender systems, and CAMRa2010: Movie recommendation in context).
[37] N. Manouselis, Nikos, R. Vuorikari, F. Van Assche, Collaborative recommendation of e-learning resources: an experimental investigation, J. Comput. Assist. Learn. 26 (4) (2010) 227–242, http://dx.doi.org/10.1111/j.1365-2729.2010.00362.x.
[38] B. Mehta, W. Nejdl, Unsupervised strategies for shilling detection and robust collaborative filtering, User Model. User Adapted Inter26action 19 (1–2) (2009) 65–97, http://dx.doi.org/10.1007/s11257-008-9050-4.
[39] B. Mobasher, R. Burke, R. Bhaumik, C. Williams, Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness, ACM Trans. Internet Technol. (TOIT) 7 (4) (2007) 23, http://dx.doi.org/10.1145/1278366.1278372.
[40] M. Montaner, B. López, J. Lluís de la Rosa, A taxonomy of recommender agents on the internet, Artif. Intell. Rev. 19 (4) (2003) 285–330, http://dx.doi.org/10.1023/A:1022850703159.
[41] M.P. O' Mahony, N.J. Hurley, N. Kushmerick, G.C.M. Silvestre, Collaborative recommendation: a robustness analysis, ACM Trans. Internet Technol. 4 (4) (2004) 344–377, http://dx.doi.org/10.1145/1031114.1031116.
[42] M.P. O'Mahony, N.J. Hurley, G.C.M. Silvestre, Promoting recommendations: an attack on collaborative filtering, in: Proc. 13th Int. Conf. Database and Expert Systems Applications, Aix-en-Provence, France, 2002, pp.494–503.
[43] M.P. O' Mahony, N.J. Hurley, G.C.M. Silvestre, Towards robust collaborative filtering, Artif. Intell. Cognitive Sci. (2002) 87–94, http://dx.doi.org/10.1007/3-540-45750-X_11.
[44] L. Ott, M.T. Longnecker, An Introduction to Statistical Methods and Data Analysis, Duxbury, 2008.
[45] C. Porcel, A. Tejeda-Lorente, M.A. Martínez, E. Herrera-Viedma, A hybrid recommender system for the selective dissemination of research resources in a technology transfer office, Inform. Sci. 184 (1) (2012) 1–19, http://dx.doi.org/10.1016/j.ins.2011.08.026.
[46] R. Ronen, N. Koenigstein, E. Ziklik, N. Nice, Selecting content-based features for collaborative filtering recommenders, in: RecSys '13 Proc. 7th ACM Conf. on Recommender Systems, Hongkong, China, 2013, pp. 407–410, http://dx.doi.org/10.1145/2507157.2507203.
[47] W.G. Rong, Q.F. Wu, Y.X. Ouyang, K. Liu, Z. Xiong, Prioritised stakeholder analysis for software service lifecycle management, in: Proc. 20th IEEE Int. Conf. Web Services (ICWS'13), Santa Clara, Calif, USA, 2013, 356–363, http://dx.doi.org/10.1109/ICWS.2013.55.
[48] J.B. Schafer, J. Konstan, J. Riedl, Recommender systems in e-commerce, in: EC '99 Proc. 1st ACM Conf. Electronic Commerce, Denver, CO, USA, 1999, pp. 158–166.
[49] J. Serrano-Guerrero, E. Herrera-Viedma, J.A. Olivas, A. Cerezo, F.P. Romero, A Google wave-based fuzzy recommender system to disseminate information in university digital libraries 2.0, Inform. Sci. 181 (9) (2011) 1503–1516, http://dx.doi.org/10.1016/j.ins.2011.01.012.
[50] W.T. Teacy, N.R. Jennings, A. Rogers, M. Luck, A hierarchical Bayesian trust model based on reputation and group behaviour, in: 6th European Workshop on Multi-Agent Systems, Bath, UK, 2008.

[51] W.T. Teacy, M. Luck, A. Rogers, N.R. Jennings, An efficient and versatile approach to trust and reputation using hierarchical Bayesian modelling, Artif. Intell. 193 (2012) 149–185.
[52] A. Tejeda-Lorente, C. Porcel, E. Peis, R. Sanz, E. Herrera-Viedma, A quality based recommender system to disseminate information in a university digital library, Inform. Sci. 261 (2014) 52–69, http://dx.doi.org/10.1016/j.ins.2013.10.036.
[53] G. Vogiatzis, I. MacGillivray, M. Chli, A probabilistic model for trust and reputation, in: Proc. 9th Int. Conf. Autonomous Agents and Multiagent Systems (AAMAS'10), Toronto, Canada, vol. 1, 2010, pp. 225–232.
[54] M.G. Vozalis, K.G. Margaritis, Using SVD and demographic data for the enhancement of generalized collaborative filtering, Inform. Sci. 15 (177) (2007) 3017–3037, http://dx.doi.org/10.1016/j.ins.2007.02.036.
[55] Z. Wang, L.F. Sun, W.W. Zhu, S.Q. Yang, H.Z. Li, D.P. Wu, Joint social and content recommendation for user-generated videos in online social network, IEEE Trans. Multimedia 15 (3) (2013) 698–709, http://dx.doi.org/10.1109/TMM.2012.2237022.
[56] J. Wang, Y. Zhang, Opportunity model for e-commerce recommendation: right product; right time, in: SIGIR '13 Proc. 36th Int. ACM SIGIRConf. on Research and Development in Information Retrieval, Dublin, Ireland, 2013, pp. 303–312, http://dx.doi.org/10.1145/2484028.2484067.
[57] L.L. Wei, J.H. Ma, R.F. Yan, An Introduction to Probability and Statistics, Science press, Beijing, China, 2012.
[58] C. Wei, R. Khoury, S. Fong, Web 2.0 recommendation service by multi-collaborative filtering trust network algorithm, Inform. Syst. Front. 15 (4) (2013) 533–551, http://dx.doi.org/10.1007/s10796-012-9377-6.
[59] C. Williams, B. Mobasher, R. Burke, J. Sandvig, R. Bhaumik, Detection of obfuscated attacks in collaborative recommender systems, in: Proc. ECAI06 Workshop on Recommender Systems, 17th European Conf. Artificial Intelligence (ECAI'06), Riva del Garda, Italy, August, 2006.
[60] C.A. Williams, B. Mobasher, R. Burke, Defending recommender systems: detection of profile injection attacks, SOCA 1 (3) (2007) 157–170, http://dx.doi.org/10.1007/s11761-007-0013-0.
[61] Z. Wu, J.J. Wu, J. Cao, D.C. Tao, HySAD: a semi-supervised hybrid shilling attack detector for trustworthy product recommendation, in: Proc. 18th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, Beijing, China, 2012, pp. 985–993, http://dx.doi.org/10.1145/2339530.2339684.
[62] S. Zhang, A. Chakrabarti, J. Ford, F. Makedon, Attack detection in time series for recommender systems, in: Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, Philadelphia, USA, 2006, pp. 809–814, http://dx.doi.org/10.1145/1150402.1150508.
[63] Q. Zhang, Y. Luo, C.L. Weng, M.L. Li, A trust-based detecting mechanism against profile injection attacks in recommender systems, in: 3rd IEEE Int. Conf. Secure Software Integration and Reliability Improvement (SSIRI2009), Shanghai, China, 2009, pp. 59–64, http://dx.doi.org/10.1109/SSIRI.2009.12.
[64] S. Zhang, Y. Ouyang, J. Ford, F. Makedon, Analysis of a low-dimensional linear model under recommendation attacks, in: SIGIR '06 Proc. 29th Annual Int. ACM SIGIR Conf. Research and Development in Information Retrieval, Seatle, USA, 2006, pp. 517–524, http://dx.doi.org/10.1145/1148170.1148259.
[65] MovieLens,2014. <http://grouplens.org/datasets/movielens/>.