Journal of Intelligent Information Systems (2022) 59:93–119
https://doi.org/10.1007/s10844-021-00689-y

Rating behavior evaluation and abnormality forensics analysis for injection attack detection

Zhihai Yang[1,2] · Qindong Sun[1,2,3] · Zhaoli Liu[1,2] · Jinpei Yan[1,2] · Yaling Zhang[1,2]

Received: 10 June 2021 / Revised: 17 November 2021 / Accepted: 17 November 2021 /
Published online: 2 December 2021

Abstract

Collaborative recommender systems (CRSs) have become an essential component in a wide range of e-commerce systems. However, CRSs are also easy to suffer from malicious attacks due to the fundamental vulnerability of recommender systems. Facing with the limited representative of rating behavior and the unbalanced distribution of rating profiles, how to further improve detection performance and deal with unlabeled real-world data is a long-standing but unresolved issue. This paper develops a new detection approach to defend anomalous threats for recommender systems. First, eliminating the influence of disturbed rating profiles on abnormality detection is analyzed in order to reduce the unbalanced distribution as far as possible. Based on the remaining rating profiles, secondly, rating behaviors which belong to the same dense region using standard distance measures are further partitioned by exploiting a probability mass-based dissimilarity mechanism. To reduce the scope of determining suspicious items while keeping the advantage of target item analysis (TIA), thirdly, suspected items captured by TIA are empirically converted into an associated item-item graph according to frequent patterns of rating distributions. Finally, concerned attackers can be detected based on the determined suspicious items. Extensive experiments on synthetic data demonstrate the effectiveness of the proposed detection approach compared with benchmarks. In addition, discovering interesting findings such as suspected items or ratings on four different real-world datasets is also analyzed and discussed.

This work was supported in part by the National Natural Science Foundation of China under Grant 62172331 and 62102310, in part by the Youth Innovation Team Construction of Shaanxi Provincial Department of Education under Grant 21JP081, in part by the China Postdoctoral Science Foundation under Grant 2020M683689XB, in part by the Natural Science Founds of Shaanxi under Grant 2020JQ-646 and 2021JQ-486, and in part by the Youth Innovation Team of Shaanxi Universities under Grant 2019-38.

✉ Zhihai Yang
zhyang_xjtu@sina.com

[1] School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China

[2] Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an, China

[3] School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China

Keywords Abnormality forensics · Malicious attack · Rating behavior · Attack detection · Recommender system

1 Introduction

Personalization collaborative recommender systems (PCRSs) play a crucial role in online e-commerce platforms, which aim to recommend a user items (e.g., products on Amazon, videos on YouTube) that match his/hers preference (Luo et al., 2017). According to collaborative filtering techniques that are utilized to analyze the user-item rating score matrix, collaborative filtering can be roughly classified to matrix-factorization-based, neighborhood-based, graph-based, and association-rule-based (Fang et al., 2018). However, PCRSs are highly vulnerable to malicious attacks, such as profile injection attacks (a.k.a., *shilling* attacks) (Burke et al., 2006; Gunes et al., 2012), pollution attacks (Xing et al., 2013), fake co-visitation injection attacks (Yang et al., 2017), poisoning attacks to graph-based recommender systems (Fang et al., 2018), practical data poisoning attacks against next-item recommendation (Zhang et al., 2020), etc. Attackers empirically inject fake user profiles with carefully crafted ratings to PCRSs, in order to spoof PCRSs for recommending target items (e.g., a new product on Amazon, a video advertisement on YouTube) to as many victim users as possible. The reputation of PCRSs and the fairness of the virtual market have been destroyed by these malicious threats. Thus, securing collaborative filtering recommender systems from malicious threats have become an important issue with increasing popularity of recommender systems. Naturally, the demand for defending malicious threats toward online recommender systems is becoming increasingly urgent.

Although previous researches have shown promising results, defending such attacks is still an unresolved issue and has not reached a full level of performance (Gunes et al., 2012; Zhang et al., 2015; Zhou et al., 2014). Specifically, how to reduce false alarm rates (Burke et al., 2006) and simultaneously retain high detection rates is a big challenge. In addition, how to referentially guide abnormality forensics analysis for real data based on heuristic knowledge learned from synthetic data is also an open issue. This research tries to explore a possible way to analyze instructively abnormality forensics on real data, and expects to put forward a reference to deal with similar threats focused on advanced recommendation techniques.

Most previous investigations have based stepwise detection frameworks on a straightforward idea that mining similarity patterns between rating behaviors (Zhou et al., 2014), and analyzing the distribution of target items (Zhou et al., 2014) can be utilized to identify anomalous profiles. Nevertheless, the challenging issues are the difficulties of (1) extracting rating behavior features facing with sparse ratings; (2) distinctively evaluating the similarity between rating behaviors which belong to the same dense region; and (3) reducing the scope of analyzing suspicious items, which may partly limit the applicability of the presented techniques. Moreover, how to construct a strategy that can be used to spot anomalous ratings for real-world application is also extremely desirable.

In this paper, we present a new stepwise detection approach to spot anomalous rating behaviors for recommender systems. First, we investigate how to eliminate disturbed rating profiles according to user's activity and item's popularity, in order to reduce the difficulty of characterizing sparse rating behaviors and the dimension of original rating matrix. Based on remaining rating profiles, secondly, rating behaviors which belong to the same dense region using standard distance measures are further partitioned from the perspective of probability

mass-based dissimilarity. To reduce the scope of determining suspicious items while keeping the advantage of target item analysis (TIA), suspected items captured by TIA are empirically converted into an associated item-item graph according to frequent patterns of rating distributions. Finally, suspicious items (nodes in a constructed graph) can be further determined by exploiting the characteristics of topological structure of nodes. In particular, analyzing abnormality forensics on unlabeled real-world data is provided to discover anomalous ratings or items. Additionally, we propose four diverse abnormality forensics metrics to comprehensively determine the existence of abnormality.

The major contributions of this paper are three-fold as follows:

1. We provide a feasible way for abnormality detection facing with the imbalanced distribution of rating behavior by eliminating disturbed profiles according to both user's activity and item's popularity.
2. To recognize anomalous users mimicked from genuine users, we propose to incorporate probability mass-based measures rather than geometric distance-based measures as the fundament to determine the closest neighbourhood. We propose to incorporate the synchronicity and normality of nodes to reduce the scope of determining suspected items.
3. We propose a new detection approach to defend different profile injection attacks. Extensive experiments have been implemented and analyzed in different cases. Additionally, to further analyze suspicious items and to discover interesting findings on real-world data, we also develop four abnormality forensics metrics including the intrinsic association between ratings and reviews, the aggregation degree of ratings, the distribution of rating intention, and the difference of degree distribution between before and after removing suspicious items.

2 Related work

Malicious attacks focused on recommender systems have been developed in the past decade (Gunes et al., 2012; Wu et al., 2014). Detecting profile injection attacks and suspicious ratings for online recommender systems have naturally received much attention. However, investigating efficient and extensible detection approaches is still desirable especially for large scale real-world datasets. This section only discusses researches related to our work from the perspective of sparse rating behavior analysis, suspicious items analysis, and abnormality detection for real-world data.

From the perspective of classification or clustering, abnormality detection can be considered as a problem of unbalanced classification or clustering. In general, the number of genuine (authentic) profiles is larger than the number of attack or anomalous profiles in recommender systems. Additionally, facing with sparse rating data, feature characterization for rating behaviors based on unbalanced distribution has always been a big challenge. Previous researches have focused on eliminating sparse or disturbed rating profiles using stepwise detection mechanisms. Firstly, Mehta et al. (2007) developed statistical methods to detect shilling attacks. After that, Chung et al. (2013) also developed an unsupervised detection method against shilling attacks based on Beta probability. Shortly, Zhou et al. (2014) proposed a detection approach for identifying group attack profiles. Additionally, Yang et al. (2016) proposed a three-phase detection method to spot anomalous ratings. They also investigated a supervised detection method to deal with the unbalanced distribution of rating behaviors (Yang et al., 2016). In addition, Yang et al. (2018) presented a detection approach

and analyzed anomalous rating behaviors on real data. However, the precision and recall of the detection method are not impressive when the filler sizes are small. Recently, Zhang and Wang (2020) investigated a group shilling attack detection approach to defend shilling attacks. Analyzing target items is proved to be an effective way for improving detection performance, which is favorable to the determination of suspected items. Previous researches have focused on suspected item analysis from different perspectives. Zhou et al. (2014) discussed the distribution of items in order to capture suspected items. By combining the advantage of stepwise detection, an impressive detection strategy has been investigated to defend shilling attacks. Xu and Zhang (2019) analyzed suspicious items and proposed a detection method based on time series analysis and trust features. Recently, Zhang et al. (2020) developed a graph embedding-based method to detect group shilling attacks. The ultimate goal of abnormality detection is to serve real-world application. Recently, Yang et al. (2017) investigated a stepwise detection method to detect shilling attacks. They developed a detection framework (Yang et al., 2020) to infer malicious attack behaviors. In addition, discovering interesting findings on real data is also analyzed.

The above efforts suggest that target item analysis, the elimination of disturbed profiles, and stepwise detection mechanisms, have considerable potential for accurate detection. This work, different from existing studies: (1) aims to explore the partition of rating behavior according to the representation of rating behavior and behavior dependence measurement; (2) incorporates synchronicity and normality analysis for accurate detection; and (3) investigates a new detection approach and discovers interesting findings on real data.

3 Overview

In this paper, we investigate a heuristic detection method as shown in Fig. 1. The ultimate goal of the detection framework is discover suspicious findings for real-world data using heuristic knowledge inspired from labeled data. Furthermore, suspected findings need to be further determined by exploiting abnormality forensics metrics. Each component of the detection framework is described below:

1. Eliminating disturbed rating profiles: To deal with the imbalanced distribution of rating data, disturbed rating profiles are eliminated in advance from the perspectives of both

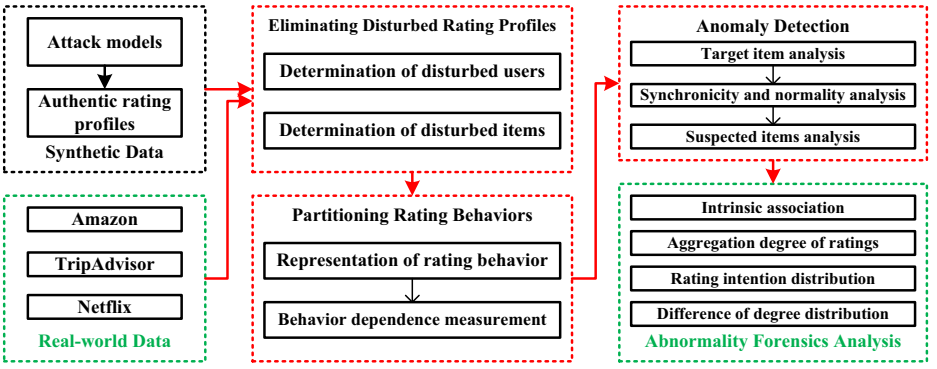


Fig. 1 The framework of the proposed detection approach

the activity of user and the popularity of item. Meanwhile, the original sparse rating matrix is reduced partly. More details will be discussed in Section 4.

2. Partitioning rating behaviors: Characterizing rating behaviors of users from multiple perspectives is desired based on the remaining data, which is favorable to construct feature space of rating behaviors. Dense regions of rating behaviors are further divided using probability mass rather than standard distance measurements in order to distinguish attack profiles from genuine profiles as far as possible.
3. Anomaly detection: Based on the result of rating behavior partition, the scope of determining suspicious items is necessary to be narrowed to reduce false alarm rates especially for the case of small attack sizes. To this end, suspicious items determined by target item analysis are mapped to an associated item-item graph. Additionally, these suspected items are further filtered by exploiting the synchronicity and normality characteristics of associated nodes, which will be introduced in Section 6.1.
4. Abnormality forensics analysis: Facing with unlabeled real-world data, analyzing anomalous ratings or items according to diverse abnormality forensics metrics is provided. It is noteworthy that heuristic knowledge such as target item distribution, frequent patterns of target items extracted from the rating matrix, etc., inspired from synthetic data can be exploited to try to discover interesting findings on real data.

4 Eliminating disturbed profiles

4.1 Determining disturbed users

Attackers and genuine users have different rating intentions on the same target items in reality. Genuine users usually represent a more natural rating behavior on target items. The final ratings on target items are depended on the interesting of genuine users. To investigate user's rating behaviors, the activity of user, which is defined as the number of items rated by a user, is calculated as shown in Fig. 2a(1). Some genuine users only rate very few items such

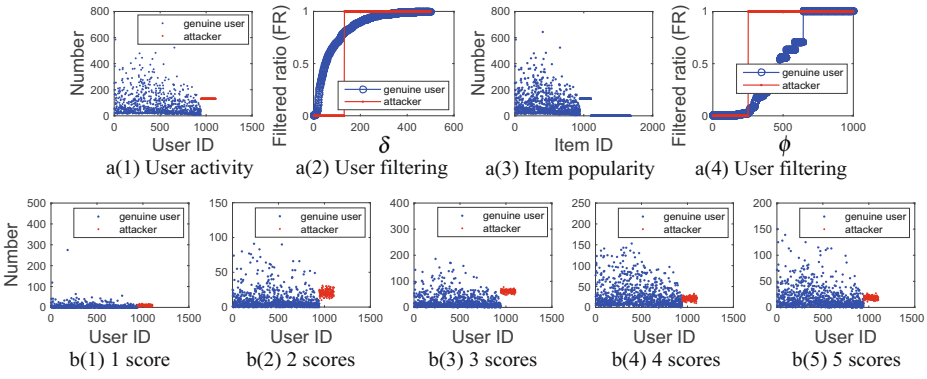


Fig. 2 Distributions of users and items, where a(1) and a(2) respectively show the activity of user and the filtered ratio of users using δ; a(3) and a(4) respectively show the popularity of item and the filtered ratio of users using ϕ; b(1)-b(5) shows the frequency of ratings of users with diverse rating scores. The Reverse Bandwagon attack is taken for example, where the attack size is 17.0% and the filler size is 7.3%

as one or two items, called *inactive* users. Similarly, some of them rate a lot of items, called *active* users. Note that, attackers mimic the rating details of *anchor* users (i.e., mimicked authentic users) for constructing attack profiles in order to manipulate final recommendations. To be the neighbors of anchor users, attackers carefully inject fake user profiles with crafted ratings into the system, in order to make higher similarities between the anchor users and these fake users (Burke et al., 2006; Mobasher et al., 2007). In this sense, neither active users nor inactive users can be considered as anomalous users who imply attack behaviors. In this paper, we call these users *disturbed* users. The main reasons are a) inactive users are difficult to be the neighbors of anchor users in recommender systems due to their limited influence, even if they have shilling rating behaviors. and b) it is unrealistic that attackers focus on active users to make attack profiles due to the cost of attacks. Moreover, few users are active in recommender systems (Yang et al., 2017).

Algorithm 1 Eliminating disturbed profiles.

Require:
Original rating matrix $\mathscr{M}_{m\times n}$;
Parameters δ and ϕ.
Ensure:
Eliminated users $\mathscr{U}^e$.
Determining disturbed users:
Initialize a set of eliminated users $\mathscr{U}^e = null$;
Calculate degree N_{u_i} for each user $u_i \in U$ in $\mathscr{M}_{m\times n}$, where U is the set of all users in $\mathscr{M}_{m\times n}$;
if $N_{u_i} < \delta$ **then**
    $\mathscr{U}^e \leftarrow u_i$;
end if
Determining sparse items:
Initialize a set of eliminated items $\mathscr{I}^e = null$;
Calculate degree N_{i_k} for each item $i_k \in I$, where I is the set of all items in $\mathscr{M}_{m\times n}$;
if $N_{i_k} < \phi$ **then**
    $\mathscr{I}^e \leftarrow i_k$;
end if
for each item $u_i \in U$ in M **do**
    if u_i only rated items belong to $\mathscr{I}^e$ **then**
        $\mathscr{U}^e \leftarrow u_i$;
    end if
end for
return $\mathscr{U}^e$;

Intuitively, abnormality detection can be considered as a process of gradually filtering out disturbed rating profiles (authentic profiles) while keeping all concerned attack profiles as far as possible. In addition, the unbalanced distribution between attack profiles and genuine profiles can not be ignored. To partly reduce the unbalanced distribution, a part of genuine profiles are eliminated in advance using an empirical threshold (i.e., parameter δ as shown in Fig. 2a(2)) of user activity. Algorithm 1 describes the process of determining disturbed users. Note that, a reasonable threshold δ can be determined through experiments.

4.2 Determining disturbed items

As aforementioned, disturbed users can be empirically determined and eliminated in advance. To investigate the distribution of items, similarly, we also calculated the popularity of each item. Inspired from the distribution of users, items can be categorized as two types, *popular* items and *unpopular* items. For the former, concretely, popular items have been rated by a lot of users. In general, popular items have high reputations. It is difficult to push or nuke a popular item to achieve the goal of shilling attacks. By considering the cost of attack, it is inadvisable to choose popular items as the target items from the perspective of attackers (Yang et al., 2016). For the latter, some items have been rated by a few users such as one or two users (see Fig. 2a(3)), called unpopular items or novel items. Unpopular items may represent a low reputation or bad quality. It is impossible to choose an unpopular item as the target item for anomaly detection. Due to the fact that, shilling attackers consistently promote or demote the target item, finally leading to a relatively high popularity of the target item. In this paper, an empirical threshold ϕ (as shown in Fig. 2a(4)) for determining unpopular items is used to eliminate disturbed items. Similarly, ϕ also needs to be determined through experiments.

Figure 2a(1) and 2a(3) also provide the frequency information of ratings in addition to the distribution of both user activity and item popularity. In shilling attacks, a group of shilling attackers consistently give well-designed ratings to a same target item (termed *single-target* attacks) or multiple same target items (called *multiple-target* attacks). Thus, the frequency that an attacker offered ratings to target items is depended on the number of target items (i.e., single-target attacks or multiple-target attacks). Noth that, we only analyze the single-target attacks in our experiments. To analyze the distribution of users that they rated a special score (e.g., r_{min} or r_{max}) on items, we implemented a list of experiments as shown in Fig. 2b(1)-b(5). The Reverse Bandwagon attack with 17% attack size and 7.3% filler size is taken for example, we provide the frequency of ratings of attackers with diverse rating scores. We can observe that the frequency of ratings of attackers is lower than that of some genuine users. In other words, it is not easy to discriminate attackers and genuine users by purely counting the frequency of their ratings.

Algorithm 1 also describes the process of determining disturbed items. Note that, the number of popular items is not too high (Yang et al., 2017). Prematurely eliminating popular items may affect the characteristics of rating behavior later. For instance, popular items are selected to construct attack profiles in Bandwagon attacks (see Section 7.1.2). In this paper, only unpopular items are investigated to determine anomalous users and are finally used to eliminate disturbed rating profiles.

5 Partitioning rating behaviors

5.1 Representation of rating behavior

Characterizing rating behaviors is a crucial task in shilling attack detection. Despite promising results in the previous researches (Gunes et al., 2012; Wu et al., 2014), several obvious issues are worth further investigation as follows:

1. Most of rating behavior features are only effective for special attacks such as segment attack, bandwagon attack, etc. The design of these features is intuitively derived from the corresponding attacks (Burke et al., 2006; Mobasher et al., 2007).

2. General features including WDA, MeanVar, etc., are favorable to get desirable detection performance. Nevertheless, the generalization of detection model is constrained according to these features (Burke et al., 2006).
3. Exploiting similarity-based rating behavior features is more effective to distinguish between anomalous and authentic rating behaviors compared with other features (Burke et al., 2006; Mobasher et al., 2007). However, calculating the similarity between rating behaviors is very time-consuming in reality (Yang et al., 2016), especially for large-scale real-world data.
4. Global and local correlation between behavior features has an impact on anomaly detection. It is not an optimistic way to solely rely on the independent contribution between features (Yang et al., 2017).

To this end, it is instructive to see that mining the difference between user rating motivations is beneficial to obtain a new perspective for measuring rating behaviors. Investigating representative and effective behavior features is also desirable. Based on the existing rating behavior features, representative features are carefully selected using an adaptive structure learning framework (Yang et al., 2017), which can be used to construct the final feature space. The details of these selected features are described as follows:

Weighted degree of agreement (WDA) is used to calculate the sum of the differences of ratings from the item's average rating divided by the item's rating frequency, which is defined as follows:

$$WDA_u = \sum_{i=0}^{N_u} \frac{\left| r_{u,i} - \overline{r_i} \right|}{NR_i}, \tag{1}$$

where $\overline{r_i}$ is the average rating of item i. NR_i is the number of ratings given to item i (Burke et al., 2006).

Mean variance (MeanVar) is to iterate through all the lowly-rated (for nuke attack) or highly-rated (for push attack) items. We choose each item in turn as the possible target and calculate the average variance between filler items (Burke et al., 2006) and the overall average rating for detecting average attacks.

$$MeanVar_u = \frac{\sum_{j \in P_{u,F}} (r_{u,j} - \overline{r_u})^2}{\left| P_{u,F} \right|}, \tag{2}$$

where $P_{u,T} = \{i \in P_u, \}$ such that $r_{u,i} = r_{max}$ (or r_{min} for nuke attacks) is the set of ratings that are potential targets. $P_{u,F} = P_u - P_{u,T}$ denotes the rest of rating profiles (Burke et al., 2006).

Filler mean target difference (FMTD) is a partitioning feature for detecting segment attacks. We calculate the rating difference of item rated by user u with r_{min} or r_{max} compared with all items rated by user u except for the items rated with r_{min} or r_{max}.

$$FMTD_u = \left| \frac{\sum_{i \in P_{u,T}} r_{u,i}}{\left| P_{u,T} \right|} - \frac{\sum_{k \in P_{u,F}} r_{u,k}}{\left| P_{u,F} \right|} \right|, \tag{3}$$

where $P_{u,T}$ is the set of items that have been rated with r_{max}. $P_{u,F}$ is the remained items (Burke et al., 2006).

Filler average correlation (FAC) calculates the correlation of rating for each user.

$$FAC_u = \frac{\sum_i (r_{u,i} - \overline{r_i})}{\sqrt{\sum_i (r_{u,i} - \overline{r_i})^2}}, \tag{4}$$

where I_u is the set of items rated by user u (Yang et al., 2016).

Filler size with popular items in itself (FSPII) is the ratio between the number of popular items rated by user u and the number of entire items rated by user u (Yang et al., 2017; Yang et al., 2016).

$$FSPII_u = \frac{\sum_{i=1}^{K} O(r_{u,i})}{\sum_{j=1}^{|I|} O(r_{u,j})}, \tag{5}$$

where K denotes the boundary of popular items and unpopular items. $O(r_{u,i})$ is 1 if user u rated item i, 0 otherwise. $|I|$ is the number of all items in the system.

Filler size with minimum rating in total items (FSMI) denotes the ratio between the number of items rated by user u with the minimum rating score and the number of entire items rated by user u.

$$FSMI_u = \frac{\sum_{i=1}^{|I|} O(r_{u,i} = r_{min})}{|I|}, \tag{6}$$

where $O(r_{u,i})$ is 1 if user u rated item i with r_{min}, 0 otherwise Yang et al. (2017, 2016).

It is noteworthy that the above behavior features may easily lead to low representation performance especially facing with sparse rating data. For instance, a user's average rating on items may not be reliable if the user did not leave several ratings on the items. In order to partly reduce the impact of data sparsity on behavior representation, we empirically eliminate as many disturbed profiles as possible including *unpopular* items (rated by few users) and *inactive* users (who have rated few items) before behavior representation. Note that, Algorithm 1 provides the basic process of eliminating disturbed profiles. Based on the above-mentioned rating behavior features, analyzing the correlation between rating behaviors from the perspective of probability distribution will be further discussed below.

5.2 Behavior dependence measurement

Only relying on the existing behavior features is not enough to achieve promising detection performance in some cases of profile injection attacks (Burke et al., 2006; Mobasher et al., 2007). Despite distance-based similarity measurement methods such as *Pearson* correlation coefficient, cosine coefficient, etc. (Mobasher et al., 2007), have been investigated to capture anomalous rating behaviors, *shilling* rating behaviors can not be fully detected especially for distinguishing between attack profiles and the mimicked genuine profiles. Furthermore, standard distance measurement methods do not possess the key property of dissimilarity. For instance, the characteristic where two instances in a dense region are less similar to each other than two instances of the same distance in a sparse region. Related studies have suggested that the dissimilarity of data dependent is a better measure compared with the data independent geometric model based distance measure (Ting et al., 2016).

Generally, shilling attackers mimic rating details of some authentic users to construct well-designed anomalous profiles in order to manipulate recommendations. The mimicked users are naturally similar to attackers, as well as similar to their original neighbors. Figure 3 shows a simple schematic process. Dissimilar users are more scattered compared with similar users which belong to a dense region. Distinguishing attackers from genuine users in a dense region is therefore desired. To demonstrate the weak points of standard distance measures, five diverse distance-based similarity measurement methods are implemented for filtering out disturbed rating profiles compared with an employed mass-based dissimilarity method (see Section 7.3). These kinds of anomalous users and authentic users are difficult to

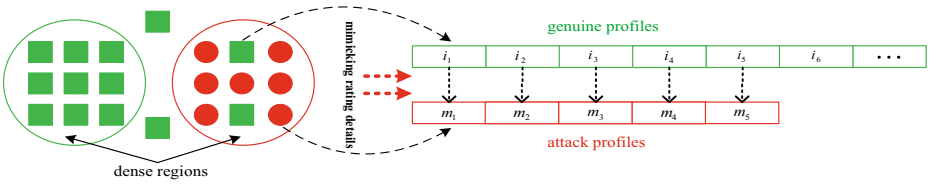


Fig. 3 A diagram for mimicking rating details in two different dense regions, where red and green nodes denote attackers and genuine users, respectively

be partitioned using both the selected rating behavior features and distance-based similarity measures, finally leading to high false alarm rates.

To reduce the limitation of both the representation of rating behavior and distance-based measurement, in this paper, we employ a probability mass based rather than the distance based as the means to find the closest match neighbourhood, in order to herald a fundamental change of perspective. Concretely, give a data sample D, a probability density function F, and let $H \in \mathscr{H}(D)$ be a hierarchical partitioning model of the space in non-overlapping and non-empty regions, the smallest local region covering instances x and y w.r.t. H and D is defined as follows:

$$R(x, y|H; D) = \arg \min_{r \subset H \ s.t. \{x,y\} \in r} \sum_{z \in D} \mathbb{I}(z \in r), \tag{7}$$

where $\mathbb{I}(\cdot)$ is an indicator function. Based on the smallest local region $R(x, y|H; D)$, the mass-based dissimilarity of x and y w.r.t. D and F is defined as follows:

$$m(x, y|D, F) = E_{\mathscr{H}(D)}[P_F(R(x, y|H; D))], \tag{8}$$

where the expectation is taken over all models in $\mathscr{H}(D)$. $P_F(\cdot)$ is the probability w.r.t. F.

To evaluate the similarity between instances, a recursive partitioning scheme termed *isolation Forest* is exploited to construct an *iForest* (Ting et al., 2016). The implementation process of measuring the similarity between instances can be summarized as follows:

1. Based on the remaining users generated from the process of eliminating disturbed profiles, an *iForest* with t *iTrees* and a partitioning structure R is firstly built. Each *iTree* is independently constructed by exploiting a subset $\mathscr{D}^s \subset D$, where $|\mathscr{D}^s| \subset \psi$. ψ denotes the sub-sampling size used to build each *iTree*;
2. A randomly selected split is utilized at each internal node of an *iTree* in order to partition the sample set at the node into two non-empty subsets, until the maximum tree height h is reached or every point is isolated.
3. Based on the constructed *iForest*, all instances in D are traversed through each tree for recording the mass of each node.
4. Calculating the sum of mass of the lowest nodes containing two test points (instances) x and y which are parsed through each *iTree*. The mass-based dissimilarity between x and y is the mean of all mass values over t *iTrees* as defined below:

$$m_e(x, y|D) = \frac{1}{t} \sum_{i=1}^{t} \tilde{P}(R(x, y|H_i; D)). \tag{9}$$

The similarity between x and y is finally calculated by $1 - m_e(x, y|D)$.

6 Detecting anomalous rating profiles

6.1 Determination of suspected items

6.1.1 Distribution and association of items

Eliminating disturbed rating profiles and using probability mass-based dissimilarity are partly effective to distinguish attack profiles from genuine profiles. Nevertheless, attackers and the mimicked genuine users which belong to a dense region can not be fully partitioned, which leads to slightly higher false alarms especially when the attack sizes are small.

Previous studies (Yang et al., 2016; Yang et al., 2017; Zhou et al., 2014) have investigated suspected items by analyzing the distribution of item. However, the number of undetermined target items that are considered as attack target items is too much to recognize authentic users. Algorithm 2 provides the basic process of analyzing the distribution of target items and spotting anomalous users. In Algorithm 2, we first calculate the number of users who have rated the minimum rating r_{min} (for nuke attacks) or the maximum rating r_{max} (for push attacks) for each item. An empirical threshold ε is used to determine suspected items (target items). Concretely, an item will be considered as a suspected item if the number of users who have rated the minimum rating r_{min} on the item is greater than ε. Based on the suspected items, attackers who have rated the suspected items can be finally detected. The underlying assumption of target item analysis is that shilling attackers focus on the same target items with the maximum or minimum rating. Group rating behaviors make a difference of rating distribution of the suspected items compared with other items. Detecting these target items which imply consistent rating behavior is naturally used to capture attackers. Unfortunately, some normal items are wrongly considered as target items, which may cause high false alarm rates.

In order to reduce the concerned range of suspected items determined by target item analysis, intuitively, filtering out unsuspected items as many as possible is feasible. Moreover, characterizing distribution features of items and rating behaviors of users is naturally a challenging task. In this paper, suspected items determined by Algorithm 2 are firstly converted into an associated item-item graph. The topological structure of graph is then exploited to further eliminate disturbed target items. Specifically, frequent patterns of both user history ratings and item distribution are firstly investigated to construct the item-item graph. As shown in Fig. 4, co-occurrence rating behaviors imply co-existence relationships between items and rating intentions. These co-existence relationships and rating intentions can be detailed in the following two aspects:

1. The rating vector of an attacker consists of selected items, filler items, and target items (Burke et al., 2006). Taking the nuke attack for example, shilling attackers focus on the same target items using the minimum rating. The corresponding ratings of selected items or filler items may also be rated with the minimum rating, such as Bandwagon attacks. As demonstrated in Fig. 4, the rating profiles of an attacker (marked as a green ellipse) are likely to include multiple target items which are determined by Algorithm 2. The frequent pattern (frequent pattern 2) of pairing (each pair of suspected items) is used to construct edges in the item-item graph, which simultaneously retains the intention of attackers.
2. A suspected item is determined by the number of users who rated the item with the minimum rating. In this way, two another situations should be considered, which may lead to a false determination for target items. Firstly, unpopular items or bad

word-of-mouth items are easily recognized as the target items due to the frequent and negative ratings such as r_{min}. Secondly, the selected items in attack profiles also are easily recognized as target items due to the group rating behavior on the selected items. Shilling ratings focused on the selected items or bad word-of-mouth items represent a frequent rating pattern (frequent pattern 1 in Fig. 4)).

Algorithm 2 Constructing an associated item-item graph.

Require:
Rating matrix of the remaining profiles M_r;
The number of attackers N_a, parameter ε.
Ensure:
An associated item-item graph G.

$\mathscr{I}^s = null$, $\mathscr{U}^s = null$;
$\{(i, N_i)\}_i^{|I|} = \{(i, N_i) \mid N_i$ is the number of ratings on item $i \in I$ with r_{min} or r_{max} in $M_r\}$;
for each item $i \in I$ **do**
  if $N_i > \varepsilon$ **then**
    $\mathscr{I}^s \leftarrow i$;
  end if
end for
for each user $u \in U$ **do**
  if u rated item i $(i \in \mathscr{I}^s)$ with r_{min} or r_{max} **then**
    $\mathscr{U}^s \leftarrow u$;
  end if
end for
for each pair of items i_m and i_n, $(i_m, i_n \in \mathscr{I}^s)$ **do**
  $count_m = 0$, $count_n = 0$;
  for each user $u \in \mathscr{U}^s$ **do**
    /* Frequent pattern 2 */
    Calculate the number of ratings, N_{u_m}, that user u rated item i_m with r_{min} (for nuke attack);
    Calculate the number of ratings, N_{u_n}, that user u rated item i_n with r_{min} (for nuke attack);
    if $count_m \geq 1$ and $count_n = 0$ **then**
      $count_m = count_m + 1$;
    end if
    if $count_m = 0$ and $count_n \geq 1$ **then**
      $count_n = count_n + 1$;
    end if
    /* Frequent pattern 1 */
    if $count_m \geq N_a$ or $count_n \geq N_a$ **then**
      Create an edge between nodes (items) i_m and i_n in graph G;
      break;
    end if
  end for
end for
return G;

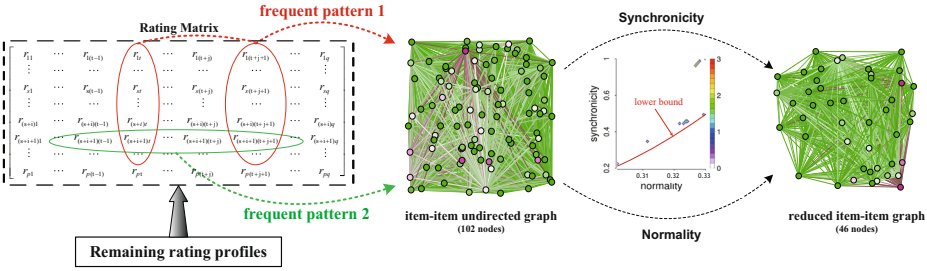


Fig. 4 Mining frequent patterns (frequent patterns 1 and 2) from a rating matrix and converting the rating matrix into an item-item graph. The characteristics of synchronicity and normality of nodes are used to comprehensively determine suspicious nodes from an original item-item graph

Finally, an item-item graph is constructed jointly based on both frequent patterns 1 and 2. A new edge is created between two items if these two suspected items appear in the pair as detailed in Algorithm 2. In the constructed graph, nodes are connected by dense edges. The connectivity between two nodes is determined by the scale of the attack as shown in Fig. 4. In other words, the greater the degree of a suspected node (item) is, the more likely it is to be an attacked target item. In Fig. 4, the greater the degree of a node, the more green the node. Likewise, the smaller the degree of a node, the more violet the node. Note that, there are more alternative target items. It is necessary to further narrow down the range of suspected items in order to reduce the false alarm rates for shilling attack detection.

6.1.2 Synchronicity and normality analysis

Converting the remaining rating matrix into an associated item-item graph expects to investigate the existence pattern of anomalous nodes from the perspective of topological structure. In this paper, synchronicity and normality of nodes are jointly incorporated to capture the concerned target items. The objective goal of exploiting synchronicity and normality is to further eliminate disturbed target items. Concretely, the synchronicity is used to qualify how synchronous the node u's targets are. The normality is used to qualify how normal u's targets are Jiang et al. (2014).

Algorithm 3 Catching suspicious nodes.

Require:
The associated item-item graph G;
Ensure:
Suspicious items G_r.
1: **for** each node $v \in G$ **do**
2: Calculate synchronicity $sync(v)$ and normality $norm(v)$ using (10) and (11), respectively;
3: **end for**
4: Give SN-plot $sync(v)$ and normality $norm(v)$;
5: Determine suspicious nodes according to the distance between nodes and lower bound;
6: Retain determined suspicious nodes and reconstruct a reduced item-item graph G_r;
7: **return** G_r;

The synchronicity of node u, $sync(u)$, as the synchronicity of u's target nodes such as the average closeness between each pair of u's targets (v, v') is defined as follows,

$$sync(u) = \frac{\sum_{(v,v')\in\mathscr{N}(u)} c(v, v')}{d(u) \times d(u)}, \tag{10}$$

where $d(u) = |\mathscr{N}(u)|$ denotes the degree of note u (the number of u' neighbors, $|\mathscr{N}(u)|$). The closeness (similarity) between two nodes v and v' is calculated based on a specific grid where each node is mapped to the specific grid (Jiang et al., 2014), which is defined as $c(v, v') = 1$ if nodes v and v' are in the same grid, 0 otherwise.

The normality of node u as the normality of u's target nodes such as the average closeness between each pair of u's targets and other nodes (v, v') is defined as follows,

$$norm(u) = \frac{\sum_{(v,v')\in\mathscr{N}(u)} c(v, v')}{d(u) \times N}, \tag{11}$$

where N is the number of all nodes in G.

Determining a suspected node (item) is based on synchronicity-normality plot (SN-plot) as shown in Fig. 4. A suspected node u has uncommonly large $sync(u)$ and abnormally small $norm(u)$. Both values of synchronicity and normality range from 0 to 1. The normal shape of SN-plot is considered as the basis for catching suspected nodes. Therefore, the SN-plot should be firstly generated. Given a SN-plot, calculating the distance between each node in SN-plot and lower limit (Jiang et al., 2014) is used to detect the outliers in the synchronicity and normality plot. The assumption is that, nodes that are too far away from the lower limit. It means that there is a parabolic lower limit in the synchronicity-normality plot for any background distribution (Jiang et al., 2014). Concretely, the residual score $r(u)$ of a suspected node u's synchronicity indicates how suspicious it is. The set of suspected nodes $\mathscr{U}_{sync}$ includes the nodes whose suspiciousness is α times standard deviations (e.g., σ) away from the mean μ, which is defined as follows,

$$\mathscr{U}_{sync} \leftarrow \{u : r(u) > \mu[r.] + \alpha \times \sigma[r.]\}, \tag{12}$$

where $r.$ denotes the set of residual scores of all nodes. Note that, the sensitivity of parameter α will be discussed in Section 7.2. A detailed process of catching suspected nodes is described in Algorithm 3. In Algorithm 3, detected suspicious nodes are finally retained. Likewise, unsuspected nodes (disturbed nodes) can be partly filtered out using the algorithm.

6.2 Abnormality forensics measurements

As aforementioned, the promising detection performance on synthetic datasets (labeled data) can be obtained based on empirical parameters. In reality, the attack scale, attack intention, and attack type are not known in advance facing with large-scale and unlabel real-world data. Moreover, the existence of attacks or anomalous rating behaviors may also not be clear. To this end, our proposed detection approach is heuristically exploited to spot abnormal ratings, items or users on real-world datasets. Nevertheless, there is no ground-truth for identifying detected results. Investigating measurements of abnormality forensics is naturally desired. This paper comprehensively analyzes the following aspects to evidence the existence of abnormality: (1) Intrinsic association between ratings and reviews; (2) Aggregation degree of rating behaviors in the rating profiles of an item; (3) Rating intention distribution between two kinds of shilling ratings including r_{min} and r_{max}; and (4) The difference of degree distribution between the rating profiles of removed and unremoved suspected items.

Firstly, analyzing the rating deviation based on both ratings and reviews is used to discover suspected items. Intuitively, mining the correlation between ratings and the corresponding reviews expects to improve the accuracy of rating prediction. Thus, a powerful prediction method which jointly exploits ratings and reviews is desirable. In this paper, a latent-factor and hidden-topics based method is employed to uncover the hidden correlation between ratings and reviews and provide an accurate rating prediction (McAuley & Leskovec, 2013). The presented prediction model discovers topics that are correlated with the hidden factors of products and users. The goal of the model is to simultaneously consider the latent factor of ratings and hidden factor of topics by globally optimizing corresponding parameters. Given a review corpus of rating, $\mathscr{T}$, the final objective function is defined as follows,

$$\underset{\Theta,\Phi,\kappa,z}{\arg\min}\left(\sum_{r_{u,i}\in\mathscr{T}}\underbrace{(rec(u,i)-r_{u,i})^2}_{\textbf{rating error}}-\underbrace{\mu\cdot l(\mathscr{T}|\theta,\phi,z)}_{\textbf{corpus likelihood}}\right), \tag{13}$$

where μ denotes a hyper-parameter that balances the importance of the above two parts, namely rating error and corpus likelihood. Φ and Θ are respectively topic and rating parameters. z denotes the set of topic assignments for each word in the corpus $\mathscr{T}$. κ is used to control the transform between rating and review parameters γ_i and θ_i, which is determined as $\theta_{i,k}=\frac{exp(\kappa\gamma_{i,k})}{\sum_{k'}exp(\kappa\gamma_{i,k'})}$, where $\theta_{i,k}$ and $\gamma_{i,k}$ denote a certain property of an item and a particular topic being discussed of the corresponding item, respectively. The exponent in the denominator enforces that each $\theta_{i,k}$ is positive, and the numerator enforces that $\sum_k\theta_{i,k}=1$.

The first part of (13) measures the rating error of latent-factor recommendation between rating $r_{u,i}$ and the recommended rating $rec(u,i)$, where $rec(u,i)$ is determined as $rec(u,i)=\alpha_0+\beta_u+\beta_i+\gamma_u\cdot\gamma_i$, where α_0 denotes a balance parameter. β_u and β_i are user and item biases, respectively. γ_u and γ_i are K-dimensional user and item factors, respectively. Based on a corpus of ratings $\mathscr{T}$, parameters $\Theta=\{\alpha_0,\beta_u,\beta_i,\gamma_u,\gamma_i\}$ are determined so as to minimize the mean squared error (McAuley & Leskovec, 2013).

The second part of (13) is used to uncover hidden factors of topics based on Latent Dirichlet Allocation (LDA) (McAuley & Leskovec, 2013). Given a review text $\mathscr{D}$, LDA associates each document $d\in\mathscr{D}$ with a $\mathscr{K}$-dimensional topic distribution θ_d. A word in document d has probability $\theta_{d,k}$ to discuss topic k. The likelihood of a review corpus $\mathscr{T}$ is defined as follows,

$$p(\mathscr{T}|\theta,\phi,z)=\prod_{d\in\mathscr{T}}\prod_{j=1}^{N_d}\theta_{z_{d,j}}\phi_{z_{d,j},w_{d,j}}, \tag{14}$$

where $\Phi=\{\theta,\phi\}$ and topic assignment z are updated via sampling (McAuley & Leskovec, 2013). $\theta_{z_{d,j}}$ and $\phi_{z_{d,j},w_{d,j}}$ denote the likelihood of seeing the particular topic and particular word for the topic, respectively. Based on the above prediction method, a relatively accurate prediction error can be obtained, which is used to explore the existence of suspicious items. More details will be discussed in Section 7.5.

Secondly, analyzing the degree of time aggregation of rating behaviors for a suspected item is used to discover anomalous ratings (Yang et al., 2017). In order to achieve attack intentions, both the effect and cost of attacks should be considered by attackers. Naturally, injecting anomalous rating profiles in a short period of time is desired for saving the cost of attacks. The group shilling rating behavior is easy to be exposed in reality. All users who have rated the suspected items are worth further certification. It is noteworthy that a short period of time is limited a few days, such as one day or two days. In reality, it depends on the actual situation. More experimental details will be analyzed in Section 7.5.

Thirdly, investigating the distribution of rating intentions is favorable to determine suspicious items. Concretely, a targeted item is easily rated with a reversed rating by shilling attackers. A good word-of-mouth item which has been rated with higher ratings is demoted with the minimum rating r_{min}, while a bad word-of-mouth item which has been rated with lower ratings is promoted with the maximum rating r_{max}. Intuitively, an item is suspected if all ratings rated on the item almost belong to r_{min} and r_{max} as well as close to the same number (Yang et al., 2017).

For the last forensics measurement, based on the item-item graph as discussed in Section 6.1, a suspected node (item) abnormally connects to other nodes, which may generate an anomalous degree distribution (a spike appears on the curve of degree distribution) (Jiang et al., 2014; Yang et al., 2017). The degree distribution is recovered after the removal of suspicious nodes. Comparing with the different of degree distribution between removed and unremoved suspected items, these suspected items deserve further evidence.

To sum up, suspicious items can be further determined by comprehensively analyzing the above four forensics measurements. Undoubtedly, the presented abnormality forensics measurements are only used for auxiliary analysis. Note that, real-world details mainly determine the final evidence of detected results.

7 Experiment and analysis

7.1 Experiment setting

7.1.1 Datasets

All datasets used in our experiments can be categorized into two different types, real-world datasets (unlabeled data) and synthetic datasets (labeled data). Five diverse datasets including Amazon, TripAdvisor, Netflix, MovieLens-20M, and MovieLens-100K datasets (McAuley et al., 2015) are exploited to investigate the problem of abnormality detection.

Each synthetic dataset (experimentally generated) consists of authentic profiles and attack profiles. For the former, the MovieLens-100K dataset is used to describe the rating behaviors of genuine users in recommender systems (Yang et al., 2016). Attack profiles are generated using corresponding attack scenarios (as introduced in Section 7.1.2), due to the fact that there is no open attack data. In our experiments, 11 different attack models are respectively implemented in different cases. For each attack, attack profiles are generated by exploiting the corresponding attack strategy with diverse attack sizes (Yang et al., 2017) {1.1%, 6.4%, 11.7%, 17.0%, 22.3%, 27.6%} and filler sizes (Yang et al., 2017) {1.2%, 4.2%, 7.3%, 10.3%, 13.3%, 16.4%}. Afterwards, each attack dataset is respectively inserted into the authentic dataset to construct a final synthetic dataset. Therefore, we have 396 ($11 \times 6 \times 6$) synthetic datasets including 11 different attack models, 6 different attack sizes and 6 different filler sizes. In the synthetic data, we just detect the nuke attacks. Push attacks can be detected in the analogous manner.

7.1.2 Attack models

In the experiments, 11 different attack models are exploited to construct attack profiles with diverse attack sizes and filler sizes. In general, rating profiles of an attacker can be briefly divided into three parts: selected items I_S, filler items I_F, and target items I_T (Burke et al.,

2006; Yang et al., 2016). The corresponding ratings are respectively $\sigma(i_k^S)$, $\rho(i_l^F)$, $\gamma(i_j^T)$, where $k > 0$, $l > 0$, and $j > 0$ (Burke et al., 2006; Gunes et al., 2012; Wu et al., 2014). For the target items, it depends on the attack intention. For nuke attacks, the target items will be rated with the lowest rating r_{min}. Similarly, the target items will be rated with the highest rating r_{max} in push attacks (Gunes et al., 2012; Wu et al., 2014). More importantly, constructing attack profiles for each attack mainly concentrates on the corresponding ratings of selected and filler items respectively. Concretely, let $N(r, \sigma^2)$ denotes the Gaussian distribution with mean r and standard deviation σ, the details of each presented attack model are described in Table 1. As is known, power users are able to influence the largest group of users in recommender systems (Wilson & Seminario, 2015). Similarly, power items (Seminario & Wilson, 2014) are also able to influence the largest group of items like popular items. These exploited attack models are detailed as shown in Table 1. In power user attacks, we choose the top 50 users based on the total number of ratings (NR) they have in their user profile, or select the top 50 users with the highest aggregate similarity (AS) are chosen as the power users. Accordingly, in power item attacks, the top-N items which have the highest aggregate similarity (AS) are empirically selected as a set of power items. In PIA-ID, power items participate in the highest number of similarity neighborhoods based on *In-Degree* (ID) centrality (Seminario & Wilson, 2014).

7.1.3 Evaluation metrics

To evaluate the performance of the presented methods, we exploit different measurement metrics for experimental results. Filtering out more disturbed rating profiles is a key task in the first stage of the proposed method. Intuitively, with the increase of empirical thresholds δ and ϕ, more genuine users and attackers will be filtered. Therefore, filtered ratio (FR) for filtered users is defined as the number of filtered users divided by the number of all genuine users. Similarly, filtered ratio for filtered attackers is defined as the number of filtered attackers divided by the number of all attackers. $FR = \frac{\#\ filtered\ users}{\#\text{all users}}$. The ratio of remaining users (RRU) is defined as the number of remaining attackers or genuine users divided by the number of all attackers or genuine users, which is written as $RRU = 1 - FR$.

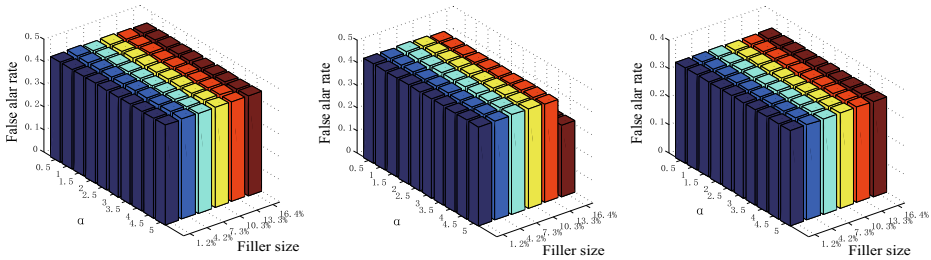
In addition, two different metrics are utilized in our experiments including detection rate and false alarm rate for measuring the detection performance of the presented methods. The detection rate (DR) is defined as the number of detected attackers divided by the number of attackers. The false alarm rate (FAR) is the number of genuine users that are predicted as attackers divided by the number of genuine users. $DR = \frac{|\text{D}\cap\text{A}|}{|\text{A}|}$, and $FAR = \frac{|\text{D}\cap\text{G}|}{|\text{G}|}$, where D, A and G denote the set of the detected users, attackers, and genuine users, respectively.

7.2 Parameter sensitivity analysis

To evaluate the sensitivity of parameter α used in (12), a series of experiments have been implemented in three different attacks (i.e., the reverse bandwagon, PIA-AS, and PUA-NR attacks are taken for example) as illustrated in Fig. 5. Parameter α is used to balance the importance of standard deviation σ and mean μ, which directly influences the determination of suspicious nodes (items) in the original graph. The goal of converting the remaining rating matrix into an associated item-item graph is to further capture the concerned target items using the characteristics of synchronicity and normality of nodes and finally reduce false alarm rates, especially when the attack sizes are small. Therefore, we analyze the influence of parameter α on reducing false alarm rates. In Fig. 5, we can observe that the false alarm rates are not sensitive to parameter α under different filler sizes in four different

Table 1 Description of all presented attack models

Attack model	I_S		I_F		I_T
	Items	Rating	Items	Rating	Push or nuke
AOP	Null		$x\%$ popular items, ratings set with normal dist around item mean Seminario and Wilson (2014).		r_{max} or r_{min}
Average	Null		Randomly chosen	Normal dist around item mean.	r_{max} or r_{min}
Love/Hate	Null		Randomly chosen	r_{min} or r_{max}	r_{max} or r_{min}
Bandwagon (average)	Popular items	r_{max} or r_{min}	Randomly chosen	Normal dist. around item mean.	r_{max} or r_{min}
Bandwagon (random)	Popular items	r_{max} or r_{min}	Randomly chosen	Normal dist. around system mean.	r_{max} or r_{min}
Reverse Bandwagon	Unpopular items	r_{min} or r_{max}	Randomly chosen	System mean	r_{max} or r_{min}
Segment	Segmented items	r_{max} or r_{min}	Randomly chosen	r_{min} or r_{max}	r_{max} or r_{min}
PUA-AS	Copy ratings and items from power user profiles Seminario and Wilson (2014).		Null		r_{max} or r_{min}
PUA-NR	Copy ratings and items from power user profiles (Seminario & Wilson, 2014).		Null		r_{max} or r_{min}
PIA-AS	Power items, ratings set with normal dist around item mean (Seminario & Wilson, 2014).		Null		r_{max} or r_{min}
PIA-ID	Power items, ratings set with normal dist around item mean (Seminario & Wilson, 2014).		Null		r_{max} or r_{min}



(a) Reverse Bandwagon attack (b) PIA-AS attack (c) PUA-NR attack

Fig. 5 Sensitivity analysis for parameter α on detection performance in 3 different attacks with different filler sizes

attacks. It is noteworthy that the detection rates are unchangeable in the stage of determining suspicious items as discussed in Section 4.

Analyzing the activity of user and the popularity of item is to determine sparse rating profiles which can be considered as disturbed rating profiles and can be eliminated in advance. As aforementioned, inactive users and novel or unpopular items can be selectively eliminated using reasonable empirical thresholds. As shown in Fig. 6 a(1)-a(3), the ultimate goal of determining a threshold for parameter δ is to filter out genuine profiles as many as possible and simultaneously retain entire attack profiles. Likewise, for parameter ϕ, a suitable threshold also is used to filter out genuine profiles as many as possible and empirically retain all attack profiles as illustrated in Fig. 6 b(1)-b(3). Note that, δ and ϕ are empirically set as 120 and 300 in our experiments, respectively. To examine the sensitivity of parameters δ and ϕ on detection performance, a list of experiments have been implemented in different attacks as demonstrated in Fig. 6 c(1)-c(3) and d(1)-d(3). One observation is that the detection rates are not sensitive to both parameters δ and ϕ. The other observation is that the false alarm rates almost are unchangeable with the increase of δ. In contrast, the false alarm rates gradually increase with the decrease of ϕ especially when ϕ is small. The main reason is that more general items such as popular items are easily wrongly considered as target items when ϕ is small. In our experiments, ϕ is set as 10, which has a small impact on the false alarm rates.

7.3 Rating behavior analysis and evaluation

The goal of partitioning rating behaviors is to further distinguish between attack profiles and genuine profiles based on the remaining rating matrix after eliminating disturbed profiles. To illustrate the effectiveness of measuring rating behaviors on behavior partition, a series of experiments have been conducted using six different similarity measurements in three different attacks as shown in Fig. 7. The advantages and disadvantages of the presented similarity or distance measurements including Chebyshev, Cosine, Euclidean, Manhattan, and Minkowski. Intuitively, geometric model based similarity measurements solely depend on geometric positions to derive their distance measures, which is partly limited for finding the closest match neighbourhood. Therefore, investigating probability mass mechanisms rather than distance mechanisms as the foundation of finding the closest match neighbourhood is concerned. To visualize the aggregation of similar neighbourhoods using a probability mass-based mechanism, additional experiments on three different attacks are provided from the perspective of similarity matrix as demonstrated in Fig. 7 g(1)-g(3). We can see that the similarities between attackers are more coherent and dense than the similarities between

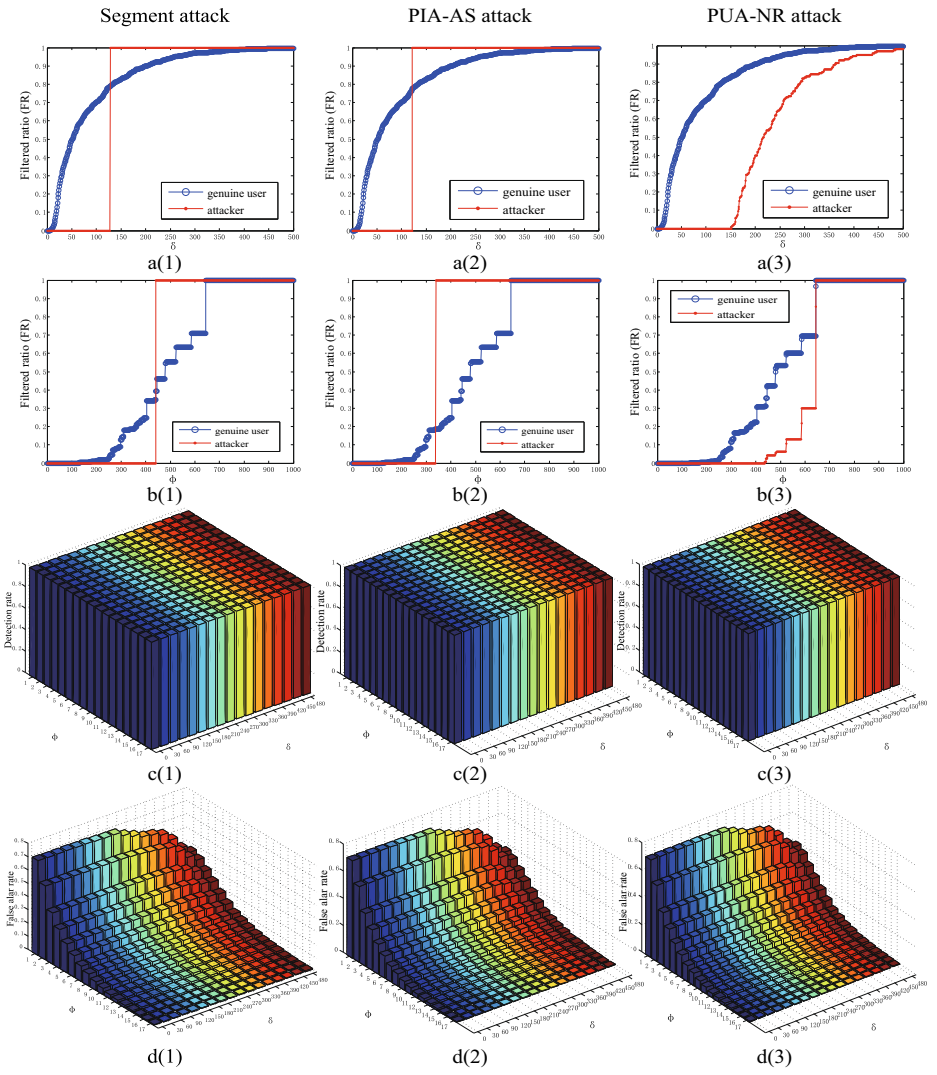


Fig. 6 The determination of empirical threshold and sensitivity analysis of parameters δ and ϕ in different attacks, where a(1)-a(3) and b(1)-b(3) are the influences of empirical thresholds on eliminating disturbed profiles; c(1)-c(3) and d(1)-d(3) analyze the sensitivity of parameters on detection performance

genuine users. This boundary in the similarity matrix provides a support for partitioning rating behaviors.

In our experiments, the ratio of remaining rating profiles including attack profiles and genuine profiles using each similarity metric is analyzed as demonstrated in Fig. 7 a(1)-f(3). The goal of measuring rating behaviors is to filter out genuine profiles and simultaneously retain attack profiles as many as possible by exploiting a reasonable similarity threshold. One observation is that the traditional similarity metrics including Chebyshev, Cosine, Euclidean, Manhattan, and Minkowski are difficult to distinguish between attackers and

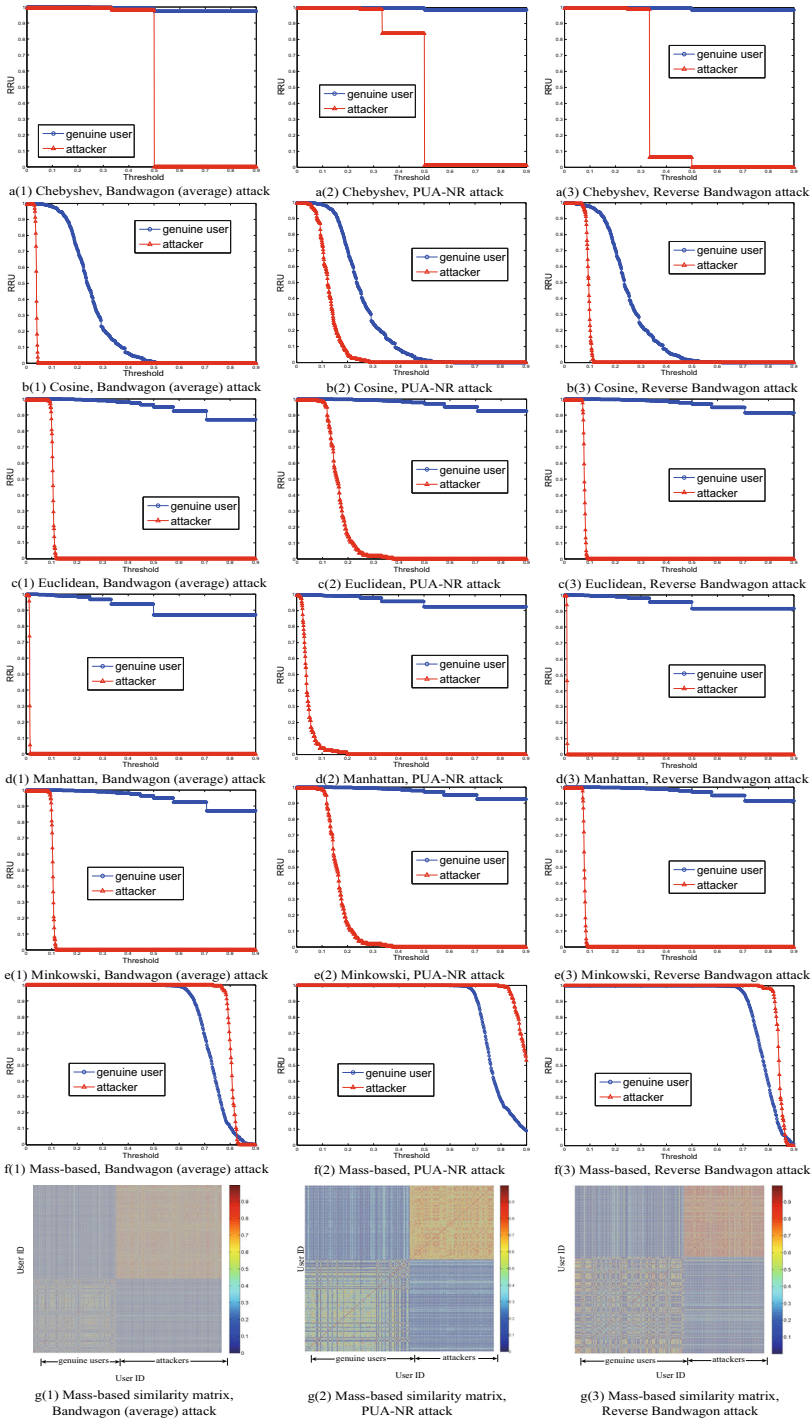


Fig. 7 Similarity analysis for partitioning rating behaviors in 6 different similarity measures. From g(1) to g(3), the similarity matrix calculated by probability mass-based method is visualized in 3 different attacks

genuine users. In contrast, the employed probability Mass-based method is relatively effective to further filter out genuine users and retain attackers simultaneously using reasonable thresholds. Note that, the threshold of similarity in Bandwagon (average), PUA-NR and Reverse Bandwagon attacks is set to 7.5. In reality, an empirical threshold can be determined via experimental analyses.

7.4 Comparison experiments

To demonstrate the effectiveness of the proposed detection method compared with benchmarks, extensive experiments have been conducted in 10 different attacks as shown in Fig. 8. We implemented five different benchmarks including PCA-based (Mehta et al., 2007; Wang et al., 2015), $\beta\mathscr{P}$-based (Chung et al., 2013), DeR-TIA (Zhou et al., 2014), EliMass-TIA (eliminating disturbed profiles with TIA), and EliMass-TIA-SynNor (our proposed method, EliMass-TIA with synchronicity and normality)

Detection results in five different attacks are firstly analyzed when the attack size is fixed (6.4%) and the filler size varies as shown in Fig. 8 a(1)-e(1). We can observe that the false alarm rates of EliMass-TIA and EliMass-TIA-SynNor (our proposed method) are significantly smaller than the false alarm rates of others in five different attacks. Meanwhile, the detection rates of DeR-TIA, EliMass-TIA, and EliMass-TIA-SynNor are higher than the detection rates of PCA-based and βP-based. Generally, detection models based on eliminating disturbed profiles and the mass-based dissimilarity have a more obvious advantage. Likewise, the detection performance of both EliMass-TIA and EliMass-TIA-SynNor is also better than the detection performance of others in different attacks when the filler size is fixed (7.3%) and the attack size varies as shown in Fig. 8 a(2)-e(2). Note that, detection models based on target item analysis such as DeR-TIA, EliMass-TIA and EliMass-TIA-SynNor

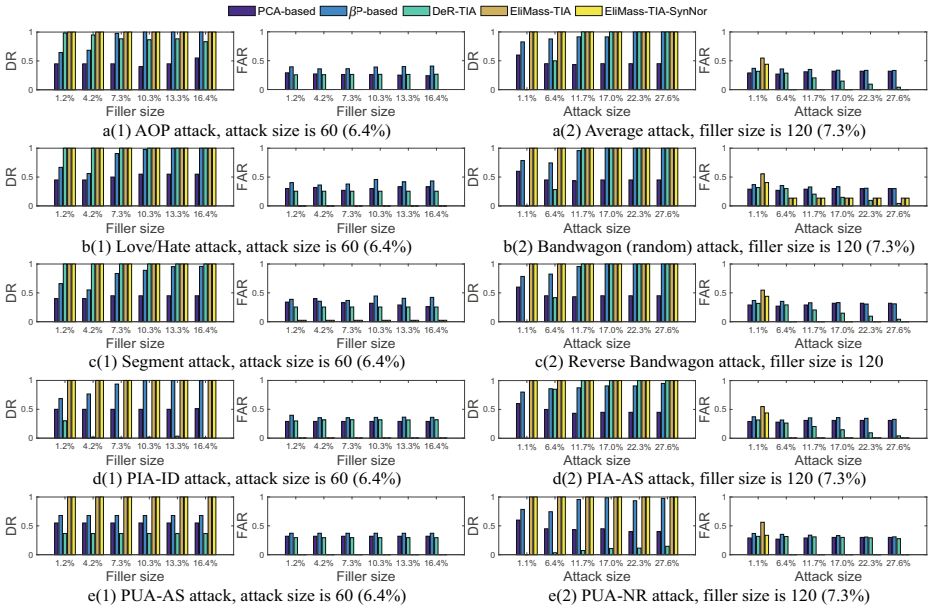
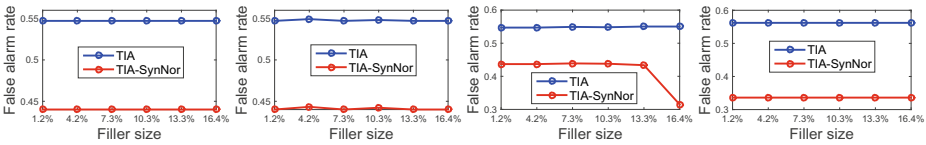


Fig. 8 Comparison of detection performance for five different methods in different attack sizes and filler sizes



(a) Average attack (b) Reverse Bandwagon attack (c) PIA-AS attack (d) PUA-NR attack

Fig. 9 Comparison of false alarm rates in the cases of solely using target item analysis (TIA) and combining target item analysis and synchronicity and normality of nodes analysis (TIA-SynNor) in different attacks and filler sizes

are favorable to capture the concerned attack profiles compared with the other presented methods. Here, δ and ϕ are set to 120 and 300, respectively. For EliMass-TIA and EliMass-TIA-SynNor, the latter has a relatively significant advantage due to the fact that suspicious items can be further determined according to the characteristics of synchronicity and normality of nodes, especially when the attack size is small as illustrated in Fig. 9. Figure 9 also demonstrates the effectiveness of reducing the false alarm rates using the characteristics of synchronicity and normality of nodes. Undoubtedly, our experimental results are for reference only.

For PCA-based, outlier points can be partly captured using rating styles including both rating assignment and item distribution. However, exploiting these rating styles may tend to be invalid when shilling attackers specially choose popular items to construct attack profiles (Wang et al., 2015). With respect to $\beta\mathscr{P}$-based, although it shows high detection rates compared with PCA-based in same cases, the false alarm rates are higher than PCA-based's. The results may be attributed to a)$\beta\mathscr{P}$-based is limited to the definition of Beta distribution (Chung et al., 2013); and b) using a single prior distribution to represent the possible values being observed for each of the user-item matrix renders the method with a low detection rate and a high false alarm rate when the genuine users have significantly different grading behaviors. Moreover, a large attack size may distort prior distribution.

7.5 Abnormality forensics for real application

To evaluate the practicability of the proposed approach, a list of experiments have been conducted on four different datasets, including Amazon, TripAdvisor, Netflix, and MovieLens-20M in order to discover anomalous rating behaviors according to heuristics knowledge learned from synthetic data. Moreover, detected results (i.e., ratings, items) are further analyzed and determined using the earlier presented abnormality forensics. In reality, it is unknown whether the real data contain anomalous rating behaviors or attacks. What is the size of attacks if anomalous rating behaviors or attacks exist, is also unknown in advance. In our experiments, first, for the datasets which contain both ratings and reviews such as Amazon and TripAdvisor data, suspicious items detected by the presented approach are analyzed by exploiting the prediction deviation of ratings one by one. For the Amazon dataset, comprehensively evaluating each suspicious items using four presented forensics measures is implemented to further determine suspected items as shown in Fig. 10. Note that, the Amazon dataset is independently detected year by year (i.e., from 1996 to 2014). Due to the limitation of space, two suspected items are analyzed in detail using four forensics measures as illustrated in Fig. 10. For the suspected item (indexed ID is 66888), the time aggregation degree of ratings is more significant in few days (i.e., from March 26 to 28, 2004).

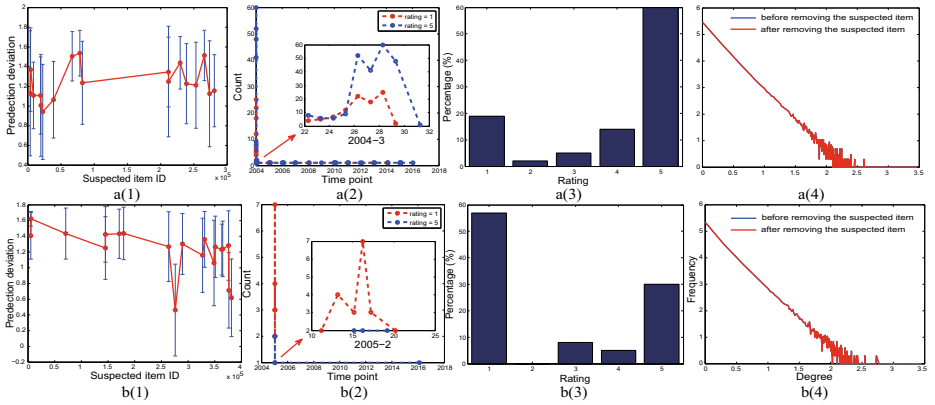


Fig. 10 Abnormality forensics analysis for Amazon data, where a(1)-a(4) denotes an item (indexed ID: 66888) in 2004; b(1)-b(4) denotes an item (indexed ID: 180571) in 2005; a(1) and b(1) show the prediction deviation of ratings; a(2) and b(2) show the time aggregation degree of ratings; c(1)-c(2) shows the distribution of ratings; d(1)-d(4) shows the distribution of rating intention

Likewise, the distribution of ratings r_{min} and r_{max} is relatively antagonistic. Nevertheless, the difference of degree distribution between the rating profiles of removed and unremoved suspected items is not significant. Similarly, for the suspected item (indexed ID is 180571) in 2005, the time aggregation degree of rating, rating distribution and rating intention distribution are similar to that presented above. It is noteworthy that all detected suspicious items and analyses are only for reference. In our experiments, the threshold ε for target item analysis is set as 100.

In contrast, analyzing suspicious items using the presented forensics measures for TripAdvisor, Netflix, and MovieLens-20M datasets is also provided. Here, we only show the prediction deviation of rating for each suspicious item as demonstrated in Fig. 11. Due to the fact that suspicious items can not be found using the other forensics measures on the three datasets. In the experiments, the thresholds for target item analysis are set as 70, 1000, and 950 in TripAdvisor, Netflix, and MovieLens-20M datasets, respectively. For the datasets which do not contain reviews including Netflix and MovieLens-20M, the prediction deviation of rating is also used for abnormality forensics. Note that, the corpus likelihood part in (13) is removed. Only the prediction deviation based on ratings is adopted in the experiments. Undoubtedly, our experimental results are only for reference.

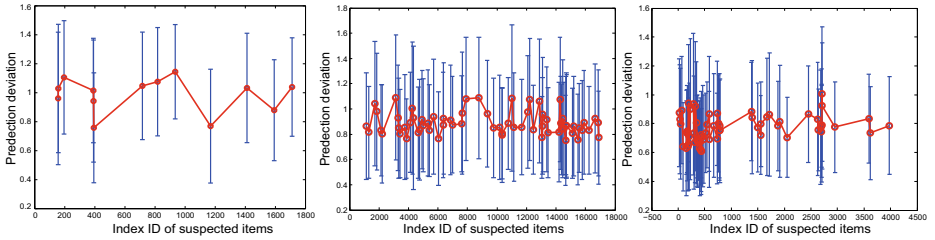


Fig. 11 Prediction deviation of ratings based on detected suspicious items in three real-world datasets

7.6 Discussion and limitation

Based on the experimental results, few insights are worth discussion as follows:

1. For dealing with the disturbed rating profiles, both the activity of user and the popularity of item are comprehensively analyzed in order to reduce the sparsity of rating matrix as far as possible. Nevertheless, intelligently eliminating disturbed rating profiles rather than purely relying on the popularity of item and activity of user needs to be further investigated due to the limited heuristic analysis of disturbed rating profiles and the single pattern of eliminating disturbed data. On the premise of providing more historical rating attributes such as user's trustworthiness, location, etc., exploiting multiple factors to remove disturbed user profiles is worth further exploration.
2. There are several advanced recommendation technologies proposed in recent years, such as the deep-learning based models (Zhang et al., 2019; Tang et al., 2019). Accordingly, data poisoning attacks including PoisonRec (Zhang et al., 2020) and LOKI (Song et al., 2020) against deep-learning based recommendation techniques have also been developed. In this paper, we only focus on the attack detection problem in collaborative filtering recommendations, which remains the malicious threats on other advanced recommendation techniques unconsidered. This work aims to explore the problem of abnormality forensics on real data according to heuristic knowledge learned from synthetic data. Investigating the detection of malicious treats based on advanced recommendation techniques and the hybrid detection of multiple types of malicious threats is part of our future work.
3. From the perspective of shilling attackers, they try to maximize the attack effect while minimizing the attack cost. To this end, attackers consistently give the highest rating score r_{max} (for push attacks) or the lowest rating score r_{min} (for nuke attacks) to target items. Intuitively, the target items are relatively easy to be identified by counting the special score (i.e., r_{max} or r_{min}) of all items especially facing with large attack sizes. In reality, there is a special group of users in collaborative filtering recommender systems, "grey-sheep" users who have inconsistent ratings with others (Gras et al., 2016; Zheng et al., 2017). Take the MovieLen data for example, the rating space of the system can be represented as {1, 2, 3, 4, 5}. The "grey-sheep" users (or termed *grey* attackers) may give the target items lower or higher scores (e.g., 2 or 4 scores). Comparatively, "grey-sheep" users are not easy to be determined due to the concealment of attack intention. For the same attack effect, the cost of grey attacks may be higher than that of promotion (or demotion) attacks with the highest rating or (the lowest rating). Due to space limitation, evaluating "grey-sheep" rating behaviors will be further analyzed and discussed in our next work.

8 Conclusion

In this paper, we develop a detection approach to defend anomalous attacks or ratings. Experimental results demonstrated that the presented detection method not only can provide relatively effective detection results compared with the benchmarks but also can be used to discover interesting findings on real-world datasets. In addition, suspected items and ratings are comprehensively evidenced by the proposed forensics metrics, which also provides a feasible way for real application.

Data Availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Code Availability Softwares or codes are not applicable to this article.

Declarations

Ethics approval All authors read and approved the final version of the manuscript.

Consent to participate All authors contributed to this work.

Consent for Publication All authors have checked the manuscript and have agreed to the submission.

Competing interests The authors declare that they have no competing interests.

References

Burke, R., Mobasher, B., & Williams, C. (2006). Classification features for attack detection in collaborative recommender systems. In *International conference on knowledge discovery and data mining* (pp. 17–20).

Chung, C., Hsu, P., & Huang, S. (2013). BP: A novel approach to filter out malicious rating profiles from recommender systems. *Journal of Decision Support Systems*, *55*(1), 314–325.

Fang, M., Yang, G., Gong, N., & Liu, J. (2018). Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th annual computer security applications conference (ACSAC)* (pp. 381–392).

Gunes, I., Kaleli, C., Bilge, A., & Polat, H. (2012). Shilling attacks against recommender systems: A comprehensive survey. *Artificial Intelligence Review*, *42*(4), 1–33.

Jiang, M., Cui, P., Beutel, A., Faloutsos, C., & Yang, S. (2014). Catchsync: catching synchronized behavior in large directed graphs. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 941–950).

Luo, X., Zhou, M., Li, S., & Shang, M. (2017). An inherently non-negative latent factor model for high-dimensional and sparse matrices from industrial applications. IEEE Transactions on Industrial Informatics.

McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *ACM Conference on recommender systems (RecSys)* (pp. 165–172).

McAuley, J., Pandey, R., & Leskovec, J. (2015). Inferring networks of substitutable and complementary products. Knowledge Discovery and Data Mining.

Mehta, B., Hofmann, T., & Fankhauser, P. (2007). Lies and propaganda: detecting spam users in collaborative filtering. In *Proceedings of the 12th international conference on intelligent user interfaces* (pp. 14–21).

Mobasher, B., Burke, R., Bhaumik, R., & Williams, C. (2007). Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology*, *7*(4), 38.

Seminario, C. E., & Wilson, D. C. (2014). Attacking item-based recommender systems with power items. In *ACM Conference on recommender systems* (pp. 57–64).

Song, J., Li, Z., Hu, Z., Wu, Y., Li, Z., Li, J., & Gao, J. (2020). PoisonRec: An adaptive data poisoning framework for attacking black-box recommender systems. In *The 36th IEEE international conference on data engineering (ICDE'20)* (pp. 157–168).

Tang, J., Du, X., He, X., Yuan, F., Tian, Q., & Chua, T. (2019). Adversarial training towards robust multimedia recommender system. *IEEE Transactions on Knowledge and Data Engineering*, *32*(5), 855–867.

Ting, K. M., Zhu, Y., Carman, M., Zhu, Y., & Zhou, Z. H. (2016). Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure. In *Proceedings of the 22nd ACM SIGKDD conference on knowledge discovery and data mining (KDD'16)* (pp. 1205–1214).

Wang, Y., Zhang, L., Tao, H., Wu, Z., & Cao, J. (2015). A comparative study of shilling attack detectors for recommender systems. In *The 12th international conference on service systems and service management (ICSSSM)* (pp. 1–6).

Wilson, D. C., & Seminario, C. E. (2015). Mitigating power user attacks on a user-based collaborative recommender system. In *Association for the advancement of artificial intelligence* (pp. 513–318).
Wu, Z., Wang, Y., & Cao, J. (2014). A survey on shilling attack models and detection techniques for recommender systems. *Science China*, *59*(7), 551–560.
Xing, X., Meng, W., Doozan, D., Snoeren, A., Feamster, N., & Lee, W. (2013). Take this personally: pollution attacks on personalized services. *USENIX Security*, 671–686.
Xu, Y., & Zhang, F. (2019). Detecting shilling attacks in social recommender systems based on time series analysis and trust features. *Knowledge-Based Systems*, *178*(15), 25–47.
Yang, G., Gong, N., & Cai, Y. (2017). Fake co-visitation injection attacks to recommender systems. *Network and Distributed System Security Symposium (NDSS)*, 1–15.
Yang, Z., Cai, Z., & Guan, X. (2016). Estimating user behavior toward detecting anomalous ratings in rating systems. *Knowledge-Based Systems*, *111*, 144–158.
Yang, Z., Cai, Z., & Yang, Y (2017). Spotting anomalous ratings for rating systems by analyzing target users and items. *Neurocomputing*, *240*, 25–46.
Yang, Z., Sun, Q., Zhang, Y., & Zhang, B. (2018). Uncovering anomalous rating behaviors for rating systems. *Neurocomputing*, *308*, 205–226.
Yang, Z., Sun, Q., Zhang, Y., Zhu, L., & Ji, W. (2020). Inference of suspicious co-visitation and co-rating behaviors and abnormality forensics for recommender systems. *IEEE Transactions on Information Forensics and Security*, *15*, 2766–2781.
Yang, Z., Xu, L., Cai, Z., & Xu, Z. (2016). Re-scale AdaBoost for attack detection in collaborative filtering recommender systems. *Knowledge-Based Systems*, *100*, 74–88.
Zhang, F., Qu, Y., Xu, Y., & Wang, S. (2020). Graph embedding-based approach for detecting group shilling attacks in collaborative recommender systems. *Knowledge-Based Systems*, *199*(8), 105984.
Zhang, F., & Wang, S. (2020). Detecting group shilling attacks in online recommender systems based on bisecting k-means clustering. *IEEE Transactions on Computational Social Systems*, *7*(5), 1189–1199.
Zhang, H., Li, Y., Ding, B., & Gao, J. (2020). Practical data poisoning attack against next-item recommendation. In *Proceedings of the web conference (WWW'19)* (pp. 2458–2464).
Zhang, Y., Tan, Y., Zhang, M., Liu, Y., Chua, T., & Ma, S. (2015). Catch the black sheep Unified framework for shilling attack detection based on fraudulent action propagation. In *Proceedings of the twenty-fourth international joint conference on artificial intelligence (IJCAI 2015)* (pp. 2408–2414).
Zhou, W., Koh, Y. S., Wen, J. H., Burki, S., & Dobbie, G. (2014). Detection of abnormal profiles on group attacks in recommender systems. In *Proceedings of the 37th international ACM SIGIR conference on Research on development in information retrieval* (pp. 955–958).
Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, *52*(1), 1–38.
Gras, B., Brun, A., & Boyer, A. (2016). Identifying grey sheep users in collaborative filtering: a distribution-based technique. In *Proceedings of the 2016 conference on user modeling adaptation and personalization* (pp. 17–26).
Zheng, Y., Agnani, M., & Singh, M. (2017). Identifying grey sheep users by the distribution of user similarities in collaborative filtering. In *Proceedings of the 6th annual conference on research in information technology* (pp. 1–6).