International Journal of Machine Learning and Cybernetics (2019) 10:1833–1845
https://doi.org/10.1007/s13042-018-0861-2

ORIGINAL ARTICLE

Shilling attack based on item popularity and rated item correlation against collaborative filtering

Keke Chen[1] · Patrick P. K. Chan[1] · Fei Zhang[2] · Qiaoqiao Li[1]

Received: 7 May 2018 / Accepted: 28 July 2018 / Published online: 4 August 2018

Abstract
Although collaborative filtering achieves satisfying performance in recommender systems, many studies suggest that it is vulnerable by shilling attack aimed to manipulate the recommending frequency of a target item by injecting malicious user profiles. The existing attack methods usually generate malicious profiles by rating the item selected randomly. However, as these rating patterns are different from the real users, who have their own preferences on items, these attack methods can be easily detected by shilling attack detection, which significantly reduces the attack ability. Although some attack methods consider disguise ability, these methods require too much information from real users. This study proposes a shilling attack which generates malicious samples with strong attack ability and similarity to real users. To imitate the rating behavior of genuine users, our attack model considers both rated item correlation and item popularity when choosing items to rate. The profiles generated by our attack model is expected to be more similar to real user profiles, which increases the disguise ability. We also investigate whether and how rated item correlation of real user profiles is different from the ones generated by our method and the existing shilling attack. The experimental results confirm that our method achieves the highest attack ability after removing the suspected profiles identified by PCA-based and SVM-based shilling attack detection. The study confirms the correlation of rated item is a critical factor of the robustness of recommender systems.

Keywords Recommender systems · Collaborative filtering · Shilling attack · Rated item correlation

1 Introduction

Making suitable recommendations to users is a challenging problem, especially in the era of big data. Collaborative filtering (CF) is a popular recommender system which makes recommendation to a user, by analyzing the preferences of other users similar to the target user in a database [1, 2]. Although CF achieves satisfying results in many applications [3–5], some studies [6, 7] indicate that CF is vulnerable in an adversarial environment, in which an adversary misleads recommendation suggested by a system on purpose by manipulating user profiles.

Shilling attack [8] is a common attack strategy against the recommender systems. The attack aims to increase (*push attack*) or decrease (*nuke attack*) the frequency of recommending a target item by injecting well-crafted user profiles to a database [8, 9]. Since the traditional CF does not consider the existence of an adversary, decisions of CF completely rely on user profiles in a database. Any change on the database may influence the recommendations [10]. The malicious sample generation in shilling attack can be separated into two steps. For each malicious sample, a number of items are firstly selected for rating, and the rating values are assigned to these selected items. Most shilling attack models [11–13] select rated items randomly or based on the item's popularity (i.e., the rating frequency). This rating behavior is different from a real user who has his/her own preference on rating items. Therefore, the malicious profiles generated by the attack models can be detected easily [14, 15], which

✉ Patrick P. K. Chan
patrickchan@ieee.org

Keke Chen
kkechen@qq.com

Fei Zhang
jfeizhang@qq.com

Qiaoqiao Li
lqeffort@163.com

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

[2] College of Computer and Information Engineering, Henan Normal University, Xinxiang, China

reduces their attack ability significantly. Although a few number of attack [11, 16] considers additional information when selecting rated items, the requirement on the knowledge of the database used in the target recommender system makes the methods impractical. For example, the exact user profiles are required [17, 18].

This study aims to propose a shilling attack with high attack and disguise ability when no complete knowledge of the recommender system is obtained. The rating behavior of real users quantified by using the correlation of rated items is considered in our model. As a user rates items according to his/her preference, some items are rated together with higher chances than others. For example, a person who likes iPhone may also like iPad but may be less interested in a smartphone with Android. However, the correlation of rated items in malicious profiles of the existing shilling attack methods should be different from the ones in real profiles since the random rating strategy is usually applied to the attack methods. The rating patterns of real users and the existing shilling attack models are investigated in Sect. 3.

A shilling attack method which utilizes not only the popularity but also the correlation of rated items is proposed in Sect. 4. A rated item of a malicious profile in our model is chosen according to its popularity and correlation with other rated items. It reduces the difference between the malicious and the real profiles explicitly, hence it increases the difficulty of attack detection. One advantage of the model is that the information of both the popularity and the rated item correlation can be approximated through the interaction with the target recommender system. The performance on the proposed attack method in different scenarios is evaluated and compared with other existing shilling attack methods experimentally in Sect. 5. The conclusion and future work are discussed in Sect. 6.

2 Background

The concepts related to the study are briefly introduced in this section. We firstly introduce the CF used in this study and its formulation. The shilling attack and its countermeasures are then discussed in Sects. 2.2 and 2.3.

2.1 Collaborative filter (CF)

CF is one of the most widely used technologies in the field of recommender systems. It predicts the interests of a user according to the preferences of other users similar to the target user in a database, represented by $\mathbf{R}$. A row of $\mathbf{R}$ represents a profile of a user, denoted by u_i where $i = 1, 2, \ldots n$ and n is the number of users in $\mathbf{R}$. $u_i = [r_{i1}, r_{i2} \ldots, r_{im}]$, where m is the number of items considered in the system, and r_{ij} represents the score of the item j given by the user i. For a memory-based method, given a target user u_t, the similarity between u_t and other users are measured by Pearson correlation coefficient [19]. The k most similar users are used to predict the preference of u_t. Items are recommended to u_t according to predicted rating, defined as follow:

$$p_{u_t,j} = \overline{r_{u_t}} + \frac{\sum_{i \in U_k} w(i, u_t)(r_{ij} - \overline{r_i})}{\sum_{i \in U_k} w(i, u_t)} \quad (1)$$

where $p_{u_t,j}$ is the predicted rating of the item j for u_t, $\overline{r_u}$ denotes the average rating value of all rated items of user u, U_k is a set of k most similar users to u_t, and $w(i, u_t)$ denotes the correlation coefficient between users u_t and i.

2.2 Shilling attack methods

Shilling attack misleads the decision of CF by injecting a number of malicious user profiles [9]. According to the objectives, the attack can be separated into two types, such as the frequency of recommending a target item increases (decreases) by *push attack* (*nuke attack*). Two parameters, *attack size* (*As*) and *filler size* (*Fs*), control the attack ability. *As* means the ratio of the number of malicious profiles injected to a system and the total legitimate profile number, while *Fs* means the proportion of items which are rated in a malicious profile. An attack method with larger *Fs* or *As* yields higher attack cost.

The existing methods follow the same model which separates a user profile into four partitions: the item targeted by attack (i_t), selected items (I_s), filler items (I_f), and unrated items (I_u). The target item is the item chosen for adversarial attack. i_t is rated with the maximum value in the push attack and minimum value in the nuke attack. I_s contains a set of items which increase the similarity between the malicious and the genuine profiles. The items in I_f are randomly selected to increase the number of rated items in profiles, while the rest of unrated items are grouped as I_u.

Both *random attack* and *average attack* do not consider I_s but only I_f in the malicious profiles, which means all the rated items are selected randomly. *Random attack* rates all items in I_f randomly according to the normal distribution with the mean and standard deviation calculated from all items in $\mathbf{R}$, while the rating of items in *average attack* are randomly assigned according to the normal distribution of each item. These two attack methods only require the mean and variance of rating of all and each item accordingly. Both I_f and I_s are considered in *Bandwagon attack* [20, 21]. I_f is rated randomly according to the normal distribution of all items, while I_s is rated with the maximum value. All the rated items are randomly selected from the $x\%$ of the most popular items in *Average over Popular (AoP) x% attack* [22, 23]. It rates the selected items according to the normal

Table 1 Strategies of the existing shilling attack models

Attack model	Selected items (I_s)	Filler items (I_f)	Knowledge on **R**
Random	∅	Rate according to the normal distribution of all items	Mean and variance of rating of all items
Average	∅	Rate according to the normal distribution of an item	Mean and variance of rating of each item
Bandwagon	Popular items, rated with the max value	Rate according to the normal distribution of all items	set of popular items, and mean and variance of rating of all items
AOP $x\%$	$x\%$ popular items, according to the normal distribution of an item	∅	Top $x\%$ popular items, and mean and variance of rating of each $x\%$ popular item

distribution of each item. In addition to the mean and variance of rating of items, *Bandwagon attack* and *AoP x%* need the information on the item popularity. The summary of the existing attack strategies is shown in Table 1.

2.3 Shilling attack countermeasures

Countermeasures of shilling attack identify malicious profiles by measuring their differences with real profiles. Several studies formulate the shilling attack detection as a supervised learning problem, i.e., classification. A number of features named as generic attributes are proposed to capture the statistical distinction between real and malicious profiles in the following aspects: the values of rating [24], the popularity of rated items [23, 25], and the similarity of a user profile and its N nearest neighbors [26]. A classifier, e.g., support vector machines (SVM), kNN or C4.5 [27, 28], is applied to distinguish the malicious from real profiles. In addition, semi-supervised learning methods [29, 30] are also used in order to reduce the number of labeled samples.

One disadvantage of the supervised or semi-supervised learning method is that the label information is required. The type of the attack should be known in advance. Unsupervised learning techniques, including K-means [31], principle component analysis (PCA) [32], and probabilistic latent semantics analysis (PLSA) [33], focus on the different characteristics of user profiles [34, 35]. The minority is identified as malicious profiles.

3 Rated item correlation analysis

It is expected to observe the correlation of rated items from real user profiles since the items are rated according to user preferences. However, malicious profiles of most shilling attack methods randomly select items. Therefore, the correlation of rated items of malicious profiles is different from the one of real users. This section firstly introduces a measure of correlation of the rated items for user profiles.

Cosine association (CA), which is an association rule in data mining [36], is applied to measure the correlation of each rated item pair. Given two items i and j, CA indicates the degree of a person who is interested in i also in j. Previous research work confirms that CA measures the correlation between item pair without a rating value effectively [37, 38]. For the items i and j, CA is defined as:

$$CA(i,j) = \sqrt{\frac{|U_{i,j}|^2}{|U_i||U_j|}} \tag{2}$$

where $|U_{i,j}|$ is the number of users who rate both i and j, while $|U_i|$ denotes the number of users who rate i. CA is a symmetric function which indicates the possibility of a user to rate i and j at the same time. When CA tends to 1, $|U_i|$ and $|U_j|$ are close to $|U_{i,j}|$, which means nearly all users who rate i also rate j. On the other hand, when CA tends to 0, a few users rate both i and j, *i.e.,* $|U_{i,j}|$ is small. Assume $|U_{i,j}| = |U_{m,n}|$. i but not j is a popular item (*i.e.,* $|U_i| >> |U_j|$), while the popularity of m and n is similar (*i.e.,* $|U_m| \simeq |U_j|$). $CA(m, n)$ is larger than $CA(i, j)$ since a user who rates i is less likely to rate j. This mechanism reduces the influence of the imbalance rating users.

MovieLens 100K dataset[1] is used as an example to illustrate the difference between real and malicious users generated by shilling attacks in terms of the CA of the rated items. The detailed description of MovieLens can be found in Sect. 5.1. CA of all item pairs in MovieLens is shown in Fig. 1. The x- and y-axis of Fig. 1a represent the movies sorted according to the rated times in ascending order. Each point in the graph denotes a value of CA of a movie pair represented by a color. A smaller (larger) value is represented by a color closer to blue (red). The white color indicates no user rates the item pair at the same time. As $CA(i, i)$ is always equal to 1, the diagnosis is in red color. The figure is symmetric. On the other hand, the distribution of CA of real profiles is illustrated in Fig. 1b. X-axis denotes the value of CA from 0 to 1 while y-axis represents the value of distribution, which indicates the frequency of the corresponding

[1] https://grouplens.org/datasets/movielens/.

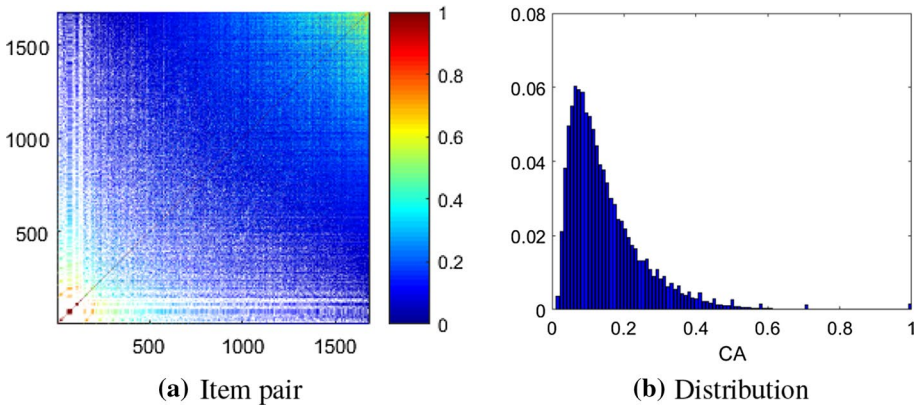


Fig. 1 CA of real user profiles in MovieLens. **a** Item pair. **b** Distribution

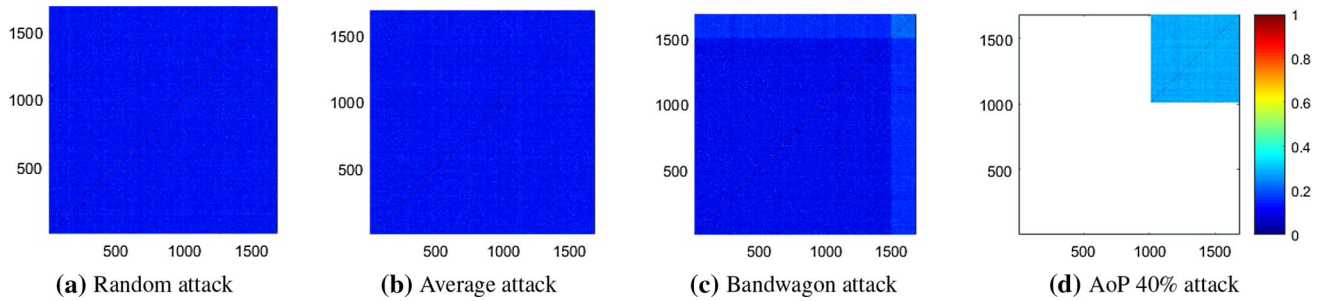


Fig. 2 CA of item pairs of different shilling attack models. **a** Random attack. **b** Average attack. **c** Bandwagon attack. **d** AoP 40% attack

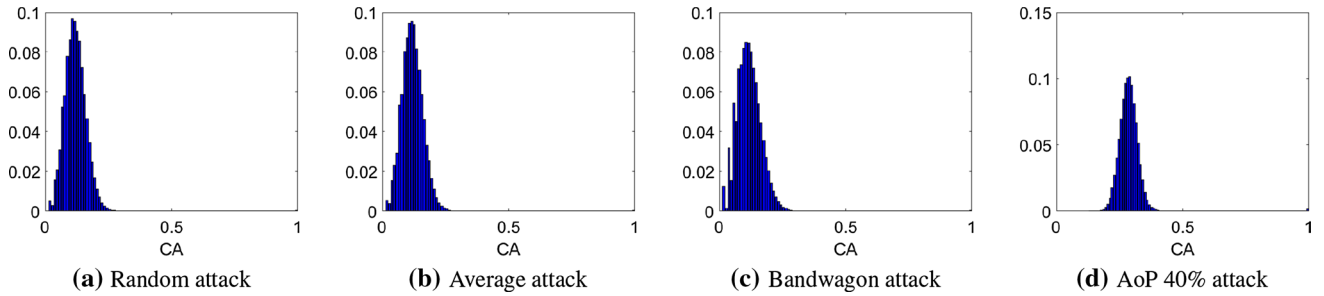


Fig. 3 Distribution of CA of different shilling attack models. **a** Random attack. **b** Average attack. **c** Bandwagon attack. **d** AoP 40% attack

CA value. Be noted that CA = 0 is not included in order to illustrate the distribution of other values clearly.

Figure 1a shows that CA of most item pairs is around 0.1 (i.e., in blue color). Small regions with yellow and red can be observed at the left bottom and right top corners. It means some items with large or small popularity contain large CA values with other items. Moreover, less popular items may not co-rated with others, i.e., white regions are found when items are less popular. As less users rate the unpopular items, the chance of no rating on unpopular items and another item is higher. The distribution of CA is similar to a F-distribution shown in Fig. 1b. The probability of CA = 0.1 is the largest. It decreases slightly with the increase of CA, and reaches 0 at CA = 0.7. It indicates that only small item pairs are highly correlated for real user profiles.

In order to compare the untainted user profiles with the attacked ones, we generate a dataset containing only one kind of malicious user profile based on MovieLens. The common shilling attack methods including *random*, *average*, *bandwagon* and *AoP 40%* attack are considered. 10% of most popular items are used as the selected item set for *bandwagon attack*. The same number of malicious users as in MovieLens are generated according to different attack methods. The number of rated items in a malicious user profile is the same as the one in a real user profile.

Figures 2a–d and 3a–d show the pairwise values and distributions of CA on the user profiles generated by different attacks respectively. The setting of those figures are the same as the ones in Fig. 1. As rated items are randomly selected in *random* and *average attack*, both Fig. 2a, b are similar and show that the values of CA are similar for all item pairs. Due to the same reason, the distributions of CA of these two attack methods follows a normal distribution illustrated in Fig. 3a. In *bandwagon attack*, 10% of the rated items are randomly chosen from only the popular items, while the rest of items are selected randomly from all items. As a result, the chance of selecting popular items is higher than unpopular items. It explains why CA of popular items is slightly larger than the ones of unpopular items in Fig. 2c. As the selection method of the rest of the rated items is the same as the one used in *random* and *average attack*, the CA values of the unpopular item pairs in Fig. 2c are similar with the ones in Fig. 2a, b. In addition, CA of all attack methods follows a normal distribution. *AoP attack* only randomly selects 40% most popular items. Therefore, no unpopular item is rated, i.e., no CA can be calculated for unpopular items in Fig. 2d. As only popular items are chosen, the chance of rating an item pair is high, which causes the values of CA to be larger as shown in Fig. 3d.

Figures 1, 2 and 3 show the dramatic difference between the untainted and attacked user profile datasets. The distributions of CA of malicious profiles follow a normal distribution while the real one follows a F-distribution. Moreover, some unpopular items are not co-rated with other items for real users. But CA of nearly all item pairs is non-zero for all attack methods expect *AoP 40%* due to random selection. CA of all unpopular item is zero in *AoP 40%* which is dramatically different from the real one. This may explain the reasons the existing shilling attack may be accurately identified by shilling attack detection methods [23]. This observation inspires us to investigate how secure the current shilling attack detection methods are in facing the attack which not only consider popularity of items but also the correlation.

4 Rated item correlation based attack model

A proposed shilling attack model which considers not only the popularity but also correlation of rated items is introduced in this section. The filled items are not selected randomly but according to rated item correlation of real users measured by CA in Eq. (2). Stronger disguise ability is expected since the malicious user profile generated by our proposed attack model is similar to the real ones in terms of rating correlation.

Our model assumes that partial knowledge on popularity and correlation of rated items of real user profiles is known. More complete knowledge increases the performance of our model. The proposed attack model generates a set of malicious profiles $\mathbf{R}^A = \left[u_1^A, u_2^A, \ldots, u_{As}^A\right]$ based on a given matrix $\mathbf{R}$ containing n real user profiles with m items, where As is the number of malicious profiles and u_i^A is the ith malicious user profile. Each malicious user profile is generated individually in our model. Rated items of a profile are selected in order to maximize the popularity of rated item as well as the rated item correlation. It is formulated as follows:

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}}(RC(\boldsymbol{\theta}), P(\boldsymbol{\theta})), \tag{3}$$

$$\text{s.t.} \sum_{k=1}^{m} \theta_k = \lfloor Fs \times m \rfloor, \tag{4}$$

$$\theta_k \geq \theta_k^0 \quad \text{for} \quad 1 \leq k \leq m \tag{5}$$

where $RC(\boldsymbol{\theta})$ and $P(\boldsymbol{\theta})$ are the functions measuring the correlation of rated items and the popularity of the selected items respectively. $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_m\}$ is a binary-valued vector representing whether an item has been selected for a malicious profile, where m is the total number of items in the dataset. When the i-th item is selected, $\theta_i = 1$; otherwise, $\theta_i = 0$. Fs denotes the filler size, which is the proportion of rated items. The number of the rated items of a malicious profile is constrained by $\lfloor Fs \times m \rfloor$ in Eq. (4). In order to increase the diversity of malicious profiles, a binary-valued vector $\boldsymbol{\theta}^0 = \{\theta_1^0, \theta_2^0, \ldots, \theta_m^0\}$ is used as the initial state of a profile. $l = \sum_{k=1}^{m} \theta_k^0$ denotes the number of 1s randomly assigned to θ^0, where $0 \leq l \leq \lfloor Fs \times m \rfloor$. When l is close to Fs, the diversity of malicious profiles increases but the attack ability decreases since most rated items are selected randomly. On the other hand, rated items of each malicious profile are similar when l is near to 0. Equation (5) forces all selected items in θ^0 also to be in θ. $\boldsymbol{\theta}^\star$ is the optimal solution. Rated item of a malicious profile in our attack method has high popularity and also similar rated item correlation with the real profiles.

$RC(\boldsymbol{\theta})$ measures the average CA of selected item $\boldsymbol{\theta}$ according to the given user profile dataset $\mathbf{R}$, and is defined as:

$$RC(\boldsymbol{\theta}) = \frac{1}{2\left(\sum_{k=1}^{m} \theta_k\right)!} \sum_{i=1}^{m} \sum_{j=1}^{m} \theta_i \theta_j CA(i,j) \tag{6}$$

where CA is calculated according to $\mathbf{R}$ by Eq. (2). $P(\boldsymbol{\theta})$ calculates the popularity of $\boldsymbol{\theta}$ as follows:

$$P(\boldsymbol{\theta}) = \frac{1}{\sum_{k=1}^{m} \theta_k} \sum_{i=1}^{m} \theta_i RT(i) \tag{7}$$

where $RT(i)$ counts the number of rating on the item i in $\mathbf{R}$. Larger $RT(i)$ indicates that the item i is more popular.

All the selected items are then rated as the average value of each item.

The multi-objective function shown in Eq. (3) is a NP-hard problem. A practical solution is also proposed in this section. In order to reduce the time complexity, the attack model is formulated as a single-objective function as follows:

$$\theta^{\star} = \arg\max_{\theta} \quad (1-\lambda)RC(\theta) + \lambda P(\theta) \tag{8}$$

where λ is a tradeoff parameter to control the influence of rated item correlation and popularity on the item selection, and $0 \leq \lambda \leq 1$. This single-objective function is also constrained by Eqs. (4) and (5). A greedy algorithm is then applied to find the sub-optimal solution. It is a simple variant of the popular forward selection algorithm, which iteratively adds an item from the current candidate set starting from an empty set [39]. The time complexity is $O(As \times m^2)$. The implementation of the proposed shilling attack method is given as Algorithm 1.

Algorithm 1 Proposed Shilling Attack Method with the Forward Selection Approach.

Input: λ: the trade-off parameter; Fs: the number of selected items in a malicious user profile; θ^0: initial state of a malicious profile.
Output: $\theta \in \{0,1\}^m$: the set of selected items for a malicious user profile.

$\mathscr{S} \leftarrow \emptyset$, $\mathscr{U} \leftarrow \{1,\ldots,d\}$;
Set $\mathscr{S} = \mathscr{S} \cup j$ and $\mathscr{U} = \mathscr{U} \setminus j$ for $\theta_j^0 = 1$;
repeat
  for each item $k \in \mathscr{U}$ **do**
    Set $\mathscr{F} \leftarrow \mathscr{S} \cup \{k\}$;
    Set $\theta = \mathbf{0}$, and then $\theta_j = 1$ for $j \in \mathscr{F}$;
    Set $RC_k = RC(\theta)$ and $P_k = P(\theta)$;
  end for
  $k^{\star} = \arg\max_k \left((1-\lambda)RC_k + \lambda P_k\right)$;
  $\mathscr{S} \leftarrow \mathscr{S} \cup \{k^{\star}\}$, $\mathscr{U} = \mathscr{U} \setminus \{k^{\star}\}$;
until $|\mathscr{S}| = Fs$
Set $\mathscr{F} \leftarrow \mathscr{S}$;
Set $\theta = \mathbf{0}$, and then $\theta_j = 1$ for $j \in \mathscr{F}$;
Return θ

5 Experiment

The proposed shilling attack method is evaluated and compared with other attack methods in different scenarios experimentally. The experimental results and discussion are given after introducing the experimental setting.

5.1 Setting

A benchmark dataset, MovieLens 100K dataset is used in our experiment. MovieLens is collected through the MovieLens website[2] in the 7-month period from September 1997 to April 1998. It contains 100,000 ratings provided by 943 users on 1682 movies. Each user rates at least 20 movies. The range of rating value is 1–5, and the degree of preference increases with the increase of the score.

For each experiment, 200 out of 943 user profile rates are randomly selected as the training set for the supervised shilling detection method. The rest of the profiles are used as the dataset $\mathbf{R}$ for a recommender system. The user-based weighted algorithm (UBW) [15], which is one of the common recommender systems, is used. UBW calculates the similarity of user profiles according to Pearson correlation coefficient with a significant weighting of n / 50, where n is the number of common rated items. The preferences of 50 users closest to a target user are recommended.

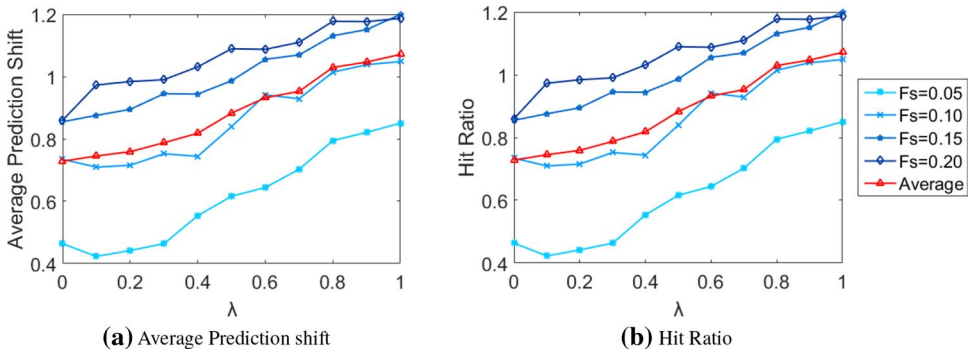
The common shilling attack methods including *random*, *average*, *bandwagon*, and *AoP 40%* methods are implemented and compared with our method. All attacks aim to increase frequency of recommending a randomly selected target item by injecting malicious user profiles to $\mathbf{R}$, i.e., push attack. For our method, 5% of the rated items are chosen randomly as the initial set θ^0. As the same as the setting used in Sect. 3, 10% of most popular items are used as the selected item set for *bandwagon attack* while 40% of the most popular items are considered in *AoP 40% attack*. The number of malicious profiles (As) is set as 10% of the size of the database (n), while the filler size (Fs) which means the proportion of the rated number in each malicious profile is set within the range of 5%, 10%, 15% and 20% of the total items (m).

Both complete and partial knowledge scenarios are considered in our method. The exact popularity and rated item correlation are provided in the complete knowledge scenario, while $\pm$ 10% and $\pm$ 20% random noise is added to both information in the partial knowledge scenario in order to simulate the situation when the information used in our model is different from the real one. We assume complete knowledge can be obtained in other attack methods.

100 items are randomly selected to evaluate the effectiveness of the attack methods in terms of average prediction shift (APS) [40] and hit ratio (HR) [41]. APS measures the average change of the predicted rating of the target items before and after the malicious profiles are injected into $\mathbf{R}$. The positive (negative) APS value indicates an attack method increases (decreases) the frequency of recommending a target item. HR examines whether the target item is listed as the top N of a user. Larger HR is better in push attack.

PCA-based and SVM-based methods are selected as the representative for unsupervised and supervised detection

[2] http://movielens.umn.edu.



(a) Average Prediction shift **(b)** Hit Ratio

Fig. 4 Attack ability of our model with different λ in terms of average prediction shift and average hit ratio. **a** Average prediction shift. **b** Hit ratio

method respectively. PCA calculates the principal components of the covariance matrix of **R**. k users with the smallest coefficient are identified as malicious users. In this study, k is set as As in order to increase the difficulty of evasion. For the supervised detection method, the generic attribute used in SVM includes: *Rating Deviation from Mean Agreement*, *Weighted Deviation from Mean Agreement*, *Weighted Degree of Agreement*, *Degree of Similarity with Top Neighbors* [27], *Mean of User Popularity Degree* , *Range of User Popularity Degree* and *Quantile of User Popularity Degree* [23, 25]. To increase the attack difficulty, we assume that the shilling attack method is known, i.e., malicious profiles in the training set are generated by using the same attack method as the one in the test set. A target item of malicious profiles is selected randomly as the real one is unknown. As a result, the training set contains 200 real user profiles randomly selected from **R**, and also 200 malicious profiles. The filler size of each profile is randomly chosen from 3%, 5%, 10%, 15%, and 20%.

F-measure [42] shown in Eq. (9) is used to evaluate the performance of shilling attack detection due to the imbalance problem. Both precision and recall are considered in F-measure. The value of F-measure is small when either precision or recall is low. Higher F-measure indicates accurate prediction on both classes. All results are obtained by calculating the average values of the experiments repeated ten times individually.

$$F\text{-measure} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{9}$$

5.2 Results and discussion

The proposed and other attack methods are evaluated and compared in different scenarios. The influence of the parameters of our proposed model is discussed in Sect. 5.2.1. The comparison between our proposed shilling attack method and other methods are discussed in terms of attack ability (Sect. 5.2.2), hardness of detection (Sect. 5.2.3), and attack ability after detection (Sect. 5.2.4). Finally, the running time of malicious profile generation of each method is discussed.

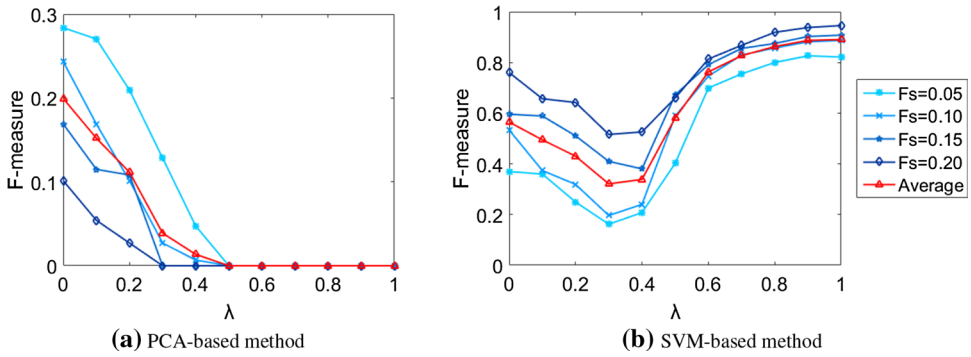
5.2.1 Parameter analysis of our proposed model

λ is the tradeoff parameter which adjusts the influence of popularity and correlation of rated items on item selection in our model. We assume that complete knowledge on popularity and rated item correlation is obtained in this discussion. When λ decreases, our model tends to consider rated item correlation but not popularity, and vise versa. This section investigates how λ affects the performance of our model in terms of average PS and HR.

The attack ability of our model with different λ and Fs are shown in Fig. 4. Y-axis represents the average values of PS and HR, while λ is shown in X-axis. The lines in different blue colors represent Fs (i.e., 0.05, 0.10, 0.15 and 0.20), and the red line indicates the average of the results with different Fs. The figures confirm that malicious profiles with larger filler size yield more influences on chance of recommending the target items in terms of both average PS and HR. However, the attack cost also increases since more items are rated in a profile.

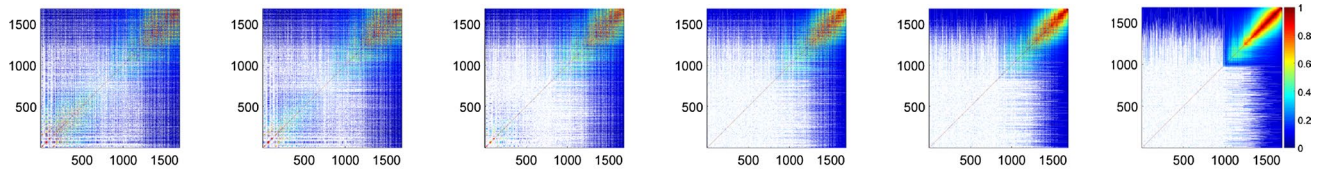
Average PS and HR of our method with different filler sizes is similar when different λs are used, i.e., both values increase with λ. Popularity of item will dominate in selection process when λ tends to 1. Therefore, many popular items are selected in a malicious user profile. As the same as the finding in previous studies [15], malicious profiles with more popular items usually have larger attack ability. This is because rating on popular items increases the similarity between malicious and real users in a recommender system.

Although larger λ yields bigger attack ability in our model, selection on λ should also consider its evasion ability. The results of the shilling attack detection methods on our proposed method are shown in Fig. 5. Similar to the attack ability discussion, our model with Fs (i.e., 0.05, 0.10, 0.15 and 0.20) represented by lines in different blue colors are also evaluated. Their average results are shown by using the red line. The trend of the blue lines are similar in



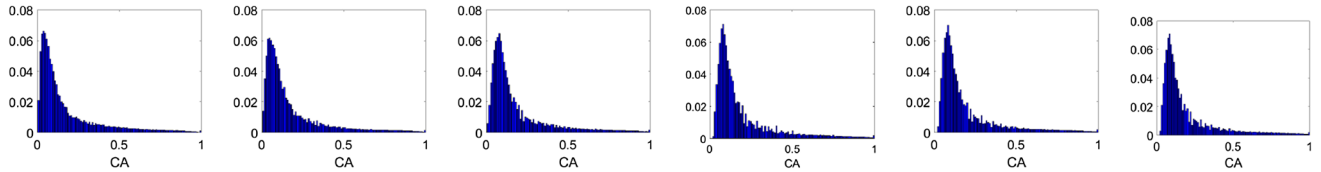
(a) PCA-based method **(b)** SVM-based method

Fig. 5 Evasion Ability of our model with different λ in terms of F-measure of PCA-based and SMV-based detection methods. **a** PCA-based method. **b** SVM-based method



(a) λ=0 **(b)** λ=0.2 **(c)** λ=0.4 **(d)** λ=0.6 **(e)** λ=0.8 **(f)** λ=1

Fig. 6 CA of item pairs of the proposed shilling attack model with different λ. **a** $\lambda = 0$. **b** $\lambda = 0.2$. **c** $\lambda = 0.4$. **d** $\lambda = 0.6$. **e** $\lambda = 0.8$. **f** $\lambda = 1$



(a) λ=0 **(b)** λ=0.2 **(c)** λ=0.4 **(d)** λ=0.6 **(e)** λ=0.8 **(f)** λ=1

Fig. 7 Distribution of CA of the proposed shilling attack model with different λ. **a** $\lambda = 0$. **b** $\lambda = 0.2$. **c** $\lambda = 0.4$. **d** $\lambda = 0.6$. **e** $\lambda = 0.8$. **f** $\lambda = 1$

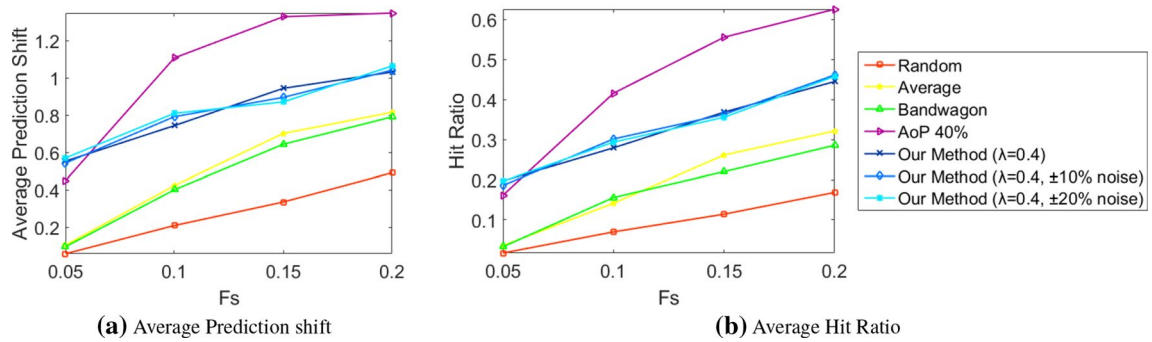
both PCA-based and SVM-based detection methods, which means they perform consistently on our models with different *Fs*.

Figure 5a, b show F-measure of PCA-based and SVM-based detection methods respectively. For PCA-based method, popularity increases the evasion ability of our model, i.e., the detection accuracy drops with the increase of λ. When $\lambda > 0.5$, the detection method cannot identify any malicious profile. This is because selecting more popular items in malicious profiles increases the covariance between them and other user profiles. This makes malicious and real users difficult to be separated by PCA. These results are consistent with previous studies [22].

On the other hand, SVM-based method accurately detects the shilling attack considering popularity of item [25]. When $\lambda = 0.3$, F-measure of the detection reaches its valley bottom (i.e., near to 0.32 in average). It may be because $\lambda = 0.3$ is the best ratio between popularity and item correlation in order to camouflage malicious user profiles. When λ is close to 0, the detection rate is higher because more unpopular items were selected, which is different from the rated item distribution of a genuine user. When λ is close to 1, the detection rate is close to 1, which means nearly all malicious profiles are identified successfully. As any defense method can be used in reality, $\lambda = 0.4$ is selected for our model by considering both attack and evasion ability.

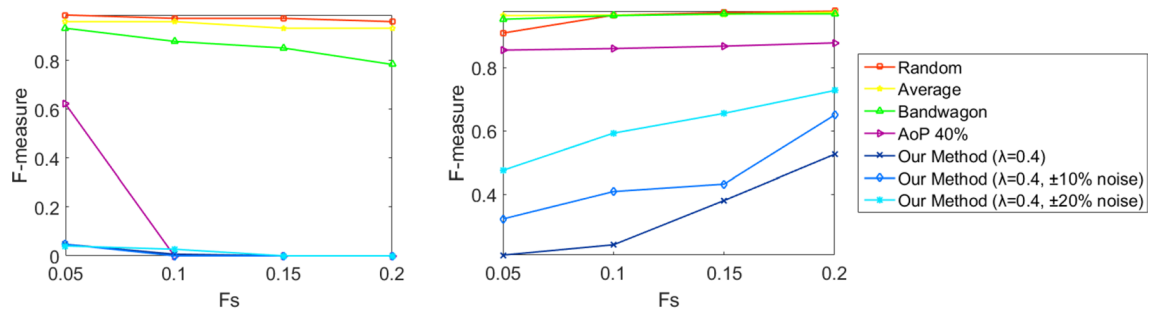
Item correlation of our proposed model is further discussed. Similar to the setting used in Sect. 3, CA of all item pairs of a dataset containing only malicious user profiles generated by our method with different λ based on MovieLens 100K dataset is illustrated in Figs. 6 and 7.

Figure 6 gives CA values of each item pair of malicious profiles generated by our method. When λ increases, less unpopular items are selected and CA of their item pairs are undefined. It causes the size of the white region at the left bottom corner increases as most rated items are the most popular items. On the other hand, items with higher CA are preferred when λ is small. Therefore, items with less popularity may also be selected. Different from other attack methods illustrated in Figs. 3, 7 indicate that all distributions of CA in our method with different λ follow F-distribution, which is as the same as the real one. In addition, the area



(a) Average Prediction shift **(b)** Average Hit Ratio

Fig. 8 Attack ability of the shilling attack methods in terms of average prediction shift and average hit ratio. **a** Average prediction shift. **b** Average hit ratio



(a) PCA-based method **(b)** SVM-based method

Fig. 9 Evasion ability of the existing attack models in terms of F-measure of PCA-based and SVM-based detection methods. **a** PCA-based method. **b** SVM-based method

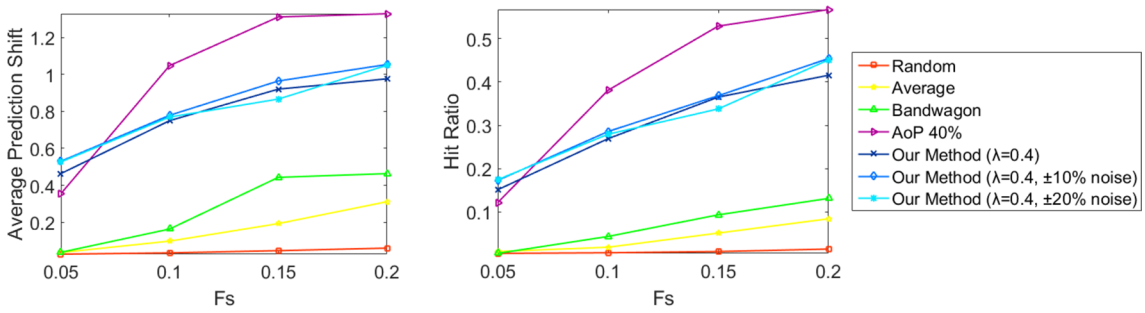
under the curve decreases when λ increases. The item without rating yields its CA equal to 0. When λ tends to 0, CA of our attack method is closer to the real one as only CA is optimized in the item selection process.

When λ is around 0.5, its evasion ability is the highest. This suggests that it is necessary to consider both popularity and the correlation of rated items in an item selection process. In the rest of the experiment, $\lambda = 0.4$ is chosen for our method. Although the attack ability of our method with $\lambda = 0.4$ is not the most desirable among all the settings, it has the highest evasion ability. This is important in practice as a defense mechanism may be used.

5.2.2 Attack ability

The performance of our proposed method with $\lambda = 0.4$ is evaluated with the existing shilling attack methods in terms of attack ability in this section. The attack ability of the attack methods evaluated by APS and HR is shown in Fig. 8a, b respectively. X-axis represents filler sizes, while the Y-axis denotes APS and HR in Fig. 8a and Fig. 8a respectively. Performances of different attack models are shown in different colors.

The performances of attack methods are consistent in terms of both evaluation methods. The attack ability of *random attack* is the lowest among all methods as all items of its malicious profiles are selected randomly. *Average* and *Bandwagon* methods have the similar attack ability, but Average attack is slightly higher than *Bandwagon* method because the average rating increases the similarity more efficiently. The difference of attack ability of our method with different levels of knowledge on real users (i.e., noise = 0%, ± 10% and ± 20%) is insignificant in both evaluations. In general, our method influences more the recommendation on target items than *Random*, *Average* and *Bandwagon* but not *AoP 40%*, which is the most efficient attack method in the experiment. Although our method is slightly better than *AoP 40%* when $Fs = 0.05$. However, when *Fs* increases, *AoP 40%* method performs better than ours significantly. This is because AoP attack selects a large proportion of popular items, which increases the similarity to the real users efficiently in a recommender system.



(a) PCA-based method + Average Prediction shift **(b)** PCA-based method + Average Hit Ratio

Fig. 10 Attack ability of shilling attack methods after sanitization. **a** PCA-based method + average prediction shift. **b** PCA-based method + average hit ratio

5.2.3 Evasion ability

Evasion ability of the shilling attack models are compared in this section. F-measure of PCA-based and SVM-based detection methods on the user profiles contaminated by different attack models are shown in Fig. 9. Figure 9a shows that PCA-based method successfully detects malicious profiles generated by *Random*, *Average* and *Bandwagon attacks*, i.e., F-measure is larger than 0.95 for *Random* and *Average attacks*, and larger than 0.78 for *Bandwagon attack*. Detection rate on our method with different levels of knowledge is similar. F-measure is slightly higher than 0 when *Fs* is small, and when $Fs \geq 0.15$, F-measure is close to 0 for all cases. Similarly, *AoP 40%* evades the detection by PCA-based method in all cases except $Fs = 0.05$. More popular items chosen in malicious profiles increase their covariance with real user profiles considered in PCA. This explains why PCA has satisfying detection accuracy on *Random*, *Average* and *Bandwagon attack* but not *AoP 40%* and our method.

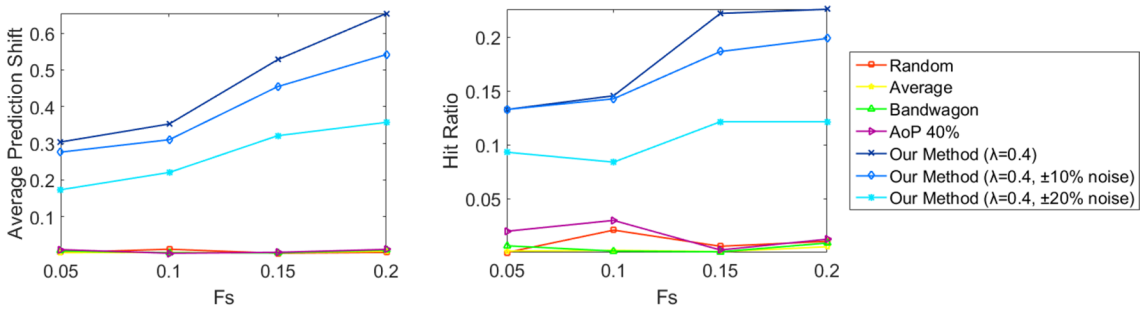
In general, SVM-based detection method achieves better performance than PCA-based method in terms of F-measure. Figure 9b illustrates that except our method, SVM-based detection distinguishes the malicious profiles generated by all attacks from legitimate ones accurately. All F-measure values for *Random*, *Average*, *Bandwagon* and *AoP 40%* are higher than 0.85. The distributions of rated items in malicious user profiles generated by *Random*, *Average*, *Bandwagon* and *AoP 40% attack* are different from the real one since their rated items are selected randomly. The difference of the distributions can be captured by the features used in the SVM model. On the other hand, F-measure of the detection on our method is significantly lower than other attack methods. Disguise ability decreases with less knowledge on the real users, i.e., lower F-measure when the noise on popularity and rated item correlation is smaller. Different from PCA-based method, the evasion ability increases when *Fs* is large. This is because when *Fs* is larger, the characteristics of the malicious profiles are much different than the real profiles captured by the features used in the SVM-based method. For example, the features *Rating Deviation from Mean Agreement*, *Weighted Deviation from Mean Agreement*, and *Weighted Degree of Agreement* measure the average rating deviation per item. Larger *Fs* increases the difference between malicious and real user profiles.

5.2.4 Attack ability after data sanitization

In practice, a shilling attack detection method may be applied to sanitize the collected user profiles. All suspected user profiles will not be used when making recommendation. The attack ability of a shilling attack method with weak evasion ability may downgrade significantly. This section investigates the attack ability after sanitization. All profiles classified as a malicious class by a detection method in the contaminated database will be removed. The attack ability is measured based on the recommendation on the target items using the sanitized database.

The attack ability of the attack methods after sanitization according to PCA-based and SVM-based detection methods is shown in Fig. 10. As nearly all malicious profiles of *Random* and *Average* methods are correctly identified, there is no influence on the chance of recommending target items after sanitization. The mechanism successfully defenses the attack. Although *Bandwagon* still affects the recommendation when filler size is large, its influence is reduced dramatically after sanitization. However, sanitization by PCA-based detection method does not reduce the influence of *AoP 40%* and our method to the recommender system. As a result, the attack ability of *AoP 40%* is still higher than our method with any attack knowledge.

Differently, when suspected profiles are identified according to SVM-based detection method, the attack ability of all methods except our proposed one drops dramatically, shown in Fig. 11a, b. This is because SVM-based method detects malicious profiles more accurately than PCA one. The malicious profiles of *Random*,



(a) SVM-based method + Average Prediction shift **(b)** SVM-based method + Average Hit Ratio

Fig. 11 Attack ability of shilling attack methods after sanitization. **a** SVM-based method + average prediction shift. **b** SVM-based method + average hit ratio

Average, *Bandwagon* and *AoP 40%* are identified successfully and their attack ability is reduced dramatically after sanitization. On the other hand, as most of profiles generated by our proposed method evades SVM-based detection, its attack ability is still reasonable after sanitization. These results confirm that the current attack methods can be easily detected by SVM-based method. However, when an adversary considers correlation of rated items, the malicious profiles are able to evade the detection and influences the recommendation of the system successfully. With more accurate information on real users, more efficient malicious profiles can be generated by our method. This is because the generated profiles are more similar to real profiles, i.e., the malicious profiles are more difficult to be detected. Although different unsupervised and supervised methods may perform differently, the mechanisms of their detection are similar. As our proposed shilling attack can successfully evade both PCA-based and SVM-based detection, which are the most effective unsupervised and supervised method respectively, it provides confidence that the proposed attack strategy will achieve the similar performance on other detection methods.

5.2.5 Running time

The running time of the attack methods is also evaluated. As all attack methods except ours select rated item randomly from all items or popular items for malicious profiles, their average running time of generating a malicious user profile with $Fs = 0.05$ is around 0.12 s. However, since a profile is generated according to Eq. (3) in our method, its average running time is 155.80 s for each sample. Although the computational complexity increases, malicious profile generation is often carried out offline. The additional running time required by our method may not be critical.

6 Conclusion and future work

In order to maintain recommendation with good quality, the security of recommender systems is a critical issue to reduce the damage of an adversarial attack. Although malicious user profiles generated by most shilling attack methods can be identified by recent shilling attack detection methods accurately, the rating pattern of these attack methods is different from real users. When an attack method increases the similarity of its malicious profiles to real users, the current shilling attack methods may not be able to defend the attack efficiently.

This study measures the rating pattern of real users in terms of correlation of rated items. We show the difference between the rated item correlation of real profiles and the existing shilling attack methods. It may explain why these methods are easily detected by the existing attack detection methods. We also demonstrate the loophole of current shilling attack detection methods by proposing a shilling attack which considers not only popularity but also correlation of related items. With incomplete knowledge of the real users, the attack ability measured by hit ratio and prediction shift of our proposed method still remains in high level in comparison with other attack methods after removing suspected profiles identified by the attack detection methods. It is because detection accuracy on the malicious profiles generated by the proposed method is low. The results suggest that considering rating pattern of real users increases disguise ability of shilling attack effectively.

One limitation of our method is the increased computational complexity with respect to other existing shilling attack methods which select items randomly. It may not be a critical issue since the malicious profiles are usually generated offline. One possible solution is to prepare two sets of items with highest popularity and rated item correlation. Items are randomly selected from these two item sets for each malicious profile. However, the attack and disguise ability of generated profiles may be lower than the method

discussed in this paper. Another possible extension of this study is to adopt rated item correlation to attack detection in order to improve the security of the recommender systems. Rated item correlation can be used as a feature in attack detection with supervised learning. A user profile may be identified as the malicious class if its rated item correlation is different from the real ones.

Acknowledgements This paper is supported by the Natural Science Foundation of Guangdong Province, China (No. 2018A030313203), the Scientific Research Foundation of Henan Normal University (qd15135) and Research Grants for Universities and Colleges in Henan (17A520037).

References

1. Ricci F (2011) Recommender systems handbook. Springer, Berlin
2. Ding L, Han B, Wang S, Li X, Song B (2017) User-centered recommendation using us-elm based on dynamic graph model in e-commerce. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-017-0751-z
3. Silveira T, Zhang M, Lin X, Liu Y, Ma S (2017) How good your recommender system is? A survey on evaluations in recommendation. Int J Mach Learn Cybern 6:1–19
4. Ortega F, Hernando A, Bobadilla J, Kang JH (2016) Recommending items to group of users using matrix factorization based collaborative filtering. Inf Sci 345(C):313–324
5. Shan L, Lin L, Sun C, Wang X, Liu B (2016) Optimizing ranking for response prediction via triplet-wise learning from historical feedback. Int J Mach Learn Cybern 8(6):1777–1793
6. Patel K, Thakkar A, Shah C, Makvana K (2016) A state of art survey on shilling attack in collaborative filtering based recommendation system. Springer, Berlin
7. Mobasher B, Burke R, Bhaumik R, Williams C (2007) Toward trustworthy recommender systems: an analysis of attack models and algorithm robustness. ACM Trans Internet Technol 7(4):23
8. Gunes I, Kaleli C, Bilge A, Polat H (2014) Shilling attacks against recommender systems: a comprehensive survey. Artif Intell Rev 42(4):767–799
9. Lam SK, Riedl J (2004) Shilling recommender systems for fun and profit. Int Conf World Wide Web 2004:393–402
10. Gunes I, Bilge A, Polat H (2013) Shilling attacks against memory-based privacy-preserving recommendation algorithms. KSII Trans Internet Inf Syst 7(5):1272–1290
11. Burke R, Mobasher B, Bhaumik R (2005) Limited knowledge shilling attacks in collaborative filtering systems. In: Proceedings of 3rd international workshop in intelligent techniques for personalization (ITWP 2005), 19th international joint conference on artificial intelligence (IJCAI 2005), Edinburgh, Scotland
12. Zhang F, Zhou Q (2012) A meta-learning-based approach for detecting profile injection attacks in collaborative recommender systems. J Comput 7(1):226–234
13. Zhang Z, Kulkarni SR (2013) Graph-based detection of shilling attacks in recommender systems. In: Proceedings of the IEEE international workshop on machine learning for signal processing, pp 1–6
14. Yang Z, Cai Z (2017) Detecting abnormal profiles in collaborative filtering recommender systems. J Intell Inf Syst 48(3):499–518
15. Xia N, Desrosiers C, Karypis G (2015) A comprehensive survey of neighborhood-based recommendation methods. Springer, Berlin
16. Luo Z, Liang C (2017) An insider attack on shilling attack detection for recommendation systems. In: 2016 7th IEEE international conference on software engineering and service science (ICSESS), Beijing, pp 277–280
17. Mobasher B, Burke R, Bhaumik R, Williams C (2005) Effective attack models for shilling item-based collaborative filtering systems. In: Proceedings of the Webkdd workshop, Chicago
18. Seminario CE (2013) Accuracy and robustness impacts of power user attacks on collaborative recommender systems. In: Proceedings of the 7th ACM conference on recommender systems, RecSys 2013. ACM Press, New York, pp 447–450
19. Mahara ST (2016) A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment. Procedia Comput Sci 89:450–456
20. O'Mahony MP, Hurley NJ, Silvestre GCM (2005) Recommender systems: attack types and strategies. In: The twentieth national conference on artificial intelligence and the seventeenth innovative applications of artificial intelligence conference, Pittsburgh, 9–13 July 2005, pp 334–339
21. O'Mahony MP, Hurley NJ, Silvestre G, CM (2006) Detecting noise in recommender system databases. In: International conference on intelligent user interfaces, Sydney, 29 Jan–1 Feb 2006, pp 109–115
22. Hurley N, Cheng Z, Zhang M (2009) Statistical attack detection. In: ACM conference on recommender systems, Recsys 2009, New York, pp 149–156
23. Zhang F, Zhou Q (2014) HHT–SVM: an online method for detecting profile injection attacks in collaborative recommender systems. Knowl Based Syst 65(4):96–105
24. Chirita PA, Nejdl W, Zamfir C (2005) Preventing shilling attacks in online recommender systems. In: ACM international workshop on web information and data management. ACM, New York, USA, pp 67–74
25. Li WT, Gao M, Li H, Xiong QY, Wen JH, Ling B (2011) An shilling attack detection algorithm based on popularity degree features. Acta Autom Sin 41(9):1563–1576
26. Su XF, Zeng HJ, Chen Z (2005) Finding group shilling in recommendation system. In: Proceedings of the 14th international conference on world wide web, Chiba, Japan, pp 960–961
27. Burke R, Mobasher B, Williams C, Bhaumik R (2006) Classification features for attack detection in collaborative recommender systems. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, Philadelphia, PA, USA, pp 542–547
28. Williams CA, Mobasher B, Burke R (2007) Defending recommender systems: detection of profile injection attacks. Serv Oriented Comput Appl 1(3):157–170
29. Chengshu LV, Wang W (2013) Semi-supervised shilling attacks detection method based on SVM-KNN. Comput Eng Appl 49(22):7–10
30. Wu Z, Wu J, Cao J, Tao D (2012) HySAD: a semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, Beijing, China, pp 985–993
31. Zahra S, Ghazanfar MA, Khalid A, Azam MA, Naeem U, Prugel-Bennett A (2015) Novel centroid selection approaches for kmeans-clustering based recommender systems. Inf Sci 320(C):156–189
32. Mehta B, Hofmann T, Fankhauser P (2007) Lies and propaganda: detecting spam users in collaborative filtering. In: International conference on intelligent user interfaces, Honolulu, 28–31 Jan 2007, pp 14–21
33. Mehta B (2007) Unsupervised shilling detection for collaborative filtering. In: AAAI conference on artificial intelligence, Vancouver, 22–26 July, pp 1402–1407

34. Sheugh L, Alizadeh SH (2018) A novel 2D-graph clustering method based on trust and similarity measures to enhance accuracy and coverage in recommender systems. Inf Sci 432:210–230
35. Ferrari DG, Castro LND (2015) Clustering algorithm selection by meta-learning systems: a new distance-based problem characterization and ranking combination methods. Elsevier, Amsterdam
36. Agrawal R, Imielinski T, Swami AN (1993) Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD international conference on management of data, Washington, DC, USA, 25–28 May 1993, pp 207–216
37. Brin S, Motwani R, Silverstein C (1997) Beyond market baskets. ACM Sigmod Rec 26(2):265–276
38. Taha A, Hadi AS (2016) Pair-wise association measures for categorical and mixed data. Inf Sci 346–347:73–89
39. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1–2):273–324
40. O'Mahony M, Hurley N, Kushmerick N, Silvestre G (2004) Collaborative recommendation: a robustness analysis. ACM Trans Internet Technol 4(4):344–377
41. Burke R, OMahony MP, Hurley NJ (2011) Robust collaborative recommendation. Springer, Berlin
42. Maratea A, Petrosino A, Manzo M (2014) Adjusted f-measure and kernel scaling for imbalanced data learning. Inf Sci 257(2):331–341