2015 IEEE 15th International Conference on Data Mining Workshops

Defending Suspected Users by Exploiting Specific Distance Metric in Collaborative Filtering Recommender Systems

Zhihai Yang
MOE KLINNS Lab
Xi'an Jiaotong University
Xi'an, China
Email: zhyang_xjtu@sina.com

Zhongmin Cai
MOE KLINNS Lab
Xi'an Jiaotong University
Xi'an, China
Email: zmcai@mail.xjtu.edu.cn

***Abstract*—Collaborative filtering recommender systems (CFRSs) are critical components of existing popular e-commerce websites to make personalized recommendations. In practice, CFRSs are highly vulnerable to "shilling" attacks or "profile injection" attacks due to its openness. A number of detection methods have been proposed to make CFRSs resistant to such attacks. However, some of them distinguished attackers by using typical similarity metrics, which are difficult to fully defend all attackers and show high computation time, although they can be effective to capture the concerned attackers in some extent. In this paper, we propose an unsupervised method to detect such attacks. Firstly, we filter out more genuine users by using suspected target items as far as possible in order to reduce time consumption. Based on the remained result of the first stage, we employ a new similarity metric to further filter out the remained genuine users, which combines the traditional similarity metric and the linkage information between users to improve the accuracy of similarity of users. Experimental results show that our proposed detection method is superior to benchmarked method.**

***Keywords*-recommender system; shilling attack; attack detection**

I. Introduction

Personalization recommender systems (RSs) become more and more popular in e-commerce websites to automatically make personalized suggestions of services or products to customers. Collaborative filtering recommender systems (CFRSs) have been proved to be one of the most successful RSs used by the well-known e-commerce companies such as Amazon, eBay etc [1], [2], [3], [4]. However, CFRSs are prone to manipulation from attackers due to its openness, which carefully inject chosen attack profiles into CFRSs to bias the recommendation results to their benefits or decrease the trustworthiness of recommendation. These phenomenons are often called "shilling" attacks or "profile injection" attacks [1]. Therefore, constructing an effective detection method to detect the attackers and remove them from the CFRSs is crucial.

While a wide range of detection techniques have been used, some of them were based on calculating similarity between users (consists of attackers and genuine users) in order to discriminate those attackers [4], [5], [6]. In practice, it is difficult to capture all concerned attackers by exploiting the similarity of users, although it can be helpful to filtering out more genuine users. Moreover, calculating directly the similarity between users on a whole dataset consumed high computation time. Current detection methods [6], [7] usually aim to construct a similarity metric for each user guided by traditional similarity measurements including Pearson Correlation Coefficient (PCC), Cosine Similarity etc. These similarity metrics calculated globally the similarity between users, however, it is important to note that such similarity metrics may be insufficient for detecting "shilling" attacks. As is known, the similarity between attackers reported higher than part genuine users, it is noteworthy that few genuine users also possessed higher similarity with the attackers. Exploiting the traditional similarity metrics for discriminating between attackers and genuine users will mislead the detection task. Thus, how to distinguish effectively between attackers and genuine users with low computation time is a key challenge for detecting "shilling" attacks.

Based on the aforementioned tasks, we propose a new detection method to make CFRSs resistant to such attacks, which exploits a novel metric for calculating similarity between users. To reduce the computation time, we firstly filter out more genuine users as far as possible determined by using suspected target items. Since the attackers will target one or more specific items with lowest or highest rating many times if they want to demote (called nuke attack) or promote (called push attack) the items to the recommendation list, we can find out all suspected target items by using an absolute count threshold. Based on the remaining users, we employ a new similarity metric inspired from the pairwised specific distance, aiming to measure effectively the similarity between users. Owing to the similarity between users can be measured partly by traditional similarity measurements, it is difficult to further discriminate the difference between some users, where attacker and few genuine users show higher similarity. Both global and local similarity between users are considered by the pairwised specific distance, which can better address the tasks that traditional methods are difficult to deal with. Experimental

978-1-4673-8493-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDMW.2015.89

results validate the effectiveness of our proposed method and also demonstrate the outperformance of our method over benchmarked method.

The rest of this paper is organized as follows. Section 2 reviews some related work. Section 3 shows attack models and attack profiles. Section 4 presents our proposed method. In Section 5, experimental results are reported and analyzed. Finally, we conclude the paper with a brief summary and prospect the directions of future works.

II. Related work

"Shilling" attacks or "profile injection" attacks are serious threats to the CFRSs. Published studies of detecting such attacks have investigated diverse detection methods. To name a few, Su et al. [5] presented a spreading similarity algorithm in order to capture groups of similar attackers. Then, [1], [8], [9], [10], [11] developed different attributes derived from user profiles for their utility in attack detection. One of their attributes is Degree of Similarity with Top Neighbors (DegSin) which calculates the similarity between users by using Pearson Correlation Coefficient. Mehta et al. [4] proposed an unsupervised detection method based on principal component analysis and performed well against "shilling attacks. The motivation behind this method is that attacker have higher similarity (by using Pearson Correlation coefficient) each other and as well as having high similarity with a large number of genuine users. However, a few of genuine users are misclassified and the detection performance in AOP attack is not satisfactory. Recently, Zhang et al. [6] proposed an online method, HHT-SVM, to detect "profile injection" attacks by combining Hilbert-Huang transform (HHT) and support vector machine (SVM). They created rating series for each user profile based on the novelty and popularity of items in order to provide basic data for feature extraction. The precision of their method shown better than the benchmarked methods, but the limitations are that the detection method should be re-trained offline when new types of attacks are generated and calculating similarity between items consumed a lot of time. After that, Zhou et al. [7] proposed an unsupervised technique for identifying group attack profiles, which uses an improved metric based on Degree of Similarity with Top Neighbors (DegSim) and Rating Deviation from Mean Agreement (RDMA). Their experimental results shown a good detection performance for their proposed method. However, calculating the DegSim consumed a lot of time. The similarity of computing requires a significant amount of computing resources which is a bottleneck for real life applications. Moreover, traditional similarity metrics reported limited capability to measure the similarity between users. In this paper, we exploit a new similarity metric to measure similarity between users, our aim is to explore an effective detection method and as well as make a low computational complexity.

III. Attack models and attack profiles

The attackers have different attack intents to bias the recommendation results to achieve their benefits in CFRSs, which promotes (called push attack) or demotes (called nuke attack) the target items with the highest or lowest rating. In order to push or nuke a target item, the attackers should know clearly the form of attack profiles [1], [8], [12]. The general form of an attack profile is shown in Table 1. The details of the four sets of items are described as follows:

I_S: The set of selected items with specified rating by the function (i_k^S) [13];

I_F: A set of filler items, received items with randomly chosen by the function (i_l^F);

I_N: A set of items with no ratings;

I_T: A set of target items with singleton or multiple items, called single-target attack or multi-target attack. The rating is (i_j^T), generally rated the maximum or minimum value in the entire profiles.

To conduct experimental data, we introduce 7 general attack models to generate attack profiles as shown in Table 2. The detail of these attack models are briefly described as follows:

1) Random attack: $I_S = \phi$ and $\rho(i) \sim N(\overline{r}, \overline{\sigma}^2)$ [13].

2) Average attack: $I_S = \phi$ and $\rho(i) \sim N(\overline{r_i}, \overline{\sigma_i}^2)$ [13].

3) Bandwagon (average) attack: I_S contains a set of popular items, $\sigma(i) = r_{max}/r_{min}$ (push/nuke) and $\rho(i) \sim N(\overline{r_i}, \overline{\sigma_i}^2)$ [14].

4) Segment attack: I_S contains a set of segmented items, $\sigma(i) = r_{max}/r_{min}$ (push/nuke) and $\rho(i) = r_{min}/r_{max}$ (push/nuke) [12].

5) Reverse Bandwagon attack: I_S contains a set of unpopular items, $\sigma(i) = r_{min}/r_{max}$ (push/nuke) and $\rho(i) \sim N(\overline{r}, \overline{\sigma}^2)$ [12].

6) Love/Hate attack: $I_S = \phi$ and $\rho(i) = r_{min}/r_{max}$ (push/nuke) [12].

7) AOP attack: A simple and effective strategy to obfuscate the Average attack is to choose filler items with equal probability from the top x% of most popular items rather than from the entire collection of items [15].

IV. Our proposed approach

Our approach consists of two stages: the stage of filtering out genuine users by exploiting suspected target items and the stage of discriminating attackers by employing a new similarity metric. At the first stage, we filter out a part of genuine users in order to reduce the computation time. At the second stage, we mainly focus on the effective similarity metric to better distinguish between attackers and genuine users based on the remaining users of the first stage.

To reduce the computation time for our proposed method, we firstly make a preprocessing to filter out more genuine users as far as possible. As is known, the attackers should promote or demote one or more target items with the highest or lowest rating to achieve their attack intentions.

Table I: General form of attack profiles

I_T			I_S			I_F			I_N		
i_1^T	...	i_j^T	i_1^S	...	i_k^S	i_1^F	...	i_l^F	i_1^N	...	i_v^N
$\gamma(i_1^T)$	...	$\gamma(i_j^T)$	$\sigma(i_1^S)$	...	$\sigma(i_k^S)$	$\rho(i_1^F)$	...	$\rho(i_l^F)$	$null$	...	$null$

Table II: Attack models summary.

Attack Models	I_S		I_F		I_N	I_T
	Items	Rating	Items	Rating		push/nuke
Random	null		randomly chosen	normal dist around system mean.	null	r_{max}/r_{min}
Average	null		randomly chosen	normal dist around item mean.	null	r_{max}/r_{min}
Bandwagon (average)	popular items	r_{max}/r_{min}	randomly chosen	normal dist around item mean.	null	r_{max}/r_{min}
Segment	segmented items	r_{max}/r_{min}	randomly chosen	r_{min}/r_{max}	null	r_{max}/r_{min}
Reverse Bandwagon	unpopular items	r_{min}/r_{max}	randomly chosen	system mean	null	r_{max}/r_{min}
Love/Hate	null		randomly chosen	r_{min}/r_{max}	null	r_{max}/r_{min}
AOP	null		x-% popular items, ratings set with normal dist around item mean.		null	r_{max}/r_{min}

In other words, the attackers will target these specific items many times if they want to push or nuke the items to the recommendation list [7]. To determine the target items, we design an absolute count threshold ε which pushes or nukes the same items with the highest or lowest at least ε times. If the count for an item is greater than ε, then the item sus_i is regarded as suspected target item. Users (consist of attackers and genuine users) who rated the sus_i with the highest r_{max} or lowest r_{min} are considered as attackers. It is noteworthy that there will be more false positives or false negative if ε is too small or large. In our work, we set ε equal to 6 which is an empirical threshold decided from a list of experiments.

Based on the remaining result of first stage, we continue to filter out all genuine users as far as possible by employing a new similarity metric. Since traditional similarity metrics such as PCC, Cosine similarity etc. are difficult to fully measure the similarity between users, we exploit a novel similarity metric to distinguish between attackers and genuine users, which combines the global and local similarity metrics and uses the linkage information between users to further improve accuracy of similarity measurement. The goal is expect to better measure the similarity between users by the new metric over other competing similarity metrics. Note that the linkages between users (contain attackers and genuine users) are formed by various reasons, which may bring about linkage for two different users, these linkages will mislead the detection task. Just as Fig. 1 illustrated, attacker A_3 can be linked to genuine user G_4 because A_3 mimics some rating details from G_4, and at the same time, genuine user G_2 also has similar rating details close to genuine user G_4 because the two genuine users have similar ratings in the same items that few co-rated by A_3 and G_4. However, G_4 and A_3 are from different classes, i.e., "genuine" and "attack". And the similarity between G_4 and G_2 are higher than between G_2 and A_3. Moreover, the similarity between attackers are higher than most of genuine users in practice. How to shrink distances within the users from the same class closer, whereas separating the users from different classes far away? Inspired from the idea of pairwised specific distance [16], we also constrain the distance between users from the same class to be smaller than their Euclidean distance, and the distance between users from different classes larger than their Euclidean distance by exploiting the following constraints:

$$d'_{uv} \begin{cases} < d_{uv}, uv \in S \\ > d_{uv}, uv \in D \end{cases} \tag{1}$$

where d_{uv} is Euclidean distance between user u and user v,

$$d_{uv}^{I_{uv}} = ((R_{uI_{uv}} - R_{vI_{uv}})^T (R_{uI_{uv}} - R_{vI_{uv}}))^{\frac{1}{2}} \tag{2}$$

where $d_{uv}^{I_{uv}}$ is Euclidean distance between user u and user v on the co-rated item set I_{uv}, $R_{uI_{uv}}$ and $R_{vI_{uv}}$ are rating vectors rated by users u and v, respectively. In addition, the relationship between A_3, G_4 and G_2 shown in Fig. 1 is non-metric [16], where A_3 is similar to G_4, G_4 is similar to G_2, but A_3 may not similar to G_2. These non-metric $NonM(A_3G_4G_2)$ properties may lead to inappropriate conclusions. In other words, a non-metric linkage implies that the two linkages within it are carrying users from different classes including genuine user and attacker.

Given a dataset, a graph $G = \{V, E\}$ can be derived, where V contains all users (consist of genuine users and attackers) and edges in E indicate the similarities between vertices. The available linkage graph for the whole data set is denoted as L. Let w_{uv} denotes the distance metric for each linked pair of users $uv \in L$, To capture the extent of difference between w_{uv} and w_{ux} for each non-metric

linkage, it is natural to minimize

$$A: \sum_{v,NonM(xuv)} d_{uv} w_{vx}^T w_{ux}$$

where d_{uv} is the Euclidean distance between user u and user v.

Algorithm 1 Detection algorithm.

Input: Data matrix **M**;
Empirical thresholds ε and τ;
Termination threshold δ;
Maximum iteration limit T;
Linkage graph for the whole dataset G;
Output: The detected result **DR**;
Process:
Step 1:
1: $\{(i, N_i)\}_i^{|I|} = \{(i, N_i)|N_i =$ the number of ratings on item i with $r_{min}\}$;
2: $item_{sus} = null$;
3: $user_{sus} = null$;
4: **if** $N_i > \varepsilon$ **then**
5: $item_{sus} \leftarrow i$;
6: **end if**
7: **for** each user u in U **do**
8: **if** u rated item i $(i \in item_{sus})$ with r_{min} **then**
9: add u to $user_{sus}$;
10: **end if**
11: **end for**
Step 2:
12: Initialize $w^{(0)} = 1$;
13: **for** $t = 1, ..., T$ **do**
14: **for** $uv \in G \wedge u, v \in user_{sus}$ **do**
15: Learn w_{uv} by solving Eq. 3;
16: **end for**
17: **if** $\|w^{(t)} - w^{(t-1)}\|_2^2 \in \delta$ **then**
18: break;
19: **end if**
20: **end for**
21: $w^{(0)} = w$;
22: Calculate similarity between remained users in $user_{sus}$ by using the new metric;
23: **for** each pair of users u and v in $user_{sus}$ **do**
24: **if** $s_{u,v} > \tau$ **then**
25: remain u and v in $user_{sus}$;
26: **else**
27: remove u and v from $user_{sus}$;
28: **end if**
29: **end for**
30: **DR** = $user_{sus}$;
31: **Return:** The detected result **DR**;

To consider two cliques $Clique(A_1A_2A_3)$ and $Clique(G_1G_2G_3)$ as shown in Fig. 1, it is obvious

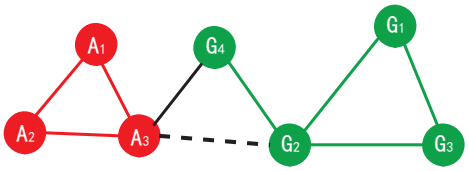

Figure 1: The diagram of link relationship between users, where red nodes denote attackers and green nodes denote genuine users.

that three linkages within each clique share the same class, genuine or attack. In other words, the smaller the value of $\|w_{ux} - w_{vx}\|_2^2$ is kept, the more similar the pariwised distance metrics for the two linkages $ux \in L$ and $vx \in L$ on their attributes, the more similar rating details should be shared by $ux \in L$ and $vx \in L$, and thus, the larger the extent that w_{ux} and w_{vx} are similar. In order to combine each pair of distance metrics and minimize all of them, it is natural to minimize

$$B: \sum_{x,NonM(uvx)} s_{uv}\|w_{ux} - w_{vx}\|_2^2 + s_{vx}\|w_{uv} - w_{ux}\|_2^2 + s_{ux}\|w_{uv} - w_{vx}\|_2^2$$

where s_{uv}, s_{ux} and s_{vx} reflect the attributes similarities between each pair of users, uv, ux and vx, respectively. $s_{uv} = exp(-\frac{{d'_{uv}}^2}{\sigma^2})$, $s_{ux} = exp(-\frac{{d'_{ux}}^2}{\sigma^2})$ and $s_{vx} = exp(-\frac{{d'_{vx}}^2}{\sigma^2})$, $\sigma = \theta \cdot \overline{d}$, $\overline{d}$ is the average distance among the dataset and θ equal to 1. Based on a distance metric w_{uv} for each linked pair $uv \in L$, the pairwised specific distance between them can be defined as: $d'_{uv} = (w_{uv}^T \delta_{R_{uI_{uv}}, R_{vI_{uv}}})^{\frac{1}{2}}$, where $\delta_{R_{uI_{uv}}, R_{vI_{uv}}} = (R_{uI_{uv}} - R_{vI_{uv}}) \odot (R_{uI_{uv}} - R_{vI_{uv}})$, $\odot$ is the element-wise product on two vectors. [16]

Finally, we combine all aforementioned cases for pairwised specific distance to formulate our problem into the following optimization framework

$$\begin{aligned} \arg\min_{w} & \sum_{uv \in L} \|w_{uv}\|_2^2 + \alpha A + \beta B \\ s.t. & \sum_{x} w_{uvx} = M \\ & d'_{uv} < d_{uv}, uv \in S \\ & d'_{uv} > d_{uv}, uv \in D \end{aligned} \tag{3}$$

where α and β are two tradeoff parameters. M is the dimensionality of vector for each metric w_{uv}. Due to the page limit, we will show more details in a longer version.

The pseudo-code of our proposed approach is summarized in Algorithm 1. In Algorithm 1, I denotes the item set in the whole system in line 1. In line 7, U denotes the user set in the whole system.

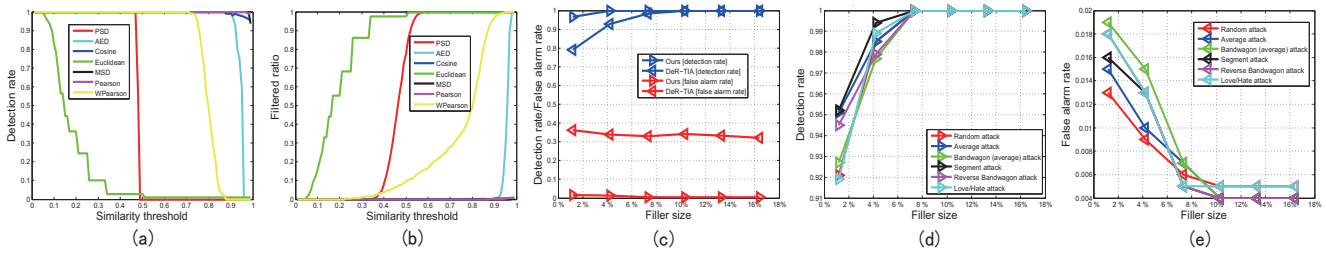

Figure 2: (a) The comparison of detection rate in 7 different similarity metrics with the similarity threshold increasing, where PSD is the proposed similarity metric; (b) The comparison of filtered ratio of genuine users in 7 different similarity metrics with the similarity threshold increasing; (c) The comparison of detection rates and false alarm rates in different detection methods when attack size is 6.4% and filler size varies, single-target AOP attack; (d) The comparison of detection rates of our method in 6 different attack models. Attack size is 6.4% and filler size varies; (e) The comparison of false alarm rates of our method in 6 different attack models. Attack size is 6.4% and filler size varies.

V. Experiments and Analysis

In this section, we firstly introduce the experimental data and settings. And then, we briefly analyze the experimental results.

A. Experimental data and settings

In our experiments, we use the MovieLens-100K [1] dataset to describe the behaviors of genuine users in RSs. It was collected by the GroupLens Research Project at the University of Minnesota. The MovieLens-100K dataset consists of 100,000 ratings on 1682 movies by 943 raters and each rater had to rate at least 20 movies. All ratings are in the form of integral values between minimum value 1 and maximum value 5. In our attack experiments, attack profiles are created according to different attack models as shown in Table 2. The attack profiles indicate the attacker's intentions that he wishes a particular item can be rated the highest or the lowest. In this paper, we just detect the nuke attacks. Of course, our proposed approach can be used to detect the push attacks. For each attack model, we generate samples of nuke attack profiles according to the corresponding attack model with diverse attack sizes [2] $\{1.1\%, 6.4\%, 11.7\%, 17.0\%, 22.3\%, 27.6\%\}$ and filler sizes [3] $\{1.2\%, 4.2\%, 7.3\%, 10.3\%, 13.3\%, 16.4\%\}$. To ensure the rationality of the results, the target item is randomly selected for each attack profile. Therefore, we have 252 (7 $\times$ 6 $\times$ 6) experimental datasets including 7 attack models, 6 different attack sizes and 6 different filler sizes. We should not forget that algorithms are strongly dependent on the characteristics of the dataset, so the results obtained in our study may not coincide with those obtained with data from other domains. All numerical studies are implemented using MATLAB R2013a and Python 2.6.8 on a server with Intel(R) Core(TM) i7-4790 3.60GHz CPU, 32G memory and Linux operating system.

To measure the effectiveness of the presented detection methods, we use detection rate and false alarm rate in this paper. Detection rate is defined as the number of detected attack profiles divided by the number of attack profiles.

$$\text{detection rate} = \frac{\#\text{Detection}}{\#\text{Attack Profiles}} \tag{4}$$

False alarm rate is the number of genuine profiles that are predicted as attack profiles divided by the number of genuine profiles.

$$\text{falsealarm rate} = \frac{\#\text{False alarm}}{\#\text{Genuine Profiles}} \tag{5}$$

B. Experimental results and analysis

To validate the effectiveness of pairwised specific distance (PSD) for measuring similarity between users, we take 6 similarity metrics to compare with the employed new metric including PCC, Euclidean Distance etc. It is noteworthy that filtered ratio is defined as the number of filtered genuine users by using a similarity empirical threshold divided by all genuine users in the system. As illustrated in Figures 2(a) and 2(b), different similarity methods show different detection rates and filtered ratios when the similarity thresholds fixed. With the similarity threshold increasing, the faster the curve of detection rate decreases monotonously, the better the similarity method as shown in Figure 2(a). Meanwhile, the faster the curve of the filtered ratios increases monotonously, the better the similarity method as shown in Figure 2(b). It is obvious that PSD are among the best of all the compared similarity metrics. To determine an effective threshold for PSD, we set τ equal to 0.45.

To investigate whether the new distance metric is beneficial, we conduct a list of experiments in AOP attack to examine the effectiveness of our detection method. As shown in Figure 2(c), it shows that our proposed method has

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

[2] The ratio between the number of attackers and genuine users.

[3] The ratio between the number of items rated by user u and the number of entire items in the recommender systems.

achieved the significantly good performance in compared with DeR-TIA [7] when attack size is 6.4% and filler size varies.

Furthermore, we conduct experiments to validate the detection performance of our method in other 6 different attack models as shown in Figures 2(d) and 2(e). It is observed that the effectiveness of the proposed method is obvious when attack size is 6.4% and filler size varies. With the filler size increasing, the detection rates achieve the highest except for the early phase (filler size $< 7.3\%$). At the same time, the false alarm rates are acceptable.

VI. Conclusion and further discussions

"Shilling" attacks and "profile injection" attacks are the main threats in CFRSs. These attack profiles have a good probability of being similar rating details to a large number of genuine profiles in order to make them hard to be detected. In this paper, we propose an unsupervised detection method for detecting such attacks, which exploits the pairwised specific distance to generate a similarity metric. Firstly, we filter our more genuine users by using determined target items as far as possible. And then , based on the remained users, we continue to filter out more genuine users used by an empirical threshold of the new similarity metric. The experimental results show that our proposed method is superior to the benchmarked method and validate the effectiveness of the proposed method. In our future work, we will consider more attack models such as power user attack and power item attack and conduct experiments on other datasets to validate the effectiveness of our proposed method.

Acknowledgment

The research presented in this paper is supported in part by the National Natural Science Foundation (61221063, U1301254), 863 High Tech Development Plan (2012AA011003) and 111 International Collaboration Program, of China.

References

[1] R. Burke, B. Mobasher, and C. Williams, "Classification features for attack detection in collaborative recommender systems," *International Conference on Knowledge Discovery and Data Mining*, pp. 17–20, 2006.

[2] K. Bryan, M. OMahony, and P. Cunningham, "Unsupervised retrieval of attack profiles in collaborative recommender systems," *ACM conference on Recommender Systems*, pp. 155–162, 2008.

[3] C. Chung, P. Hsu, and S. Huang, "A novel approach to filter out malicious rating profiles from recommender systems," *Journal of Decision Support Systems*, pp. 314–325, 2013.

[4] B. Mehta, T. Hofmann, and P. Fankhauser, "Lies and propaganda: detecting spam users in collaborative filtering," *In: IUI07: Proceedings of the 12th International Conference on Intelligent User Interfaces*, pp. 14–21, 2007.

[5] X. Su, H. Zeng, and Z. Chen, "Finding group shilling in recommendation system," *WWW*, p. 960C961, 2005.

[6] F. Zhang and Q. Zhou, "HHT-SVM: An online method for detecting profile injection attacks in collaborative recommender systems," *Knowledge-Based Systems*, 2014.

[7] W. Zhou, Y. S. Koh, J. H. Wen, S. Burki, and G. Dobbie, "Detection of abnormal profiles on group attacks in recommender systems," *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 955–958, 2014.

[8] C. A. Williams, B. Mobasher, and R. Burke, "Defending recommender systems: detection of profile injection attacks," *SOCA*, pp. 157–170, 2007.

[9] C. A. Williams, B. Mobasher, R. Burke, and R. Bhaumik, "Detecting profile injection attacks in collaborative filtering: a classification-based approach," *Advances in Web Mining and Web Usage Analysis*, pp. 167–186, 2007.

[10] M. Morid and M. Shajari, "Defending recommender systems by influence analysis," *Information Retrieval*, pp. 137–152, 2014.

[11] Z. Zhang and S. R. Kulkarni, "Detection of shilling attacks in recommender systems via spectral clustering," *International Conference on Information Fusion*, pp. 1–8, 2014.

[12] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, "Shilling attacks against recommender systems: A comprehensive survey," *Artificial Intelligence Review*, pp. 1–33, 2012.

[13] Z. Zhang and S. Kulkarni, "Graph-based detection of shilling attacks in recommender systems," *IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, 2013.

[14] Z. A. Wu, Y. Q. Wang, and J. Cao, "A survey on shilling attack models and detection techniques for recommender systems," *Science China*, vol. 59, no. 7, pp. 551–560, 2014.

[15] C. E. Seminario and D. C. Wilson., "Attacking item-based recommender systems with power items," *ACM Conference on Recommender Systems*, pp. 57–64, 2014.

[16] J. Hu, D. C. Zhan, X. Wu, Y. Jiang, and Z. H. Zhou, "Pairwised specific distance learning from physical linkages," *ACM Trans. Knowl. Discov. Data*, 2014.