PROCEEDINGS OF SCIENCE

Defending Suspected Ratings in Collaborative Filtering Recommender Systems: a Fast Detection Method

Zhihai Yang[1][2]
Xi'an Jiaotong University, Xi'an, 710000, China
E-mail: zhyang_xjtu@sina.com

Aghil Esmaeilikelishomi
Xi'an Jiaotong University, Xi'an, 710000, China
E-mail: ashil_im@sina.cn

Yuan Yang
Xi'an Jiaotong University, Xi'an,710000, China
E-mail: yuanyang@sei.xjtu.edu.cn

Xinyuan Chen
Xi'an Jiaotong University, Xi'an, 710000, China
E-mail: chenxinyuan1994@gmail.com

Collaborative filtering recommender systems (CFRSs) are key components of the well-known E-commerce websites such as Amazon, Yelp etc., to make personalized recommendations. In practice, CFRSs are highly vulnerable to "shilling" attacks. Detection methods based on such attacks have received much attention. However, their detection accuracy is not fully acceptable especially when the attack size or filler size is small. In this paper, we solve the following task: Given the rating dataset, how can we spot abnormal ratings of users as well as keeping reasonable time-consumption? We propose a fast and effective detection method to detect such attacks, which consists of two phases. We firstly capture all suspected users by employing a top-k similarity method for calculating the similarity between users. Finally, we continue to filter out more genuine users by analysing target items as far as possible. In addition, extensive experiments demonstrate that the detection performance of our method outperforms benchmarked methods. It is noteworthy that the recently published attack, PIA (power item attack) including PIA-AS, PIA-ID and PIA-NR can be detected by our proposed method.

ISCC 2015
18-19, December, 2015
Guangzhou, China

[1] Speaker and corresponding author

[2] 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 People's Republic of China.

1. Introduction

Existing popular E-commerce services have not only gained higher customer satisfaction about products or services but derived more benefits from customer ratings since being successfully armed with personalized recommendations. Personalization recommender systems (RSs) become more and more popular in some well-known websites such as Amazon, eBay etc. [1-4]. Collaborative filtering recommender systems (CFRSs) have proved to be one of the most popular RSs. However, CFRSs are highly vulnerable to shilling attacks due to their openness, which are injected with chosen profiles of abnormal ratings in order to control recommendation results to their benefits or decrease the trustworthiness of recommendation [5-6]. Therefore, constructing an effective detection method is crucial to detect attackers and remove them from CFRSs.

Considering that the similarity between attackers is higher than that between genuine users, estimating the user similarity is a fundamental issue for detecting abnormal users. Generally speaking, there are two different principles to measure the similarity between users. Firstly, given a graph, two vertices are equivalent structurally if they have the same structural role. The other is that two vertices are also equivalent structurally if they share a lot of common neighborhoods [7]. In the literature, few similarity methods have been considered by using the first principle such as Pearson Correlation Coefficient (PCC), Cosine Similarity etc. [8]. However, these similarity metrics just calculate the similarity by following the local fashion. Furthermore, those metrics based on the first principle are difficult to handle the large-scale graph. Besides, little previous work is focused on the second principle. Fortunately, Zhang et al. [7] proposed a new method by combining these two principles.

While a wide range of detection approaches have been presented, some were based on calculating similarity by following the aforementioned first principle [3, 9, 10]. The problem remains largely unsolved. In practice, it is difficult to capture all concerned attackers by calculating the similarity of users, although it can be helpful to spot some attackers. Moreover, it is clearly infeasible to apply traditional similarity metrics based on the first principle to large-scale graphs. Exploiting traditional similarity metrics for discriminating between attackers and genuine users will mislead the detection task. Therefore, how to capture abnormal users in large-scale networks is a key challenge for detecting shilling attacks.

According to the aforementioned tasks, in this paper, our goal is to construct a sample and effective detection method that is flexible enough to estimate quickly the similarity between users in a large-scale graph. First of all, we employ a fast Top-k similarity metric [7] to address the similarity calculation task, which considers both two principles mentioned before. It is noteworthy that the Top-k similarity metric measures weight between vertices in the user-user graph by calculating the number of items rated by common neighbors. In reality, both the ratings and items rated by users are important to measure the similarity between users. In other words, the Top-k similarity metric is enough for perfectly measuring the similarity between users. In light of this situation, we continue to capture potential attackers by analyzing target items based on the result at the first stage. As is known, attackers will target one or more specific items with the lowest or highest rating frequently if they want to demote (called nuke attack) or promote (called push attack) their items to recommendation lists, so we can decide suspected target items by using an absolute count threshold [11]. We evaluate the efficiency of our method on Movielens dataset with diverse attack models. In addition, the recently published attack, PIA (power item attack) including PIA-AS, PIA-ID and PIA-NR can be detected by our proposed method.

The paper is structured as follows. In Section 2, we introduce some related work. Section 3 shows attack models and attack profiles. Section 4 describes the proposed method. Experimental results are reported and analyzed in Section 5. Finally, we conclude the paper with a brief summary and predict the direction of the future work.

2. Related Work

This work aims at detecting abnormal users from user profiles for defending “shilling” attacks. Here, we just discuss detection methods related to the work. Su et al. [9] presented a spreading similarity algorithm to detect groups of similar attackers. Then, developed different features extracted from user profiles to capture attackers by exploiting classification-based methods[2, 4, 12, 13, 14,15]. 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. [3] developed an unsupervised detection method based on the principal component analysis which proved to perform well against shilling attacks. The motivation behind this method is that attackers have higher similarity (by using Pearson Correlation coefficient) while a part of genuine users have higher similarities. However, a few of genuine users are misclassified and the detection performance in AOP attack is not satisfactory. After that, Zhou et al. [11] proposed an unsupervised detection method to spot attack profiles by using an improved method based on both DegSim (Degree of Similarity with Top Neighbors) and RDMA (Rating Deviation from Mean Agreement). Their experimental results showed a good detection performance of their proposed method. However, calculating the DegSim consumed a lot of time. Recently, Gnnemann et al. [5] proposed a detection technique by analyzing temporal rating distributions. In addition, Gnnemann et al. [6] presented a new detection approach based on the sound Bayesian framework to detect concerned anomalous ratings.

3. Attack Models and Attack Profiles

Those attackers have different attack intents to control recommendation results to achieve their benefits in CFRSs, which demotes (called nuke attack) or promotes (called push attack) target items with the lowest or highest rating. For the attack profile, its general form is shown in Table 1 [2, 12, 16]. Details of each item set are described as follows:

I_T : A set of target items with singleton or multiple items, is called the single-target attack or multi-target attack. The rating $\gamma\left(i_j^T\right)$ generally assesses the maximum or minimum value in the entire profile [17].

I_S : A set of selected items with a specified rating by the function $\sigma\left(i_k^S\right)$;

I_F : A set of filler items receiving items randomly chosen by the function $\rho\left(i_l^F\right)$;

I_N : A set of items with no ratings;

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
$\gamma\left(i_1^T\right)$	...	$\gamma\left(i_j^T\right)$	$\sigma\left(i_1^S\right)$	...	$\sigma\left(i_k^S\right)$	$\rho\left(i_1^F\right)$	...	$\rho\left(i_l^F\right)$	null	...	null

Table 1: General Form of Attack Profiles

To conduct experimental data, we introduce 9 general attack models to generate attack profiles as shown in Table 2 [16-19].

Attack Models	I_S		I_F		I_N	I_T push or nuke
	Items	Rating	Items	Ratin		
Random	nul		randomly chosen	normal dist around system 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}
PIA-AS	power items		null		null	r_{max}/r_{min}
PIA-ID	power items		null		null	r_{max}/r_{min}
PIA-NR	power items		null		null	r_{max}/r_{min}

Table 2: Attack Models Summary

4. Our Proposed Method

Given an undirected weighted user-user graph G, our goal is to find out suspected users with anomalous ratings on products. In contrast, anomalies represent irregularities which are different between the observed ratings and normal ratings. Since attackers aim to push (rate the highest rating) or nuke (rate the lowest rating) on target items, ratings on target items can not represent the actual situation. To reach our objective, we propose an efficient detection method to address this task. Our approach consists of two stages: the stage of finding the most similar users and the stage of filtering out genuine users by analyzing target items. At the first stage, we employ an effective similarity metric to deal with normal small scale datasets and to handle large-scale datasets for the purpose of shrinking the range of suspected users. Based on the remaining result of first stage, we continue to capture the concerned users by focusing on suspected target items.

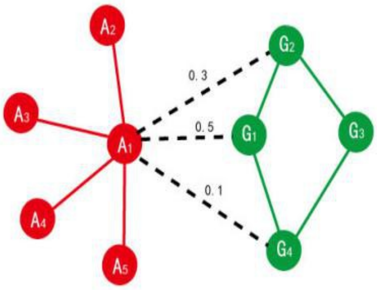

Figure 1: the diagram of link relationship between users in the disconnected network, where red nodes denote attackers and green nodes denote genuine users.

In practice, attackers demote or promote one or more target items with the lowest rating or highest rating to achieve their attack intentions by exploiting similar attack behaviors. In other words, attackers have high similarity to each other while a part of genuine users have high similarities as well. It is noteworthy that traditional similarity metrics are difficult to evaluate accurately the similarity between users (consisting of attackers and genuine users) especially for estimating the similarity between users from disconnected networks, such as Pearson Correlation Coefficient (PCC), Cosine Similarity etc. Just as Fig. 1 shows, a number of attackers (red nodes) have certain similarity with some genuine users (green nodes), although they belong to different networks. Moreover, these metrics are also difficult to scale up to handle large datasets. Thus, how to evaluate effectively the similarity between users in both connected and disconnected network as well as handling large-scale networks is a key challenge. To address these problems, in the first step of our method, we employ a new metric to calculate the similarity between users, called Panther [7], which designs an unified method to quantify the similarity between users by considering the structural equivalence in both common neighbors and the same structural roles. As to applications, we do not know more details in a network, such as connected network or disconnected network, that two vertices in the network are equivalent structurally if they share a lot of common neighbors, etc. In addition, handling large-scale datasets is also a big challenge. It is infeasible to apply traditional similarity metrics to evaluate the similarity between vertices in a large-scale network. The goal of our method in the first stage is to calculate quickly the similarity between users in a large user-user graph by exploiting the presented method.

The newly employed similarity metric is a fast method to evaluate similar vertices. Given a network G, Panther [7] generates randomly R paths with length T. In Panther, we select randomly a vertex in G as the starting vertex and generate random walks of T steps starting from the vertex v_i by exploiting the transition probability [7]. Therefore, the more two vertices appear on the same path, the more the similarity shows between them. Based on the discussion, the path similarity between two vertices v_i and v_j is defined as follows

$$S_{RP}(v_i, v_j) = \frac{|P_{v_i, v_j}|}{R} \tag{4.1}$$

where P_{v_i, v_j} is a subset of paths in G that contain both v_i and v_j . R is the number of random paths [7].

To avoid that Panther favors too much close neighbors, the extension of the Panther is presented by J. Zhang et. al.[7]. Constructing a feature vector for each vertex v_i is a new extension. The similarity between v_i and v_j is re-calculated as follows:

$$S_{RP}(v_i, v_j) = \frac{1}{\|\theta(v_i) - \theta(v_j)\|} \tag{4.2}$$

where $\theta(v_i) = (S_{RP}(v_i, v_{(1)}), S_{RP}(v_i, v_{(2)}), \ldots, S_{RP}(v_i, v_{(D)}))$, S_{RP} denotes the *n*-th largest path similarity between v_i and v_n . *N* is the number of top similarities calculated by Panther.

Although the employed similarity metric can quickly calculate the similarity between vertices in a large-scale network, it is also difficult to fully evaluate the similarity between users in a large-scale user-user graph. The weight between two users at the first stage is dependent on the number of items rated by the two users, in reality, however, ratings on these items are also important to evaluate the similarity between users. Based on this observation, we design naturally the second stage with our method for further capturing concerned attackers. As aforementioned discussed, attackers demote or promote one or more target items with the lowest or highest rating to achieve their attack intentions. It means that attackers will target suspected items repeatedly if they want to demote or promote items to recommendation lists [11]. To capture these target items, we use an absolute count threshold ε , which nukes or pushes the same item with the lowest or highest at the least times. In other words, an item is regarded as a suspected target item i_{sus} if the count for an item is greater than ε . Ultimately, users who are rated i_{sus} with the lowest or highest are considered as attackers (suspected users).

5. Experiments and Discussions

We firstly introduce the experimental data and settings, then we briefly analyze experimental results.

5.1 Experimental Data and Settings

We use the MovieLens-100K dataset to describe behaviors of genuine users in our experiments. The dataset is constructed by 100,000 rating profiles rated by 943 users on 1,682 different movies. Each user had to rate at least 20 movies (from the minimum value 1 to the maximum value 5). In our attack experiments, attack profiles are created according to different attack models as shown in Table 2. For each attack model, we generate attack profiles according to the corresponding attack model with diverse attack sizes {1.1%, 6.4%, 11.7%, 17.0%, 22.3%, 27.6%} and filler sizes {1.2%, 4.2%, 7.3%, 10.3%, 13.3%, 16.4%}, where attack size is the ratio between the number of attackers and genuine users, filler size is defined as the number of items rated by one user divided by the number of all items in the whole system.. In addition, we randomly choose an item from original profiles as the attack target. 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 evaluate the effectiveness of the presented method, we use two evaluation indexes, detection rate (DR) and false alarm rate (FAR). They are defined as follows:

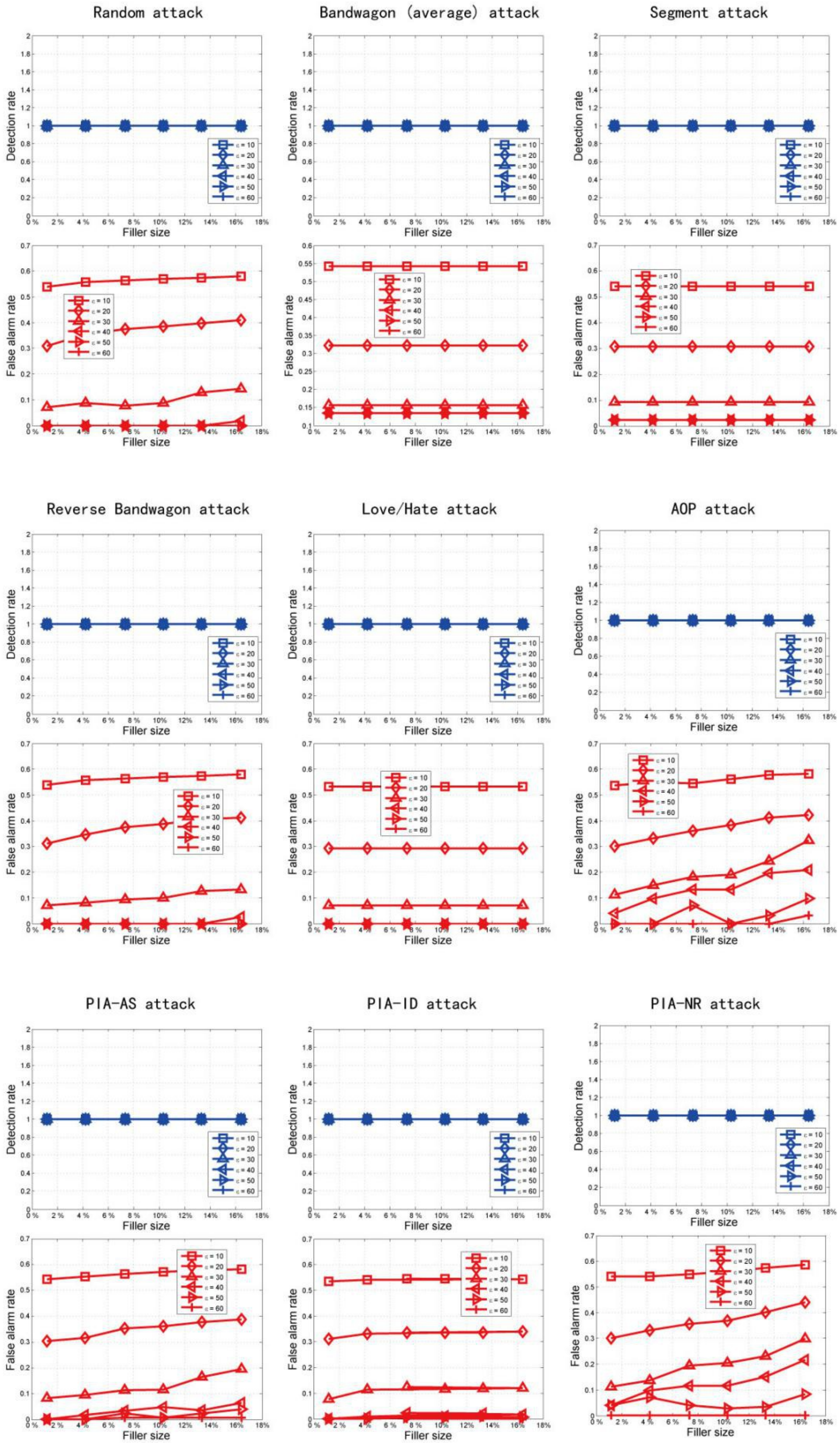

Figure 2: the comparison of detection performance of our method in 9 attacks, where the attack size is 16.4% and the filler size varies.

$$detectoin\ rate = \frac{|D \cap A|}{|A|} \qquad (5.1)$$

$$false\ alarm\ rate = \frac{|D \cap G|}{|G|} \tag{5.2}$$

where D denotes the set of detected user profiles, A denotes the set of attack profiles and G denotes the set of genuine profiles.

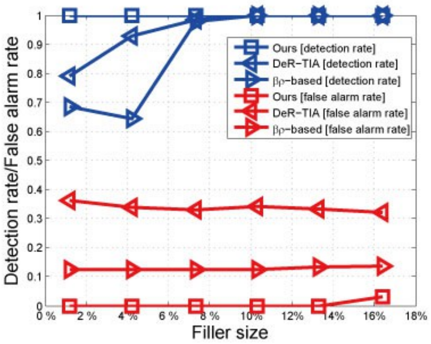

Fig. 3: the comparison of detection rate and false alarm rate in the presented methods, where the attack size is 6.4% and the filler size varies; single-target AOP attack.

5.2. Experimental Results and Analysis

To validate the effectiveness of our method for detecting attacks, we conduct a list of experiments in 9 attack models to show the detection performance. As illustrated in Fig. 2, the detection rates almost keep the highest with the filler size increasing. For the false alarm rates, the curves gradually become close to zero when ε increases from 10 to 60. This may indicate that the count of suspected target items is useful to capture concerned attackers. The false alarm rates are sensitive to the count threshold, although curves of false alarm rates are linear in some attacks. Note that, by exploiting an effective count threshold such as 60, the detection performance of the proposed method is acceptable regardless of different attacks.

Attack size	**Methods**	
	Generating graph+Top-k searching	**PCC**
1.10%	8.498(m)+0.0403s	39.477m
6.40%	9.915(m)+0.0401s	49.426m
11.70%	10.850(m)+0.0403s	61.789m
17.00%	12.552(m)+0.0405s	66.781m
22.30%	14.710(m)+0.0405s	69.492m
27.60%	16.186(m)+0.0399s	58.172m

Table 3: the comparison of time-consumption of alternative approaches with diverse attack sizes. Before "+" denotes the computational cost for the generating graph, after "+" denotes the top-k similarity search, where PCC denotes Pearson Correlation Coefficient and the filler size is 4.2%.

We conduct a list of experiments to compare the detection performance of our method with two alternative methods, DeR-TIA [11] and $\beta\rho-based$ [20]. Fig. 3 shows that the proposed method significantly outperforms the benchmarked one when the attack size is 6.4% and the filler size varies. It is obvious that our method almost keeps the absolute outperformance with the highest DR and near zero FAR. To compare the computational cost of our method with PCC, we conduct a list of experiments on MovieLens-100K dataset in diverse attack sizes (take the AOP attack for example). From Table 3, we can see that PCC cannot compete with our method for all attack sizes within a reasonable time.

6. Conclusion

Shilling attacks are main threats facing CFRSs. In this paper, we propose a new detection method for spotting such attacks or anomalous ratings, which exploits the fast top-k similarity method to calculate the similarity between vertices in a graph. Firstly, we filter out more genuine users by adopting an empirical threshold of the new similarity metric. Based on the remaining users, we continue to filter out more genuine users by using suspected target items as far as possible. It is noteworthy that our proposed method can scale up to handle the large network for detecting anomalous ratings. Experiments demonstrate that our proposed method is superior to benchmarked methods and validate the effectiveness of the proposed method. In the future, we will explore interesting discoveries on the real-world large-scale datasets including Amazon, Yelp etc.

References

[1] K. Bryan, M. OMahony, and P. Cunningham. *Unsupervised retrieval of attack profiles in collaborative recommender systems*[C]. ACM conference on Recommender Systems, ACM, New York, pp. 155-162(2008)

[2] R. Burke, B. Mobasher, and C. Williams. *Classification features for attack detection in collaborative recommender systems*[C]. ACM International Conference on Knowledge Discovery and Data Mining, ACM, New York, pp. 17-20(2006)

[3] B. Mehta, T. Hofmann, and P. Fankhauser. *Lies and propaganda: detecting spam users in collaborative filtering*[C]. In: IUI07: Proceedings of the 12th ACM International Conference on Intelligent User Interfaces, ACM, New York, pp. 14-21(2007)

[4] Z. Zhang and S. R. Kulkarni. *Detection of shilling attacks in recommender systems via spectral clustering*[C]. IEEE International Conference on Information Fusion, IEEE, Salamanca, pp. 1-8(2014)

[5] N. Gnnemann, S. Gnnemann, and C. Faloutsos. *Robust multivariate autoregression for anomaly detection in dynamic product ratings*[C]. Proceeding WWW'14 Proceedings of the 23rd ACM international conference on World Wide Web, ACM, New York, pp. 361-372(2014)

[6] S. Gnnemann, N. Gnnemann, and C. Faloutsos. *Detecting anomalies in dynamic rating data: A robust probabilistic model for rating evolution*[C]. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, pp. 841-850(2014)

[7] J. Zhang, J. Tang, C. Ma, H. Tong, Y. Jing, and J. Li. *Panther: Fast top-k similarity search on large networks*[C]. In Proceedings of the Twenty-First ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’15), ACM, Sydney, pp. 1445-1454(2015)

[8] K Choi and Y Suh. *A new similarity function for selecting neighbors for each target item in collaborative filtering*[J]. Knowledge-Based Systems, Elsevier, 37(1), pp. 146-153(2013)

[9] X. Su, H. Zeng, and Z. Chen. *Finding group shilling in recommendation system*[C]. Proceedings of the 14th ACM international conference on World Wide Web, New York, pp. 960-961(2005)

[10] F. Zhang and Q. Zhou. *HHT-SVM: An online method for detecting profile injection attacks in collaborative recommender systems*[J]. Knowledge-Based Systems, Elsevier, 65(7), pp. 96-105(2014)

[11] W. Zhou, Y. S. Koh, J. H. Wen, S. Burki, and G. Dobbie. *Detection of abnormal profiles on group attacks in recommender systems*[C]. Proceedings of the 37th international ACM SIGIR

conference on Research & development in information retrieval, ACM, New York, pp. 955-958(2014)

[12] C. A. Williams, B. Mobasher, and R. Burke. *Defending recommender systems: detection of profile injection attacks*[J]. Service Oriented Computing and Applications, Springer-Verlag, 1(3), pp. 157-170(2007)

[13] C. A. Williams, B. Mobasher, R. Burke, and R. Bhaumik. *Detecting profile injection attacks in collaborative filtering: a classification-based approach*[J]. Advances in Web Mining and Web Usage Analysis, Springer Berlin Heidelberg, 4811, pp. 167-186(2007)

[14] M. Morid and M. Shajari. *Defending recommender systems by influence analysis*[J]. Information Retrieval, Springer Netherlands, 17(2), pp. 137-152(2014)

[15] B. Mehta. *Unsupervised shilling detection for collaborative filtering*[C]. Association for the Advancement of Artificial Intelligence, AAAI, Vancouver, pp. 1402-1407(2007)

[16] I. Gunes, C. Kaleli, A. Bilge, and H. Polat. *Shilling attacks against recommender systems: A comprehensive survey*[J], Artificial Intelligence Review, 42(4), pp. 1-33(2012)

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

[18] Z. A. Wu, Y. Q. Wang, and J. Cao. *A survey on shilling attack models and detection techniques for recommender systems*[J]. Science China, 59(7), pp. 551-560(2014) (In Chinese)

[19] C. E. Seminario and D. C. Wilson. *Attacking item-based recommender systems with power items*[C]. ACM Conference on Recommender Systems, ACM, New York, pp. 57-64(2014)

[20] C. Chung, P. Hsu, and S. Huang. *A novel approach to filter out malicious rating profiles from recommender systems*[J]. Decision Support Systems, 55(1), pp. 314-325(2013)