Abnormal Group User Detection in Recommender Systems Using Multi-dimension Time Series

Wei Zhou(✉), Junhao Wen, Qingyu Xiong, Jun Zeng, Ling Liu, Haini Cai, and Tian Chen

College of Software Engineering, Chongqing University,
174 Shazheng Street, Chongqing, China
{zhouwei,wjhcqu,xiong03,zengjun,liuling,hainic,chengtian}@cqu.edu.cn

Abstract. Collaborative filtering based recommender systems are capable of generating personalized recommendations, which are tools to alleviate information overload problem. However, due to the open nature of recommender systems, they are vulnerable to shilling attacks which insert forged user profiles to alter the recommendation list of targeted items. Previous research related to robustness of recommender systems has focused on detecting malicious profiles. Most approaches focus on profile classification but ignore the group attributes among shilling attack profiles. A method for detecting suspicious ratings by constructing multi-dimension time series *TS-TIA* is proposed. We reorganize all ratings on each item sorted by time series, each time series is examined and suspicious rating segments are checked. Then statistical metrics and target item analysis techniques are used to detect shilling attacks in these anomaly rating segments. Experiments show that our proposed method can be effective and less time consuming at detecting items under attacks in greater datasets.

Keywords: Abnormal group users · Shilling attack detection · Time series · Recommender system

1 Introduction

Information overload [1] is a problem people have to face in modern society. Collaborative Filtering based recommender systems play an increasing role in information filtering, which is an important tool to alleviate this contradiction [2]. In recent years, recommender systems have become extremely popular and are applied in a variety of areas. Recommender systems have been developed to recommend movies, music, news, books, research articles, social tags, and other products.

However, recent research has shown that traditional collaborative filtering based techniques are vulnerable to attacks [3–5]. Recent work has shown that

S. Wang and A. Zhou (Eds.): CollaborateCom 2016, LNICST 201, pp. 373–383, 2017.
DOI: 10.1007/978-3-319-59288-6_34

even modest attacks are sufficient to manipulate the behavior of the most commonly used recommendation algorithm [6]. Attackers change the recommendation list by introducing biased profiles into the rating matrix, which is called shilling attack. Attacks against recommender systems can affect the quality of predictions, resulting in crisis of confidence.

In order to preserve order and fairness of recommender systems, attack profiles should be detected and removed. In this paper, we propose a novel technique based on statistical metrics and rating time stamps. The main contribution of this technique is that we divide the rating matrix into rating segments (windows) and find suspicious rating segments. We examine the suspected rating segments instead of the whole rating matrix, which reduces the algorithm complexity and time consuming. The paper is organized as follows. In next section we look at previous work in the area. In Sect. 3 we discuss preliminary knowledge that used in this paper, including detecting metrics and how to construct time series. Section 4 describes the detailed metrics used in detecting attacks and algorithms and experimental methodology for our detection model. In Sect. 5 we present our experimental results and a conclusion in Sect. 6.

2 Related Work

Attack profiles that are introduced into recommender systems in order to alter recommendation lists of a set of target items. There are two types of main attacks according to the intent of attackers. A push attack is an attack that aims to promote an item and boost its ranking, whereas a nuke attack is an attack designed to demote an item and lower its rankings.

The word "shilling" was first termed in [7]. There have been some recent research efforts aiming at detecting and reducing the effects of profile injection attacks [8,9]. These attacks consist of a set of attack profiles, each containing biased rating data associated with a fictitious user identity. Since "shilling" profiles look similar to authentic profiles, which is difficult to identify. Many attack profiles are based on random and average attack models which were introduced originally in Lam and Reidl [10]. There are three categories of attack detection algorithms: supervised, unsupervised, and semi-supervised.

In the first category, attack detection techniques are modelled as a classification problem. Most early detection algorithms [11] exploited signatures of attack profiles. These techniques were considered less accurate, since they looked at individual users and ignored the combined effect of such malicious users. Moreover, these algorithms do not perform well when the attack profiles are obscured. Some of these techniques use nearest neighbours classifiers, decision tree methods, rule based classifiers, Bayes classifiers, Neural Networks classifiers, or *SVM* based classifiers [12,13].

In the second category, unsupervised detection approaches address these issues by training on an unlabeled dataset. The benefit of this is that these techniques facilitate online learning and improve detection accuracy. Some of

the techniques use clustering, association rules methods [14] and other statistical approaches [15,16]. Zhang et al. [17] used a Singular Value Decomposition (*SVD*) method to learn a low-dimensional linear model. Wei et al. [18,19] proposed a novel 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*) using statistical strategy. They also extend a detailed analysis of target item rating patterns. The proposed methods improve the detection accuracy of detection using target item analysis method.

In the third category, semi-supervised detection approaches make use of both unlabelled and labelled user profiles for multi-class modelling. Wu et al. [20] proposed a system called *HySAD* for hybrid attack detection. In general, *HySAD* is a semi-supervised learning system that makes use of both unlabeled and labeled user profiles for multi-class modeling.

Time series have been used to detect shilling attacks in recommender systems. The intuition of the idea is that all attack models cause changes in the rating distributions of target items. [17] postulates that the distribution of item ratings in time can reveal the presence of a wide range of shilling attacks given reasonable assumptions about their duration. In this paper, we borrow this idea, all ratings on each item are sorted by time series, each time series is examined and suspected rating segments are checked. Then techniques of our previous study are used to detect shilling attacks in these anomaly rating segments using statistical metrics and target item analysis.

3 Preliminary Knowledge

In this section, some preliminary knowledge are introduced. Section 3.1 introduces the detecting structure of the proposed method. Section 3.2 introduces how a multi-time series data stream are constructed. Section 3.3 reviews two different detecting metrics in collaborative filtering algorithms.

3.1 Structure of Detecting

There are millions of profiles and items in real time recommender systems; it is time consuming to carry out detecting on the whole system. Group attributes and time clustering characteristics are found exist in shilling attacks. Most of profiles and ratings are low suspected of being attackers. We intend to divide the whole dataset into a group of subsections, and then suspected subsections are checked. In the end, techniques we proposed in our previous studies are used focusing on rating matrix that composed of suspected subsections. The scope of the target dataset will be reduced greatly, which would be time saving and efficiency.

There are two phases in the detecting structure. In the first phase, we find the suspected ratings on an item and then find suspicious rating segments that rated on the item during the specific time. Suspected rating segments are determine by

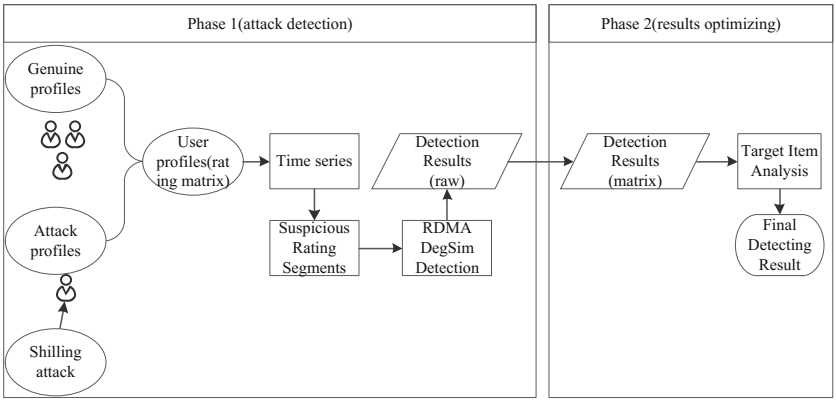

Fig. 1. Detecting structure of the proposed method *TS-TIA*

constructing a time series. The number of profiles are reduced by a wide margin in this phase. Then we detect shilling attacks in these anomaly rating segments using statistical metrics and target item analysis method. In the second phase, a fine-tuning phase whereby the target items in the potential attack profiles set are analyzed. Figure 1 shows the detecting process of the proposed method *TS-TIA*.

3.2 Constructing Time Series

To construct the time series of the above measure for an item, we first sort all the ratings for the item by their time stamps into a data stream, and then group every disjoint w consecutive ratings into a window. w is referred as the window size. In this paper, window size is set to 20. According to the attribute of characteristic of abnormal groups, multi-dimension time series are chose, including number of rating in unit time, average rating and review frequency. For example, in a online store, sort rating on an item according to the time stamps in ascending order. We can get:

$$R(s) = \{r_1, r_2, r_3, \cdots, r_i, \cdots, r_j, \cdots, r_{n_s}\} \tag{1}$$

$$TS(s) = \left\{t_{s_1}, t_{s_2}, t_{s_3}, \cdots, t_{s_i}, \cdots, t_{s_j}, \cdots, t_{s_n}\right\} \tag{2}$$

While $R(s)$ is the rating series of an item by all users that rate on the item, and $TS(s)$ is the time series of ratings on the item. For all ratings, $1 \leq i \leq j \leq n$, $t_{s_i} \leq t_{s_j}$. For example, t_{s_i} is the time stamp of rating r_i.

A time window Δt is used to divide the rating time interval $I = [t_0, t_0 + T]$ into $n = T/\Delta t$ time windows, while the length of every time window is Δt, t_0 is the starting time stamp. For the ith time window I_i, $I_i = [t_0 + (i-1)\Delta t, t_0 + it_0]$, while $I = \bigcup_{i=1}^{n} I_i$. For every time window in the time series I_i, the value of characteristic is calculated. So, in a rating dataset, given

the time interval $[t_0, t_0 + T]$ and the time window Δt, a time series can be achieved:

$$F_s(I, \Delta t) = \begin{bmatrix} I_1(1), I_1(2), \cdots, I_1(i), \cdots, I_1(j), \cdots I_1(n) \\ I_2(1), I_2(2), \cdots, I_1(i), \cdots, I_1(j), \cdots I_2(n) \\ I_3(1), I_3(2), \cdots, I_1(i), \cdots, I_1(j), \cdots I_3(n) \end{bmatrix} \tag{3}$$

Abnormal group user discovery based on multi-dimension time series can disclose the group attribute of abnormal users. If multi-dimension characteristics are abnormal in the time series, then all the ratings in the time interval is recognized as abnormal. For example, time interval $[t_0, t_0 + T]$ is abnormal in Fig. 2. A set of suspicious ratings can be get in this step. It is necessary to filter normal ratings from the suspicious ratings set. Figure 2 is the schematic diagram that shows how suspicious rating segments (windows) are located.

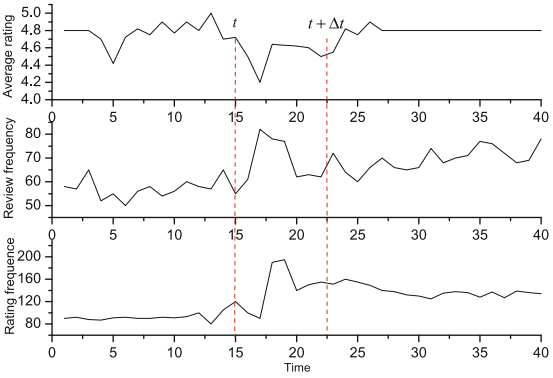

Fig. 2. Abnormal group users discovery based on multi-time series

3.3 Detecting Metrics

Attack profiles differ from that of genuine profiles in a statistical way. There are two main differences: the rating given to the target item (items); the rating distribution among the filler items. Due to this there are different metrics that have been proposed to measure the difference between rating profiles. In this section we will look at two metrics, *RDMA* and *DegSim* [6]. RDMA value of attack profiles is higher than that of genuine profiles; while DegSim value of attack profiles is lower than that of genuine profiles.

Rating Deviation from Mean Agreement. *RDMA* measures the deviation of agreement from other users on a set of target items, combined with the inverse rating frequency for these items. *RDMA* can be calculated in the following way:

$$RDMA_u = \frac{\sum_{i=0}^{N_u} \frac{|r_{u,i} - \overline{r_i}|}{NR_i}}{N_u} \tag{4}$$

where N_u is the number of items user u rated, $r_{u,i}$ is the rating given by user u to item i, NR_i is the overall number of ratings in the system given to item i.

Degree of Similarity with Top Neighbours. The *DegSim* attribute is based on the average Pearson correlation of the profile's k nearest neighbours and is calculated as follows:

$$DegSim = \frac{\sum_{u=1}^{k} W_{uv}}{k} \tag{5}$$

where W_{uv} is the Pearson correlation between user u and user v, and k is the number of neighbours.

In paper [18], we proposed a novel 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*). We also extended our work with a detailed analysis of target item rating patterns. Experiments show that the combined methods can improve detection rates when the dataset is in a small scale. However, the efficiency becomes lower when the datasets increase. In the next section, time series will be constructed and suspected rating segments are checked.

4 Detecting Profile Injection Attacks

In the first phase, a suspicious rating segment is get by constructing a time series. The scope of attack profiles are greatly narrowed, which saves time and reduces the computing complexity. However, genuine profiles and attack profiles are mixed together. We use statistical metrics and target item analysis techniques in our previous research to filter out genuine profiles.

Overall attackers should have a high influence in the system in order to promote the target items effectively. However, there are three different features in attack profiles, which enable us to differentiate between genuine and attack profiles. Firstly, filler items are randomly chosen thus the similarity based on these filler items between attack and genuine profiles should be lower. Secondly, since shilling attacks usually try to push items with low ratings or vice versa in nuke attacks, the users mounting such an attack will assign a rating that deviates from the average rating value assigned by the genuine profiles. Last but not least, all target items are assigned a highest or lowest value, the count number of this value should be greater than other values among items. Based of these three reasons, we choose the *RDMA* and *DegSim* metrics, which reveal these distinctive features in the rating patterns. Attackers should therefore have relatively high values for *RDMA*, as well as very low values in *Degsim*. The pseudocode of the method is shown in Algorithm 1.

In this phase, an *RDMA* value for each profile is calculated. If the *RDMA* value for a profile is above a maximum ϵ_{RDMA} threshold then we consider this profile as a suspicious profile.

$$RDMA_u = \frac{\sum_{i=0}^{N_u} \frac{|r_{u,i} - \overline{r_i}|}{NR_i}}{N_u} \geq \epsilon_{RDMA}$$

From this process, we get a pool of suspicious profiles, SP_{RDMA}, that had *RDMA* values above the assigned threshold. We also calculate the *DegSim* value for each

Algorithm 1. Applying target item analysis method on abnormal rating segments.

Input: The set of suspected profiles $SUSSEG$;item set I;
Output: Final detect result set $DetectedResult$;
1: $SUSSEG = SUSSEG$ - $RDMA_u \geq \epsilon_{RDMA} \cap DegSim_u \leq \epsilon_{DegSim}$
2: $DetectedResult = \emptyset$;
3: $\forall i \in I, count_i \leftarrow$ number of ratings in $item_i$ equal to r;
4: **while** $\max(count) > \theta$ **do**
5: $item_t \leftarrow \{item_i \,|count_i = \max(count)\}$;
6: $\forall p \in SUS_{RD}, P \leftarrow p$ rate $item_t$ with r;
7: $DetectedResult \leftarrow P \cup DetectedResult$;
8: $SUS_{RD} \leftarrow SUS_{RD} - P$;
9: **end while**
10: **return** $DetectedResult$.

of the profiles. If the *DegSim* value for a profile u is below a minimum ϵ_{DegSim} threshold then we consider this profile as a suspicious profile.

$$DegSim = \frac{\sum_{u=1}^{k} W_{uv}}{k} \leq \epsilon_{DegSim}$$

From this process, we get a pool of suspicious profiles, SP_{DegSim}, that had *DegSim* values below the assigned threshold. Lastly we consider the intersection between the pool of SP_{RDMA} and SP_{DegSim}, as our *SuspectedAttackers*.

$$SuspectedAttackers = SP_{DegSim} \cap SP_{RDMA}$$

We set generous thresholds ϵ_{RDMA} and ϵ_{DegSim}, allowing more profiles to be considered as suspicious. We then filter out the misclassified profiles in the second phase.

For example, Table 1 is an example of a rating matrix and attack profiles. The matrix is an $m \times n$ matrix. Each row in the matrix is the rating for the m items by a user. Table 1 shows genuine user profiles from $User_1$ to $User_m$ and attackers profiles from $Attacker_1$ to $Attacker_p$. The last row is the count number of rating 5, in this example, $Item_5$ is the target item.

5 Experiments and Results

5.1 Experiment Setup

The datasets used in the experiments are the widely used *MovieLens* Datasets, Including MovileLens 100k Dataset, 1 Million and 10 Million Dataset by the GroupLens Research Project at the University of Minnesota and a subset of Netflix dataset. The platform we implement all the experiments as flows: Hardware: CPU is Intel Core i7 processors, Windows 7, with 16 G RAM. Software: All of our tests is on Matlab 2012b.

Table 1. An example of rating matrix and attack profiles.

	Item_1	Item_2	Item_3	Item_4	Item_5	....	Item_n
User_1	5	2	3	0	0	....	5
User_2	2	0	4	1	2	....	3
User_3	4	2	3	0	5	....	0
User_4	0	3	0	3	4	....	3
....	....	....	....	....	....	....	....
User_m	2	0	4	1	2	....	3
Attacker_1	2	1	0	0	5	....	4
Attacker_2	2	2	0	0	5	....	3
Attacker_3	1	2	0	0	5	....	2
....	....	....	....	....	....	....	....
Attacker_p	2	0	0	0	5	....	4
Count(5)	2	2	2	2	9	....	3

5.2 Experiment Results

In order to simulate real attacks in recommender systems, we injected attack profiles generated by certain attack models. In the experiments we varied two different variables: the attack size and the filler size. Because the median filler size of all profiles is 3%, we did not consider situations where the filler size is greater than 10%. In order to get certain prediction shift, the minimum number of attack profiles are set to 20. The attack size are varied from 20 to 200 and the filler size are varied from 3% to 9% (Table 3).

Experiments Result and Comparisons. Table 2 shows the attack detection results when attack size and filler size varies. The detection rate increase while the false positive rate decrease with the increase of attack size and filler size.

Table 2. Detection results of *TS-TIA* when the filler size and attack size varies

Attack size	Filler size	20	40	60	80	100	120	140	160	180	200
Detection rate	3%	65%	74%	78%	85%	86%	88%	91%	93%	95%	98%
	5%	68%	76%	80%	86%	87%	92%	93%	94%	95%	98%
	7%	78%	80%	86%	89%	91%	93%	93%	95%	95%	98%
	9%	82%	84%	87%	90%	92%	93%	94%	96%	97%	99%
False positive rate	3%	5%	5%	4%	4%	3%	3%	2%	2%	3%	2%
	5%	4%	4%	4%	4%	3%	3%	3%	2%	2%	2%
	7%	4%	4%	3%	4%	3%	5%	2%	1%	2%	2%
	9%	4%	4%	3%	4%	2%	2%	2%	1%	2%	1%

Table 3. Comparison of detection results of different methods when filler size is 7% and attack size varies

Attack size	Method	20	40	60	80	100	120	140	160	180	200
Detection rate	DeR-TIA	78%	80%	90%	98%	100%	100%	100%	100%	100%	100%
	RD-TIA	92%	95%	100%	100%	100%	100%	100%	100%	100%	100%
	TS-TIA	60%	82%	86%	89%	91%	93%	93%	95%	95%	98%
	bp-based	70%	60%	78%	85%	90%	96%	100%	100%	100%	100%
False positive rate	DeR-TIA	3%	3%	2%	0%	0%	0%	0%	0%	0%	0%
	RD-TIA	2%	2%	2%	0%	0%	0%	0%	0%	0%	0%
	TS-TIA	4%	4%	3%	4%	3%	5%	2%	1%	2%	2%
	bp-based	12%	11%	11%	12%	11%	12%	12%	12%	12%	12%

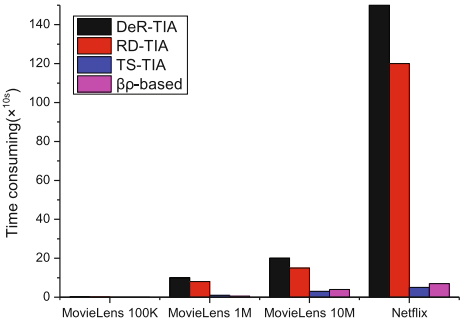

Fig. 3. Time consuming of different detecting methods when the datasets are different

The detection rate reaches 90% or more when the attack size greater than 140. The false positive rate is less than 10%, and became stable around 3% when the attack size is greater than 100. The false positive rate of algorithms using target item analysis method is lower than that of $\beta\rho$-based method.

Table 2 shows comparison of detection rate of different methods. The detection rate increase while the false positive rate decrease with the increase of attack size. The detection rate of the proposed algorithm is lower than that of other detecting algorithms. The false positive rate of the proposed algorithm is higher than that of *DeR-TIA* and *RD-TIA*, which all of them using target item analysis method, but lower than that of $\beta\rho$-based algorithm. The false positive rate is less than 5%, and became stable around 2% when the attack size is greater than 100.

Figure 3 shows time consuming of different algorithms when attack size varies. All Algorithms consume more time when detecting greater datasets. *DeR-TIA* consumes the most time in all detections and *RD-TIA* gets the second most time consuming, which is intolerable in greater datasets. *TS-TIA* consumes the least time. $\beta\rho$-based method consumes more time than *TS-TIA* but less than other two target item analysis based methods. Generally speaking, even the detection

rate of *TS-TIA* is not better than that of $\beta\rho$-based method, but consumes less time and the false positive rete is lower.

6 Conclusion

Collaborative filtering based recommender systems suffer from shilling attacks. In most cases, attackers inject masses of forged profiles in order to get a substantial prediction shift. When a large number of forged profiles are injected into the rating matrix in a short period of time, the group feature would stand out. In this paper, a data stream by sorting ratings of an item is constructed, time series are built and suspicious profiles are filtered, then two statistical metrics are used to detect forged profiles; last but not least, target item analysis method reduce the false positive rate of the final detecting result. Experiments showed that the proposed method gets lower precision in great datasets, but occupies less computing capacity.

Acknowledgement. This research is supported by NSFC under grant No. 61602070, 61502062, 61379158 and China Postdoctoral Science Foundation under Grant No. 2014M560704.

References

1. Ma, Y., Wang, S., Yang, F., Chang, R.N.: Predicting QoS values via multi-dimensional QoS data for web service recommendations. In: 2015 IEEE International Conference on Web Services (ICWS), pp. 249–256. IEEE (2015)
2. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. (TOIS) **22**(1), 5–53 (2004)
3. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295. ACM (2001)
4. Xia, H., Fang, B., Gao, M., Ma, H., Tang, Y., Wen, J.: A novel item anomaly detection approach against shilling attacks in collaborative recommendation systems using the dynamic time interval segmentation technique. Inf. Sci. **306**, 150–165 (2015)
5. Cao, J., Wu, Z., Mao, B., Zhang, Y.: Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system. World Wide Web **16**(5–6), 729–748 (2013)
6. Chirita, P.-A., Nejdl, W., Zamfir, C.: Preventing shilling attacks in online recommender systems. In: Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management, pp. 67–74. ACM (2005)
7. Lam, S.K., Riedl, J.: Shilling recommender systems for fun and profit. In: Proceedings of the 13th International Conference on World Wide Web, pp. 393–402. ACM (2004)
8. Zhang, S., Ouyang, Y., Ford, J., Makedon, F.: Analysis of a low-dimensional linear model under recommendation attacks. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 517–524. ACM (2006)

9. O'Mahony, M.P., Hurley, N.J., Silvestre, G.: Detecting noise in recommender system databases. In: Proceedings of the 11th International Conference on Intelligent User Interfaces, pp. 109–115. ACM (2006)
10. Fu, L., Goh, D.H.-L., Foo, S.S.-B., Na, J.-C.: Collaborative querying through a hybrid query clustering approach. In: Sembok, T.M.T., Zaman, H.B., Chen, H., Urs, S.R., Myaeng, S.-H. (eds.) ICADL 2003. LNCS, vol. 2911, pp. 111–122. Springer, Heidelberg (2003). doi:10.1007/978-3-540-24594-0_10
11. O'Mahony, M.P., Hurley, N.J., Silvestre, G.C.M.: Promoting recommendations: An attack on collaborative filtering. In: Hameurlain, A., Cicchetti, R., Traunmüller, R. (eds.) DEXA 2002. LNCS, vol. 2453, pp. 494–503. Springer, Heidelberg (2002). doi:10.1007/3-540-46146-9_49
12. Grčar, M., Fortuna, B., Mladenič, D., Grobelnik, M.: KNN versus SVM in the collaborative filtering framework. In: Batagelj, V., Bock, H.H., Ferligoj, A., Žiberna, A. (eds.) Data Science and Classification. Studies in Classification, Data Analysis, and Knowledge Organization, pp. 251–260. Springer, Heidelberg (2006)
13. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. 2009, 4 (2009)
14. Lee, C.-H., Kim, Y.-H., Rhee, P.-K.: Web personalization expert with combining collaborative filtering and association rule mining technique. Expert Syst. Appl. **21**(3), 131–137 (2001)
15. Hurley, N., Cheng, Z., Zhang, M.: Statistical attack detection. In: Proceedings of the Third ACM Conference on Recommender Systems, pp. 149–156. ACM (2009)
16. Wang, S., Ma, Y., Cheng, B., Chang, R., et al.: Multi-dimensional QoS prediction for service recommendations (2017)
17. Zhang, S., Chakrabarti, A., Ford, J., Makedon, J.: Attack detection in time series for recommender systems. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 809–814. ACM (2006)
18. Zhou, W., Wen, J., Koh, Y.S., Alam, S., Dobbie, G.: Attack detection in recommender systems based on target item analysis. In: 2014 International Joint Conference on Neural Networks (IJCNN), pp. 332–339. IEEE (2014)
19. Zhou, W., Koh, Y.S., Wen, J., Alam, S., Dobbie, G.: Detection of abnormal profiles on group attacks in recommender systems. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 955–958. ACM (2014)
20. Wu, Z., Wu, J., Cao, J., Tao, D.: 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, pp. 985–993. ACM (2012)