Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 17(2): 1558–1577.
DOI: 10.3934/mbe.2020081
Received: 12 September 2019
Accepted: 01 December 2019
Published: 04 December 2019

Research article

A novel Kalman Filter based shilling attack detection algorithm

Xin Liu, Yingyuan Xiao*, Xu Jiao, Wenguang Zheng and Zihao Ling

Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, Tianjin 300384, China; Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology, Tianjin 300384, China

* **Correspondence:** E-mail: yyxiao@tjut.edu.cn.

Abstract: Collaborative filtering has been widely used in recommendation systems to recommend items that users might like. However, collaborative filtering based recommendation systems are vulnerable to shilling attacks. Malicious users tend to increase or decrease the recommended frequency of target items by injecting fake profiles. In this paper, we propose a Kalman filter-based attack detection model, which statistically analyzes the difference between the actual rating and the predicted rating calculated by this model to find the potential abnormal time period. The Kalman Filter filters out suspicious ratings based on the abnormal time period and identifies suspicious users based on the source of these ratings. The experimental results show that our method performs much better detection performance for the shilling attack than the traditional methods.

Keywords: collaborative filtering; recommendation system; shilling attack; attack detection; Kalman Filter

1. Introduction

With the advent of the Big Data Age, the recommendation system has become an important tool for users to choose potential projects of interest. The recommendation system can make recommendations for users based on the implicit connection between users and items. However, this system is relatively vulnerable to malicious users. The attacker deliberately inserts attack profiles into the recommender system to bias the predicted rating of a specific item. As a result, this item can be recommended to the influenced users more or less frequently. For example, if a user likes to watch comedy films and is not likely to watch horror movies, there should be no horror movies in his recommendation list. After the attack, his recommendation list showed horror movies that he did not really like. This user is called the influenced user, and this kind of attack is called “profile injection attack” or “shilling attack”.

In order to prevent shilling attacks, researchers tried to use some algorithms to detect attackers. The main challenge is how to detect attack profiles accurately and efficiently. Existing detection methods mainly use some features and models to separate suspicious users from normal users. Specifically, some methods only consider the user’s rating of the item when finding attacker, and others consider more factors, such as time and user group. Although effective, the existing methods still have some limitations. Most of the detection algorithms are from the user's point of view, rarely from the perspective of the item, and the recall is poor.

According to the issues mentioned above, we explore a novel model for attack detection. We assume that an anomaly condition will occur when an attack occurs. In the case of sufficient data, genuine user's rating behavior is basically stable. However, the attack profile injected by an attacker is usually done centrally in a certain period of time. Based on the hypothesis, we try to seek a time-related model to detect unusual behavior and determine the attack profiles.

In this paper, we propose a novel Kalman Filter model for detecting shilling attacks. In order to achieve the effect of detection, the main challenge is finding the attack profiles. According to the above analysis of the attack behavior, we use the Kalman Filter's predicted value to compare with the actual value to calculate the deviation value. We define two kinds of deviation values and then determine the suspicious time period according to the abnormal condition of the deviation. Then the attack profiles are found from the suspicious time period. We compared the proposed Kalman Filter with some existing methods. The experimental results show that our method performs much better detection performance for the shilling attack than the traditional methods.

The rest of paper is organized as following. In Section 2, we briefly introduce the related work. In Section 3, we introduce some preliminaries, including attack profiles, models, etc. In Section 4, we describe our detection method in detail. In Section 5, the results of the experiment are reported and analyzed. In the last section, we make a summary of the paper and prospect in the direction of future work.

2. Related work

Collaborative filtering is a popular recommendation algorithm which is very susceptible to shilling attacks. The attacker falsifies the user profile and makes the fake users become close neighbors of genuine users as more as possible. Since collaborative filtering is recommended based on the interests of neighbors, the attacker can influence the system's recommendation results and increase or decrease the recommended frequency of the target object. There are already many researches on the detection methods of the shilling attacks in CFRS. Chirita et al. [1] proposed the RDMA attribute, which is the earliest defined attribute used to characterize the difference in user’s rating vector. Then develop an algorithm to calculate the probability of a user becoming a shilling attacker using RDMA and average similarity metrics. Burke et al. [2] present a number of detection features which are extracted from user profiles to classify users. They show that classifiers built using these features can detect attacks well and improve the stability of a recommender. Then, they utilize KNN model to complete the classification. Williams et al. [3] also extracted some features from user profiles. But they suffered misclassification and low precision. In addition, they tried the time interval detection scheme and present an empirical evaluation of the anomaly detection methods which can be quite successful in identifying items under attack and time periods in which attacks take place. But its robustness needs to be improved. Mehta et al. [4] provided an in-depth analysis of shilling profiles and put forward two unsupervised algorithms based on PLSA soft clustering and PSA variable selection. The algorithm has better detection performance, but it needs to know the size of the attack in advance, and the utility is not high. Peng et al. [5] proposed the "interest kurtosis coefficient" according to the concentration of interest of users, and used unsupervised detection methods to select effective detection indicators for different types of attack attacks, and then designed an unsupervised detection algorithm based on feature subsets. Zhang et al. [6] proposed a novel graph-based unsupervised learning detection algorithm, which transforms the problem of detecting attacks into the problem of searching for the largest subgraph. It is difficult for the algorithm to choose a suitable threshold and the performance is not good when the fill size is small. Wu et al. [7] proposed a semi-supervised detection algorithm, which combines the naive Bayesian classifiers and augmented expectation-maximization base on several selected metrics. The algorithm is time-consuming, low recall, and narrow applicable breadth. Lv et al. [8] proposed a detection method based on SVM and KNN algorithm, which can achieve good detection results when the mark data is small, but the accuracy and recall are relatively low when the fill size is small. Gao et al. [9] analyzed two common features and defined four types of items, then proposed a time intervals detection approach. They have achieved good results on four types of items, but the practicality is low. Yang et al. [10] find a mapping model between rating behavior and item distribution. They use 8 item attributes and 12 rating attributes to construct a corresponding relationship between ratings and items. Then use these presented features to construct a mapping model for detecting shilling attacks. The detection results demonstrate the outperformance of the proposed method, but cannot reach a full level when the attack size is small. Bhebe et al. [11] propose a combiner strategy that combines multiple classifiers in an effort to detect shilling attacks. The proposed Meta-Learning classifier has a nice detection performance but also has a low recall at low attack sizes. Zhou et al. [12] propose a detection algorithm based on SVM and target item analysis method. The method has high precision but low recall. Conclusions that can be drawn from existing research, including the attacker usually rated the target project as the highest or lowest rating, and the injected attack files are usually injected in a short period of time. So, the average rating of the item will produce a significant deviation in a short time. If this abnormal deviation can be detected, then the time of the injection attack can be determined to determine the attack profile.

3. Preliminaries

3.1. Attack profile and attack model

Attackers with different attack intentions will adopt different attack methods. Shilling attacks can be classified into two types: push attack and nuke attack [13]. In the push attack, the attacker will give target items the highest rating. Conversely, in the nuke attack, the attacker will give target items the lowest rating. The attack profile in a general attack model is shown in Table 1 [14].

In an attack profile, I^S represents the set of selected items. It is determined by function δ. For some types of attacks, the selected item is not used. I^F represents a collection of filler items which are randomly selected by the attacker is determined by function σ. $I^\emptyset$ represents the unrated items which are not rated by the attackers. In addition, i_t is the target item that is promoted or demoted.

Each attack profile has a target item. i_t is determined by function γ.

Table 1. The General Attack Profile.

I^S			I^F			$I^\emptyset$			
i_1^S	…	i_k^S	i_1^F	…	i_k^F	$i_1^\emptyset$	…	$i_k^\emptyset$	i_t
$\delta(i_1^S)$		$\delta(i_k^S)$	$\sigma(i_1^F)$		$\sigma(i_k^F)$	null		null	$\gamma(i_t)$

In a single attack, the target item i_t usually rated the highest rating (in the push attack) or the lowest rating (in the nuke attack). The choice of I^S and I^F is determined by the different types of attack models. There are three most commonly used models: random attack, average attack, and bandwagon attack [14].

- In random attack, the filler items I^F are randomly chosen from the non-target items, and the rating value of each item is subject to a normal distribution with a mean rating value of the entire data set. The selected item is not used in this attack. The target item i_t is rated for r_{max}.
- The average attack is similar to random attack. The difference is that the distribution of the filler item ratings is determined by the normal distribution of mean values of the ratings of each item i.
- In bandwagon attack, the popular items with the highest number of ratings are chosen as the selected items and rated at the highest rating value. Filler items are randomly selected in non-target items and are rated near the average of each item's rating as the average attack. The target items i_t for all of these attacks are rated for the highest or lowest rating (depending on whether it is a push attack or a nuke attack).

3.2. Kalman Filter

The Kalman Filter algorithm estimates the value of the unknown variable for the next time period by analyzing the observed data over time. It can predict the next state based on the previous state (Assume that the input measured value and measurement noise are both normally distributed). Kalman filter can be represented by the following two equations.

$$x_t = F_t x_{\mathrm{t}-1} + w_t \tag{1}$$

$$z_t = H_t x_{t-1} + v_t \tag{2}$$

Equation (1) is called the state equation and indicates the internal state of this system. x_t represents the state vector of time t with a noise w_t, and w_t is a normal distribution, $w_t \sim N(0, Q_t)$. Q_t is a covariance matrix. x_{t-1}, the state vector of time $(t-1)$, is the vector of the previous state of x_t and F_t represents a matrix of time conversion of the system. Equation (2) is called observation equation. It can output the observation value z_t according to the system state. z_t is an observation vector with noise v_t, which is an observation noise vector that conforms to the normal distribution $v_t \sim N(0, R_t)$. R_t is a covariance matrix. Then, H_t is the mapping from the state vector to the observation vector.

Kalman Filter has two steps: the prediction step and the update step. In the prediction step, it estimates the state at time t according to the state at time $(t-1)$. Then in the update step, it estimates the state of a more accurate t-time state based on the observed value at time t. The two steps are as follow.

Prediction step:

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1} \tag{3}$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \tag{4}$$

Update step:

$$Kg_t = \frac{P_{t|t-1} H_t^T}{H_t P_{t|t-1} H_t^T + R_t} \tag{5}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + Kg_t (z_t - H_t \hat{x}_{t|t-1}) \tag{6}$$

$$P_{t|t} = (I - Kg_t H_t) P_{t|t-1} \tag{7}$$

Kg_t represents Kalman gain. $\hat{x}_{t-1|t-1}$ and $P_{t|t-1}$ are the value of them at time $(t-1)$. $\hat{x}_{t|t-1}$ and $P_{t|t-1}$ are the predicted value acquired from the prediction step. Then, after the correction of the update step, the correction values $\hat{x}_{t|t}$ and $P_{t|t}$ are obtained.

There are already precedents for applying the Kalman Filter model to find bias in user ratings system from Amazon.com in [15]. This technique is computationally feasible and can update the estimation of bias with every new review without having to store all the past ratings information.

An example of Kalman filtering in the recommendation system was proposed in [16]. Model was used to predict user preference in consideration of user preference changes and generate the recommendation list. In their model, F_t and H_t are unit vector I. Noise is a standard normal distribution. The two-step formula is updated to:

$$\hat{x}_{n_{t|t-1}} = \hat{x}_{n_{t-1|t-1}} \tag{8}$$

$$P_{t|t-1} = P_{t-1|t-1} + I \tag{9}$$

$$Kg_t = \frac{P_{t|t-1}}{P_{t|t-1} + I} \tag{10}$$

$$\hat{x}_{n_{t|t}} = \hat{x}_{n_{t|t-1}} + Kg_t(z_t - \hat{x}_{n_{t|t-1}}) \tag{11}$$

$$P_{t|t} = (I - Kg_t)P_{t|t-1} \tag{12}$$

We improved on this model and further transformed this model with the characteristics of the shilling attack.

3.3. Confidence interval estimate

Confidence interval estimation is based on the central limit theorem which shows that the distribution of the sample mean becomes more standardized as the sample size increases. So, if we have a large enough sample, we can use a normal distribution to describe the sample mean from any group, even a non-normal population [17]. The confidence interval or interval estimation is based on the point estimate and the sampling standard error, and the interval containing the parameter to be estimated is established according to the given probability value. The given probability value is called the confidence or confidence level. For example, 95% confidence interval indicates that 95% of the data in a given data will fall within this interval. The two values delineating the confidence interval are called the lower confidence limit and the upper confidence limit.

Suppose we have some ratings $r_1, r_2, \dots, r_{\mathrm{k}}$. The purpose of interval estimation is to find two statistics y_1, y_2, so that (y_1, y_2) can cover these samples as much as possible. y_1 and y_2 can be calculated by the following formulas.

$$y_1 = \bar{r} - \sigma Z_\alpha \tag{13}$$

$$y_2 = \bar{r} + \sigma Z_\alpha \tag{14}$$

where $\bar{r}$ and σ is the mean and standard deviation of these ratings. Z_α is the z-value at the confidence level $(1-\alpha)$. For example, when the confidence level is 95%, the value of Z is 1.96.

3.4. Item classification

In order to target the characteristics of different items, in [9], they are classified into four categories according to the z-score of the items and the average number of ratings, fad item, fashion item, style item and scallop item. Z-score is calculated from the life cycle of each item. The life cycle represents the time span from the start of rating time t_S to the final rating time t_E. That is, the life cycle is the value of $t_E - t_S$. Here is the formula for z-score:

$$\mathrm{z-score}(x) = \frac{lc - \overline{lc}}{\sigma_{lc}} \tag{15}$$

Where lc is a life cycle of an item; $\overline{lc}$ and σ_{lc} are the mean and standard deviation of the life cycle of all these items, respectively. The specific division rules of the items are shown in Table 2. The $\bar{n}$ in the table represents the average number of ratings for each item.

Table 2. Item Division Rule.

Category	z-score (z)	Number of ratings (n)
Fad item	$z < 0$	$n \le \bar{n}$
Fashion item	$z < 0$	$n > \bar{n}$
Style item	$z \ge 0$	$n \le \bar{n}$
Scallop item	$z \ge 0$	$n > \bar{n}$

4. Our approach

4.1. Problem analysis

We see the shilling attack as an anomaly detection problem, and then we need to find the exception. It is mentioned in [17] that the normal behavior of the recommendation system can be characterized by a series of observations over time. In other words, the rating in the normal recommendation system is regular. When an abnormality occurs, the regular rating will be disturbed. The biggest challenge we face is to find the potential unusual time period when the normal ratings became abnormal. Then we propose a detection algorithm based on Kalman Filter model for the characteristics that the attack profiles are injected in a short time.

4.2. Detection process

The process of detection is as shown in Figure 1.

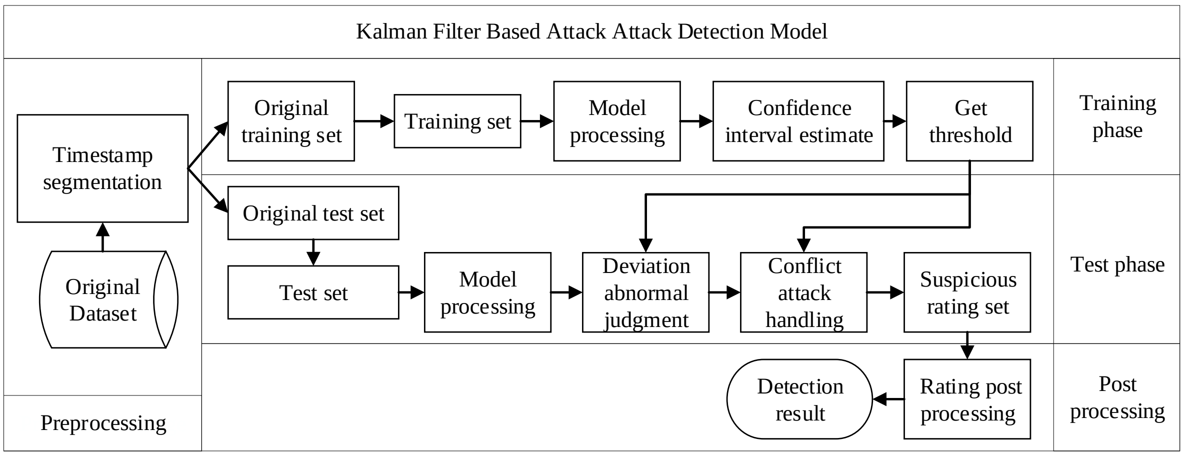

Figure 1. The process of detection model.

The detection process is divided into four phases, pre-processing phase, training phase, test phase and post-processing phase. The focus of the detection process is on training phase and test phase.

In the training phase, our goal is to get the threshold for detection. First, we extract a part of the original data as a training set. Then, we inject the attack profile into the training set and divide it into time. After that, calculate the predicted value for the next time period of each time period feature by our prediction model. The feature is the statistical indicator of the current time period data. Here, the total rating of the current time period and the average value of the rating are used as features. The specific content of the model is as described in section 4.3. Then, in section 4.4, we compare the predicted value with the actual value to calculate the deviation and provide a standard for judging abnormal deviations. In section 4.5, we count these deviations as a “deviation set”, then, carry out interval estimation on the deviation value in the set to obtain a threshold value, and finally, the choice of threshold is described.

There are two situations in shilling attack. One is that there is only a push attack or a nuke attack. Even if it occurs at the same time, it will act on different items and have an impact on the recommendation. This kind of attack is called a “normal attack” in this paper. The other is that there are two opposing users, who are for the same item, one for the highest rating and one for the lowest rating. When the number of profiles they inject is close, it does not have a big impact on the recommendation effect, but it is also an attack, which may have a potential impact on the recommendation system. This kind of attack is called a “conflict attack” in this article. Both types of attacks may exist in the data, so we should take corresponding measures in both cases.

In the test phase, our goal is to get the time period in which the anomaly rating occurs. We add the attack profile in a part of the original data set to form a test set. Then, get the "deviation set" of the test phase after the same processing as in the training phase. Screen this "deviation set" according to the threshold obtained in the training phase. The screening method is described in section 4.4. In this way, abnormal time period can be found based on the abnormal deviation value. For conflict attack, after detecting the abnormal deviation, we further process the normal data to deal with conflict attack. This part is described in section 4.6. Finally, the suspicious rating set is post-processed to obtain the detection result. The post-processing process is described in section 4.7.

4.3. Model processing

In this paper, we propose a Rating Detection Adapted Kalman Filter model (RDAKF). We have made some improvements to the traditional Kalman Filter model.

Prediction step:

$$\hat{x} = \frac{x(n_A + n_P)}{n_A} \tag{16}$$

$$\hat{P} = P + q \tag{17}$$

In the prediction step, first, we get the preliminary prediction value of rating sum x (the sum of the values of all ratings up to this state) and the model’s standard deviation P. Then, n_A represents the total number of ratings that have been calculated before the start of this condition. n_P represents the total number of ratings in the current time block. q represents the standard deviation of the model error. Here, $[(n_A + n_P)/n_P]$ is equivalent to F_t in (1).

Update step:

$$Kg = \frac{\hat{P}}{\hat{P} + r} \tag{18}$$

$$x^{+} = \hat{x} + Kg(z - \hat{x}) \tag{19}$$

$$P^{+} = (1 - Kg)\hat{P} \tag{20}$$

$$n_A = n_A + n_P \tag{21}$$

In the update step, x and P and preliminary prediction values $\hat{x}$ and $\hat{P}$ are further processed to obtain more accurate prediction values x^{+} and P^{+} as output prediction results. In (18), Kg represents Kalman gain and r represents the standard deviation of measurement error. In (19), z is the sum of the values of all ratings within the current time block (observation value). Equation (21) is to update the value of n_A.

In [16], they assume that user preferences do not change due to purchase actions, treating F_t and H_t in (1) and (2) as I (a unit vector), and taking into account the shift in user preferences, w_t and v_t in (1) and (2) are regarded as the standard normal distribution $N{\sim}(0, I)$. In our case, the ratings' standard deviation of the analyzed data sets is very close to 1, and the filler items of the injected attack files are also randomly generated by the model with a standard deviation of 1. So here, we assume that q and r in (17) and (18) are 1.

4.4. Deviation scheme

A recommendation system is basically stable when the amount of data is large enough. The deviation of the data is kept within a certain range. First, we declare the meaning of the deviation mentioned in this article here. The deviation mentioned in this paper refers to the difference between the value predicted by the model and the actual value of the data set.

The key point in judging anomalies is to determine whether the deviation is abnormal. So, we propose two deviations.

The total deviation:

$$v = y - \hat{x} \; where \; y = x + z \tag{22}$$

The average deviation:

$$v_A = v/n_P \tag{23}$$

y in the (22) represents the sum of the results of the previous step and the observed data of this step. In (23), v_A represents how much each rating in this step deviates from this step.

With these two parameters, we can determine which deviations are abnormal. There are four possible situations.

Both v and v_A are normal, indicating that the system has not been attacked at this time.

v is normal but v_A is abnormal, that is, the total deviation is normal and the average deviation is abnormal. In this case, it is possible that the rating data is small (the total deviation is normal) during this time period, and both are extremely rating values (average deviation abnormality). This situation is normal and cannot be judged as abnormal.

v is abnormal and v_A is normal. The average deviation at this moment is normal and the total deviation is abnormal. This phenomenon occurs at the time of the rating set, the number of ratings is large (the total deviation is abnormal), and both are normal rating (average deviation is normal).

Both v and v_A are abnormal. There are many ratings and extreme ratings, and we believe that this is the time of the attack.

In summary, when both v and v_A are abnormal, it is determined that the system is under attack.

4.5. Threshold selection

In order to judge whether the deviation exceeds the standard, it is necessary to set a threshold. To this end, we combine the interval estimation knowledge in section 3.3 to select a reasonable threshold.

We treat each deviation as a sample of data. Confidence interval estimates for individual samples were performed on these samples. Choose a suitable confidence level to get a confidence upper bound. And use this upper bound as the critical value, which is the threshold, as the basis for judging the abnormality.

Two thresholds are needed to determine the anomaly of two deviations, and these two thresholds need to be trained on the training set. Two upper bounds of confidence intervals η and η_A are obtained as thresholds. If $v > \eta$ and $v_A > \eta_A$, it is regarded as an abnormal point.

4.6. Conflict attack handling

The recommendation system may encounter competing attacks at the same time period when it encounters a hosted attack (Two attackers take the opposite strategy attack on the same item at the same time period). If the number of profiles injected by the two types of attackers is similar, the two attacks will cancel each other out. In this case, although the average rating of the item is hardly affected, it will affect the accuracy of the recommendation. Therefore, the detection algorithm should also respond to this situation.

We have devised a method to further screen normal data in the deviation detection.

We count the number of ratings and the number of extreme ratings (the highest or the lowest rating) for each time period of each item. The total number of ratings is n_T, and the number of extreme ratings is n_E. According to (24), α can be obtained for each time period of each item.

$$\alpha = \frac{n_E}{n_T} \tag{24}$$

Afterwards, it is similar to the processing method of the deviation value in section 4.5, and the interval threshold is used to select the appropriate threshold to judge whether the α is abnormal. This threshold is recorded as η_α.

Since some scattered true ratings are also treated abnormally when the number of ratings is too small during the time period, we count the average number of ratings $n_{Alltime}$ over all time periods. When $n_T > n_{Alltime}$ and $\alpha > \eta_\alpha$, we treat this condition as a exception.

In summary, the set of suspicious ratings obtained in section 4.5 and section 4.6 constitute the final set of suspicious ratings.

4.7. Post-processing

The detection effect of the model can only be locked to the time period during which the shilling attack occurs, and the ratings within the time period need to be further filtered. When we get a series of ratings for a suspicious time period, we need to remove the ratings that are not extreme ratings (if it is a push attack, the rating that is not the highest rating is removed, and if it is a nuke attack, the non-minimum rating is removed). For a shilling attack, if the attacker does not use extreme rating to attack the system, the effect of the attack will not be easy to achieve the ideal situation of the attacker. Then find users who rated these extreme ratings and convert the suspicious rating set into a suspicious user set as the final detection result.

5. Experiments

5.1. Experiment setup

In the experiment, we use the 100K Movielens dataset, which contains 100,000 ratings from 943 users and 1682 items, records a total of 215 days of data from September 20, 1997 to April 23, 1998. Each user in this dataset rates at least 20 movies. The rating ranges from 1 to 5. We treat all the data in the original data set as the data evaluated by genuine users and divide this data into parts every four days, for a total of 54 parts. We used 100 items including 5328 deviations as a training set for training. According to the proportion of the item types, 12 of them are fad items, 6 of them are fashion items, 32 of them are style items, and 50 of them are scallop items. Total deviation selects 99% confidence and average deviation selects 90% confidence.

First we explain the attack size and filler size [17]. Attack size refers to a percentage of the pre-attack user profiles that attack profiles account for. Suppose there are 1000 user profiles initially, and 100 attack profiles are injected, the attack size is 10%. Filler size is a percentage of filler items in all items.

This experiment is mainly about push attack. We select 5 items as target profiles. Insert different scales of attack size with filler size 5%, and make them the same size as the genuine users. For random attacks and average attacks, we generate ratings from $N{\sim}(\bar{r},\bar{\sigma}^2)$ and $N{\sim}(\bar{r_i},\bar{\sigma_i}^2)$. $\bar{r}$ and $\bar{\sigma}$ are the mean rating value and the standard deviation value of the entire data set. $\bar{r_i}$ and $\bar{\sigma_i}$ are the mean rating value and the standard deviation value of the current item *i*. In the bandwagon attack, we select movie 50 as the selected item and rate it $r_{max} = 5$.

We tested our method on three attacks and then compared it with KNN, Bayes [11], SVM [12] and NMFSAD [18] algorithms. KNN is a traditional classification algorithm. It is the grouping of the k most similar users. Bayesian detection algorithm is mentioned in [11] uses the combiner strategy that combines multiple classifiers in an effort to detect shilling attacks. The SVM algorithm in [12] combines target item analysis method on the basis of the SVM model to improve the detection performance. NMFSAD algorithm in [18] applies non-negative matrix factorization techniques for feature extraction of data and combines clustering ideas of unsupervised learning detection algorithms to cluster data.

5.2. Evaluation metrics

To measure the performance of the detection algorithm, we use two metrics for precision and recall.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \tag{25}$$

Precision reflects the proportion of a category of targets detected by the detector that really belong to that category. Where TP is the number of attack profiles detected correctly, FP is the number of misclassified genuine profiles into attack profiles.

Recall reflects the true number of categories in a category of targets detected by the detector. FN is the number of attack profiles misclassified into genuine profiles.

5.3. Experimental results and discussion

In this section, we test the performance of our detection algorithm. First, we test the performance of our algorithm under different attack models. Then, we test the impact of different confidence levels on detection performance. Finally, we perform performance comparisons with the other four algorithms.

5.3.1. Performance of different attack models

We use our algorithm to test its precision and recall in dealing with random attack, average attack and bandwagon attack. The filler size is set to 5%.

Figure 2 shows the precision of our approach under different attack sizes. It can be seen that as the size of the attack increases, the precision of the detection algorithm is increased and the precision remains above 0.8. However, compared to the other two attacks, the algorithm has a poor effect on the low attack size of bandwagon attack. It can be seen that our algorithm has more misclassification in bandwagon attack. Bandwagon attack contains selected item, which is not included in the other two types of attacks. From the experiment result, we can conclude that the selected item has a certain impact on the detection performance of our algorithm.

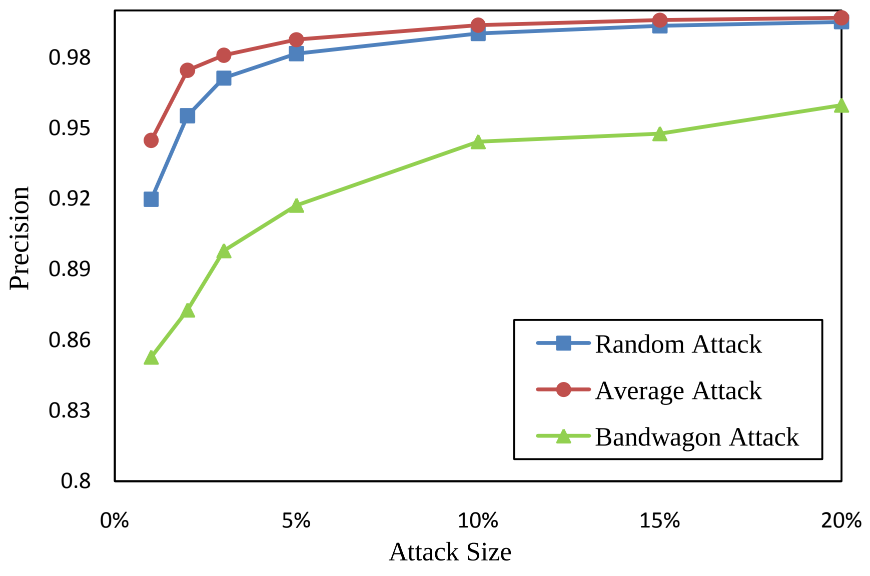

Figure 2. Precision of different attack models when the attack size varies and filler size is 5%.

Then, Figure 3 is the recall of our method under different attack sizes. Our algorithm has the best detection rate for average attack, and it is acceptable for random attack, and the same problem with precision is not effective for bandwagon attack. It can be seen that the selected item also has an impact on the recall.

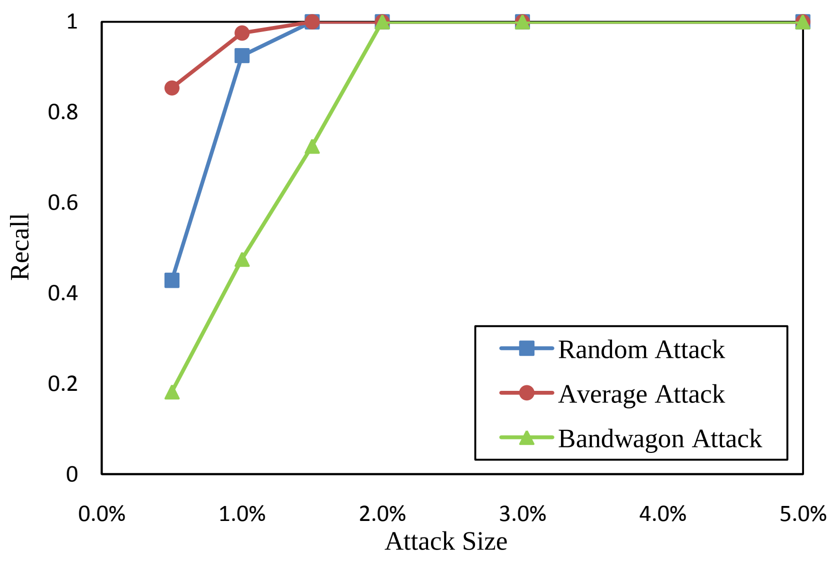

Figure 3. Recall of different attack models when the attack size varies and filler size is 5%.

In addition, we also test our algorithm's ability to detect special cases, as shown in Figure 4 and Figure 5.

The “mixed attack” in the figure represents a combination of normal push attack and conflict attack. For example, 5% of mixed attack represents 5% push attack and 5% conflict attack.

Although the precision have dropped a little under the low attack size, it can be seen that our algorithm can still maintain good detection results in the face of conflict attack.

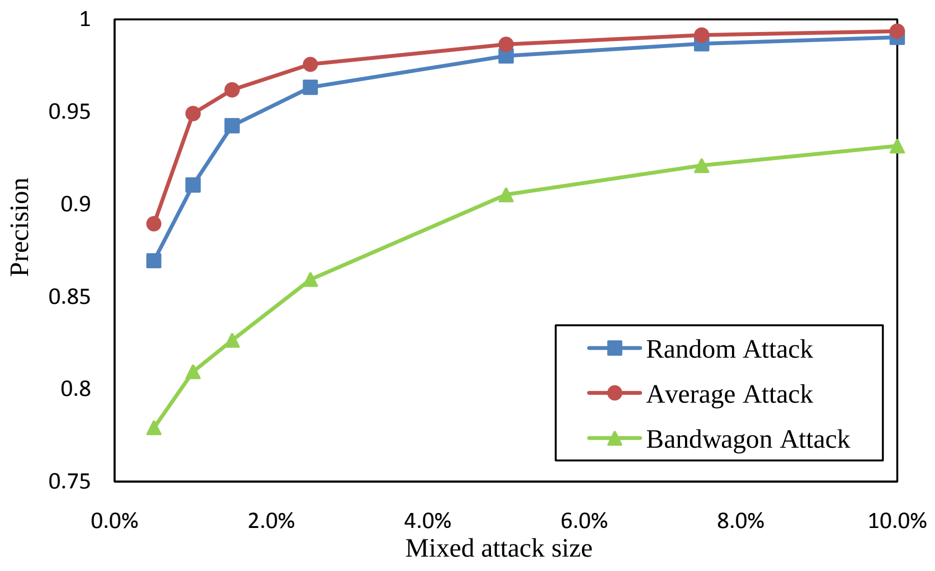

Figure 4. Precision of different attack models when encountering a mixed attack.

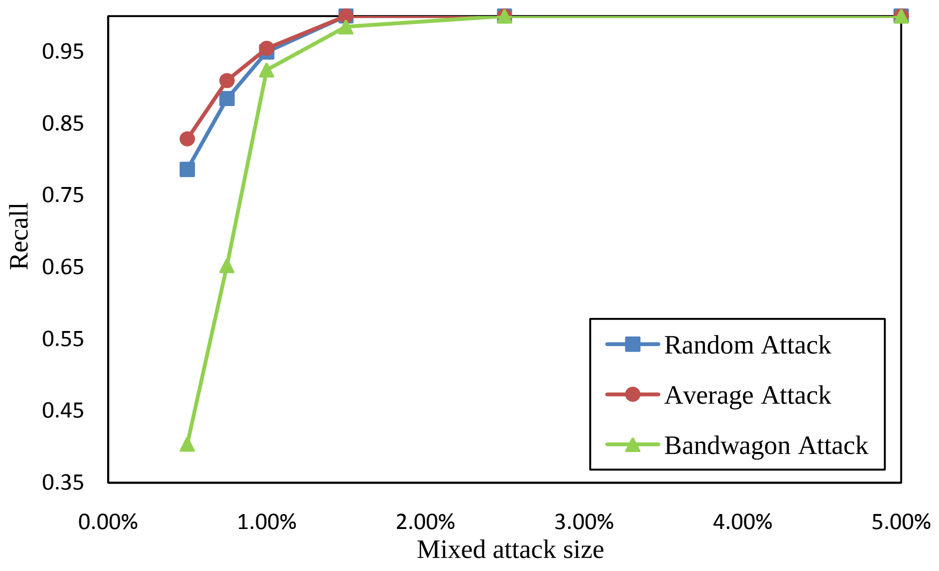

Figure 5. Recall of different attack models when encountering a mixed attack.

5.3.2. Impact of confidence levels

The choice of confidence level will affect the precision and recall. If the confidence level is high, then the range it accepts will expand. It means that a larger deviation can be accepted. The deviation caused by some attacks may not be detected, that is, the recall is reduced. However, due to the increase in acceptability, the probability of a normal deviation being misjudged is reduced, that is, the precision is improved. Conversely, if the confidence level is low, the recall will increase and the precision will decrease. So we compare the impact of different confidence level selections on performance to choose an appropriate one.

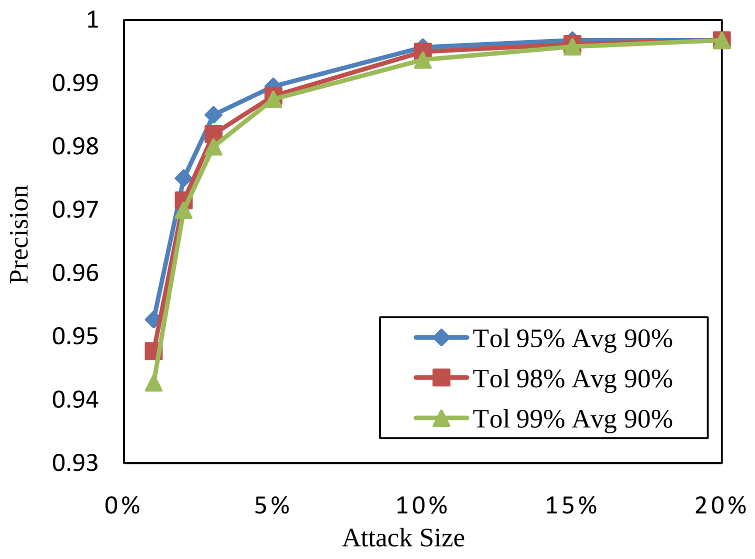

Figure 6. The effect of different confidence levels of total deviation on detection.

Figure 6 shows the detection effect of the confidence levels for the different total deviations when the confidence level of the average deviation is 90%. It can be seen that the higher the confidence level of the total deviation, the higher the accuracy when the confidence level of the average deviation is fixed.

Figure 7 and 8 show the detection effect of the confidence level of the total deviation and the confidence level of the different average deviations. Figure 7 shows the average attack and Figure 8 shows the bandwagon attack. In average attack, 90% confidence level and 85% confidence level are comparable, while 95% confidence level is less effective. In bandwagon attack, 95% confidence level highest, 90% second, 85% worst.

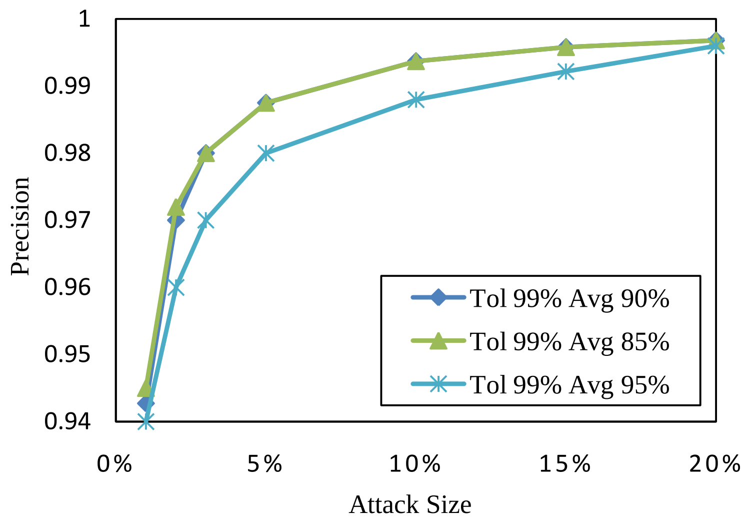

Figure 7. The effect of different confidence levels of average deviation on detection (average attack).

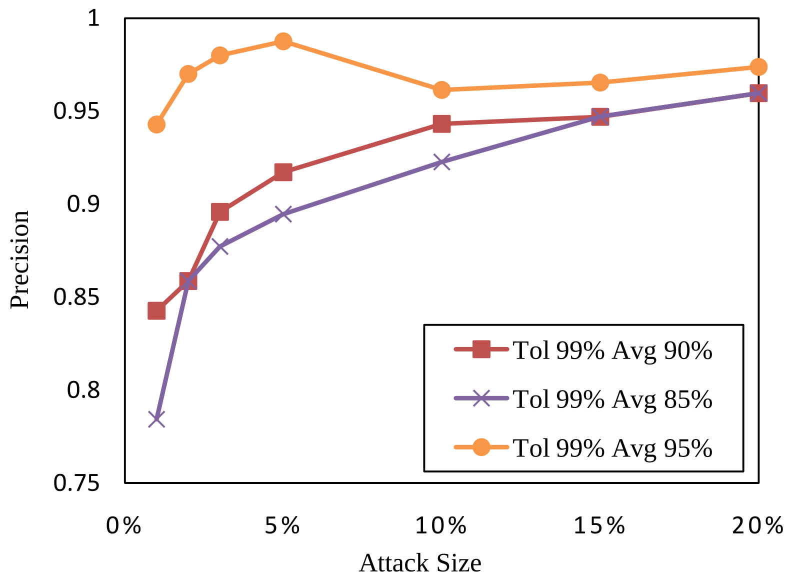

Figure 8. The effect of different confidence levels of average deviation on detection (bandwagon attack).

Figure 9 is the recall of the five combinations mentioned above.

Based on the above comparison, for the confidence level of the total deviation, high confidence level brings high precision. The recall is almost the same, so we choose a 99% confidence level.

For the confidence level of the average deviation, both the 85% and 95% confidence levels have significant disadvantages under certain circumstances. So we choose a more balanced 90% confidence level.

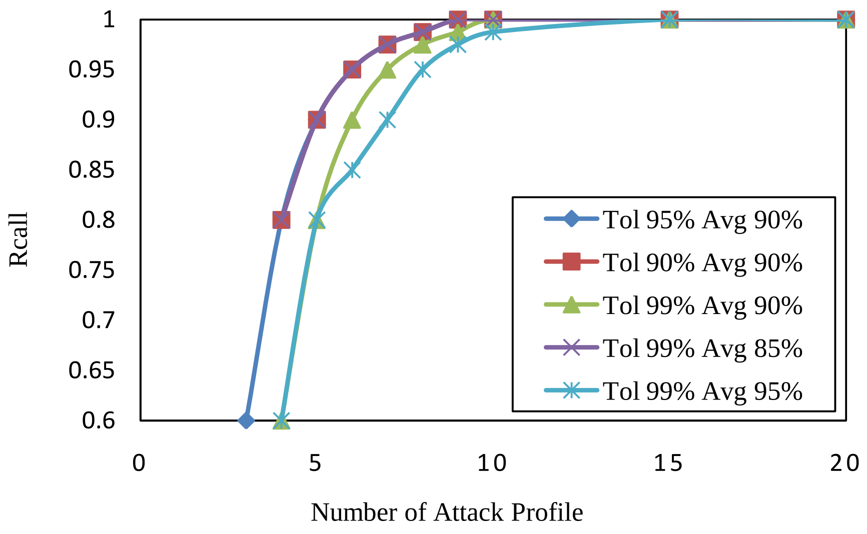

Figure 9. The recall of different confidence levels (bandwagon attack).

5.3.3. Comparison of detection performance

We compared our approach with four algorithms. As shown in Figures 10, in terms of the precision of the average attack, our method is slightly better than the improved SVM algorithm (slightly less than it in the case of low attack size) and is significantly better than the other two algorithms. Bayes algorithm is not suitable for detection on low attack size. The performance of NMDSAD is similar to KNN.

The precision contrast in the bandwagon attack is shown in Figure 11, the precision of other algorithms does not change much compared to it in the average attack, and the precision of our algorithm has a certain decline due to the influence of the selected item.

Figure 12 is the recall comparison of the four algorithms in the average attack. Our algorithm shows the best performance. NMDSAD is a bit less than our algorithm. The improved SVM algorithm has reached a very high precision level due to the addition of the target item analysis strategy. However, this strategy is mainly to reduce the situation that genuine users are classified as attack users. Therefore, its recall is limited to the SVM model itself does not reach a higher level. The combiner strategy used in the Bayes detection algorithm is also for misclassification. So, its recall is also based on the performance of the Bayesian model itself. The KNN algorithm responds to the assumption that the attacker has a high similarity, and can clearly distinguish between the genuine user and the attack user, but for those attack users whose attack effect is not obvious, there is no good detection effect (low recall).

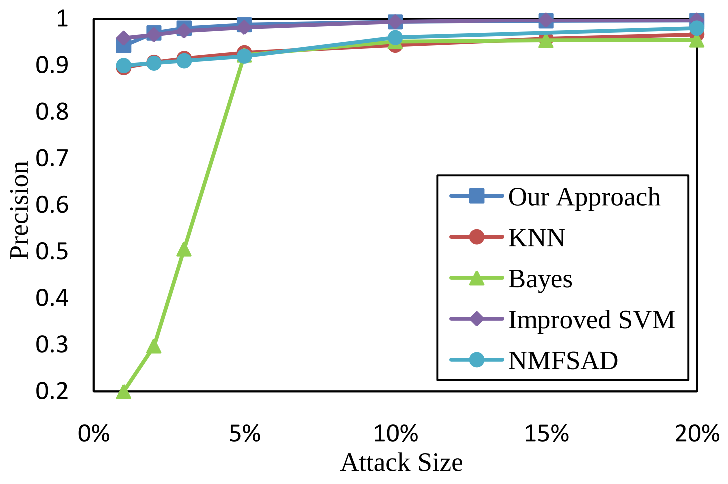

Figure 10. Precision of different detection algorithms on average attack with a filler size of 5%.

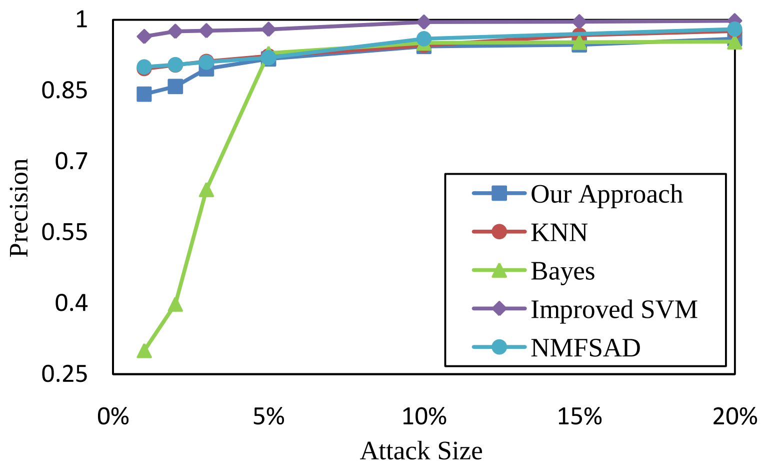

Figure 11. Precision of different detection algorithms on bandwagon attack with a filler size of 5%.

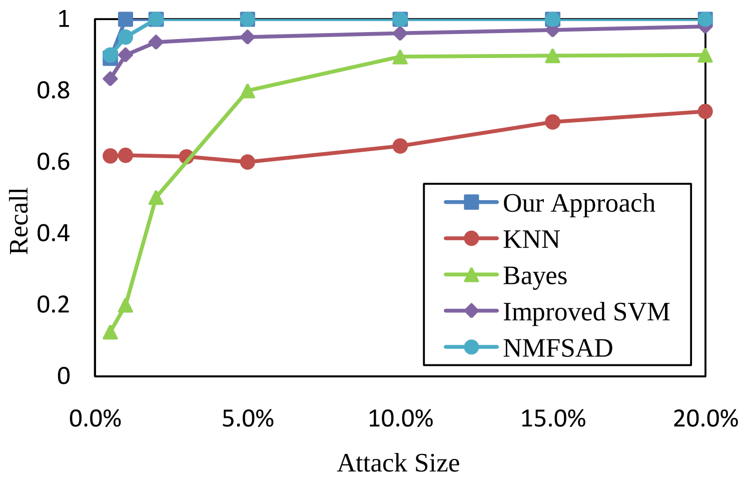

Figure 12. Recall of different detection algorithms on average attacks with a filler size of 5%.

As shown in Figure 13, we also test the effects of mixed attack on these algorithms.

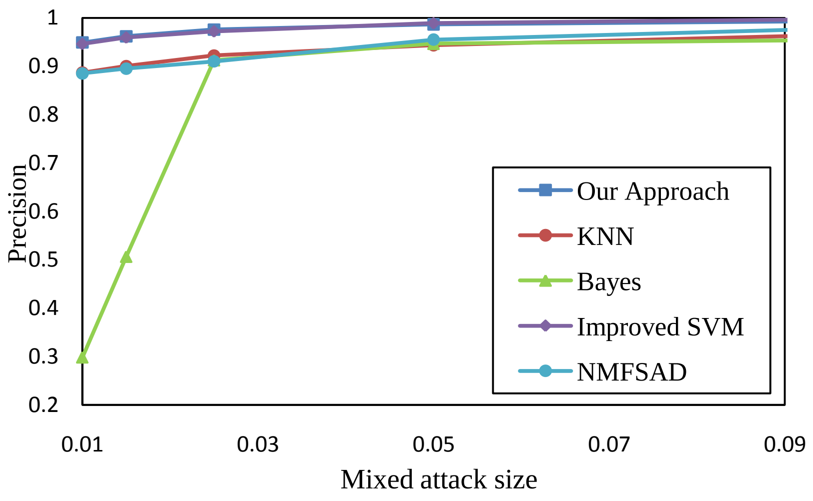

Figure 13. Precision of different detection algorithms on average attack (mixed attack).

The detection effect of all algorithms has a slight decrease, and our algorithm can still maintain a high level.

Our method is different from other algorithms in terms of user vectors but detects the true and false of the rating itself. It is highly sensitive to abnormal conditions, which is reflected in the high recall. Since it is targeted at rating data, the likelihood of misclassification increases when the rating is mapped to the user, that is, the precision is lower in some cases.

6. Conclusion and future work

In this paper, we propose an improved Kalman filter model RDAKF. It calculates the deviation between the predicted value and the actual value and measures the abnormal rating data according to the confidence interval estimation. The experiment uses the dataset of Movielens to analyze the performance of our approach under different attack models. Then, we compare our algorithm with other algorithms. The experimental results show that our algorithm has a good effect in detecting the abnormal ratings, but it is not effective for some attack models with selected items.

In future work, we will focus on the impact of selected items and consider more types of attacks. Combine some implicit features with more datasets to come up with a better solution.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (61702368), Major Research Project of National Nature Science Foundation of China (91646117) and Natural Science Foundation of Tianjin (17JCYBJC15200, 18JCQNJC0070).

Conflict of interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. P. A. Chirita, W. Nejdl and C. Zamfir, *Preventing shilling attacks in online recommender systems*, 2005 ACM International Workshop on Web Information and Data Management (WIDM), 2005. Available from: https://dlnext.acm.org/doi/abs/10.1145/1147376.1147387.
2. R. Burke, B. Mobasher, C. Williams and R. Bhaumik, *Classification features for attack detection in collaborative recommender systems*, 2006 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2006. Available from: https://dlnext.acm.org/doi/proceedings/10.1145/1150402.
3. C. A. Williams, B. Mobasher and R. Burke, Defending recommender systems: detection of profile injection attacks, *SOCA*, **1** (2007), 157–170.
4. B. Mehta, *Unsupervised shilling detection for collaborative filtering*, 2007 National Conference on Artificial Intelligence, 2007. Available from: https://www.aaai.org/Library/AAAI/2007/aaai07-222.php.
5. F. Peng, X. Zeng, H. Deng and L. Liu, Unsupervised detection of shilling attack for recommender system based on feature subset, *Comput. Eng.*, **40** (2014), 109–114.
6. Z. Zhang and S. R. Kulkarni, *Graph-based detection of shilling attacks in recommender systems*, 2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2013. Available from: https://ieeexplore.ieee.org/xpl/conhome/6648476/proceeding.
7. Z. Wu, J. Cao, B. Mao and Y. Wang, *Semi-SAD: Applying semi-supervised learning to shilling attack detection*, 2011 ACM Conference on Recommender Systems (RecSys), 2011. Available from: https://dlnext.acm.org/doi/proceedings/10.1145/2043932.
8. C. Lv and W. Wang, Semi-supervised shilling attacks detection method based on SVM-KNN, *Comput. Eng. Appl.*, **49** (2013), 7–10.
9. M. Gao, Q. Yuan, B. Ling and Q. Xiong, Detection of abnormal item based on time intervals for recommender systems, *Sci. World J.*, **2014** (2014), 1–8.
10. Z. Yang and Z. Cai, Detecting abnormal profiles in collaborative filtering recommender systems, *J. Intell. Inf. Syst.*, **48** (2017), 499–518.
11. W. Bhebe and O. P. Kogeda, *Shilling attack detection in collaborative recommender systems using a meta learning strategy*, 2015 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), 2015. Available from: https://ieeexplore.ieee.org/xpl/conhome/7164876/proceeding.
12. W. Zhou, J. Wen, Q. Xiong, M. Gao and J. Zeng, SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems, *Neurocomputing*, **210** (2016), 197–205.
13. M. O'Mahony, N. Hurley, N. Kushmerick and G. Silvestre, Collaborative recommendation: A robustness analysis, *ACM Trans. Internet. Technol.*, **4** (2004), 344–377.
14. B. Mobasher, R. Burke, R. Bhaumik and C. Williams, Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness, *ACM Trans. Internet. Technol.*, **7** (2007), 23–60.

15. R. T. Sikora and K. Chauhan, Estimating sequential bias in online reviews: A Kalman Filtering approach, *Knowledge-Based Syst.*, **27**(2012), 314–321.
16. K. Inuzuka, T. Hayashi and T. Takagi, *Recommendation system based on prediction of user preference changes*, 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2016. Available from: https://ieeexplore.ieee.org/xpl/conhome/7814734/proceeding.
17. C. Williams and B. Mobasher, Profile injection attack detection for securing collaborative recommender systems, *DePaul Univ. CTI Technol. Repo.*, **2006** (2006), 1–47.
18. K. Fang and J. Wang, Shilling attacks detection algorithm based on nonnegative matrix factorization, *Comput. Eng. Appl.*, **53** (2017), 150–154.