SVM-TIA A Shilling Attack Detection Method Based on SVM and Target Item Analysis in Recommender Systems

Wei Zhou^a, Junhao Wen^{a,b}, Qingyu Xiong^b, Min Gao^b, Jun Zeng^b

^a*College of Computer Science, Chongqing University, China*

^b*School of Software Engineering, Chongqing University, China*

Abstract

Due to the open nature of recommender systems, collaborative recommender systems are vulnerable to profile injection attacks, in which malicious users inject attack profiles into the rating matrix in order to bias the systems' ranking list. Recommender systems are highly vulnerable to shilling attacks, both by individuals and groups. Most of previous research focuses only on the differences between genuine profiles and attack profiles, ignoring the group characteristics in attack profiles of an attack. There also exist class unbalance problems in supervised detecting methods, the detecting performance is not as good when the amount of samples of attack profiles in training set is smaller. In this paper, we study the use of SVM based method and group characteristics in attack profiles. A two phase detecting method SVM-TIA is proposed based on these two methods. In the first phase, Borderline-SMOTE method is used to alleviate the class unbalance problem in classification; a rough detecting result is obtained in this phase; the second phase is a fine-tuning phase whereby the target items in the potential attack profiles set are analysed. We conduct tests on the MovieLens 100K Dataset and compare the performance of SVM-TIA with other shilling detecting methods to demonstrate the effectiveness of the proposed approach.

Keywords: recommender system, shilling attack detection, unbalanced data, SVM, target item analysis

1. Introduction

In recent years, we live in the information age. Information Overload is when you are trying to deal with more information than you are able to process to make sensible decisions. Recommender systems have become an effective tool to overcome information overload problem. Recommender systems predict a rating or preference that a user would give to an item [1, 2]. Collaborative filtering approaches typically build models from a user’s past behaviour coupled with similar decisions made by other users. This is then used to build a model to predict items or ratings for items that a user may be interested in. In this paper, we mainly talk about shilling detection in user-based collaborative filtering recommender systems. Collaborative filtering techniques have been one of the most popular personalized recommendation techniques. There are some performance advantages using collaborative filtering techniques: The first one is that collaborative filtering can filter information which computers are unable to automatically analyze, such as books, music, text and so on. The second one is that collaborative filtering can share experiences, making content analysis more accurate; and the ability to filter some complex, elusive concepts (such as user interests). The third one is that collaborative filtering has the capabilities to recommend new information, helping users find potential interests.

But studies have shown that collaborative filtering technology itself has serious shortcomings and deficiencies. Collaborative filtering techniques rely on user preferences files to provide users with personalized recommendations, which will make collaborative filtering algorithm could be attacked easily by injecting false user profiles. Due to the open nature of collaborative filtering recommender systems, they suffer vulnerabilities of being attacked by malicious users by injecting profiles consisting of biased ratings [3]. These attacks are carried out in order to influence the system’s behavior, and have been termed “shilling” or “profile injection” attacks, and attackers as shillers or attackers [4].

In some e-commerce websites, there are a team of shillers who can push a specified item to the recommended list in a short period of time for money [1]. Individuals may be interested in promoting or demoting an item, known as a target item, by manipulating the recommender system. Most attacks can be implemented as follows. The attacker takes on different identities within the system, and creates a user profile for each identity, which is referred as attack profiles. Within each of the profiles created, the attacker would

then manipulate the recommendation by rating or recommending a particular target item. In order to obfuscate themselves and appear as genuine users in the system, the attack profiles will contain ratings for non-target items. These ratings can be selected in different ways either randomly or more intelligently if the attacker has prior knowledge of the ratings in the system. Recent work has shown that even modest attacks are sufficient to manipulate the behaviour of the most commonly used recommendation algorithms [5, 6].

There are several hazards of attacks in recommender systems. Attacks can cause different losses to unprotected systems. The first is it will be unfair representation of users in recommender systems. The second is that the recommender systems failed to produce proper recommendations to users, which could ruin the reputation of recommendation systems. Under some conditions, a large number of attack profiles can lead to a breakdown of a system [7]. It is difficult to prevent unscrupulous users from injecting fake data (profiles) into a system. To ensure the trustworthiness of recommender systems, attack profiles need to be detected and removed accurately.

The first contribution of this research is that we use an adaptive synthetic sample fitting method Borderline-SMOTE to alleviate class unbalance in shilling attack detection. The second contribution of this paper is a proposed hybrid attack detection method, Target Item Analysis, which uses group features of attack profiles in an attack. The rest of the paper is organized as follows. In the next section, we examine previous work and background in the area of attack detection in recommender systems; in the Section 3 we describe the details of our approaches. Our experimental results are presented in Section 4. We discuss and summarize our research in Section 5.

2. Related Work

In this section, we introduce the background knowledge which will be used in this paper. Firstly, several common attack models and their characteristics are described. Secondly, the theory of target item analysis are introduced. Finally, class unbalance problem of supervised detecting method in shilling detection is analyzed.

2.1. Background

The word “shilling” was first coined by [4]. There have been some recent research efforts aimed at detecting and reducing the effects of profile injection attacks [5, 8, 9, 10, 11, 12, 13]. These attacks consist of a set of attack profiles,

each containing biased rating data associated with a fictitious user identity. In order to make themselves hard to be detected, the attackers make shilling profiles look similar to genuine profiles. Intentional attacks can cause the recommender system to become unreliable and untrustworthy, which can result in user distrust. In this section we concentrate on researches in attack detection in recommender systems. Detecting method can be divided into different categories according to different dimensions. In this paper, we divide attack detection algorithms into three categories: supervised, unsupervised, and semi-supervised.

In the first category, attack detection techniques are modelled as a classification problem. A lot of research has been undertaken to employ supervised learning for shilling attack detection [3, 14, 15]. Three classification algorithms, k NN-based, C4.5-based and SVM -based, are used to improve the robustness of the recommender system [16]. These supervised algorithms need a large number of labeled users to enhance the accuracy [17]. Classification-based methods require balanced numbers of attack and normal profiles to train the classifiers. Most early detection algorithms 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 Network classifiers, or SVM based classifiers [18, 19, 20].

In the second category, unsupervised detection approaches address these issues by training on an unlabeled dataset. These methods involve far less computational effort as compared to supervised approaches. The benefit of this is that these techniques facilitate online learning and improve detection accuracy. There has been significant research interest focused on detecting attack profiles using the unsupervised approach. Some of the techniques use clustering, association rules methods and statistical approaches [21, 22, 23]. Zhang et al. [23] used a Singular Value Decomposition (SVD) method to learn a low-dimensional linear model. Hurley et al. [21] utilizes Neyman-Pearson theory to construct both supervised and unsupervised detectors. An unsupervised shilling attack detection algorithm using principal component analysis (PCA) was proposed in [24]. Statistical detection techniques [7] are also used to detect profile injection attacks.

In the third category, semi-supervised detection approaches, such as [25, 26], make use of both unlabelled and labelled user profiles for multi-class mod-

elling. Cao et al. [25] proposes a new semi-supervised method called *Semi-SAD* shilling attack detection algorithm using both types of data. HySAD introduces MC-Relief to select effective detection metrics, and semi-supervised Naive Bayes (SNBL) to precisely separate random-filler model attackers and average-filler model attackers from normal users.

2.2. Attack Models

An attack consists of attack profiles that are introduced into the system in order to alter recommendation lists of a set of target items. Based on different assumptions about the attacker’s knowledge and purpose, a number of attack models have been identified [3]. There are four popular attack models: random attack, average attack, bandwagon attack, and segment attack models. Ratings in an attack profile can be divided into three sets of items: a target item set I_T , a selected item set I_S , and a filler item set I_F . Figure 1 shows the structure of an attack profile. For each attack profile, depending on the attack type, one or more items are chosen and given either the maximum and minimum rating value. Selected set I_S is the set of selected items that have special features, I_S is not necessary for some attack models; I_F is the set of filler items usually chosen randomly. Filler items in an attack profile are a set of items that make the profile look similar to genuine profiles. Features of the attack models are shown in Table 1.

Figure 1: Structure of attack profiles

Table 1: Features of the attack models

Attack model	I_S (Selected Items)	I_F (Filler Items)	I_T (Target Items)
Random Attack	$\emptyset$	random ratings	r_{max}/r_{min}
Average Attack	$\emptyset$	mean of each item	r_{max}/r_{min}
Bandwagon Attack	r_{max}	random ratings	r_{max}/r_{min}
Segment Attack	r_{max}	random ratings	r_{max}/r_{min}

The quality of the filler items depends on the existing knowledge gathered from the recommender system. As more knowledge is obtained, an attack generated is more sophisticated. The major difference of attack models is

how the ratings of filler items and the selected items are determined. The differences among attack models are the variance rating distribution in filler items and the selected items.

2.3. Profile Attributes Extraction

In order to distinguish attack profiles from genuine profiles, researchers divide profile attributes into general attributes and attack model specific attributes according to the attack models [27]. General attributes try to capture the property from the perspective of descriptive statistics of the profiles, which distinguish the genuine profiles and attack profiles. Specific attributes are those that try to distinguish the genuine profiles and attack profiles of a particular attack model. Due to the sparsity and rating high-dimensional property of rating matrix in recommender systems, it is unrealistic to apply shilling attack methods directly on the original rating matrix. Therefore, researchers will focus on profile attributes extraction techniques and dimensionality reduction techniques, detection algorithms based on supervised learning methods are then implemented on a set of attributes extracted from these profiles. Figure 2 show the schematic diagram of feature extraction of profiles. q attributes are extracted in the process.

$$\begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ p_{3,1} & p_{3,2} & \cdots & p_{3,n} \\ \cdots & \cdots & \cdots & \cdots \\ p_{m-1,1} & p_{m-1,2} & \cdots & p_{m-1,n} \\ p_{m,1} & p_{m,2} & \cdots & p_{m,n} \end{pmatrix} = \begin{bmatrix} Attribute_1 & \cdots & Attribute_q \\ Attribute_1 & \cdots & Attribute_q \\ Attribute_1 & \cdots & Attribute_q \\ \cdots & \cdots & \cdots \\ Attribute_1 & \cdots & Attribute_q \\ Attribute_1 & \cdots & Attribute_q \end{bmatrix}$$

Figure 2: Feature extraction of profiles

The training set is created as a combination of user profiles from the MovieLens rating matrix and attack profiles generated using the attack models described in Section 2.2. Each profile is labeled as either being an attack profile or as a genuine user profile. (We assume that all profiles in MovieLens rating matrix are genuine profiles.) A binary classifier is then created based on the training set using the attributes described below. Each profile is labeled as either being part of an attack or as coming from a genuine user.

2.4. *Borderline-SMOTE method*

Previous research apply SVM method to attack detection in recommender systems by extracting user profile features and building attack detection model [16, 28]. Comparing with other supervised learning methods such as decision trees and neural networks, SVM has a better overall performance than other methods [15, 17, 29]. Class unbalance problem exists in classification when using machine learning methods [30, 31]. Classification performance of the traditional classifiers is always restricted when facing unbalance data [32]. Chawla N.V [33] came up with a synthetic minority over-sampling technique SMOTE. The idea of the method is: first find k nearest neighbours of the minority class, choose one neighbour from k nearest neighbours, construct new sample set by random linear interpolation between the new sample and the origin sample. SMOTE can solve classification fitting problem well and promote the performance of the classifier. But SMOTE construct the same amount samples for each sample, and dose not consider distribution of border samples, which increases the probability of repeated samples. In order to overcome these drawbacks, some improved algorithm are proposed [34]. S-MOTE construct synthetic sample for every sample, but Borderline-SMOTE only construct samples for samples nearby the border. In this paper, classification algorithms for unbalanced data is borrowed and applied in shilling attack detecting, thereby increasing the recall of detecting results. By using target item analysis method, reduce false positive rate of detecting results.

3. The proposed method

In this section, the idea of target item analysis is introduced. Based on target item analysis, a detection method based on Borderline-SMOTE method is proposed. SVM-TIA consists of two stages: stage of classification and stage of target item analysis.

3.1. *Framework of shilling attack detection*

A single profile can not affect the recommendation list too much. Attack profiles reaches a certain quantities can alter the recommendation list of the target item. Based on this consideration, we can abstain a better overview of attack profiles when consider them as a whole. In this paper, we propose a shilling detecting framework, which contains two phases. In the first phase, a rough detection method is used to get a suspicious profile set contains attack profiles and genuine profiles. The second phase is a fine-tuning phase

whereby the target items in the potential attack profiles set are analysed. Based on the analysis we can then remove genuine profiles from this set, which can reduce false positives in the final result. Figure 3 shows the two phase shilling detecting framework in recommender system.

Figure 3: Framework of two phase shilling detecting method

3.2. Shilling Attack Detection Based on SVM and Target Item Analysis

In this section, we study the use of SVM based method and group characteristics in attack profiles. A two phase detecting method SVM-TIA is proposed based on these two methods. In the first phase, Borderline-SMOTE method is used to alleviate the class unbalance problem in classification; a rough detecting result is obtained in this phase; the second phase is a fine-tuning phase whereby the target items in the potential attack profiles set are analysed.

There are three parts in the process. The first part is attribute extraction. A rating matrix is composed by ratings on all items in the recommender system. In the matrix, every row of data stands for all the ratings a user rated on all items; similarly, every column of data stands for all the ratings on the item by all the users in the system. Rating data of a user is called the user profile. Different profile attributes are extracted from every profile. In

the second part, a SVM-based classifier is constructed base on Borderline-SMOTE method. In this part, a rough detecting result is got using the classifier. In the third part, target item analysis method is a fine-tuning phase whereby the target items in the potential attack profiles set are analysed. Profiles that are misjudged as attack profiles can be filter out. Figure 4 is the flow diagram of SVM-TIA shilling detecting method based Borderline-SMOTE.

Figure 4: Flow diagram of SVM-TIA shilling detecting method based Borderline-SMOTE

3.3. Target item analysis

The key idea of target item analysis is to find the target item(s) ID, and consider all profiles rate on that target item as attack profiles. The goal of attackers is to change the recommendation frequency of target items, in other words, prediction shift of target items. All shilling attack profiles rate on target item with maximum or minimum rating when carrying out an attack. In the rating matrix composed by profiles of an attack, ratings on target item would be maximum or minimum rating. If the target item IDs are found, then all the attack profiles rate on that target item can be retrieved according to the attack intent.

The rating matrix of recommender system is sparse. For example, in MovieLens 100K Dataset, about only 5% elements are rated. The count number of maximum or minimum rating on target item would be greater than that of other items. Taking push attack as an example, assume there are m genuine profiles and n attack profiles in the rating matrix we get; the sparsity of the rating matrix is λ ; The value of expectation of rating 5 is $E(item_i | rating = 5)$ equals to $(m + a)\lambda\gamma$. In the MovieLens 100K Dataset, $m = 943$; $a < 0.1m$; $\lambda < 5\%$; $\gamma = 20\%$.

$$E(item_i | rating = 5) = 0.001m < 1$$

This means if there are no attack profiles in the rating matrix, only a few genuine profiles are misjudged, which is acceptable. In previous research, in order to achieve a considerable prediction shift, a number of attack profiles θ are needed. In this paper, we set the threshold value θ equals 6. In the process of shilling attack detecting, we first filter out profiles that have low possibility of being attacks, and get a suspicious profile set SUS_{RD} . We use target item analysis to filter genuine profiles in SUS_{RD} . The pseudocode of target item analysis is shown in Algorithm 1.

Algorithm 1 SVM-TIA Phase 2, Filter out genuine profiles.

Input: The set of suspected profiles SUS_{RD} ; highest rating r ; item set I ;

Output: Final detect result set $DetectedResult$;

```

1:  $DetectedResult = \emptyset$ ;
2:  $\forall i \in I, count_i \leftarrow$  number of ratings in  $item_i$  equal to  $r$ ;
3: while  $\max(count) > \theta$  do
4:  $item_t \leftarrow \{item_i | count_i = \max(count)\}$ ;
5:  $\forall p \in SUS_{RD}, P \leftarrow p$  rate  $item_t$  with  $r$ ;
6:  $DetectedResult \leftarrow P \cup DetectedResult$ ;
7:  $SUS_{RD} \leftarrow SUS_{RD} - P$ ;
8: end while
9: return  $DetectedResult$ .
```

Table 2 is an example that illustrates how target item analysis works. Let the table be the rating matrix composed by genuine profiles and attack profiles. $User_1$ to $User_m$ are genuine profiles, while $Attack_1$ to $Attack_p$ are attack profiles injected. $Item_1$ to $Item_n$ are items in the recommender system. Numbers from 1 to 5 are ratings of a user to an item, 5 means the

user likes the item, while 1 means the user dislike it. 0 means the user dose not rate on the item. In this example, the count number of maximum rating is $Item_5$, $Item_5$ is recognized as the target item. All profiles rate $Item_5$ with maximum rating is considered as attack profiles. $User_3$ is misjudged in this example.

Table 2: An example of rating matrix and attack profiles.

	Item₁	Item₂	Item₃	Item₄	Item₅		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

4. Experiments and analysis

In this section, datasets used in the tests are introduced. Then the evaluation metrics of false positive rate, recall and precision are introduced. Finally, experimental results are analyzed.

4.1. Experimental data and settings

The datasets used in the experiments are the widely used *MovieLens* 100k Dataset by the GroupLens Research Project in the University of Minnesota. There are 100,000 ratings (1-5) from 943 users on 1682 movies in MovileLens 100k Dataset. Each user has rated at least 20 movies. Each user can rate a movie from 1 to 5. Where 1 is the lowest and 5 is the highest.

Filler size is defined as the ratio between the number of items rated by a user and the number of entire items in the recommender system. Attack size is defined as the ratio between the number of attack profiles in an attack and the number of entire profiles in the recommender system. The distribution of filler size for genuine users in the Movielens 100K Dataset indicate that the majority of genuine users only rate a small number of items. In order to

make the attack profiles hard to be detected, the attackers may use the same or similar filler sizes as genuine users do. In our experiments, the range of filler sizes for attack profiles is set between 1% and 20%.

The platform we implement all the experiments as follows: Hardware: CPU is Intel Core i7 processors, Windows 7, with 16G RAM. Software: All of our tests is on Matlab 2012b.

4.2. Evaluation metrics

To evaluate the performance of our proposed method, detection rate, false positive rate, recall and precision are used in this paper, which is defined as follows.

False positive rate is the ratio of the number of false positives with the number of attack profiles.

$$False\ Positive\ Rate = \frac{\#False\ Positives}{\#Genuines} \quad (1)$$

Recall is the ratio of the number of attack profiles in detecting result with the number of attack profiles.

$$Recall = \frac{\#True\ Positives}{\#True\ Positives + False\ Negatives} \quad (2)$$

Precision is the ratio of the number of attack profiles in detecting result with the number of profiles of detecting result.

$$Precision = \frac{\#True\ Positives}{\#True\ Positives + False\ Positives} \quad (3)$$

4.3. Analysis of information gain

As one of the most widely used measures for evaluating how informative an attribute is, information gain evaluates the importance of an attribute to a classification system, where larger information gain denotes more importance of the attribute. Therefore, we first use information gain to evaluate the importance of the attributes in this paper. We also analysis information gain of the attribute we proposed in paper [35]. Clearly, greater information gain value of a feature indicates greater contribution to the classification, thus attributes of greater information gain value are chosen as eigenvectors.

Profile attributes including RDMA, DegSim, WDMA, WDA, LengthVar, MeanVar, FillerMeanDiff [27], $DegSim'$ and so on, $DegSim'$ is a new profile

attribute we have proposed in our previous research [35]. Information gain is different when the filler size and attack size are different. In the experiments below, we calculate the mean information gain when the attack size is 5%, and filler size varies from 1% to 50%. Information gain of profile attributes in different attack models is shown in Figure 5. Figure 5(A) is from push attack type while Figure 5(B) is from nuke attack type. Information gain is different when the attack type and filler size different.

Figure 5: Information gain of profile attributes when filler size varies

4.4. Experimental results and analysis

In this section, detection results of the proposed shilling detecting method SVM-TIA is shown. In order to study performance of SVM-TIA when filler size and attack size varies, two experiments are designed.

To create the training set, we randomly select 200 genuine profiles from the Movielens 100K Dataset as samples of genuine profiles. Samples of attack profiles are generated by random, average, segment and bandwagon attacks with filler sizes varies from 1%, 3%, 5%, 7%. To balance the proportion between genuine profiles and attack profiles in the training set, we construct attack profiles with attack size varies from 1%, 3%, 5%, 10%, 20% for each attack. For each detecting result, detecting is repeated 20 times and the average values of detection results are reported for the experiments.

In the tests, we use Libsvm 3.0 [36] to generate the classifier. Target items are chose randomly when generating each attack profile. Push attacks are introduced in this paper, however, the proposed method can be put into effect to detect the nuke attacks.

Figure 6: Detection rate of different attack models when attack size and filler size vary in push attack

Figure 6 shows the recall of detecting different attack models when attack size and filler size vary in push attack. Four subfigures are the recall when detecting different attack models when the filler size is 1%, 3%, 5%, 7% respectively. In each subfigure, recall efficiency increase with the increasing of attack size. While recall increases with the increasing of filler size when the attack size is the same.

While Figure 7 shows false positive rate of SVM-TIA. Figure 7 shows the false positive rate when attack size and filler size vary in push attack while detecting different attack models. Four subfigures are the false positive rate when detecting different attack models when the filler size is 1%, 3%, 5%, 7% respectively. In each subfigure, false positive rate efficiency decrease with

Figure 7: False positive rate of different attack models when attack size varies in push attack

the increasing of attack size. While false positive rate decreases with the increasing of filler size when the attack size is the same. We can come to a conclusion that the false positive of detecting result using SVM-TIA is high when the quantity of attack profiles is small; the precision rate can be promoted by employing target item analysis method.

In order to test the effectiveness of our proposed method SVM-TIA when attack size varies, we also compare the detection result using different detecting method in the second experiment, including SVM, RSVM, C4.5, KNN shilling detecting method. In the experiments, we use the same profile attributes in SVM based detecting method and SVM-TIA based detecting method. In MovieLens 100K Dataset, the filler size is 3%, and the attack size varies from 1%, 2%, 5%, 10%, 20% respectively. Some attack models are used including random attack model, average attack model and bandwagon attack model. The detecting result is shown in Table 3 and Table 4.

Table 3 shows that the comprehensive result is best in all three attack

Table 3: Detection result of different methods when filler size is 3% and attack size varies in push attack

Attack size		1%	2%	5%	10%	20%	
Random attack	Recall	SVM	0.512	0.685	0.724	0.86	0.91
		RSVM	0.987	0.992	0.993	0.995	0.996
		C4.5	0.98	0.985	0.99	0.994	0.997
		KNN	0.715	0.753	0.804	0.95	0.996
		SVM-TIA	0.642	0.856	0.92	0.93	0.92
	Precision	SVM	0.55	0.62	0.73	0.82	0.91
		RSVM	0.79	0.82	0.9	0.95	0.98
		C4.5	0.853	0.86	0.867	0.874	0.88
		KNN	0.972	0.973	0.972	0.97	0.978
		SVM-TIA	0.98	0.991	0.992	0.994	0.995
Average attack	Recall	SVM	0.489	0.55	0.698	0.82	0.9
		RSVM	0.987	0.992	0.993	0.995	0.996
		C4.5	0.98	0.985	0.99	0.994	0.997
		KNN	0.715	0.753	0.804	0.95	0.96
		SVM-TIA	0.586	0.814	0.915	0.92	0.91
	Precision	SVM	0.532	0.602	0.715	0.806	0.889
		RSVM	0.79	0.82	0.9	0.95	0.98
		C4.5	0.853	0.86	0.867	0.874	0.88
		KNN	0.972	0.973	0.972	0.97	0.978
		SVM-TIA	0.974	0.982	0.986	0.992	0.993
Bandwagon attack	Recall	SVM	0.523	0.69	0.75	0.852	0.9
		RSVM	0.984	0.991	0.992	0.992	0.995
		C4.5	0.95	0.955	0.989	0.994	0.995
		KNN	0.702	0.723	0.824	0.954	0.965
		SVM-TIA	0.684	0.884	0.93	0.92	0.91
	Precision	SVM	0.53	0.6	0.69	0.79	0.88
		RSVM	0.77	0.81	0.9	0.95	0.97
		C4.5	0.823	0.835	0.856	0.868	0.879
		KNN	0.984	0.981	0.972	0.977	0.987
		SVM-TIA	0.985	0.986	0.991	0.993	0.995

types. The precision increases with the increasing of attack size. The precision reaches 100% when the attack size is over 10%.

From Table 3 and Table 4 we can see that, the recall of SVM based method and SVM-TIA method increase with the increase of attack size; and SVM-TIA has the higher recall than that of SVM method. The recall of SVM-TIA method is not as good as other methods. The precision of SVM-TIA is high, which means the false positive in the detecting result is low.

Table 4: Detection result of different algorithms when filler size is 3% and attack size varies in nuke attack

Attack size		1%	2%	5%	10%	20%	
Random attack	Recall	SVM	0.522	0.676	0.753	0.884	0.931
		RSVM	0.997	0.994	0.996	0.995	0.997
		C4.5	0.972	0.975	0.991	0.993	0.994
		KNN	0.725	0.756	0.818	0.953	0.997
		SVM-TIA	0.712	0.862	0.9	0.914	0.92
	Precision	SVM	0.56	0.64	0.74	0.83	0.92
		RSVM	0.79	0.83	0.92	0.96	0.99
		C4.5	0.823	0.865	0.857	0.874	0.882
		KNN	0.968	0.971	0.974	0.981	0.983
		SVM-TIA	0.981	0.992	0.993	0.995	0.996
Average attack	Recall	SVM	0.46	0.56	0.692	0.739	0.878
		RSVM	0.981	0.984	0.988	0.99	0.993
		C4.5	0.981	0.975	0.993	0.992	0.996
		KNN	0.725	0.733	0.824	0.942	0.996
		SVM-TIA	0.604	0.826	0.91	0.9	0.913
	Precision	SVM	0.51	0.62	0.72	0.81	0.91
		RSVM	0.78	0.81	0.87	0.93	0.95
		C4.5	0.833	0.839	0.847	0.846	0.858
		KNN	0.962	0.953	0.961	0.958	0.964
		SVM-TIA	0.978	0.985	0.988	0.992	0.992
Bandwagon attack	Recall	SVM	0.46	0.525	0.658	0.782	0.886
		RSVM	0.988	0.993	0.994	0.993	0.996
		C4.5	0.954	0.965	0.979	0.984	0.991
		KNN	0.732	0.742	0.814	0.924	0.947
		SVM-TIA	0.718	0.816	0.906	0.91	0.93
	Precision	SVM	0.54	0.695	0.78	0.825	0.93
		RSVM	0.79	0.82	0.91	0.95	0.98
		C4.5	0.825	0.815	0.844	0.868	0.88
		KNN	0.964	0.961	0.972	0.969	0.98
		SVM-TIA	0.978	0.982	0.991	0.991	0.995

There is no big difference in the detecting result between push attack type and nuke attack type.

5. Conclusions and future work

In this paper, we propose a method to detect shilling attacks based on SVM and target item analysis method. Experiments on the MovieLens Dataset demonstrate the effectiveness of SVM-TIA in detecting shilling

attacks. The limitation for SVM based detection method is that class unbalance problem exists in SVM classifier. Borderline-SMOTE method is used to alleviate the class unbalance problem. A SVM classifier combined with Borderline-SMOTE method using profile extraction attributes is proposed. On the other side, target item analysis method is used to reduce false positive rate of the detection result. The false positive of detecting result using SVM-TIA is high when the quantity of attack profiles is small; the precision rate can be promoted by employing target item analysis method. Even recall of SVM-TIA method is not as good as other methods, but the precision of SVM-TIA is always high, which means the false positive in the detecting result is low. There is no big difference in the detecting result between push attack type and nuke attack type.

In our future work, time factor is an important factor for detecting attack behavior, time factor should also be considered in detection algorithms. and we will enhance the function of target item analysis method to detect more complex attack models. On the other side, the proposed method should be tested in online systems.

Acknowledgments

We would like to declare that part of the content has been published in KSEM 2015 (International Conference on Knowledge Science, Engineering and Management) and ACM SIGIR (The 37th Annual ACM SIGIR CONFERENCE). The research reported in this paper has been supported by the National Science Foundations under grant No.61379158, and the Ph.D. Programs Foundation of Ministry of Education under grant No. 20120191110028, 2014M560704 and the Fundamental Research Funds for the Central Universities under Grant 106112014CDJZR095502.

References

- [1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *Knowledge and Data Engineering, IEEE Transactions on* 17 (6) (2005) 734–749.
- [2] T. Zhou, M. Medo, G. Cimini, Z.-K. Zhang, Y.-C. Zhang, Emergence of scale-free leadership structure in social recommender systems, *PLoS One* 6 (7) (2011) e20648.

- [3] R. Burke, B. Mobasher, C. Williams, R. Bhaumik, Classification features for attack detection in collaborative recommender systems, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 542–547.
- [4] S. K. Lam, J. Riedl, Shilling recommender systems for fun and profit, in: Proceedings of the 13th International Conference on World Wide Web, ACM, 2004, pp. 393–402.
- [5] B. Mehta, T. Hofmann, W. Nejdl, Robust collaborative filtering, in: Proceedings of the 2007 ACM Conference on Recommender systems, ACM, 2007, pp. 49–56.
- [6] G. Feng, Robust recommendation algorithm based on metadata fusion., International Journal of U-& E-Service, Science & Technology 7 (6).
- [7] R. Bhaumik, C. Williams, B. Mobasher, R. Burke, Securing collaborative filtering against malicious attacks through anomaly detection, in: Proceedings of the 4th Workshop on Intelligent Techniques for Web Personalization (ITWP’06), 2006.
- [8] B. Mehta, W. Nejdl, Attack resistant collaborative filtering, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2008, pp. 75–82.
- [9] R. Burke, B. Mobasher, C. Williams, R. Bhaumik, Detecting profile injection attacks in collaborative recommender systems, in: The 8th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, IEEE, 2006, pp. 23–23.
- [10] R. Burke, B. Mobasher, R. Bhaumik, C. Williams, Segment-based injection attacks against collaborative filtering recommender systems, in: Fifth IEEE International Conference on Data Mining, IEEE, 2005, pp. 4–pp.
- [11] B. Mobasher, R. Burke, R. Bhaumik, C. Williams, Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness, ACM Transactions on Internet Technology (TOIT) 7 (4) (2007) 23.

- [12] Z. Cheng, N. Hurley, Effective diverse and obfuscated attacks on model-based recommender systems, in: *Proceedings of the third ACM Conference on Recommender systems*, ACM, 2009, pp. 141–148.
- [13] W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García, F. García-Sánchez, Social knowledge-based recommender system. application to the movies domain, *Expert Systems with Applications* 39 (12) (2012) 10990–11000.
- [14] N. Zheng, Q. Li, A recommender system based on tag and time information for social tagging systems, *Expert Systems with Applications* 38 (4) (2011) 4575–4587.
- [15] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, C. D. Perkasa, A novel intrusion detection system based on hierarchical clustering and support vector machines, *Expert Systems with Applications* 38 (1) (2011) 306–313.
- [16] C. A. Williams, B. Mobasher, R. Burke, Defending recommender systems: detection of profile injection attacks, *Service Oriented Computing and Applications* 1 (3) (2007) 157–170.
- [17] F. Zhang, Q. Zhou, Hht-svm: An online method for detecting profile injection attacks in collaborative recommender systems, *Knowledge-Based Systems* 65 (2014) 96–105.
- [18] M. Grčar, B. Fortuna, D. Mladenič, M. Grobelnik, knn versus svm in the collaborative filtering framework, in: *Data Science and Classification*, Springer, 2006, pp. 251–260.
- [19] G. Wang, J. Hao, J. Ma, L. Huang, A new approach to intrusion detection using artificial neural networks and fuzzy clustering, *Expert Systems with Applications* 37 (9) (2010) 6225–6232.
- [20] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, K. Dai, An efficient intrusion detection system based on support vector machines and gradually feature removal method, *Expert Systems with Applications* 39 (1) (2012) 424–430.

- [21] N. Hurley, Z. Cheng, M. Zhang, Statistical attack detection, in: Proceedings of the third ACM Conference on Recommender Systems, ACM, 2009, pp. 149–156.
- [22] L. Fu, D. H.-L. Goh, S. S.-B. Foo, J.-C. Na, Collaborative querying through a hybrid query clustering approach, in: Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access, Springer, 2003, pp. 111–122.
- [23] S. Zhang, A. Chakrabarti, J. Ford, F. Makedon, Attack detection in time series for recommender systems, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 809–814.
- [24] B. Mehta, W. Nejdl, Unsupervised strategies for shilling detection and robust collaborative filtering, *User Modeling and User-Adapted Interaction* 19 (1-2) (2009) 65–97.
- [25] J. Cao, Z. Wu, B. Mao, Y. Zhang, Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system, *World Wide Web* 16 (5-6) (2013) 729–748.
- [26] Z. Wu, J. Wu, J. Cao, D. Tao, 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, ACM, 2012, pp. 985–993.
- [27] C. Williams, B. Mobasher, Profile injection attack detection for securing collaborative recommender systems, Tech. rep., Technical report, DePaul University (2006).
- [28] J.-S. Lee, D. Zhu, Shilling attack detection-a new approach for a trustworthy recommender system, *INFORMS Journal on Computing* 24 (1) (2012) 117–131.
- [29] S. Ch, S. Sohani, D. Kumar, A. Malik, B. Chahar, A. Nema, B. K. Panigrahi, R. Dhiman, A support vector machine-firefly algorithm based forecasting model to determine malaria transmission, *Neurocomputing* 129 (2014) 279–288.

- [30] H. Zhang, A. C. Berg, M. Maire, J. Malik, Svm-knn: Discriminative nearest neighbor classification for visual category recognition, in: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 2, IEEE, 2006, pp. 2126–2136.
- [31] F. Zhang, B. Wang, Approach of detecting user profile attacks based on svm and ough set theory, Journal of Chinese Computer Systems 35 (1) (2014) 108–113.
- [32] S. Lv, W. Wang, A shilling attacks detection method of recommender systems based on hybrid strategies, Computer Engineering & Science 35 (8) (2013) 174–179.
- [33] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (1) (2002) 321–357.
- [34] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: a new over-sampling method in imbalanced data sets learning, in: Advances in intelligent computing, Springer, 2005, pp. 878–887.
- [35] W. Zhou, Y. S. Koh, J. Wen, S. Alam, G. Dobbie, 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, ACM, 2014, pp. 955–958.
- [36] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology (TIST) 2 (3) (2011) 27.