Survey on shilling attacks and their detection algorithms in Recommender Systems

Miriam El Khoury Badran  
Computer Science Department  
Notre Dame University  
Zouk Mosbeh, Lebanon  
mabadran@ndu.edu.lb

Wissam Jurdi  
Computer Science Department  
Notre Dame University  
Deir el Qamar, Lebanon  
wsjurdi@ndu.edu.lb

Jacques Bou Abdo  
Computer Science Department  
Notre Dame University  
Deir el Qamar, Lebanon  
jbouabdo@ndu.edu.lb

Abstract—Shilling attacks are malicious noise that target recommender systems to modify the recommendation process to promote or demote certain items. These attacks put the user’s trust in the system at risk. Therefore, many detection techniques are developed to identify those anomalies that are caused by attackers. The following paper is the first to survey all shilling attacks on recommender systems and their detection algorithms.

Keywords—Recommender Systems, shilling attacks

I. INTRODUCTION

We live in the information age where societies, governments, enterprises and individual strive for information. Day after day, the amount of data is rising exponentially with the integration of technology in every aspect of the human life. People spend most of their waking hours in front of screens, surfing the net, on social media, receiving and most importantly giving a widespread range of information. The use of technology, increase in communication, globalization, deregulation and downsizing are main factors for information overload [1].

With all the data in the cloud, at the service of the public, the problem of information overload rises. Information overload is “the dilemma of having more information than one can assimilate, or being burdened with a large supply of information” [2]. The human brain is receiving more information that it is able to process in a certain timeframe: the conversion of data into knowledge is challenging due to the speed and the quantity of data received [3]. People use online services to save time and effort, yet, with the many choices and possibilities that they have, the task of taking decisions is made harder. The problems caused by information overload are several: psychological stress, wrong decisions, disregarding relevant information, etc. When users are faced with high levels of information, dysfunctional consequences, such as confusion and cognitive fatigue, result due to the overload of their limited processing capacity. Victims of information overload require longer time for consumption which impacts the enterprises of course. This time factor can lead to distraction, time waste or the reduction of decision quality due to a certain time-limit to make the choice. The user usually loses the ability to separate valuable information from what is unnecessary, basing his/her decision on wrong information: this will result in a low decision quality [2].

One widespread solution for the information overload problem is the use of recommender systems since they are adaptable to different environments. In order to reduce the consumer’s cognitive efforts and increase the quality of his/her decisions (and consequently, his/her satisfaction), decision-support systems (DSS) are implemented in the shape of recommendations. This recommender system aims at filtering the wide information to match the user’s taste and support him in efficient decision making. These DSS alleviate the processing effort of the user, thus protecting him from information overload, while maintaining an acceptable level of choice accuracy, through the use of heuristics [3].

The system guides users that lack a detailed product domain knowledge to useful or relevant items in a personalized way among a huge space of possibilities. In return to this “favor” that the system is doing to users, users build up a trust towards it.

This accuracy is at risk from the existence of noise. Recommender systems, like any other system, are prone to error, especially that it involves human activity that is, itself prone to error. Two types of noise exist:

A. Natural Noise

It is linked to the human error whether it is unintentional or due to moodiness. It is witnessed when the user makes an unusual rating, i.e. incompatible with his/her profile and previous activity. It is hard to detect since everything related to the human psychology is hard to measure: how can the system say that the user is moody or committed an error in this specific rating? Studies are done to solve this issue, detect, and remove this type of noise.

B. Malicious noise

This type of noise is intentionally inserted by attackers. That is why this category is also known for shilling attacks. This deliberate insertion of noise has for goal the manipulation of the recommendations. The aim can be to increase or increase the recommendation of a targeted item. It is usually done by the insertion of fake profiles, according to certain patterns, which is why detecting and correcting malicious noise is somewhat easier. Many techniques and algorithms exist.

This paper focuses on malicious noise caused by shilling attacks. It presents a survey of different attacks and detection algorithms. It is organized as follows. Section II provides the terminology used in the attack. Section III introduces different attack models and compares them, while Section IV presents an overview of a multitude of shilling noise detection techniques. Finally, the paper is concluded in Section V.

II. RECOMMENDATION SYSTEM ISSUES

In order to generate accurate recommendations, the majority of collaborative filtering methods employ a user-item matrix. This user-item matrix comprises the profiles of the users and its integrity is essential for the generation of meaningful recommendations. However, this task is hard and many issues can arise. These issues can be classified into...
three categories: cold start, data sparseness and shilling attacks (or profile injection attacks). [4] The first issue consists of the problem that arises when new items or new users are introduced to the system. Data sparseness happens when ratings are sparse. The third issue is what concerns us the most in our studies.

![Fig. 1. Recommender systems as solution for information overload.](image1)

Profile injection attacks, like its name indicates, consist of the introduction of biased profiles into the system. These attacking profiles consist of biased ratings targeting a particular item. The first type of attacks is the pushing attack where ratings for the attacked items are extremely high. On the other hand, nuke attacks oppositely are formed of very low ratings. The main goal is to shift the predicted ratings of this item, affecting consequently the overall recommendations. The problem is, if the attacks take their way frequently, the users' trust towards the recommender system victim to these attacks will be reduced, resulting in a reduction in the use of the system. [4] Many shilling attacks were marked in the past. Two know attacks at least on Amazon.com were published [5] [6].

In the attack profile, ratings can be divided into four sets of items: a selected item, a target, a filler item set and unrated items [7].

![Fig. 2. Structure of attack profile](image2)

Through the researches, and based on assumptions about the attacker’s purpose and knowledge, a multitude of attack models have been established. The main differences between different attack types are: the way selected items are identified, the proportion of filler items which determines the attack profile size since the selected item set is small and how the ratings are assigned to items. We can count many popular models [10]:

**A. Random attack model**

The random attack is the basic, simplest, and less effective of them all. It generates profiles where all items to rate, except the target item, and their ratings are chosen randomly.

**B. Average attack model**

The profiles generated by the average attack share the tendencies of the system’s users. This is possible by drawing the created malicious profile’s ratings from the rating distribution that is associated with every item. This attack requires that the attacker has a somewhat complete and detailed knowledge of the data set on which the recommender system is based: its effectiveness is closely related to this.

**C. Probe attack model**

Some recommender systems are configured to produce predicted ratings for the items. Attackers relying on the probe attack use this feature to fill out their profile. That way, their ratings will be naturally similar to other users in the system, especially their neighbors.

**D. Bandwagon attack model**

The bandwagon attack is also known as the popular attack; the attacker determines the popularity of the item independently from the system. After pinning the well-known items, he will associate the attacked items with a handful of them. This type of attack achieves many of the average’s attack benefits without requiring the knowledge of the data set underlying the recommender algorithm.

**E. Segment attack model**

The segment attack targets a specific set of users.
These previously described attacks can be adapted to decrease the rating of a certain item rather than increasing it. However, reverse-bandwagon and love/hate attacks are specifically designed to “nuke” a target item [10].

1) Reverse-bandwagon attack model

Like its name indicates, it is a variation of the bandwagon attack. In this attack a low rating is assigned to both the target items and the other items used to create the malicious profile. These chosen items tend to be poorly rated by a large number of users. Associating the target item with widely disliked products increases the probability that the system will generate a very low predicted rating for the attacked item.

2) Love/hate attack model

In the love/hate attacks no previous knowledge is required. The attack profile consists of a target item with minimum rating while fillers get the maximum rating. This simple way of generating attack profiles has surprisingly a high effectiveness.

These attacks are considered by Bryan et al. [11] standard attacks. They mention another category called obfuscated attacks. Attackers try to obfuscate their attack signature to decrease their visibility from detection algorithms. Multiple models which can be applied on the random attack, the average and the bandwagon attack are proposed.

a) Noise injection

It is the addition of noise to the ratings. It is done according to a standard normal distribution multiplied by a constant. The degree of obfuscation of this method is governed by this constant.

b) User shifting

It involves shifting by a constant amount the ratings of a subset of items inside the profile. Shifts have a positive and a negative form depending on the amount of shift that is governed by a standard normally distributed number.

c) Target shifting

It is a branch of user shifting where the target item’s rating is shifted. Usually in attack profiles, target items receive either the maximum rating or the minimum whether it is a push or a nuke attack respectively: this is a particular signature common for all attack profiles. Target shifting consists of setting the rating of the target item to maximum - 1 or minimum + 1.

Yang et al. [12] and Bhauamik et al. [13] would add to these obfuscation methods the AOP (Average Over Popular Items) attack, an effective yet simple strategy to obfuscate average attacks. It is based on choosing filler items with equal probability from the most popular items, the top x%, instead of choosing from the entire set of items.

Bhauamik et al. also adds the mixed attack. As its name shows it consists of mixing the same amount of segment, bandwagon, random and average attack profiles. A spam detection technique should perform efficiently against mixed attacks.

Yang et al. [14] adds two additional types of attacks: the power item and the power user attacks. Power items in the collaborative filtering recommender systems influence the largest group of items. The influence is the ability of the item to affect the predictions that the system offers for other users.

In a similar way, power users influence the widest group of users. The Power Item Attack (PIA) can itself be divided into three kinds. In PIA-AS, the top-N items with highest Aggregate Similarity (AS) are selected to be the set of power items. The PIA-ID is based on In-Degree centrality. The third type of PIA is the PIA-NR where items are chosen to be power items if they have the highest number of users. Similarly, the Power-User Attacks (PUA) consist of three kinds: PUA-AS, PUA-ID and PUA-NR.

IV. DETECTION ALGORITHMS

To fight against the shilling attacks detection algorithms have been proposed. Their aim is to detect the attack profiles and remove them from the recommender system before the generation of recommendations. There are two types of approaches: profile-based and item-based. Item based algorithms aim at detecting the attacked items. To do so, they use the item vectors within the user-item matrix. While the profile based algorithms screen the rating vectors of every user in the user-item matrix in order to identify abnormal profiles [4].

Furthermore, profile-based algorithms can be divided into two more categories. The first approach is the classification-based approach. It exploits a classification model that was built previously to predict whether a new profile is a genuine user or an attacker. This method requires a balanced number of negative and positive cases. In the real world, collecting a balanced number of cases is difficult: negative cases are less frequent and can be unnoticed [4].

The second approach is the Principal Component Analysis based method. It transforms the user-item matrix to a hyper-plane; that way, every profile can be represented using three principal components. If a profile is close to the origin of the hyper-plane, then, it will be marked as an attacker. The PCA method is unable to handle missing values in the user-item matrix. The missing data is computed from values around it and estimated. However, due to the sparsity of the matrix, since users might not know all offered items, the estimation’s quality is questionable [4].

In order to detect items under stack, the item-based approach picks up items whose preferences are beyond boundaries. This approach relies on the Statistical Process Control, SPC technique, in order to detect anomaly ratings for every item. The boundary is formed of the upper and the lower limits: two horizontal lines estimated with the use of historical ratings. This technique unfortunately alerts of the presence of the attack, but it is unable to spot the culpable rater [4].

According to Zhou et al. [7] attack detection algorithms can be divided into three categories: unsupervised, supervised and semi-supervised. In the supervised category, the detection techniques are modeled as classification problem. Three main classification algorithms are used: kNN-based, SVM-based and C4.5-based. They aim to improve the robustness of the system and enhance its accuracy. To be able to achieve this, they need a large number of labeled users, to make sure that the number of attack and genuine profiles are balanced to be able to train the classifiers. The earliest techniques were less accurate because they exploited the signatures of the attack profiles, looking at individual users and ignoring the combined impact of these malicious users. Some of those techniques used the
decision tree methods, the nearest neighbor classifiers, Bayes classifiers, rule based classifiers or Neural Network classifiers.

The issues faced in the supervised approaches are addressed in the unsupervised detection approaches by training on an unlabeled dataset. Compared the previous approaches, these methods have less computational effort involved in them, improve the detection accuracy and facilitate online learning. Some of the techniques use association rules methods, clustering and statistical approaches.

In their work, Williams et al. [15] prove that a supervised classification learning approach improves the profile injection attacks detection technique by adding robustness significantly. A crucial factor for the protection of the recommender system is the choice of the classifier algorithm. This algorithm, if combined with a robust classification algorithm like SVM can increase the robustness even more. Their technique utilizes a three branched strategy combining first, similarity to reverse engineered attacks, second, attributes for general ratings anomalies detection and, finally, target concentration. This model above mentioned includes many dimensions: critical mass information, time series... However, its limitations are caused by using the profiles in isolation (i.e. without identifying common items under attack) and not incorporating temporal properties.

Zhang et al. [16] proposed a meta-learning-based approach with the aim of improving the precision of detecting profile injection attacks. The model contains a base-level and a meta-level training that both use SVM as learning algorithm. The base-level training set is generated through the use of the attack profiles and the rating database. The diversities of the base-classifiers reduce in an effective way the correlation of the misclassifications but also improves the predictive capability of the meta-level; after all, the output of the base-level is integrated as the input of the meta-level that will generate the final result. The underlying concept of the meta-learning approach is relearning the existing knowledge in order to boost the overall predictive effectiveness. This proposed algorithm holds a high recall and improves the precision effectively.

Since the number of attackers is generally way less than genuine users, the problem of imbalanced classification arises, affecting the supervised learning based detection (SVM, kNN, etc.) that are weak in handling this type of issues. Boosting proved to be efficient in this scenario since weak learners are fitted iteratively to the training data. The emphasis on the observations modeled weakly by these weak earners is gradually increased by using appropriate methods. AdaBoost presented in [12] apply weights to the different observations to emphasize the poorly modelled samples and iteratively strengthening the correction of misclassifications. AdaBoost’s performance can be improved to be consistent, creating the re-scale AdaBoost (RAdaBoost).

Chung et al. [4] suggested an unsupervised algorithm based on Beta probability distribution to detect as many attackers as possible while keeping the genuine users as intact as possible. The proposed algorithm is immune from the issue of data skew, since it does not require negative training data, but also immune from the problem of null values. Additionally, it is able to detect multiple-target attacks launched simultaneously, which is not an option in many of the other detection algorithms. This method called Beta-Protection $\beta P$ aims on having both, a low alarm rate and high detection rate. It is composed of three phases based on the characteristics of the Beta distribution. The first phase consists of marking users with extremely low numbers of rated items as attackers. After all, attackers have to give scores to the filler items that can be few in some attacking profiles. The Beta-distribution characteristic is applied on the distribution of the items that are being rated following a procedure called rating_extremely_few_items. The second and third phase are linked respectively to raters with extreme scores positively and negatively. The characteristic is applied on the scores of the raters: rating_extreme_score. If a rater fits in any of the three previously mentioned categories, he/she will be judged as abnormal.

This method nevertheless presents some limitations that are directly linked to the definition of Beta distribution itself. An assumption is made: a single prior distribution can be used in order to describe the possible values that are observed in every cell of the user-item matrix for genuine users. In consequence, the method will score a high false alarm rate and a low detection rate in case the genuine users present different grading behavior significantly. A second limitation is derived from the size of the attack: if the attack was large, it can distort prior distribution, causing the generation of poor results.

Bryan et al.[11] propose the UnRAP algorithm that uses a sparse variation of the Hv-score metric. This metric was developed in the area of gene expression data analysis and performs well in separating genuine profiles from attackers. In comparison with the standard metrics, it can be used to improve the performance of detection methods. UnRAP that is built on the strength of the Hv-score metric shows a good performance over many standard and obfuscated attacks. Furthermore, the unsupervised nature of Unrap means that it is probably capable of detecting future novel attacks which is not even a considerable option with supervised approaches that fit the current attack models.

Bhaumik et al. [13] present an unsupervised attribute-based k-means clustering approach to identify attack profiles despite the type of the attack. This algorithm detects spam users with fewer misclassified authentic users and higher degree of accuracy, and is mostly effective against segment attacks. This technique is based on the hypothesis that the attack profiles are less in numbers and will consequently dominate one clusters due to their similarity. The algorithm generates profiles for each user in the database, and then, using the k-means clustering algorithm, the profiles are partitioned into two groups of users sharing similarities. According to the assumption that the smaller clusters corresponds typically to the attack profiles, this smaller cluster will be marked as anomalous and all profiles in this cluster will be given low preference when recommendations are generated.

Bilge et al. [17] rely also on the fact that shilling profiles share high similarity among themselves. For that reason, the correlation-based clustering algorithms are more prone heuristically to group attack profiles together. Their study is based on bisecting k-means clustering approach to gather the attack profiles in a leaf node of a binary decision tree. This method is extremely successful against average segments and bandwagon attack profiles.
Yang et al. [18] present an unsupervised detection method consisting of two phases. In the first phase, a density-based clustering method based on some selected features is exploited in order to determine suspected users. The second phase aims to find out suspicious items through the use of some selected features of item in order to spot further the attackers raised on the first phases result. That way the attackers can be detected. Features are selected effectively based on the adaptive structure learning that takes advantage of adaptive global and local structure learning.

Zhang et al. [19] propose the UD-HMM, an unsupervised approach for detecting chilling attacks based on hidden Markov model and hierarchical clustering. The hidden Markov model is first used in order to model user's history rating behaviors. Then, by analyzing the user's preference sequence and the difference between malicious users and authentic ones in the rating behaviors, each user's suspicious degree is calculated. Afterwards, according to this user's suspicious degree, users are grouped using the hierarchical clustering method. As a result, the set of attack users is obtained. This detection technique does not require the knowledge of the attack size in advance, nor does it require to label candidate attack users manually. However, it performed poorly when detecting collusive spammers.

Xia et al. [20] propose a novel dynamic time interval segmentation technique based item anomaly detection approach to detect shilling attacks. This segmentation technique can detect attacks despite their type, with linear time complexity, and is effective in a large range of sizes with strong robustness. It is first inspired by the common features of attacks. The first common feature according to this study is that the attack schedules are short bearing in mind leveraging the costs and benefits. Second of all, the scales of the attack are large enough so that the statistical characteristics of the attacked target item are able to change with the attackers' intention. The final common feature is that the attackers like to mark their target with ratings considerably different from the mean value that is the profile characteristics of the item. The detection algorithm uses the rate of change of skewness quantities for time interval segmentation. This technique can cluster the consecutive attack ratings of the same type together successfully, create the intervals with multiple scales and identify the intervals that are subject to attack in an effective way.

Zhang et al. [21] base their detection technique on a common characteristic of the shilling attack: over the period of the attack, they induce change in the distribution of the ratings of items, particularly the target item. Therefore, according to their approach, detecting a wide set of attacks is made possible through the examination of the rating distribution for every item over a time series. This approach has two main advantages. The first one is that it allows the detection of attacks that were difficult to isolate when using methods where every attack profile is studied alone. The second benefit is that it can reveal attacks that were previously unknown or undefined. It presents some disadvantages, like the fact that ratings to a certain item in a large-scale recommender system might have a seasonality or a trend over time.

Away from profile-based detection approaches, Bhaumik et al. [22] present an item-based detection approach that aims at the identification of what items might be victims of an attack based on the item’s rating activity. Instead of detecting attacker profiles, items with suspicious trends are identified independently from users. Three statistical anomaly detection techniques were investigated. Two SPC (Statistical Process Control) techniques are presented and evaluated: the Confidence Interval control limit technique and the X-bar control limit. The third approach is a time interval detection scheme.

SPC is usually used for long term monitoring of feature values that are related to the process. When a feature of interest is constructed or chosen, an estimation of the distribution of this feature and future observations can be monitored automatically. For the quality monitoring in SPC, control charts are routinely used. Generally, SPC is formed of two phases. The first phase consists of estimating the process parameters. To achieve that, historical events are used. Then, these parameters are employed to detect out-of-control anomalies for recent events. The X-bar chart is one frequently used type of control chart. It plots where the current measurement falls compared to the average value. Upper and lower limits are set: a new item is probably under attack, if the average rating is greater than the upper limit or less than the lower one. In the confidence interval control limit, the upper and lower boundaries correspond respectively to the signal threshold for pus attacks and nuke attacks. If the item falls inside the range, it is not subject to attack; contrarily, if it falls outside, it is marked as suspicious.

A series of observations over time characterize the usual behavior of a recommender system. If an attack occurs, it is crucial to notice it as fast as possible before allowing significant performance degradation to happen. In order to assure that, a continuous monitoring of the system can be done, searching for deviations from past behavior. The time interval detection algorithm detects the time period over which a potential attack is targeting the item.

Moreover, Bhaumik et al confirm that the rating distribution of an item affects the detection performance. Items with low average rating or few ratings can be manipulated easier than items with high average rating or many ratings.

Morid et al. [9] suggest that instead of applying the attack detection methods to the entire user sets, applying them to the influential users would improve the detection performance and the stability of the recommender system. In order to detect the influential user, the loo (leave-one-out) strategy is employed. The target profile is removed from the system and the impact of his absence is observed. The more users are affected by his/her absence, the more influential he/she is. Once located, testes to detect if this user is a malicious attacker are made only on him/her.

V. CONCLUSION

In order to solve the problem of information overload, recommender systems are implemented to help the user narrow down the infinite choices that he/she has. The user in return build a trust towards the system. This trust is endangered by the existence of noise in his two types: natural noise that is cause unintentionally by the user himself due to a simple error or moodiness, and shilling noise cause my malicious profiles. This paper presents a survey of attacks that are done against recommender systems and the detection algorithms that allow the systems to defend themselves.
REFERENCES


