Identifying Abnormal Behavior of Users in Recommender Systems

Homa Tafakori, Soheila Karbasi*, Mehdi Yaghoubi
Department of Computer Engineering
Golestan University, Gorgan, Iran
homa_tafakori@yahoo.com, s.karbasi@gu.ac.ir, m.yaghoubi@gu.ac.ir

Received:2019/08/11 Revised:2019/09/07 Accepted:2019/12/23

***Abstract*— Nowadays, we deal with a large volume of information that we may have wrong choices without appropriate guidance. To this end, recommender systems are proposed which are a type of information filtering system that acts as a filter and displays information that is useful and close to the user's interests. They reduce the volume of the retrieved information and help users to select relevant products from millions of choices available on the internet. However, since these systems use explicitly and implicitly collected information about the user's interests for different items to predict the user's favorite items, the adversaries due to their openness nature might attack them. Therefore, identifying them is essential to improve the quality of the recommendations. For this purpose, in this paper, a method based on two criteria of a maximum number of users with the equal length and the degree of novelty of their profiles is presented and finally, the DBSCAN clustering algorithm is used to distinguish genuine users from fake users. In order to improve the DBSCAN algorithm, we proposed a new method to determine the values of Eps and MinPts automatically.**
The results of the proposed method are compared with a new comparative study on shilling detection methods for trustworthy recommendations, which shows that the proposed method independent of the type of attack can identify fake users in most cases with accuracy close to 1.

Keywords*— *Recommender Systems; Shilling Attack; Abnormal Behavior; Novelty Degree Of User's Profile

1. INTRODUCTION

A large and growing volume of the information on the web and internet has challenged many users in selecting the required information and products. This issue has attracted the researchers to find solutions to handle the information overload. There are two approaches so far. The first approach is to use information restoration and information filtering. The main limitation of these two concepts is that they cannot distinguish items of high quality and low quality. The mentioned problem motivated the scientists to present a second approach known as recommender systems [1].

Recommendation systems can extract patterns and offer products to users using a variety of information from users' behavior, such as the number of customer purchases, types of user-friendly goods, and so forth. However, researches show that although this information makes the systems provide good services, they can be exploited, and so identifying the attackers and attacks that is called shilling attacks are essential. An attack against a collaborative recommender system consists of a set of attack profiles. An attack profile consists of an m-dimensional vector of ratings, where m is the total number of items in the system.

In these attacks, malicious users are inserted into the existing dataset in order to influence the results of recommender systems. Mostly, product sellers or developers who aim to promote their own products or demote their competitor's products generate these attacks.

Based on different assumptions, the attack models can be divided into different categories such as standard or obfuscated attacks and push or nuke attacks. Push attacks try to make one or more target items recommended to more users, while nuke attacks try to cause them less likely to be recommended. Therefore, the most important challenge in the recommender systems is privacy and identifying fake users. Because these users alter the recommendation lists and reduce the accuracy of the recommendations [2].

In recent years, a large number of studies have been carried out on detection of shilling attacks, but most of them are dependent on the type of attacks and some related parameters.

In this paper, to address the above challenges, we propose an approach based on two features of the maximum number of users with equal length and novelty degree of their profile that is independent of the type of attacks.

In fact, the length of profiles is equal for each attack that includes multiple attack profiles. This makes the number of fake users with equal length of profile be more than genuine users because the majority of genuine users in the real world rate only a small number of items. Therefore, first we identify the maximum number of genuine users with equal length and extract suspicious users. Then we use the concept of novelty items to investigate the discrepancies between items in user profiles that have not been used so far to distinguish genuine users misclassified as attack profile from fake users.

We summarize our main contributions as follows:

- We generate attack profiles based on different attack models, including Random, Average, Bandwagon attack model with different values of attack size (2%, 3%, 5%, 10% and 20%) and filler size (1%, 3%, 5%, 7%, 10% and 15%). After that, the attacks data are respectively inserted into the pure datasets to construct the finally experimental datasets.
- We propose a feature based on attack profile for extract suspicious users (the maximum number of genuine users with equal length).
- We use the concept of novelty items to investigate the inconsistencies between items in user profiles that have not been used to detect shilling attacks so far to distinguish genuine users misclassified as attack profiles from suspicious users.
- We propose a new method to determine the input parameter values of the DBSCAN algorithm, which are neighborhood radius Eps and a minimum number of points MinPts, automatically.
- We conduct experiments on the MovieLens 1M dataset and the MovieLens 100K dataset and compare the performance of proposed method with the methods presented in the paper [3].

The rest of paper is organized as follows: the next section introduces preliminaries and reviews attack types. In section 3, we discuss details about our proposed approach, section 4 deals with the experiments performed and their analysis and finally the conclusion is presented in section 5.

2. Literature Review

2.1. Challenges of recommendation systems

Recommender systems analyze previous behavior of the users and collect information explicitly or implicitly about users' interest in different items to predict how the users think so that it can identify and recommend the most suitable products to the users [1, 4]. Figure 1 shows the architecture of recommender systems. These systems employ content-based filtering, collaborative filtering, and hybrid filtering to achieve their objective, which is finding the useful information fast and appropriate to the users' interests [5].

- Content-based Filtering: this technique compares attributes of items with users' profiles that reflect the properties of their favorite items, and ultimately recommends items that are close to the users' interests. Figure 2 shows the content-based filtering algorithm [4].
- Collaborative Filtering: is the most well-known technique of recommender systems that determines similar users or items to predict the recommendations. That is, considering users' ratings of items; it considers users with similar preferences as a group, and offers the most-liked items to the specified user.
- Hybrid technique: this technique is a combination of two or more techniques to obtain better performance. In most cases, it is a combination of model-based collaborative filtering and memory-based collaborative filtering to overcome the limitations and weaknesses of collaborative filtering algorithms such as sparsity and so on.

Despite advantages like related recommendation, new and various items, increasing satisfaction and reducing the searching time of user to find the items of interest, these systems have the following disadvantages [2]:

- Cold Start: this problem is associated with new users. Due to that, a new user has not rated any items, so it is difficult to identify his/her interests. Thus, accuracy of the recommendations reduce significantly.
- Scalability: collaborative filtering algorithm based on users or items, should investigate the whole database of the recommendation system to calculate similarities. If number of users or items increases, computational complexity is increased, which means the efficiency of system reduces.
- Gray-Sheep Users: indicates the users, which their ideas and comments do not match with any other group of users. Therefore, these users cannot benefit from advantages of collaborative filtering.
- Sparsity: this problem usually occurs when number of items is larger than number of users and users are not desired to rate the items. Thus, most elements of the item-user matrix would be empty and accuracy of the recommendations reduces.
- Synonymy: this problem occurs if there are similar items with different names in the system. Because, the system would not be able to detect their relationship and system efficiency decreases significantly. For instance, although the terms "kids film" and "children film" seem different but they are the same. While, most memory-based collaborative filtering systems cannot detect them and calculate similarity.
- Privacy: the recommender systems try to guess how the user thinks by analyzing previous behavior of the users and collecting information and detect the most proper items to his/her interest. Although these systems apply the personal information of their users to offer better services, but due to their openness nature, this information might be misused by the jobbers and create security problems (generating fake recommendations) and affect quality of the. recommendations

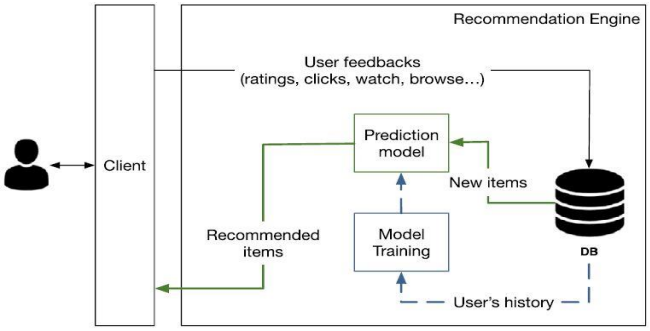

Fig. 1. Architecture of the recommender systems

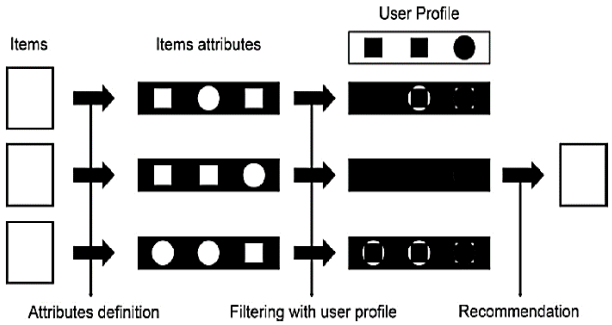

Fig. 2. Content-based filtering algorithm

2-2. Attack models

Today, E-commerce websites employ collaborative recommendation systems to offer recommendations to their customers to increase their sales and profit. However, as these systems have become popular, they have been more important to various attacks as shilling attacks or fake profile injection. In these attacks, the attackers try to affect the performance of the system and dissatisfy the users through fake rating and changing list of the recommendations based on their objectives. There are two main push and nuke attacks which aim to increase and decrease the popularity of the target items [7, 8].

An attack model is an approach for attackers to construct a set of attack profiles to alter recommendation lists of a set of target items. The general form of an attack profile is shown in Figure 3. Rating in an attack profile can be divided into four sets of items:

- Target item: for each attack profile, there is usually a single target item that depending on the attack type will be given either the maximum or minimum rating value.
- Selected items: it is a small group of items for special treatment during the attack. These are randomly chosen and not necessary for some attack models.
- Filler items: filler items in attack profiles are a set of randomly selected items that are assigned ratings according to the proportion of the attack and make the profiles look similar to genuine profiles and be harder to detect.

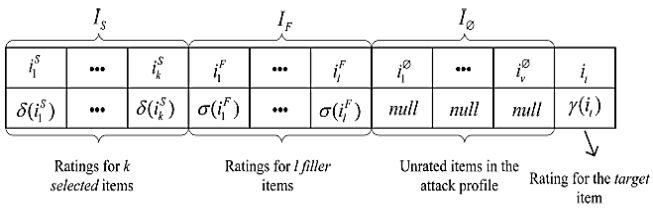

Fig. 3. The general form of a single-target attack profile

- Unrated items: the set of items that are not rated by the attacker.

Inserting these profiles in the database of the recommender system is manually or automatically. Based on the attacker's knowledge and purpose, a number of attack models have been identified. We will introduce three most well-known attack models [8]:

Random Attack:
This attack generates profiles in which the items and their ratings are chosen randomly based on the overall distribution of user ratings in the database, expect for the target item. This attack is very simple to implement, but it has limited effectiveness.

Average Attack:
In the average attack, each assigned rating for a filler item corresponds to the mean rating for that item, across the users in the database who have rated it. This is a very effective attack; however, it requires knowledge about the system. Table 1, shows the attack profile for the random and average attacks.

Bandwagon Attack:

This attack is the extended version of the random attack and its main idea is to use a set of popular items. Items which are more liked by the people and they are rated more often; for example, blockbuster movies in a movie recommender system. Using these items in the fake profile and giving them the highest rate, increases the similarity of the fake profiles and real ones.
Compared to the average attack, this attack requires less knowledge and its implementation is easy because the determination of the popular items is simple. However, it is efficient as the average attack and it does not affect the item-based algorithm.

For Movie Lens 100K and Movie Lens 1M datasets, popular items are those with more than 300 rates and 506 rates respectively [9, 10, 11].

Any of the above attacks, which specifies how the user rates the items in its fake profile, is determined using the following two parameters [8, 12].

- Attack Size: this parameter specifies the ratio of the number of profiles injected to the system by the attacker and the number of entire genuine users in the recommender system. For example, 5% attack size in a system with 1000 users means that 50 attack profiles are injected into the system. Usually, this parameter is set to 1-15%.
- Filler Size: is the ratio between the number of items rated by an attacker on its attack profile and the number of entire items in the recommender system. For example, 5% filler size means that in the system with 1000 items, the attacker has rated 50 items in its attack profile. Usually, this parameter is set to 1-20%.

Table 2 shows the examples of reported attacks on several websites [13, 14].

2-3. Shilling Attacks detection

A large number of studies have been carried out to improve the accuracy of recommendation systems and reduce the effect of shilling attacks. Major approaches are categorize into two groups: attack detection methods and robust CF algorithms. Detection methods include supervised classification, semi-supervised classification, unsupervised clustering, and time series [3].

Supervised Classification

This type of detectors is feature-based, that is, it first defines a set of metrics and thus transforms each profile (i.e., rating record) into a vector in the feature space, which could be labeled as normal or shilling profile. Then some classification algorithms, such as KNN, C4.5, SVM are used to detect shilling attacks in recommender systems.

Among advantages and disadvantages of these methods, the followings can be mentioned [15]:

- Although the overall performance is satisfactory, they are unstable because they depend on training datasets.
- They are unsuccessful in detecting hybrid shilling attacks and successful in detecting attacks with large profiles.

Unsupervised Clustering

Unsupervised clustering methods, which perform learning on unlabeled data to find hidden patterns, automatically can discover clusters based on the similarity and proximity of the samples. These include:

PLSA-based clustering, which each user is assigned to the cluster with the highest probability of membership, then the radius of the clusters is calculated and the cluster with the minimum radius is considered as the cluster of fake users.

TABLE 1. THE STRUCTURE OF RANDOM AND AVERAGE ATTACK

Attack model	I_S	I_F	I_Φ	I_T
Random	Null	Random ratings with a normal distribution around the mean rating value across the whole database	Null	r_{max}/r_{min}
Average	Null	Random ratings with a normal distribution around the mean rating for item i in I_F	Null	r_{max}/r_{min}

TABLE 2. SAMPLE ATTACKS REPORTED ON SEVERAL WEBSITES

Item	*Attack Type*	*Site*	*Year*
Six Steps to a Spiritual Life book (written by the evangelist Pat Robertson)	***Push***	***Amazon***	***2002***
Gay men or sex manual	***Push***	***Amazon***	***2002***
Item with ID:0385519478	***Push***	***Amazon***	***2014***
6% of reviews on sites like Yelp and TripAdvisor	***Push***	***Yelp and TripAdvisor***	-

Graph-based approach, which consists of three stages. Firstly, an undirected user-user graph is constructed from original user profiles. Based on the graph, a graph mining method is employed to estimate the similarity between vertices for creating a reduced graph. Then, similarity analysis is used to distinguish the difference between the vertices in order to rule out a part of genuine users. Finally, the remained genuine users are further filtered out by analyzing target items and the attackers can be detected [16].

Semi-supervised

Most of shilling detection algorithms are supervised learning or unsupervised learning, which ignore the fact that the majority of the users in real recommender systems are unlabeled and only a few of them are labeled, so semi-supervised methods have been proposed.

Semi-supervised methods are a class of learning methods that uses both unlabeled data as well as labeled data to improve learning accuracy. These techniques perform better than other methods. Because they first use labeled training data to construct the model and then use unlabeled data to improve the performance of the constructed model [15].

Time Series

According to the reviews, there are two general features to all of the shilling attacks. First, the rate of the target item is usually maximum or minimum. Second, all attacks occur in a short time interval. Hence, another class of detection methods called time series are presented. In these methods, abnormal intervals are detected through analysis of time series; then, the attacked items are found and the attacks are eliminated. Since in the basic models, the initial time window size was fixed and the detection rate depended on the time window size, a method has been proposed as dynamic partitioning over time series. This method focuses on finding the abnormal items through obtaining important points of the dynamic time series, but due to the variety of the items and the unpredictability of ascending or descending the rates, the false alarm rate of this method is high.

All of the mentioned methods use a set of features for evaluation. These features are divided into three types: general, specific, and inter-profile features.

1. General features: These attributes are based on simple phenomena that the overall signature of attack profiles will be different from the original profiles. This difference comes from the rating behavior of user i.e. rating given to the target item and rating distribution of ratings among the filler items. For example, Rating Deviation from Mean Agreement (RDMA) measures rate deviation for each user profile.

2. Specific features: these attributes are used to detect a specific attack type, not all the attacks. For example, segment and bandwagon attacks assign the highest rate to the selected items for more impact, which increases the difference between the mean of the selected items and the mean rate of other items.

3. Inter-profile features: User profiles with different number of ratings will generate different features. Since an attack profile, whose number of ratings deviates far away from the genuine profile, can be detected easily, especially when the filler size of attack profiles are higher than genuine For example, the ratio of the rated items to total items of the recommender system (FSTI[1]).

3. Proposed Approach

As mentioned previously, recommender systems are one of the most efficient information processing methods and applicable solution to the information overload problem. These systems increase purchases and user satisfaction through offering best recommendations but they have some problems due to their openness nature. One of these problems is shilling attacks or attack profile injection that changes the results of the system and dissatisfies the users.

Therefore, in order to attract users' trust, we must identify and remove the attackers that changes the results of the recommendation list. The flowchart of our approach to detect these users is shown in Figure 4, which includes three steps:

- Calculate the length of user profiles.
- Extract the suspicious users.
- Discriminating the genuine users from the fake users based on the novelty degree of users' profile

Calculate the length of user profiles and extract the suspicious users

In shilling attacks, the attacker inserts a large number of attack profiles, which all attack profiles have the same set of selected items and target items, which make all attack profile inserted into the system have equal length. Therefore, in the first step, the length of the users' profile (number of rates for any user) is calculated and compared with other users and the maximum number of users with equal length is selected to identify suspicious users. However, because in the real world users rate a small fraction of the items, we may have the maximum number of users for the minimum profile length that makes it difficult to identify suspicious users in the attacks with small attack size. Because According to the first step, if there are 25 fake users with length 30 and 30 genuine users with length 20, 30 genuine users are considered as fake users incorrectly.

In order to address this problem, the maximum number of users with equal profile length is calculated and compared with the users with minimum profile length. If the maximum number of users with the same profile length is greater than the number of users with

[1] Filler Size with Total Items

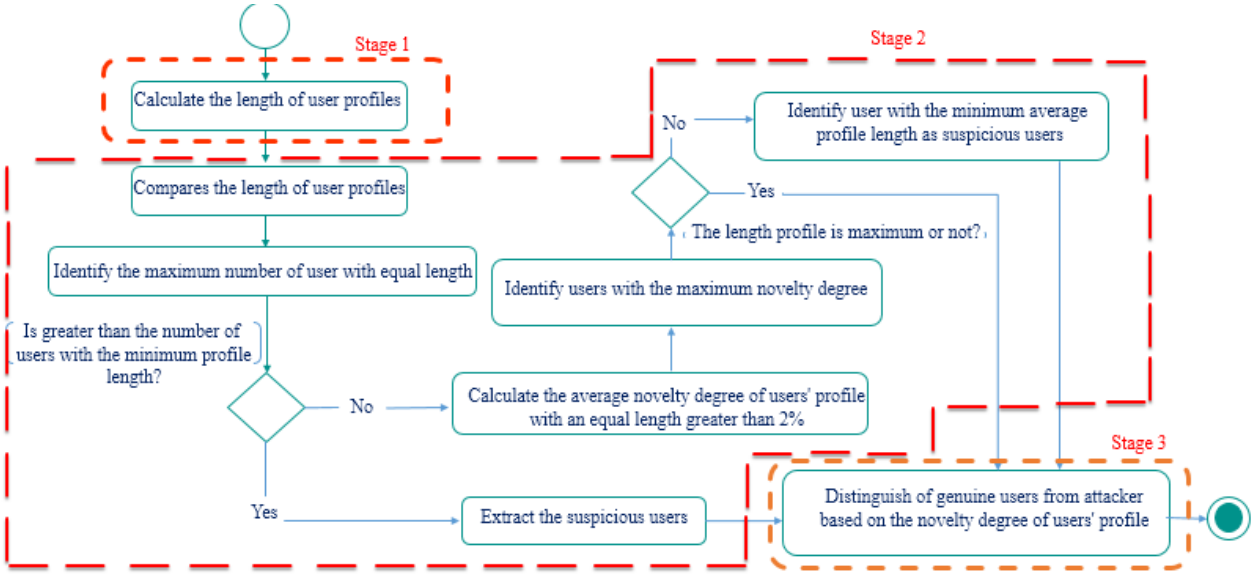

Fig. 4. The flowchart of proposed method

the minimum profile length, the second step is performed.

In order to detect the fake users which are fewer than users with the minimum profile length, since minimum attack size considered in the system is 2%, maximum number of users is detected up to 2% and the average novelty degree of the profiles (described in the second step) is calculated for each profile length. Therefore, users whose profile lengths and average novelty degree profiles are maximized as fake users.

Because as shown in Figures 5 and 6, the filler sizes of more than 90% of genuine users are below 10%. There are only a small number of genuine users whose filler sizes are between 10% and 40%. There are no genuine users whose filler sizes are greater than 40%. The results indicate that the majority of genuine users in the collaborative recommender system only rate a small number of items. Therefore, if filler size is maximum since the minimum attack size considered in system is 2% which is greater than the number of users with the same length, the mean novelty degree is increased. Otherwise, for small filler size, the users with the minimum average novelty degree are considered suspicious users and the second step is taken. The example of Figure 7 presents the details of this step.

Discriminating the Genuine Users Incorrectly Identified as Fake Users from Fake Users

In the second step, in order to discriminate the genuine user incorrectly identified as fake users, we use the novelty item concept to calculate the novelty degree of the users' profiles [17, 18].

In fact, by studying the popularity of the items (number of rates of each item) as shown in Figure 8, most of the tail items are rated by a small group of users. Therefore, we are more likely to find novelty items that are similar to user profiles and have been targeted by attackers.

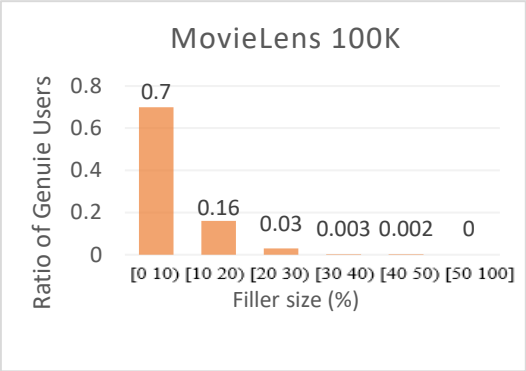

Fig. 5. Users profile length of Movie lens 100K Dataset

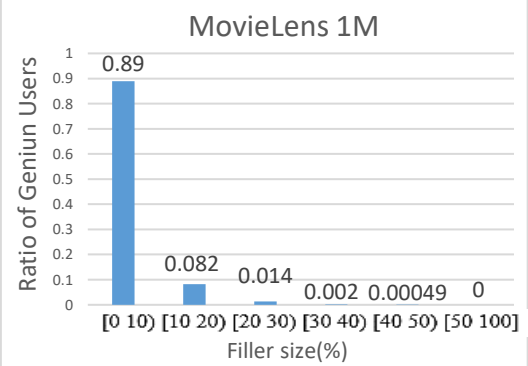

Fig.6. Users profile length of Movie lens 1M Dataset

A novelty item is unknown to the user, but the user might be interested in it if it is recommended to him/her. Because it has minimum distance and maximum similarity when it is compared with the user profile-items rated by the user [19].

In order to calculate the novelty degree of the users' profile and discriminate the users incorrectly identified as fake users from real attackers, the novelty degree of the items is calculated for the total users according to Eq. (1), (2) and (3) [20]. Then, the total novelty of the

items rated by each user determines the novelty degree of its profile. The novelty degree obtained for real users is greater than fake users. Because, real users are interested in the items, which have higher similarity to their interest while fake users, which rate the items randomly, are less probable to use items with high novelty. Figure 9 shows this step with an example for better understanding.

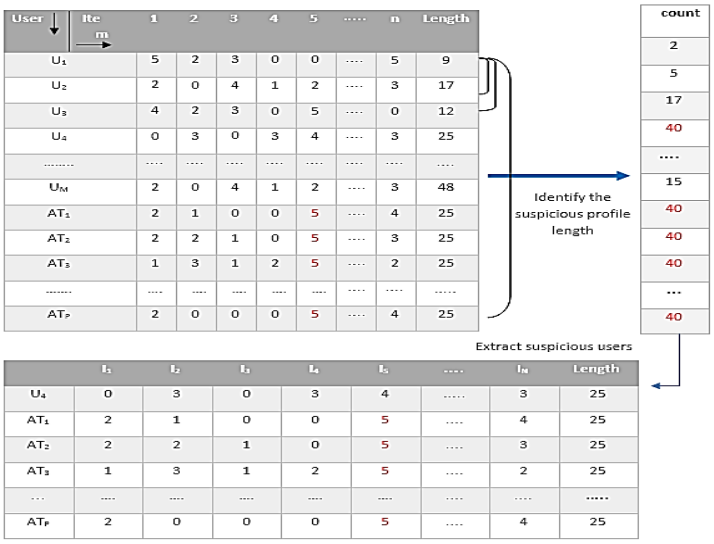

Fig. 7. Example of identifying suspicious users

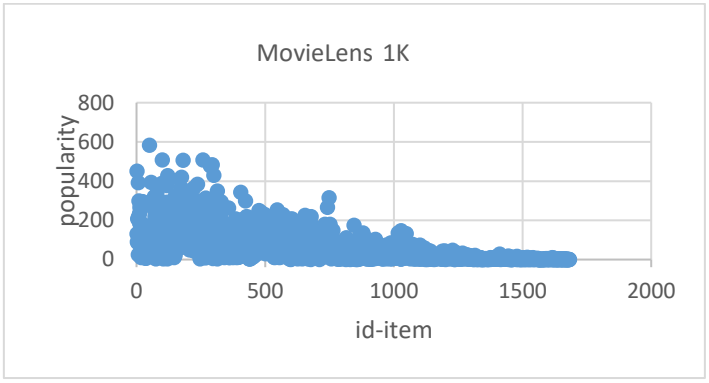

Fig. 8. The popularity of items of Movie lens 100K Dataset

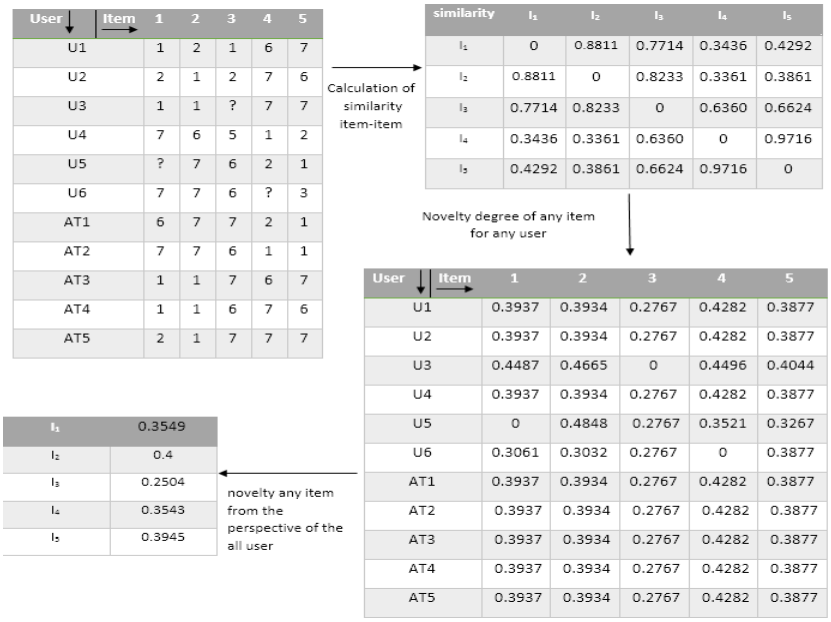

Fig. 9. Example of discriminating the real users incorrectly identified as fake users

$$nol_i = \frac{1}{|R_g|} \sum_{u \in R_g . r_{u.i \neq o}} nol_{u.i} \quad (1)$$

where $r_{u,i}$ denotes the rating of user u on item i, R_g denotes all users in dataset, nol_i denotes novelty of item i and $nol_{u,i}$ denotes the novelty of item i to all users, which is computed in the succeeding Equation:

$$nol_{u.i} = \frac{1}{p-1} \sum_{j \in L} (1 - sim(i.j)) \quad 0(2)$$

Where L denotes the set of items rated by user u, P denotes the items except item i rated by the user u, and Sim(i,j) denotes the similarity between item i and item j, which is computed in the succeeding Equation:

$$sim(i.j) = \frac{\sum_{u \in R_g} r_{u.i} . r_{u.j}}{\sqrt{\sum_{u \in R_g} r_{u.i}^2} . \sqrt{\sum_{u \in R_g} r_{u.j}^2}} \quad 0(3)$$

Where $r_{u,i}$ and $r_{u,j}$ are ratings of user u on items i and j.

Finally, when suspicious users and novelty degree of their profiles are specified, DBSCAN density-based clustering algorithm is used to discriminate the users based on the novelty of their profiles. This algorithm requires two parameters:

Eps: It defines the neighborhood of a point.

MinPts: It defines the minimum number of neighbors within eps radius.

DBSCAN algorithm can be abstracted in the following steps:

1. The algorithm starts with an arbitrary point which has not been visited and its neighborhood information is retrieved from the eps parameter.
2. If this point contains MinPts within eps neighborhood, cluster formation starts. Otherwise, the point is labeled as noise. This point can be later found within the eps neighborhood of a different point and, thus can be made a part of the cluster.
3. If a point is found to be a core point, then the points within the eps neighborhood is also part of the cluster. So all the points within eps neighborhood are added, along with their own eps neighborhood, if they are also core points.
4. The above process continues until the density-connected cluster is completely found.
5. The process restarts with a new point which can be a part of a new cluster or labeled as noise.

However, since the values of the parameters significantly affect clustering performance of the algorithm, it is a challenge to determine the input parameters values of DBSCAN algorithm. For this limitation, in the literature methods the automatic determination of these parameters were proposed. For example, Ozkok et al. proposed AE-DBSCAN algorithm to automatically determine the epsilon value by utilizing k-dist list. The proposed AE-DBSCAN algorithm requires a dataset and a k value (or MinPts) as inputs. The proposed algorithm has two stages, such as determining the value of Eps and clustering the dataset. In the first stage, this algorithm assigns the first sharp change in the k-dist plot as epsilon value. To find the first sharp change, it first generates the k-dist plot of the dataset, and then, takes the first slope, which is above the mean + standard deviation of all non-zero slopes. Then this Eps value is used in the second stage with k (or MinPts) value to discover the clusters out of the dataset [21].

However, in the multi-density data set, DBSCAN may merge different clusters and neglect other clusters that assign them as noise. Because with a single global parameter Eps, it is impossible to detect some clusters using one global-MinPts, so the user must specify the different range of Eps values.

Gaonkor et al. proposed a new approach to determine the different range of Eps values automatically to identify the number of clusters of different densities including noise, which first draw a k-dist graph for all the points (k entered by users) then use the "knees" for estimating the set of Eps parameters. After determining the different Eps values, to estimate the value of MinPts, the number of data objects in Eps neighborhood of every point in the dataset is calculated one by one. For each different value of Eps the corresponding MinPts value is calculated by Equation 4:

$$\text{Minpts} = \frac{1}{n} \sum_{i=1}^{n} p_i \quad (4)$$

Where p_i is the number of points in Eps neighborhood of point i and n denotes the total number of points [22].

As it is explained, two methods require discriminating the value of k. Therefore, we proposed a new method to determine the values of Eps and MinPts automatically without requiring the value of k.

In order to determine the value of eps, first, the novelty profile values for all the suspicious users are calculated, then the novelty values are sorted. Using sorted novelty values, the novelty plot is drawn. Where the slope of the plot increases (absolute the difference between novelty profiles of two adjacent users), it indicates that the user belongs to another dense region. Because users belonging to a dense region have the least slope changes relative to each other, while users belonging to a region with different densities have the maximum changes.

Therefore, two dense areas, their centers, and the distance of each point from the center of its area are calculated. The farthest point of each area, which indicates its radius, is considered and the radius with the minimum value, which indicates the smallest area, is considered as eps. The pseudo-code is given in Figure 10.

As well as, in order to determine the value of Minpts, first, the average similarity values of the

suspicious users are calculated, and then the average similarity values are sorted. Using these sorted average similarity values the average similarity plot is drawn. where the slope of the plot increases (absolute the difference between average similarity of two adjacent users), discriminate users of two areas and center of each area is determined; unlike eps method, difference of the users in each area compared to each other should be calculated and their average is used as the radius. Now, the number of the points which their distance from the center of each area is less than or equal to the calculated radius is determined as Minpts. The pseudo-code is given in Figure 11.

4. EXPERIMENTS AND ANALYSIS

4-1. Experimental Data

We used two publicly available datasets to evaluate our work: MovieLens 100K and MovieLens 1M. The MovieLens 100K includes 100000 ratings on 1682 movies by 943 users and the MovieLens 1M includes 1000000 ratings on 3952 movies by 6040 users. The ratings are a positive integer between 1 and 5, where 1 indicates that the users do not like the movies and 5 indicates that the users enjoy the movies very much.

4-2. Attack Experimental Design

Collaborative filtering recommender systems (CFRSs) are widely used in the well-known E-commerce websites such as Amazon, eBay, and etc. However, they are vulnerable to profile injection shilling attack or shilling attack because attack profiles inject into the rating system to affect the user's opinion.

According to the motivation of attackers, shilling attacks can be divided into push and nuke attacks. Push attacks aim to increase the popularity of a target item to make the target item more likely to be recommended for users. Conversely, nuke attacks try to decrease the popularity of a target item to make it less likely suggested.

we mainly aim at push attack and generate attack profiles based on different attack models, including Random, Average, Bandwagon attack models with different values of attack size (2%, 3%, 5%, 10% and 20%), filler size (1%, 3%, 5%, 7%, 10% and 15%). After that, the attack datasets are respectively inserted into the authentic data to construct the finally experimental datasets. In three attacks, target item is an randomly choose from the dataset and is assigned to r_{max}. In random attack, filler items are given system's mean as a rating. In average attack, mean of each item is given as rating for filler items. Bandwagon attack is one in which selected items are the popular items for which maximum ratings are given.

we use Matlab 2017 to implement these attack models in a personal computer with Intel(R) Core(TM) i5-4110 2.40 GHz CPU, 6GB memory and Microsoft Windows 7 operating system.

Input: Suspected users
Output: Determination of Epsilon value
Algorithm:

1. Calculate the novelty profile of all suspected users.
2. Sort the novelty value and draw a novelty plot.
3. Calculate the mean of novelty.
4. Find the first slope, which is above mean (novelty).
5. Identify users whose novelty values are above mean (novelty) as group1.
6. Identify users whose novelty values are below mean (novelty) as group2.
7. Calculate the center of each group and the point's difference from the center of the group.
8. Identify the farthest points of each group's center.
9. The minimum radius is as Eps.

Fig. 10. The pseudocode determines the value of Eps parameter

Input: Suspected users
Output: Determination of MinPoints value
Algorithm:

1. Calculate the average similarity of suspected users.
2. Sort the average similarity of suspected users and draw an average similarity plot.
3. Calculate the mean and standard deviation of the average similarity of users.
4. Find the first slope, which is above mean + standard deviation.
5. Identify users whose average similarity values are above mean + standard deviation as group1.
6. Identify users whose average similarity value are below mean + standard deviation as group2.
7. Calculate the center of each group and the difference between the points of each group relative to each other.
8. The number of points whose distance from each group is less than the calculated radius is as MinPts.

Fig. 11. The pseudocode determines the value of MinPts parameter

4-3. Evaluation Metric

To evaluate the performance of our proposed approach, we use the precision, recall and f-measure parameters.

Precision is the ratio of the number of attackers identified to the total number of users who are identified as attackers and is defined as:

$$\text{Precision} = \frac{nTP}{nTP+nFP} \quad (5)$$

Where TP denotes the number of attackers (profiles) correctly classified as attackers and FP denotes the number of authentic profiles misclassified as attack profiles.

Recall is the ratio of the number of attackers identified to the total number of attackers in the system and is defined as:

$$\text{Recall} = \frac{nTP}{nTP+nFN} \quad (6)$$

Where FN denotes number of attack profiles misclassified as authentic profiles.

4-4. Experimental Results and Analysis

Due to the high number of experiment (2 * 5 * 6 * 3) including 2 different datasets, 5 different attack sizes, 6 different filler sizes and 3 different attack models, only the results of clustering algorithm for three attack models with attack size 10% and filler size 5% and 10% are presented in Figures 12-17.

To further examine the detection performance of the proposed method, all experiments are conducted on two datasets. The outputs are evaluated using Recall, Precision, and F-Measure metrics where their results are shown in Tables 3 and 4.

As shown in Figures 12-17 and Tables 3 and 4, with the increasing attack size, the number of fake user increases which makes easier detection of fake users and accuracy increases. Also with increasing filler size, the number of genuine profiles misclassified as attack profiles is reduced which makes easier detection of fake users and accuracy increases.

In addition, it is important to note that although average and bandwagon attacks are more efficient than random attack and detecting attacks should be difficult, but since the first step of the proposed approach is based on the length of the user profiles and independent from users' rate, it can independent of type of attack, detect fake users with an accuracy close to 1.

The Bandwagon attack uses popular items, which increases the similarity of the fake user's profile to the real user's profile, and increases profile length in a smaller number of real users incorrectly identified as

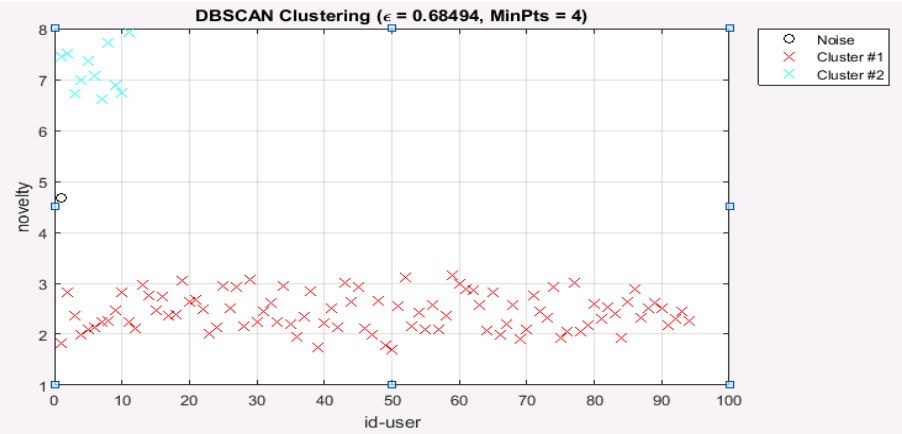

Fig. 12. The result of random attack with Attack size = 10%, Filler size = 3% (Movie Lens 100K)

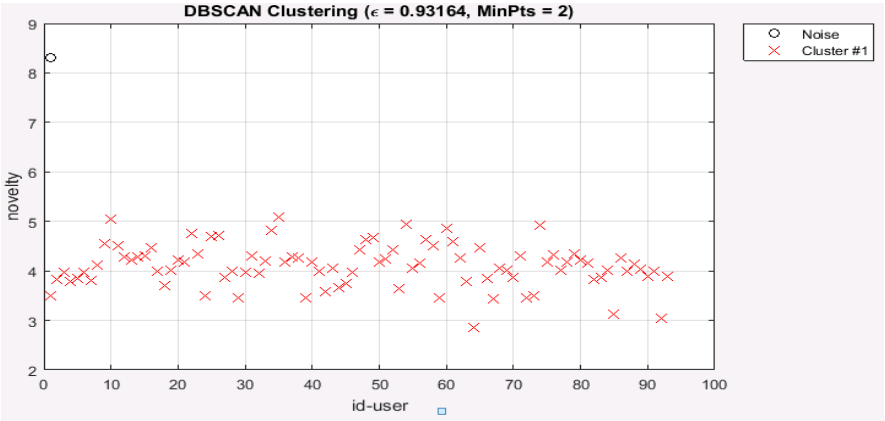

Fig. 13. The result of random attack with Attack size = 10%, Filler size = 5% (Movie Lens 100K)

fake users. Wherefore the accuracy is still maintained.

In the following, we compare the performance of the proposed approach and the various attack detection techniques presented in [3] to demonstrate the effectiveness of the proposed approach.

Figures 18 and 19 show comparative results of NB, C4.5, PCA, MDS, HySAD, SDF detectors and proposed approach against three types of attacks with 94 attacker profiles injected.

As shown in Table 5 and 6, although the overall performance of supervised detectors is satisfactory, but they have behaved unstably because they heavily depend on the training dataset. The performance of supervised detectors fluctuates considerably, especially that of C4.5. NB is more stable than C4.5, of which the reason is that NB is affected by the joint probability of all selected features, yet C4.5 is commonly to use a few features for constructing a decision tree.

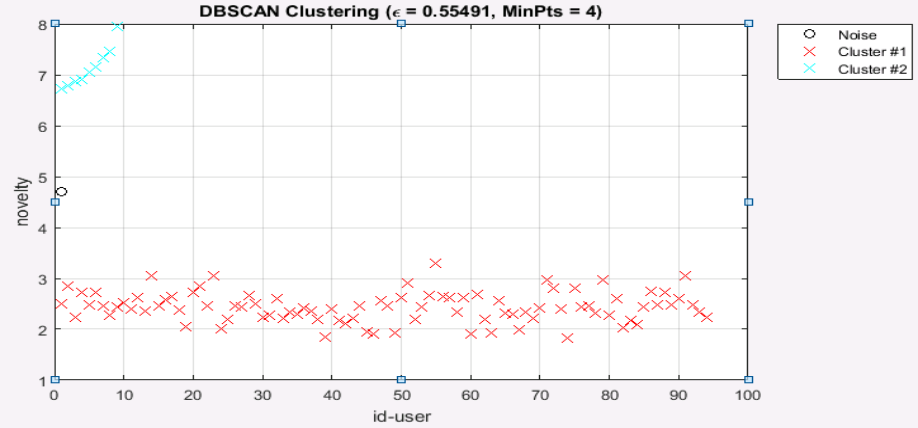

Fig. 14. The result of average attack with Attack size = 10%, Filler size = 3% (Movie Lens 100K)

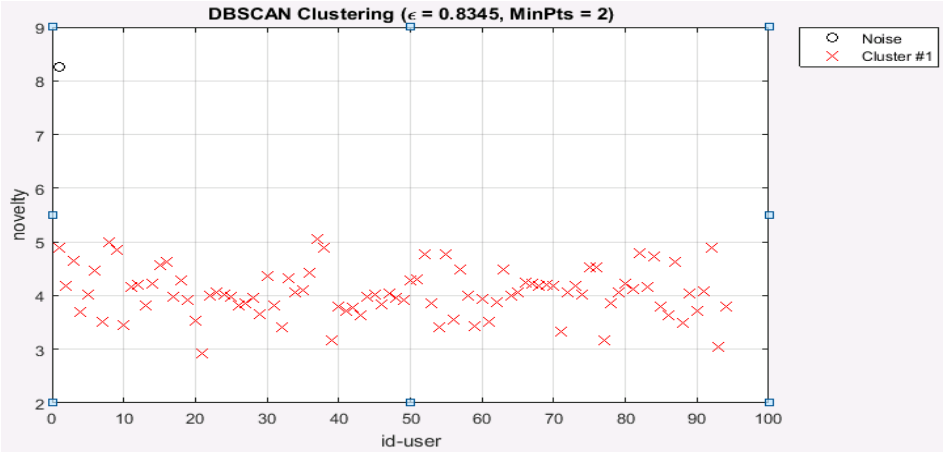

Fig. 15. The result of average attack with Attack size = 10%, Filler size = 5% (Movie Lens 100K

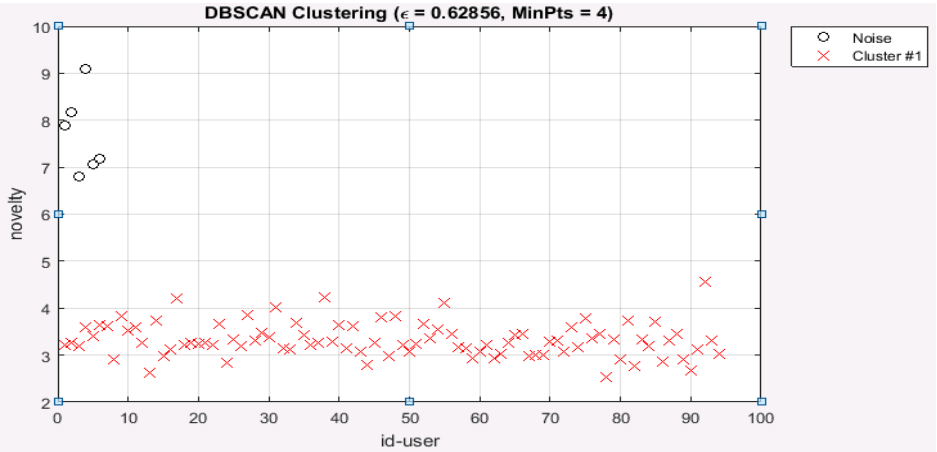

Fig. 16. The result of bandwagon attack with Attack size = 10%, Filler size = 3% (Movie Lens 100K)

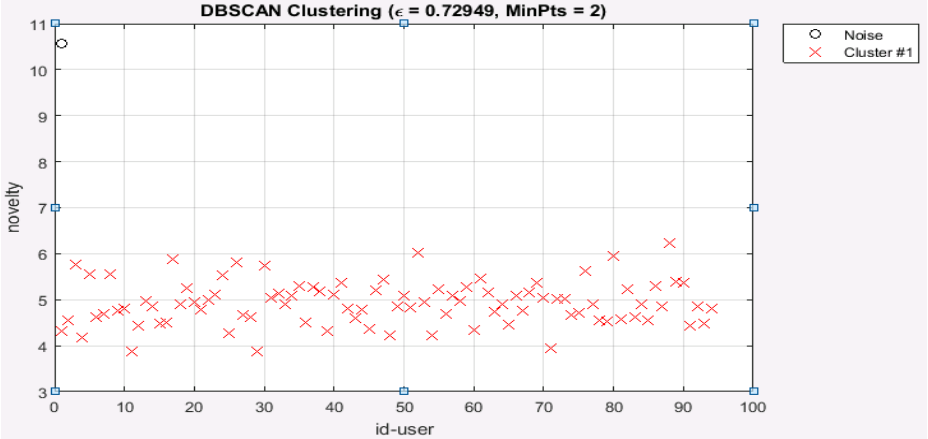

Fig. 17. The result of bandwagon attack with Attack size = 10%, Filler size =5% (Movie Lens 100K)

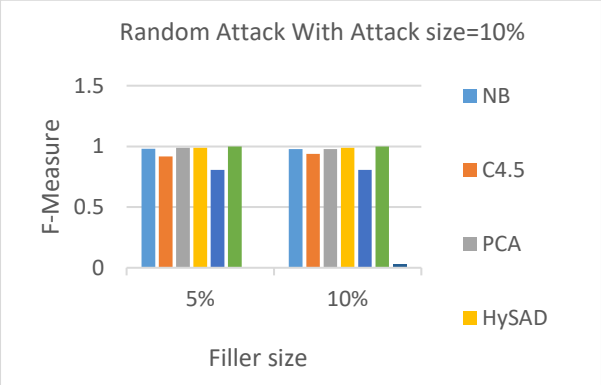

Fig. 18. The random attack Performance Analysis of the proposed method and method presented in [3] with Attack size = 10% (Movie Lens 100K)

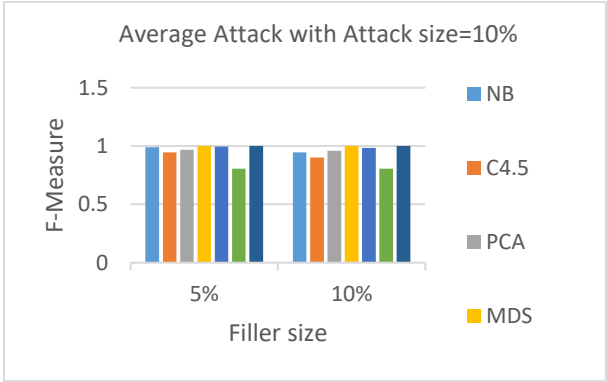

Fig. 19. The average attack Performance Analysis of the proposed method and method presented in [3] with Attack size = 10% (Movie Lens 100K)

TABLE 3. PERFORMANCE ANALYSIS OF PROPOSED METHOD ON MOVIE LENS 100K DATASET

	Measure	Ranandom Attack					Average Attack					Bandwagon Attack with 5 popular item				
		AS=2%	AS=3%	AS=5%	AS=10 %	AS=20 %	AS=2%	AS=3%	AS=5%	AS=10 %	AS=20 %	AS=2 %	AS=3%	AS=5 %	AS=10 %	AS=20 %
FS=1.2%	P	0.952	0.965	0.94	1	1	0.952	0.965	0.979	0.989	0.994	0.952	0.965	0.979	0.979	0.984
	R	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	F	0.979	0.982	0.969	1	1	0.975	0.982	0.989	0.994	0.996	0.975	0.982	0.989	0.989	0.991
FS=3%	P	1	1	1	0.998	0.994	1	1	1	1	0.994	1	1	1	0.998	0.999
	R	1	1	1	1	1	1	1	1	1	1	0.95	1	1	١	١
	F	1	1	1	0.998	0.996	1	1	1	1	0.996	0.974	1	1	0.998	0.999
FS=5%	P	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
FS=7%	P	1	0.965	0.979	0.998	0.994	0.952	0.965	1	0.989	0.994	1	1	1	1	1
	R	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	F	1	0.982	0.989	0.998	0.996	0.975	0.982	1	0.994	0.996	1	1	1	1	1
FS=10%	P	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R	1	0.964	1	1	0.994	1	1	1	1	1	1	1	1	1	1
	F	1	0.981	1	1	0.996	1	1	1	1	1	1	1	1	1	1
FS=15%	P	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Unsupervised detector cannot effectively identify every type of Shilling attack, because in PCA, suspicious users are selected from outlier users, i.e. those users who have entirely different rating styles with other users and lower similarity with other users, i.e., exert smaller effects to other users. Thus, PCA is apt to detect shilling attackers with smaller attack power. Thus, the accuracy of the random attack is higher

TABLE 4. PERFORMANCE ANALYSIS OF PROPOSED METHOD ON MOVIE LENS 1M DATASET

	Measure	Ranandom Attack					Average Attack					Bandwagon Attack with 5 popular item				
		AS=2%	AS=3%	AS=5%	AS=10 %	AS=20 %	AS=2 %	AS=3%	AS=5%	AS=10 %	AS=20 %	AS=2%	AS=3 %	AS=5 %	AS=10 %	AS=20 %
FS=1.2%	P	0.992	0.994	0.993	1	0.998	0.997	0.989	0.990	0.995	0.998	0.992	0.994	0.996	0.993	0.995
	R	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	F	0.995	0.996	0.996	1	0.998	0.998	0.994	0.994	0.997	0.998	0.995	0.996	0.997	0.996	0.997
FS=3%	P	1	0.994	0.996	0.998	1	1	0.994	0.996	0.998	0.999	0.992	0.994	0.996	0.998	0.999
	R	1	1	1	1	1	1	1	1	1	\	1	1	1	1	\
	F	1	0.996	0.997	0.998	1	1	0.996	0.997	0.998	0.999	0.995	0.996	0.997	0.998	0.999
FS=5%	P	0.992	0.994	0.996	0.998	0.999	0.992	0.994	0.996	0.998	0.999	1	1	1	1	1
	R	1	1	1	1	\	1	1	1	1	\	1	0.994	1	1	1
	F	0.995	0.996	0.997	0.998	0.999	0.995	0.996	0.997	0.998	0.999	1	0.996	1	1	1
FS=7%	P	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
FS=10%	P	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R	1	1	1	1	1	0.992	1	1	1	1	1	1	1	1	1
	F	1	1	1	1	1	0.995	1	1	1	1	1	1	1	1	1
FS=15%	P	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TABLE 5. F-MEASURE VALUE OF THE PROPOSED METHOD AND METHODS PRESENTED IN [3] WITH ATTACK SIZE = 10% IN RANDOM ATTACK (Movie Lens 100K)

	NB	***C4.5***	***PCA***	***MDS***	***HySAD***	***SDF***	***Proposed Method***
FS=5%	***0.981***	***0.916***	***0.989***	***0***	***0.989***	***0.807***	***1***
FS=10%	***0.977***	***0.937***	***0.979***	***0.032***	***0.989***	***0.807***	***1***

TABLE 6. F-MEASURE VALUE OF THE PROPOSED METHOD AND METHODS PRESENTED IN [3] WITH ATTACK SIZE = 10% IN AVERAGE ATTACK (Movie Lens 100K)

	NB	***C4.5***	***PCA***	***MDS***	***HySAD***	***SDF***	***Proposed Method***
FS=5%	***0.989***	***0.945***	***0.968***	***1***	***0.994***	***0.807***	***1***
FS=10%	***0.945***	***0.902***	***0.959***	***1***	***0.984***	***0.807***	***1***

than average attack. Because in the random attack, fake profiles are lower similar to other users. MDS unlike PCA tries to seek shilling attackers among effective users, rather than outlier users. It is reasonable that if shilling attackers cannot influence other users, i.e., resulting in little damage to recommender systems, they could be left out. Thus, it is apt to detect shilling attackers with high attack power. Since in the average attack, the mean of each item is given as rating for filler items unlike the random attack, it has higher power and accuracy.

Statistical method of Segmented Dynamic Framework (SDF) detects spam users through detecting the target items. Since it is independent of attack type, its results are uniform for all filler sizes.

HySAD detectors can effectively extend the supervised classifiers to make full use of both labeled and unlabeled user profiles for the categorization model. Since HySAD method firstly uses the labeled dataset to train a NB classifier, and predicts the posterior probabilities of the unlabeled data, then the initial classifier will be improved by using an expectation-maximization-like iterative process with unlabeled dataset.

As regards, test results show the detection performance of the proposed method performs an acceptable result in most of the presented attacks with diverse attack sizes and filler sizes. In addition, it performs better than all of the methods mentioned above due to independency of attack type and user rating in the first step.

5. CONCLUSION

As mentioned, most e-commerce websites use recommender systems to increase their sales and attract the trust of the users by recommending the best items in proportion to the user's interest. However, a shilling attack is a significant threat to these systems because it distrusts the users' trust. In this paper, a new approach based on the maximum profile length and novelty

degree of the users' profile is proposed to detect fake users so that accuracy of the recommendations is improved and increase the users' trust which is difficult for attack sizes less than 3% because proportionality of the real and fake users is not suitable. However, when attacking size increases, the number of fake users increases which causes easier detection and higher accuracy.

In attacks with filler sizes less than 10%, according to Figures 5 and 6, since the profile length of most real users lie in this interval, the number of genuine users misclassified as fake user increases and the accuracy value reduces. As filler size increases, since the number of real users misclassified as fake users decreases, discrimination improves and accuracy is increased. The proposed approach can be applied for various nuke attacks with a small change in the feature set, which can be an objective for future studies.

REFERENCES

[1] S. Badsha, X. Yi, and I. Khalil, "A practical privacy-preserving recommender system", *Data Science and Engineering*, vol. 1, Issue. 3, pp. 161-177, 2016.

[2] R. Burke, "Hybrid recommender systems: Survey and experiments", *User modeling and user-adapted interaction*, vol. 12, Issue. 4, pp. 331-370, 2002.

[3] Y. Wang, L. Qian, F. Li, and L. Zhang, "A comparative study on shilling detection methods for trustworthy recommendations", *Journal of Systems Science and Systems Engineering*, vol. 27, Issue. 4, pp. 458-478, 2018.

[4] M. G. Campana, and F. Delmastro, "Recommender systems for online and mobile social networks: A survey", *Online Social Networks and Media*, vol. 3, pp. 75-97, 2017.

[5] F. Mansur, V. Patel, and M. Patel, "A review on recommender systems", in *Innovations in Information, Embedded and Communication Systems (ICIIECS)*, IEEE, 2017.

[6] J. Bobadilla, F. Ortega, A. Hernando and A. Gutiérrez "Recommender systems survey", *Knowledge-based systems*, vol. 46, pp. 109-132, 2013.

[7] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, "Shilling attacks against recommender systems: a comprehensive survey", *Artificial Intelligence Review*, vol. 42, Issue. 4, pp. 767-799, 2014.

[8] M. Si, and Q. Li, "Shilling attacks against collaborative recommender systems: a review", *Artificial Intelligence Review*, pp. 1-29, 2018.

[9] Z. Yang, Z. Cai, and X. Guan, "Estimating user behavior toward detecting anomalous ratings in rating systems", *Knowledge-Based Systems*, vol. 111, pp. 144-158, 2016.

[10] H. Abdollahpouri, R. Burke, and B. Mobasher, "Managing Popularity Bias in Recommender Systems with Personalized Re-ranking", *arXiv preprint* arXiv:1901.07555, 2019.

[11] Z. Wu, L. Zhang,Y. Wang,J. Cao, Identifying Spam in Reviews. In: Alhajj R., Rokne J. (eds) (2018), Encyclopedia of Social Network Analysis and Mining. Springer, New York, NY.

[12] R. Bhaumik, B. Mobasher, and R. Burke. "A clustering approach to unsupervised attack detection in collaborative recommender systems", in *Proceedings of the International Conference on Data Mining (DMIN)*, Citeseer, 2011.

[13] F. Zhang, "Robust Analysis of Network based Recommendation Algorithms against Shilling Attacks", *International Journal of Security and Its Applications*, vol. 9, Issue. 3, pp. 13-24, 2015.

[14] B. Mehta and T. Hofmann, "A Survey of Attack-Resistant Collaborative Filtering Algorithms", *IEEE Data Eng. Bull*, vol. 31, Issue. 2, pp. 14-22, 2008.

[15] J. Cao, Z. Wu, B. Mao, and Y. Zhang, "Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system". *World Wide Web*, vol. 16, Issue. 5-6, pp. 729-748, 2013.

[16] W. Zhou, "Attack detection in recommender systems based on target item analysis", *International Joint Conference Neural Networks (IJCNN)*, IEEE, 2014.

[17] P. Castells, N. J. Hurley, and S. Vargas, "Novelty and diversity in recommender systems", in *Recommender Systems Handbook*, pp. 881-918, Springer, 2015.

[18] P. Castells, S. Vargas, and J. Wang, "Novelty and diversity metrics for recommender systems: choice, discovery and relevance", In *Proceedings of International Workshop on Diversity in Document Retrieval (DDR)*, 2011.

[19] L. Zhang, "The Definition of Novelty in Recommendation System", *Journal of Engineering Science & Technology Review*, vol. 6, Issue. 3, 2013.

[20] F. Zhang, and H. Chen, "An ensemble method for detecting shilling attacks based on ordered item sequences", *Security and Communication Networks*, vol. 9, Issue. 7, pp. 680-696, 2016.

[21] F. O. Ozkok, and M. Celik, "A new approach to determine Eps parameter of DBSCAN algorithm", *International Journal of Intelligent Systems and Applications in Engineering*, vol. 5, Issue. 4, pp. 247-251, 2017.

[22] M. N. Gaonkar, and K. Sawant, "AutoEpsDBSCAN: DBSCAN with Eps automatic for large dataset", *International Journal on Advanced Computer Theory and Engineering*, vol. 2, Issue. 2, pp. 11-16, 2013.

Homa Tafakori received the B.S. degree from Bojnourd University in 2016 and the M.S. degree from Golestan University in 2019.

Soheila Karbasi is an assistant professor of Information Systems in Golestan University. Her main research areas are Information retrieval process and Data mining.

Mehdi Yaghoubi is an assistant professor of Artificial Intelligence in Golestan University. His work focuses mainly on the Data mining, Process mining and BPMS.