Advances in Science, Technology and Engineering Systems Journal Vol. 6, No. 1, 684-691 (2021)

www.astesj.com

Special Issue on Multidisciplinary Sciences and Engineering

ASTESJ

ISSN: 2415-6698

Dismantle Shilling Attacks in Recommendations Systems

Ossama Embarak[*]

Computer Information Science, Higher Colleges of Technology, Abu Dhabi, 51133 UAE

ARTICLE INFO

Article history:
Received: 18 December, 2020
Accepted: 24 January, 2021
Online: 05 February, 2021

Keywords:
Malicious users
Push & nuke attacks
Collaborative systems
Shilling attacks

ABSTRACT

Collaborative filtering of recommended systems (CFRSs) suffers from overrun false rating injections that diverge the system functions for creating accurate recommendations. In this paper, we propose a three-stage unsupervised approach. Starts by defining the mechanism(s) that makes recommendation vulnerable to attack. Second, find the maximum-paths or the associated related items valued by the user. We then rule out the two attacks; we will need to pull two different measures. (a) We will pull user ratings across all reviews and measure their centre variance. (b) We will then pull each individual user rating and measure them according to the original rating. Detected attack profiles are considered untrusted and, over time, if the same user is detected as untrusted, the profile is classified as completely untrusted and eliminated from being involved in the generation of recommendations. Thus, protect CFRS from creating tweaked recommendations. The experimental results of applying the algorithm to the Extensive MovieLens dataset explicitly and accurately filter users considering that a user could seem normal and slightly diverge towards attack behaviours. However, the algorithm used assumes that the framework has already begun and manages user accounts to manage the cold start scenario. The proposed method would abstractly protect users, irrespective of their identity, which is a positive side of the proposed approach, but if the same user reenters the system as a fresh one, the system will reapply algorithm processing for that user as a normal one.

1. Introduction

This paper is an extension of work originally presented in 2019 Sixth HCT Information Technology Trends (ITT) [1]. Personalization Collaborative Filter Recommending Systems (CFRSs) is becoming increasingly popular in well-known e-commerce services such as Amazon, eBay, Alibaba, etc. [2]. Most people are rating products or services without even realizing it is something they can do [3]. However, CFRSs are highly vulnerable to "profile injection" attacks; often referred to as "shilling" attacks [4]. It is common for attackers to pollute the recommended systems with malicious ratings. They either demote a target item with the lowest rating; called a nuke attack or promote a target item with the highest rating; called a push attack to reach their target or minimize the recommendation's accuracy. It is therefore necessary to develop an effective detection system for detecting and removing attackers before the recommendation [5–7].

Detection methods based on the attacks gained much coverage. Since the similarities between attackers are higher than the actual users, some have been presented based on the similarity between users. Traditional similarity metrics, including the Pearson Correlation Coefficient (PCC),[8,9]. However, the detection efficiency of these methods relies largely on the estimation of similarity. How to reduce the time consumption of the measurement of similarity is also a difficult problem, particularly when dealing with large datasets[10]. In addition, some attackers imitate the rating data of some legitimate users in order to increase their reliability. Only the use of similarities is difficult to discriminate incomplete [11]. In order to overcome these problems, a more efficient form of detection should be considered in the following aspects:

A. Applied algorithm complexity should be acceptable.

B. The proposed method should be able to defeat various kinds of "shilling" attacks.

This paper proposes an unsupervised detection method for detecting such attacks, consisting of three phases. The purpose of the proposed method is to filter out more legitimate users, and at the same time hold, all attackers step by step. Firstly, the recommendation system admin needs to understand the attacker's strategy based on the recommendations working mechanisms. Secondly, we constructed users' maximal paths that represent his/her loop less-visited nodes or rated items in an association, then calculate the maximal path's mean and standard deviation considering all ratings done by users visited that particular maximal path. Thirdly, we detected untrusted users by detecting the user's zone (normal, freeze or melt zone area) which reflect the gap between the user's ratings in the

[*] Corresponding Author: Ossama Embarak, oembarak@hct.ac.ae

https://dx.doi.org/10.25046/aj060174

maximal path (items) and the genuine profiles' ratings of the same path(items) considering the maximum expected calculated deviation. The proposed algorithm prevents untrusted users' maximal sessions should not be added to the system profiles used for generating recommendations, and the user is labelled as untrusted profiles in which the system should not consider such profiles towards generating recommendations.

Recommendation systems are massively used to support online systems provides services through the internet. As an alternative to passive search engines that simply accept as "true" the displayed results, these collaborative filtering systems attempt to take the perspective of the user's needs and what they are searching for and identify similar results based on that [12]. Decisions that depend on a user's previous demands can be made with a recommendation system; a recommendation system is a set of techniques used to process and generate recommendations. In order to inject the ratings of biased users into recommender systems, users attempt to influence the system by injecting biased ratings for a specific product they are interested or associated with. Through a process known as a rating bot attack, which can plough the ratings up or down, change the ratings of services or items to the highest or the lowest available levels, or launch attacks against them [13]. In order to avoid shilling attacks, the company has attempted to implement various suggestions, such as asking users to rate items by using certain code, or by only allowing ratings to occur after a certain amount of time, or by simply making profiles harder to create or by increasing the price of creating a profile? Because of the ways in which these methods can eliminate attacks, the participants in rating and evaluating services became much smaller than normal [14]. E-commerce web-systems suffer from attackers who push specific items to the recommendations list and promote the target item via manipulating recommendation systems [15]. The attackers create numerous profiles by injecting unusual identities and putting very high-ratings for their target items. An attacker's profile will camouflage itself by keeping track ratings for other non-target items, and the system will use that information to predict the attacker's targets. The manipulations of recommendations systems of injecting misleading and false data make a recommendation system's operation is negatively affected [16]. Recommendations systems powered by user-generated reviews will only be representative of individuals with very few reviews. Initial profiles will greatly affect future processing and output. Moreover, this system does not generate suitable recommendations, and occasionally, it breaks down the whole system; It causes the system to lose its credibility and respect by people in the community. Therefore, several research groups' main goal is to develop a system that can filter out fake profiles that could make those systems able to provide accurate recommendations [17].

This paper looks into a different type of attacks for recommendation systems and propose a solution to detect and isolate attackers. It contains the following sections: Section 1 introduction to recommendation systems and its attacks. Section 2, the background of recommendation systems different forms of attacks. Section 3, we discuss the previous researches and approaches for handling different attacks. Section 4 identify the study problem. In section 5, we explore the proposed solution. Section 6 demonstrate experimental results. Finally, The paper and the future work will be concluded.

2. Background

Numerous different sorts of recommenders are used on different sites. Studies in the past have shown dramatic growth in the methods used to improve recommendations. Recommendation systems can be broadly classified into content-based and collaboration-based [18]. Content-based filtering suggests products based on users browsing history. There are downsides to this type, such as over-specialization, which recommends only similar products to what was consumed before. Collaborative filtering takes into account past behaviour, considering that users will behave similarly. Shilling attacks manipulate recommendation systems by inserting malicious user profiles into the filter data [19]. The target item is either promoted or demoted. Attacks can be categorized as push or nuke strikes, a product may therefore be promoted or dismissed for advantages over competitors [20]. Over time, numerous attacks models have been developed. There are many detection techniques and algorithms to counter such attacks. All of the attack models are designed to generate vicious users. The discrepancies are recognized in how the strike profiles are established. Malicious users' domain items can be considered as S = { I_s, I_p, I_n, $I_Ø$}, where I_s is the set of items that matched with the genuine users. *while* I_p is the set of items the malicious user promote, and I_n is the set of items he/she demote, and $I_Ø$ items that are not ranked yet by the trouble user.

Over the years, several attacks were developed. All such attacks can be classified either as a high-level attack or as a poor-knowledge attack based on their motivation and knowledge [21, 22]. Attacks with little knowledge are more practical and are more likely to impact the real world, but such attacks' effectiveness is also low. On the other hand, high-level attacks can have a huge impact on Recommender Systems' performance, but they are more difficult to pull out. *RandomBot attack* is the plainest form of a shilling attack. Items rated by the attackers selected in random, except for the target item. The rankings for such items are based on the overall system norm [22].

A maximum or minimum rating will be given on the basis that it is a push or a nuke attack. The purpose of a random attack is usually more effective in disrupting the Recommender Systems' performance than in promoting the target item. *Average Attack* is similar to Random Attack when selecting items. The randomized distribution of individual items is based on their distribution of ratings. The average rating of each filling item is assigned. This attack can only be carried out if the attacker knows the system's dataset. While random attack and average attack differ only from filler ratings, the average attack impact is much higher. *Bandwagon Attack* is the kind of attack where attacker profiles are filled with popular high rating items. Naturally, attack profiles are closer to many people. The highest value is given to the target item. Depending on the rating scheme used to provide the filler items, this attack can further be divided into random and bandwagon averages. Bandwagon is also classified as a low-knowledge attack since the attacker only needs public data. *Bandwagon Reverse Attack* is the exact reversal of an attack on a bandwagon. This attack is used to demote the target product by rating the items with high negative ratings low and giving the target item the least rating. *Segmented Attack* targets a particular group of users who can buy the target item in the e-commerce system. Segment attacks typically occur in collaborative item-based filtering [23, 24].

The ratings and items are based on knowledge of the segment from the attacker. The important benefit of this method is its ability to reach potential clients over other approaches. *Probe attack* occurs when an attacker provides genuine ratings for items based on their knowledge of the predicted rating scores. The attacker uses this detail to rate items, enabling it to appear similar to other ratings. The attacker composes the acceptable items list based on the rated items. This collaborative scheme allows attack profiles to remain close to their neighbours. It also facilitates an attacker in gaining more information about the system. *Love/Hate Attack*, where the attacker randomly chooses filler items and gives them the highest rating and lowest rating on the target item. It can also be known as a push attack by changing the highest ratings [25]. *Noise Injection attack* generates random numbers multiplied by a constant to each rating for a subset of rated profiles. The more obfuscation, the more constant. It can be effectively applied to obfuscate its signature to all standard methods. A slightly decreased ability to withstand an attack is observed in response to noise injection. *User-Shifting attack* where a subset of users revises ratings [26]. The attack subsets were selected randomly to reduce their similarities. Ratings are scaled differently for different subsets of the ratings [27]. *Target Shifting attacks* shift the targeted item rating to a lower level in push attacks than maximum. For nuke attacks, the minimum rating is shifted up by one. *The mixed attack* is made using the same proportions of random, average, bandwagon and segmented attacks. The detection technique should be able to detect all the standard attacks successfully. The various methods for attacking the same target item are used. It helps to avoid various methods of detection [28].

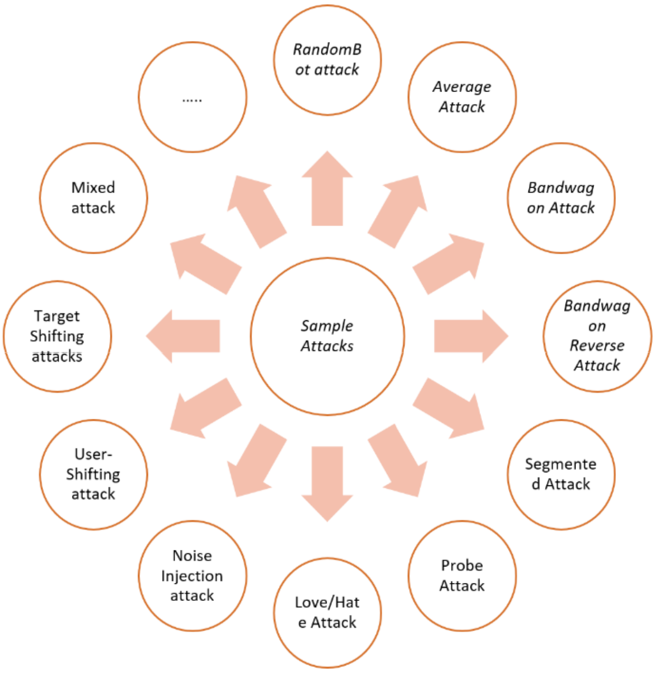

Figure 1: Sample Attacks Types

3. Related Works

A well-known class of attacks is called the Shilling attacks, which attempts to inject some profiles to influence the targeted system's performance [13]. This attack is divided into two attacks: push attack- which uses a malicious profile to rate a specific item highly (greater than most items) and nuke attack- which uses malicious profiles to lower rating for a specific item (s). Playbooks attacks reflect a series of actions undertaken to maximize a certain item's importance and increase its profits [29]. Some systems produce low-quality recommendations, so that once the users discover such attacks, the system loses its users' respect and loyalty [30]. Unfair rating attack occurs when a system uses trusted agents' ratings and compares them against each other for a service entity. This attack leads to predictions that are not actually correct [31]. Re-entry attacks occur when an agent is transferred between accounts using different names to avoid low ratings [29]. Sybil attacks reflect a fake identity to give different ratings. So far, a robust recommendation system is essential for any application of a recommendation system. Robustness measures a model's ability to produce good predictions with noisy data [32]. Whenever a user rates something, the system updates its databases, which means we have no way of knowing if the ratings are real or not. Ratings can sometimes be tedious, which could be caused by users' carelessness, and malicious users may attack. In robustness, there are two different aspects, *the first* of which is the accuracy of recommendation, which are the products that were recommended by the systems after the attack took place. *The second thing a system must do is be very stable*, which means won't recommend slightly different products if it detects an anomaly or attack.

Before storing user ratings into the system, the attack type must be detected to maintain its good reputation before affecting its performance. Injecting profiles another example of forged data being injected by malicious users into recommendations systems to manage recommendations [33]. Collaborative filters are susceptible to profile attacks by user-based and item-based recommendations. In this sort of attack, the attackers try to evaluate target and non-target items to make ratings appear normal by forging rating profiles and injecting them in rating systems [34]. Studies show that group attack profiles manipulate the target object ranking recommendation. Many attackers work together in a given time frame to attack certain target items to quickly put some items into a preferred list. The next section will provide a breakdown of the experiment and its results.

4. Identifying the Problem

Recommendation systems may provide health recommendations that reflect the preferences of users. However, injecting false ratings or trying to push or nuke ratings of the offered items differs in the system's performance. Many proposed solutions are almost implemented after attacking or generating recommendations that burden many computations and increase system time complexity. Furthermore, depending on the specific fixed average rating, the dynamic change in items' ratings over time is not reflected. Therefore, we propose a new approach to filter users into normal, fully trusted and untrusted users based on their maximal traversing path or items movable threshold.

5. The Proposed Solution

Details of the proposed method are implemented in three stages in this section. In the first phase, we need to understand the attacker's strategy based on recommendations generation mechanisms. In the second phase, we constructed users' maximal paths and calculated the maximal path's mean and variance, considering all users' ratings visited that particular maximal path. In the third stage, we detected untrusted users by detecting the user's zone (normal, freeze or melt zone area), which reflect the gap between the user's ratings in the maximal path (items) and the genuine profiles ratings of the same path(items) considering the maximum expected calculated deviation. All detected untrusted users' maximal sessions should not be added to the system profiles used for generating recommendations, and the user is labelled as untrusted profiles in which the system should eliminate his future participations.

5.1. Users Profiles

Recommendation systems collect user ratings – items used to form an object rating matrix that includes three elements: users, items and ratings.

$$U = \{ u_1, u_2, \ldots., u_{m-1}, u_m \}$$

where *U* refers to set of *m* users.

$$I = \{ I_1, I_2, \ldots., I_{n-1}, I_n \}$$

where *I* refers to a set of n items.

All users can rate few or all items; a user-item rating matrix generated from all the ratings is used to find attackers and make many recommendations.

$$\begin{bmatrix} I_{u_1}^{1} & \cdots & I_{u_1}^{n} \\ \vdots & \ddots & \vdots \\ I_{u_m}^{1} & \cdots & I_{u_m}^{n} \end{bmatrix}$$

Figure 2: Items-users ratings matrix

where $I_{u_1}^1$ reflect the rating of the first item by the first user, while $I_{u_m}^n$ is the rating of the item number n by the user number m.

A regular user weight items in the overall round the norm. Similarly, both the push and nuke attackers aim to rate items extremely far from the item threshold. We call it a *freezing* zone, as the rates are very far from the item average rates measured, and the nuke attackers aim to rate items that are extremely low from the item thresholds, we call it a *melting* zone, as the rates are very far back from the item thresholds. We measure each user's ratings and can detect any change in weight in either "Freeze" or "Melting" zones by measuring the weighted mean of all ratings.

5.2. Attackers Background

The attackers usually deliberately attempt to change the efficiency of the recommendation system. They know how the system operates and use their expertise and understanding to affect the system's efficiency.

There are three types of systems in terms of user awareness of the Recommendation Mechanisms.

1. Systems that depend on the popularity threshold (like Insider) to produce recommendations are simply showing popularity to users. As a result, attackers know exactly what will be recommended, attempt to modify popularity by revisiting the same path(route) or increase or decrease other paths' ratings.
2. Systems that depend on each item's popularity to produce a recommendation, the popularity displayed to users with a list of recommendations can also use the association rule to show that items x are frequently selected with item y, such as YouTube videos. As a result, attackers have full knowledge of how ratings are allocated in the recommended device profiles. Attackers attempt to manipulate the specific item(s) by raising or decreasing the item(s) ratings or increasing or decreasing the related items' ratings.
3. Systems that show recommended items without a strong indication of popularity or affiliation between items, such as Amazon. As a result, attackers have no understanding of the system mechanisms.

Recommendation system developers must consider the attacker tactics and program the system to identify and remove any untrusted users from being part of the profiles used to produce recommendations.

5.3. Constructing Maximal User Path

As shown in the figure below, each user targets items and gives their ratings of those items. The ratings influence those items to be the top choices. Others will probably allocate very little to nuke individual objects at the lowest level far below the average weighted µ of the item. An average is normally relative to all items rated.

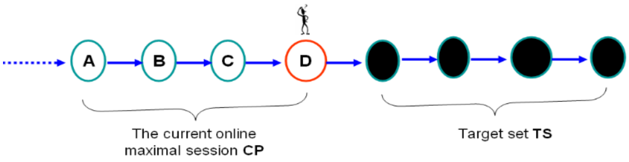

Figure 3: User-Targeted Items

We consider a maximal route occurred during a visit (session), as shown in the figure below, where there is no loop happened in a maximal path. The user may have several visits during which the rating occurred, as shown in the below figure, the user might be a genuine or an attacker. Whether a genuine person or an attacker, a user tries to check certain nodes, so we need to assess his traversing rates.

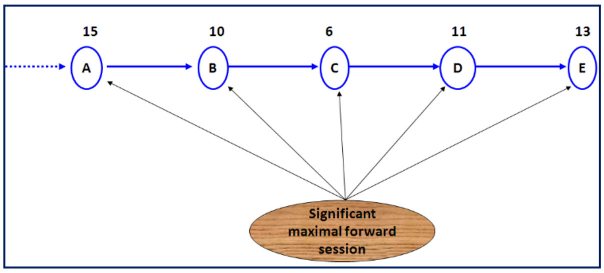

Figure 4: User Maximal Path

We begin with each user session, and gradually explore the data, moving from s_i. And as for all of the rates R_i, they do not exceed the limits. The user session is entered into the system of that session.

$$s_i = \{ R_1, R_2, \ldots., R_{n-1}, R_n \}$$

By measuring the session maximal path average rate, we can find the rate deviation of the user's rating of the node's visited and rate by looking at the other user's ratings of the user rated objects.

$$\mu_{s_i} = \frac{\sum_{i=1}^{n} R_i}{n} \qquad (1)$$

$$\sigma(s_i) = \sqrt{\frac{\sum_{j=1}^{l}(|\mu_{s_i} - R_{u_j}^{i}|)^2}{n}} \qquad (2)$$

where μ_{s_i} represents the mean weight of that path by all users and $\sigma(s_i)$ represents the calculated standard deviation of all rates happened on that maximal path.

5.4. Detect Untrusted Users

The two types of attacks are very severe from average users, but this does not mean that a normal user does not score an object at a low or high rate. However, it is uncommon for the same user to rate multiple items with very low or very high ratings relative to other users that rate the same item on similar maximum paths.

We then found the following rule to use every rate state (melting, freezing, or normal).

where F(s_i) is the evaluation function of each session as a collection of rankings made through a visit, $\mathfrak{I}_+$ represents the case when the user aims to push ratings far from the normal ratings (the freeze status), and $\mathfrak{I}$ reflects ratings happened within the threshold boundaries, while $\mathfrak{I}_-$ reflects the case when the user aims to downgrade target items (the melting status) as shown in the figure below.

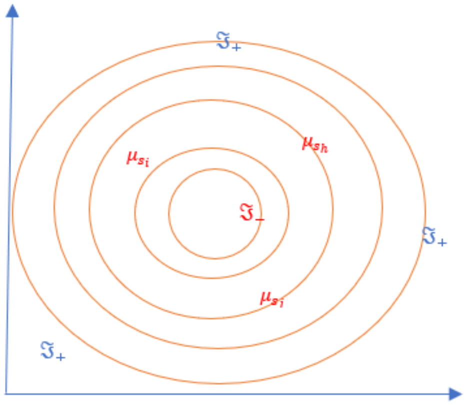

Figure 5: User Rates Zone

where,
Freeze case ($\mathfrak{I}_+$) = $\{R_i \mid R_i > \mathfrak{I}_+\}$
Meting case ($\mathfrak{I}_-$) = $\{R_i \mid R_i < \mathfrak{I}_-\}$
Normal case ($\mathfrak{I}$) = $\{R_i \mid R_i \cong \mathfrak{I} \; or \; R_i \cong \mu_{s_i}\}$

Therefore, $\mathfrak{I}_+$ & $\mathfrak{I}_-$ users should be eliminated from being part of recommendation generation processing.

Table 1: Attack Status

Attack model	Zone	Status
push attack	Freeze zone	$\mathfrak{I}_+$
No attack	Normal	$\mathfrak{I}$
Nuke attack	Melting zone	$\mathfrak{I}_-$

Table 1 shows that $\mathfrak{I}_+$ reflects the highest rates could be reached by attackers in the rating; for specific items or maximal path, far beyond the standard deviation. And $\mathfrak{I}_-$ reflect the customer's minimum prices relative to the measured average rate and the selected products' standard deviations. While $\mathfrak{I}$ reflects the normal rates of the trusted user obtained by the recommendation systems.

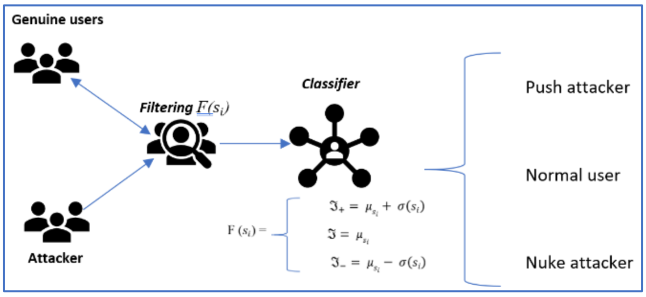
Genuine users
Filtering $F(s_i)$
Classifier
Attacker
$F(s_i) =$
$\mathfrak{I}_+ = \mu_{s_i} + \sigma(s_i)$
$\mathfrak{I} = \mu_{s_i}$
$\mathfrak{I}_- = \mu_{s_i} - \sigma(s_i)$
Push attacker
Normal user
Nuke attacker

Figure 6: Suggested Model Stages

5.4.1. Algorithm for Detecting an Attack

The following algorithm demonstrates the followed steps to detect a user rating zone.

```
Input: users ratings or maximal path in a session.
Output: filter users into trusted, untrusted (melting, freeze).
1. Read a user $u_i$ session $s_i$
2. $\forall_i \in s_i$, calculate $\mu_{s_i}$, and $\sigma(s_i)$
3. Read UT
4. While not UT
     For R in $s_i$:
        a. if $R_i > \mathfrak{I}_+$
              i. Detected Freeze case
              ii. Refuse the whole session
              iii. Untrusted user is highlighted
              iv. Update UT ← $u_i$
        b. Else if $R_i < \mathfrak{I}_-$
              i. Detected Melting case
              ii. Refuse the whole session
              iii. Untrusted user is highlighted
              iv. Update UT ← $u_i$
        c. Else
              i. Detected Normal case
              ii. Accept the whole session ratings
              iii. Trusted user is highlighted
              iv. Update U ← $u_i$
5. Return zone
```

where R reflects items ratings in a session s, and ***UT*** reflects the untrusted users discovered and recorded on the system. The system would automatically maintain its functionality by upgrading any identified untrusted user to untrusted profiles that make sense of removing any ratings received from these users in the future. In contrast, the normal user ratings should be updated to the normal profiles used to produce recommendations.

5.5. The Contribution of the Manuscript

This paper's main contribution is implementing a new algorithm to filter user rating injections and item rating injections to prevent false similarities. A matrix of ratings is used in the processing to identify the trust of the user. Therefore, if the user ratings exceed a particular computed ceiling, the system routinely denies the user scores and moves them in an untrustworthy category, and the user ratings are not considered in the recommendation creation.

6. Experimental Results

We use the available online Movie Lens 1M data set to apply and test the proposed algorithm, comprising 1,000,000,000 unidentified reviews of roughly 3,900 films received by 6,040 users of Movie Lens. All scores are 1 to 5 integers, with 1 being the lowest, and 5 being the highest. There are 18 separate domains in the data set and all users with at least 20 films rated. The proposed algorithm (see section 3.4.1) was able to detect cases near the freezing, and those were very close to the melting zone. Total cases were 11364): Normal ratings (2677), Malting near (3943), Freeze near (4611), found attacks (133) as demonstrated in the table below.

Table 2: Users- Malicious Detection

Domain name	Normal	Malting touch	Freeze touch	Detected Attacks
Action	240	158	154	3
Adventure	123	216	349	2
Animation	157	226	216	0
Children	125	152	173	9
Comedy	151	267	248	50
Crime	104	348	247	0
Documentary	177	326	536	4
Drama	173	158	195	15
Fantasy	123	234	165	0
Film-Noir	140	217	300	3
Horror	157	154	172	8
Musical	125	150	150	0
Mystery	141	148	141	11
Romance	177	178	272	0
Sci-Fi	173	353	467	12
Thriller	176	277	286	0
War	105	211	380	7
Western	110	170	160	9

In each movie domain, the following figure shows user patterns; it is clear that several freezes touch cases when user ratings exceed the normal ratings; the same is obvious in the melting zone. The experimental results showed that the algorithm detected 50 attacks from the comedy group, while many cases were under border conditions. The clearly reported attacks in the trained data showed that the attached percentage are as follows, in Action (2%) , Adventure(2%),Animation (0%), Children (7%), Comedy(38%), Crime(0%), Documentary(3%), Drama(11%), Fantasy(0%), Film Noir(2%), Horror(6%), Musical(0%), Mystery(8%), Romance(0%), Sci-Fi(9%), Thriller(0%), War(5%), Western(7%).

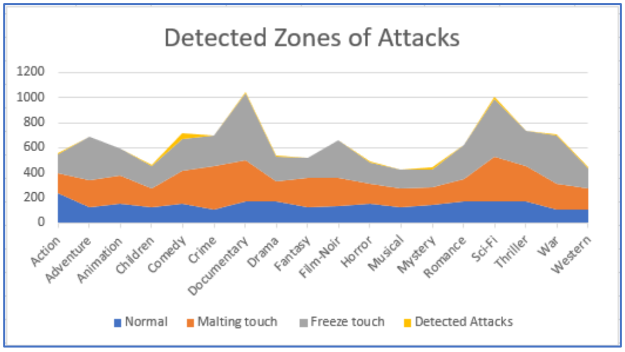

Figure 7: Attacks Detections For 300 Users

We expanded the data set and considered 450 users; the algorithm could detect more attacks, as seen in the figure below. This represents the incremental number of maximum paths that are being tested and the rise in the number of attacks observed by improvements in the slander of the deviation and the zones' size.

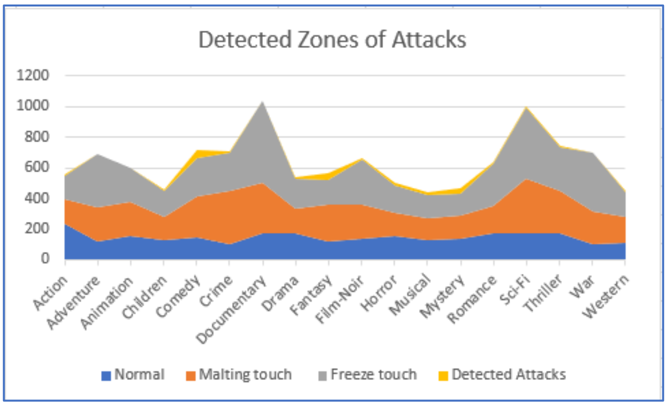

Figure 8: Attack Detection For 450 Users

The following table demonstrates a comparison between the proposed algorithm and another two algorithms were used to detect malicious attacks in recommendation systems.

Table 3: Algorithms Comparison

Points	Proposed algorithm	Rating behaviour	Profile similarity
Processing	Find the maximal path or sequence of items ratings	Pinpoint lower rating values considering that it has a minimal contribution to the systems	Discover profiles similarity
System behaviour	Classify users in three zones Normal, Melting and Freeze before updating their datasets	Detect all lower rating profiles.	Detect all identical profiles in ratings
Actions	Use trustworthy users to make recommendations	Ignore low rate profiles and only use moderately to highly qualified profiles	Ignore all identical profiles
Privacy concerns	Users' demographic data are not collected, only their maximum routes or sequence of items are collected. No privacy concerns.	The issue of privacy is valid.	Privacy is a valid issue.
Effective against	Various attack forms.	Random and Reverse	Average, Segment,

		Bandwagon attacks	and Bandwagon attacks
System reacts	The system will continue to serve untrusted users but will ignore their preferences for making recommendations.	The system eliminates profiles of an attacker	The system removes the profiles of attackers

Different attacks can be conducted using the proposed algorithm by finding the maximum path and measuring how far it is from the abstract user path. This helps classify users into different three zones; Normal, Melting (extreme below the threshold), and Freeze (extreme high from the threshold). The system then can ignore both Melting& Freeze cases and maintain only the normal cases for recommendation processing.

7. Conclusion and Future Works

Shilling attacks are a significant challenge to collaborative filtering recommendation systems. These attack profiles are likely to have similar rating information to many legitimate profiles to make them difficult to identify. This paper proposed an unsupervised detection method for detecting attacks (or irregular ratings) consisting of three phases. In the first phase, we need to understand the attacker's strategy based on the recommendation generation mechanisms. In the second phase, we built users' maximal paths and calculated the maximal path's mean and standard deviation considering all users' ratings visited that particular maximal path. In the third stage, we detected untrusted users by detecting the user's zone (normal, freeze or melt zone area) which reflect the gap between the user's ratings in the maximal path (items) and the genuine users' ratings of the same path(items) considering the maximum expected calculated deviation. All detected untrusted users' maximal sessions should not be added to the system profiles used for generating recommendations, and the user is labelled as untrusted profiles in which the system should eliminate any future participation. Experimental results showed that with a greater number of maximum routes used by the recommendation systems, the model could detect more attacks and thus discover untrusted users and prevent them from affecting their performance. The proposed method's main limitations include the following two aspects: (a) How the system could work in the launching phase when there are no ratings in the system collected from users. However, in the beginning, various methods can be used to overcome such systems' cold start, such as developing a suggestion considering the like-minded users when navigating the system. (b) The proposed approach abstractly handles users regardless of their identities, representing a good side of the proposed approach, but if the same user joins the system as a new user, the system will perform the calculations again for that user. One of our future objectives is to apply the suggested algorithm on a larger data set and apply it to live stream data in collaborative systems.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] O. Embarak, "Demolish falsy ratings in recommendation systems," in 2019 Sixth HCT Information Technology Trends (ITT), IEEE: 292–295, 2019.

[2] N. Sivaramakrishnan, V. Subramaniyaswamy, L. Ravi, V. Vijayakumar, X.-Z. Gao, S.L.R. J I.J. of B.-I.C. Sri, "An effective user clustering-based collaborative filtering recommender system with grey wolf optimisation," International Journal of Bio-Inspired Computation, **16**(1), 44–55, 2020. https://doi.org/10.1504/IJBIC.2020.108999

[3] Y. Pan, D. Wu, D.L. J D.S.S. Olson, "Online to offline (O2O) service recommendation method based on multi-dimensional similarity measurement," Decision Support Systems, **103**, 1–8, 2017. ttps://doi.org/10.1016/j.dss.2017.08.003

[4] Z. Yang, L. Xu, Z. Cai, Z. %J K.-B.S. Xu, "Re-scale AdaBoost for attack detection in collaborative filtering recommender systems," Knowledge-Based Systems **100**, 74–88, 2016.

[5] W. Zhou, J. Wen, Q. Xiong, M. Gao, J. J N. Zeng, "SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems," Neurocomputing **210**, 197–205, 2016.

[6] M. Si, Q. %J A.I.R. Li, "Shilling attacks against collaborative recommender systems: a review," Artificial Intelligence Review **53**(1), 291–319, 2020.

[7] S. Alonso, J. Bobadilla, F. Ortega, R. J I.A. Moya, "Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems," EEE Access, **7**, 41782–41798, 2019. ttps://doi.org/10.1016/j.ds.2017.08.003

[8] Z. Yang, Z. Cai, X. %J K.-B.S. Guan, "Estimating user behavior toward detecting anomalous ratings in rating systems," Knowledge-Based Systems **111**, 144–158, 2016.

[9] K.G. Saranya, G.S. Sadasivam, M. %J I. journal of science Chandralekha, Technology, "Performance comparison of different similarity measures for collaborative filtering technique," Indian journal of science and Technology, **9**(29), 1–8, 2016.

[10] B. Li, Y. Wang, A. Singh, Y. Vorobeychik, "Data poisoning attacks on factorization-based collaborative filtering," in Advances in neural information processing systems, 1885–1893, 2016.

[11] M. Fang, G. Yang, N.Z. Gong, J. Liu, "Poisoning attacks to graph-based recommender systems," in Proceedings of the 34th Annual Computer Security Applications Conference, 381–392, 2018.

[12] M.N. TEKLEAB, RECOMMENDATION SYTEM ANALYSIS AND EVALUATION, NEAR EAST UNIVERSITY, 2019.

[13] A.M. Turk, A. %J E.S. with A. Bilge, "Robustness analysis of multi-criteria collaborative filtering algorithms against shilling attacks," **115**, 386–402, 2019.

[14] N. Nikzad–Khasmakhi, M.A. Balafar, M.R. %J E.A. of A.I. Feizi–Derakhshi, "The state-of-the-art in expert recommendation systems," **82**, 126–147, 2019.

[15] E. Çano, M. %J I.D.A. Morisio, "Hybrid recommender systems: A systematic literature review," **21**(6), 1487–1524, 2017.

[16] C. Wang, Y. Zheng, J. Jiang, K. %J E. Ren, "Toward privacy-preserving personalized recommendation services," **4**(1), 21–28, 2018.

[17] M. Sappelli, S. Verberne, W. %J J. of the A. for I.S. Kraaij, Technology, "Evaluation of context-aware recommendation systems for information re-finding," **68**(4), 895–910, 2017.

[18] J. Su, Content based recommendation system, 2017.

[19] L. Yang, W. Huang, X. %J I.E.T.I.S. Niu, "Defending shilling attacks in recommender systems using soft co-clustering," **11**(6), 319–325, 2017.

[20] V.W. Anelli, Y. Deldjoo, T. Di Noia, E. Di Sciascio, F.A. Merra, "Sasha: Semantic-aware shilling attacks on recommender systems exploiting knowledge graphs," in European Semantic Web Conference, Springer: 307–323, 2020.

[21] A.P. Sundar, F. Li, X. Zou, T. Gao, E.D. %J I.A. Russomanno, "Understanding Shilling Attacks and Their Detection Traits: A Comprehensive Survey," **8**, 171703–171715, 2020.

[22] P. Kaur, S.G. Goel, Shilling Attack Detection in Recommender Systems, 2016.

[23] X. Li, M. Gao, W. Rong, Q. Xiong, J. Wen, "Shilling attacks analysis in collaborative filtering based web service recommendation systems," in 2016 IEEE International Conference on Web Services (ICWS), IEEE: 538–545, 2016.

[24] M. Ebrahimian, R. Kashef, "Efficient Detection of Shilling's Attacks in Collaborative Filtering Recommendation Systems Using Deep Learning Models," in 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), IEEE: 460–464, 2020.

[25] I. Gunes, H. %J I.R.J. Polat, "Detecting shilling attacks in private environments," **19**(6), 547–572, 2016.

[26] Y. Hao, F. Zhang, J. Wang, Q. Zhao, J. %J S. Cao, C. Networks, "Detecting shilling attacks with automatic features from multiple views,"

2019, 2019.

[27] V. Mohammadi, A.M. Rahmani, A.M. Darwesh, A. %J H.C. Sahafi, I. Sciences, "Trust-based recommendation systems in Internet of Things: a systematic literature review," **9**(1), 1–61, 2019.

[28] S. Khusro, Z. Ali, I. Ullah, "Recommender Systems: Issues, Challenges, and Research Opportunities," Lecture Notes in Electrical Engineering, **376**, 1179–1189, 2016, doi:10.1007/978-981-10-0557-2_112.

[29] O. Embarak, M. Khaleifah, A. Ali, "An Approach to Discover Malicious Online Users in Collaborative Systems," in International Conference on Emerging Internetworking, Data & Web Technologies, Springer: 374–382, 2019.

[30] Y. Cai, D. %J D.S.S. Zhu, "Trustworthy and profit: A new value-based neighbor selection method in recommender systems under shilling attacks," **124**, 113112, 2019.

[31] D. Lee, P. Brusilovsky, Recommendations based on social links, Springer Verlag: 391–440, 2018, doi:10.1007/978-3-319-90092-6_11.

[32] K. Christakopoulou, A. Banerjee, "Adversarial attacks on an oblivious recommender," in Proceedings of the 13th ACM Conference on Recommender Systems, 322–330, 2019.

[33] C.-M. Lai, Attackers' Intention and Influence Analysis in Social Media, University of California, Davis, 2019.

[34] O.H. Embarak, "Like-minded detector to solve the cold start problem," in 2018 Fifth HCT Information Technology Trends (ITT), IEEE: 300–305, 2018.