IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 13, 2020, accepted September 1, 2020, date of current version September 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022962

Understanding Shilling Attacks and Their Detection Traits: A Comprehensive Survey

AGNIDEVEN PALANISAMY SUNDAR[1], FENG LI[2], XUKAI ZOU[3], TIANCHONG GAO[4], (Member, IEEE), AND EVAN D. RUSSOMANNO[5]

[1]Department of Computer and Information Science, Indiana University–Purdue University Indianapolis, Indianapolis, IN 46202, USA
[2]Department of Computer Information and Graphics Technology, Indiana University–Purdue University Indianapolis, Indianapolis, IN 46202, USA
[3]Department of Computer and Information Science, Indiana University–Purdue University Indianapolis, Indianapolis, IN 46202, USA
[4]School of Cyber Science and Engineering, Southeast University, Nanjing 210000, China
[5]Department of Information Systems and Operations Management, Ball State University, Muncie, IN 47306, USA

Corresponding author: Feng Li (fengli@iupui.edu)

This work was supported in part by the U.S. National Science Foundation under Grant CNS-1852105, Grant CNS-1560020, and Grant OAC-1839746.

ABSTRACT The internet is the home for huge volumes of useful data that is constantly being created making it difficult for users to find information relevant to them. Recommendation System is a special type of information filtering system adapted by online vendors to provide recommendations to their customers based on their requirements. Collaborative filtering is one of the most widely used recommendation systems; unfortunately, it is prone to shilling/profile injection attacks. Such attacks alter the recommendation process to promote or demote a particular product. Over the years, multiple attack models and detection techniques have been developed to mitigate the problem. This paper aims to be a comprehensive survey of the shilling attack models, detection attributes, and detection algorithms. Additionally, we unravel and classify the intrinsic traits of the injected profiles that are exploited by the detection algorithms, which has not been explored in previous works. We also briefly discuss recent works in the development of robust algorithms that alleviate the impact of shilling attacks, attacks on multi-criteria systems, and intrinsic feedback based collaborative filtering methods.

INDEX TERMS Collaborative filtering, detection traits and algorithms, profile injection attacks, robust algorithms, shilling attacks.

I. INTRODUCTION

We live in the information age where there is an overload of information generated by individuals, companies, and governments. The internet has become a common platform for all of this information to be shared and stored. Multiple e-commerce platforms have come into existence, selling all kinds of products and services. With this information overload, it has become increasingly difficult for online users to find content relevant to them. As a means of addressing this problem, many websites are utilizing the recommender system [1]. The recommender system is an information filtering mechanism to provide customers with products/services based on their requirements.

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

Multiple Recommender System approaches are employed to cater to different kinds of needs in different websites. Over the years, there has been a drastic growth in the methods used to improve recommendation results for different purposes [2]–[8]. Recommendation systems can be broadly classified into two types, content-based [9]–[14] and collaborative filtering-based [15]–[20].

Content-based filtering recommends products to users by comparing the content of the products to the users' profiles. The downside of using content-based filtering is the over-specialization; they tend to recommend only the products that are very similar to what has already been consumed by the user which wasn't the case with collaborative filtering. The collaborative filtering recommender system works by analyzing the past behavior of a user. The key idea is that users with similar behavior have similar needs and interests. Recommendations made using collaborative filtering depend

on relationships between the users and items. Unfortunately, due to its openness and dependency on user ratings, collaborative filtering is prone to shilling attack, also known as a profile-injection attack.

Shilling attack [21]–[25] is a particular type of attack where a malicious user profile is inserted into an existing collaborative filtering dataset to alter the outcome of the recommender system. The injected profiles explicitly rate items in such a way that the target item is either promoted or demoted. It has been a topic of study for over a decade, and multiple survey papers have covered different parts of this domain. In [26], Mehta *et al.* focus exclusively on robust collaborative filtering techniques and not on detection techniques or attack strategies. In [27], the types of attacks and the detection techniques discussed are limited. In 2014, [28] produced one of the most comprehensive surveys on the topic, but it presents details on the attacks only until 2011. The survey in [23] focuses only on the statistical measures used in the detection and the basic shilling attack methods. Kaur and Goel [29] perform experimental evaluation comparing the most commonly used shilling attack methods. In [30] and [22], the discussions do not consider the different detection attributes used in supervised and unsupervised detection methods. Both [24] and [25] briefly discuss the various attack and detection methods. There is no discussion on robust algorithms, and the detection methods are not categorized.

This paper aims to be a comprehensive survey of different attack models and detection attributes for shilling attacks on collaborative filtering recommender systems. Since shilling attacks are more prominent in explicit rating systems, this survey's scope is limited to methods that work on explicit rating systems where the user explicitly gives one rating for each item. Shilling attacks are possible in both nearest-neighbor-based and matrix factorization-based recommender systems; it is predominantly tested in nearest-neighbor settings, which will be used in our explanations. We explain the collaborative filtering with examples in Sect. 2. Sect. 3 discusses the attack profiles and the various attack models, and Sect. 4 contains the detection attributes. Sect. 5 details the detection algorithm along with the targeted traits which are not discussed in earlier works. We also briefly introduce the impact of shilling attacks on multi-criteria and implicit feedback systems in Sect. 6. Finally, we conclude our paper and give some future directions in Sect. 7.

II. COLLABORATIVE FILTERING

Collaborative filtering uses the user-item interaction data related users and items to make recommendations. It can further be broadly divided into user-based and item-based collaborative filtering. Typically, a user-based collaborative filtering system consists of an $m \times n$ matrix with m users and n items. Each element in the matrix represents the ratings given by the user for that item/product. The User-Item matrix, also referred to as the utility matrix, is incomplete as most users would not have rated all the items. Each line of the utility matrix denotes the behavioral history of one user. Consider a

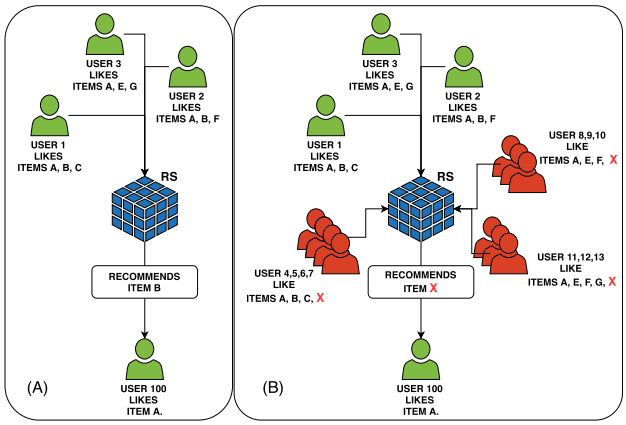


FIGURE 1. A toy example of the influence of a shilling attack.

system with only two users A and B, who have given similar ratings to products p_1, p_2 and p_4. If user B gives a high rating to product p_3, then p_3 will also be recommended to user A. The process is to find top X similar users to the target user u, then calculate the product ratings for user u based on similar users' ratings. The top N products with high ratings that have not yet been rated by user u are recommended. On the other hand, item-based collaborative filtering functions by forming an item-item matrix to determine the relationship between each pair of items. Here, the recommendations are based on the other items that the user has purchased. For example, consider that multiple users give high ratings to both product p_1 and product p_2. This causes p_1 and p_2 to have high correlation in item-based CF. If a new user gives a high rating to p_1, then product p_2 will also be recommended to that user. The ability to work with sparse data and easier maintenance are some of the advantages that item-based CF had over user-based CF. Both these methods are widely used in different recommendation tasks depending on the system's requirements.

III. SHILLING ATTACKS

Shilling attacks can be classified based on intent as a push or nuke attack, where a product is either promoted or demoted, respectively, to gain an economic advantage over competitors. Fig.1 gives an example of the impact of a shilling attack on a recommender system. Here, item X is the target item that is promoted by the shilling attack. Over the years, multiple attack profiles and models have been developed [21], [31]–[36]. Simultaneously, many detection techniques and algorithms have emerged to counter such attacks [37]–[42]. Almost all of the attack models use the same attack profile while generating malicious users. The attack models' differences are attributed to how the individual elements of the attack profiles are formed.

A. ATTACK PROFILE

The attack profile is segmented into four sets: Selected items I_s, Filler items I_f, Null items $I_\emptyset$, and the Target item(s) I_t. I_t is the set of items or an individual item which needs to be pushed or nuked. I_s is the set of items carefully chosen so

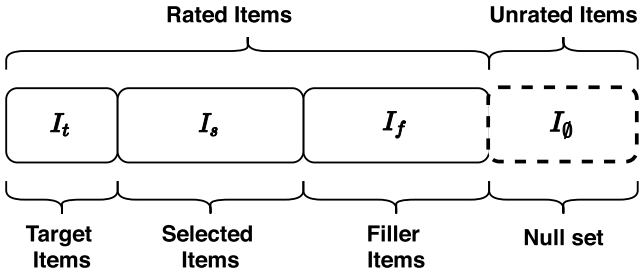


FIGURE 2. Attack profile.

that a malicious profile has a similarity with the maximum possible number of genuine users. The efficiency of an attack is decided by how many users are recommended with the target item. I_s plays a crucial role in attack efficiency. I_f is the set of filler items chosen and rated in such a way that the malicious profiles can camouflage with the genuine profiles. $I_\emptyset$ is the set of items that are not rated by the malicious user [43]. Fig.2 illustrates an attack profile.

1) ATTACK SIZE

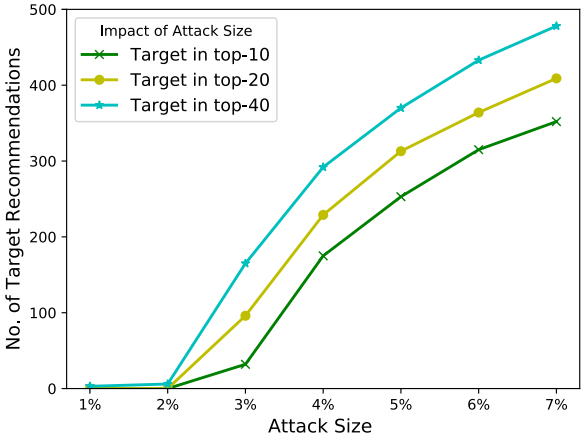
The number of injected profiles and the number of items rated per profile considerably influences an attack's reach. The number of injected profiles, also known as attack size, should be large enough to have any impact on the system. Fig.3a shows the increase in the reach of the target item with respect to the attack size. The MovieLens dataset [44] with 100,000 movie ratings from 943 users on 1682 items were used for the generation of this graph. A random attack discussed in the next section, with various attack sizes (1% to 7% of the number of authentic users), was implemented. A movie with an average rating of 1.9 calculated from 31 authentic ratings was chosen as the target item. Before the attack, the target item was not part of the top-40 recommendations made to any of the authentic users using a kNN-based algorithm. Fig. 3a shows the number of users who have the target item in their top-10,20 and 40 recommendations after the attack. The graph shows that the target item reaches more people as the attack size increases. The number of filler items per attack profile was fixed at 2% of the total number of items.

2) FILLER LENGTH

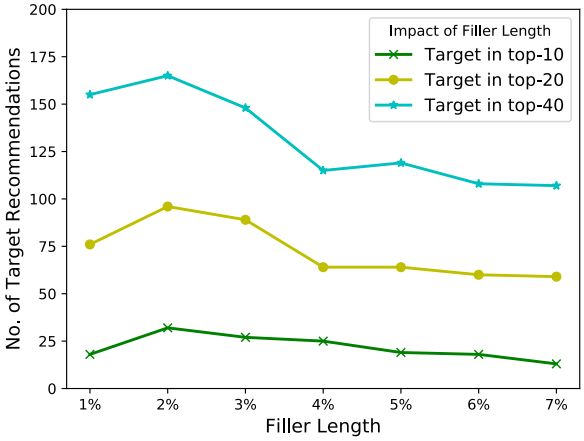
The number of filler items rated per injected profile is known as the filler size. Fig.3b shows the impact of increasing the number of rated items, also known as filler length, on the target recommendation. For the evaluation of this graph, the number of injected profiles was fixed at 3% of total users. From this graph, it can be seen that increasing the filler length can be detrimental to attack efficiency, implying that high filler length can cause the attack profiles to be less similar to authentic users.

B. ATTACKS MODELS

Based on the attackers' motivation and knowledge, multiple attack models have been developed over the years. All these attacks can be categorized either as a high-knowledge attack or a low-knowledge attack. Low-knowledge attacks are more



(a) Reach of the target item in various attack sizes



(b) Influence of filler length for a fixed attack size

FIGURE 3. Number of users with the target item in their top-N recommendations after shilling attack.

practical and have a higher chance of having a real-world impact, but the efficiency of such attacks is also low. On the other hand, high-knowledge attacks can have a massive effect on Recommender Systems' performance, but they are harder to pull off. From a practical standpoint, an inside job is a viable option to execute a high-knowledge attack, the chances of which are negligible. So, in real-world applications, a moderately efficient low-knowledge attack poses a more significant threat than a highly efficient high-knowledge attack. Based on how the selected items and filler items are chosen, multiple attack models exist which can further be classified as standard or obfuscated, based on the attacks' ability to go undetected.

1) STANDARD ATTACKS

These are the attack models that do not make an exclusive attempt to go undetected in a recommender system. Many detection algorithms have a higher chance of detecting the shilling attack profiles injected using these attacks.

Random Attack [21], also known as the RandomBot attack, is the simplest form of shilling attack. In this model, the items rated by the attack profile are chosen at random

TABLE 1. Attack models.

Attack Models	I_s		I_f		I_t	Knowledge
	Selection	Rating	Selection	Rating	Rating	
Random	∅		Random	System mean	max/min	Low
Average	∅		Random	Item mean	max/min	High
Bandwagon	Popular	max/min	Random	System mean	max/min	Low
Reverse Bandwagon	Unpopular	min	Random	System mean	-/min	Low
Segment	Segmented	max/min	Random	System mean	max/min	Moderate
Love/Hate	∅		Random	max	-/min	Low
User Shifting	∅		Random	System mean + random	max/min	Low
Target Shifting	∅		Random	System mean	max-1/min+1	Low
AoP	∅		Top X% of popular	Item mean	max/min	Low
PIA	Power items	Item mean	∅		max/min	High
PUA	Power users' items & ratings		∅		max/min	High

except for the target item. The ratings for these items is around the system overall mean. The target item gets the maximum or minimum rating based on whether it is a push or a nuke attack. Some attacks are intended to disrupt the trustworthiness of a recommender system, known as random vandalism [30]. Being the most straightforward attack, it is also the least effective. The purpose of a random attack is usually more effective in disrupting the performance of a Recommender System rather than promoting the target item. The ease of execution of random attacks is because of its low-knowledge requirement. All that the attacker needs are the overall system mean which can be easily empirically calculated. Being the simplest attack, it is not very effective.

Average Attack [21] is similar to the random attack in terms of the item selection process. The randomly chosen items are rated based on the rating distribution of the individual items. Each filler item is assigned the mean rating of that item. This attack is feasible only if the attacker has immense knowledge about the dataset on which the recommender system is built. The effectiveness of this model is proportional to the attacker's knowledge. Though the only difference between random attack and average attack is the filler ratings, the average attack's effectiveness is much better.

Bandwagon Attack [31], [33] is the type of attack where the profiles generated by attackers are filled with popular items with high ratings. The attack profiles are naturally closer to a large number of users. The target item is given the highest rating. This attack can be further divided into bandwagon-random and bandwagon-average depending on the rating scheme used for the filler items. Bandwagon also falls under the low-knowledge attack category since the attacker only needs publicly available data.

Reverse Bandwagon Attack [32], [33] is the exact reversal of a bandwagon attack. This attack is used to nuke the target product by giving low ratings to the items with high negative reviews and giving the least rating to the target item. It is also a low-knowledge attack, just like the bandwagon attack. Though it is highly similar to the bandwagon attack, the efficiency of the reverse bandwagon attack is slightly better.

Segmented Attack [45] targets a specific group of users who are likely to purchase the target item in an e-commerce setup. Segment attacks are usually deployed in item-based collaborative filtering. The rated items and the ratings are based on the attacker's knowledge about the segment. The significant advantage that this method has over other methods is its ability to reach potential customers. For example, if the target item is a book in the science fiction genre, then the selected items will also be from the same genre. Such selection increases the chances of the target book reaching more fans of science fiction. Since the attack is deployed only in a segment of the system, the impact is high.

Probe Attack [46] is not an attack that can be generalized for all systems. Some recommender systems project a predicted rating score for each of the items. The attacker uses this detail to rate the items, enabling it to be similar to other users. The attacker gives genuine ratings to some seed items. Then, when the recommender suggests more items, the attacker forms the rated items list based on these items. This scheme ensures that the attack profiles stay close to its neighbors. It also enables the attacker to learn more about the system.

Love/Hate Attack [32] is a highly effective nuke attack. Here, the attacker randomly chooses filler items and gives them the highest ratings and the least rating to the target item. Despite the simplicity of this model, the effectiveness is surprisingly high. Though it was predominantly designed for nuke attacks, it can also be used for a push attack by altering the ratings. Push attack is not as effective as a nuke attack. Table 1. comprehensively summarizes the differences in various attack models.

2) OBFUSCATED ATTACKS

To go undetected from detection algorithms, attackers try to obfuscate their attack signature. Many models incorporate

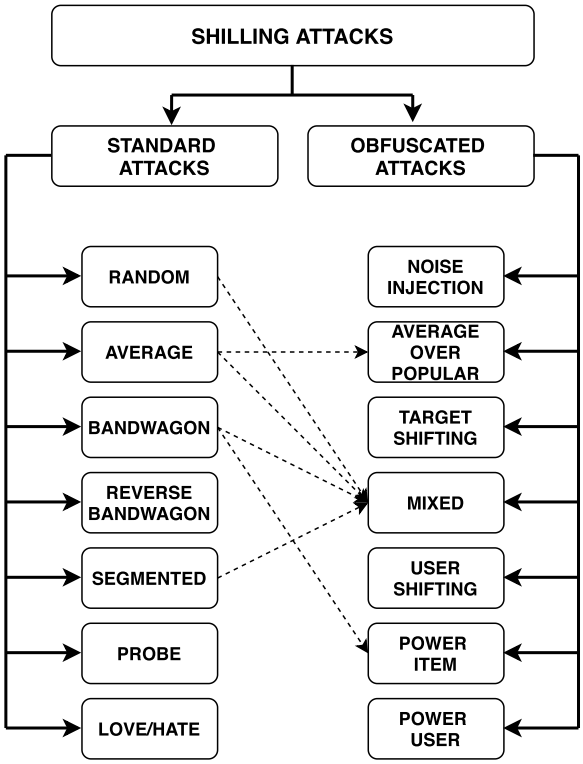


FIGURE 4. Types of shilling attack.

slight modifications to the standard attack techniques to achieve obfuscation. Fig. 4 shows which of the standard attacks have influenced which of the obfuscated ones. The dotted lines indicate a direct influence between the attacks. The ones that are not derived from specific standard attacks can be incorporated with any standard attack. Though obfuscation might slightly reduce the impact of the attack, it is better than being detected.

Noise Injection [47] adds to a Gaussian distributed random number multiplied by a constant to each rating, for a subset of injected profiles. The degree of obfuscation is dependent on the constant that is multiplied. It can be effectively applied to all of the standard attack methods to obfuscate its signature. Since the rating scheme is affected by noise injection, a slight but observable drop in the attack efficiency can be noticed.

User Shifting [47] is an obfuscation tactic where a subset of the rated item of each injected profile is modified. The ratings of this subset of items are either increased or decreased to reduce the similarity between attack profiles. For different groups of the injected profiles, different subsets of rated items have their ratings modified.

Target Shifting [47] shifts the rating of the target item to one level lesser than the highest possible in push attacks. In nuke attacks, the target rating is shifted to one rating higher than the least possible rating. This strategy is specifically useful in evading the detection methods that penalizes users that give an extreme rating to items. If the target item is already popular, it will be harder to push while employing target shifting obfuscation. In such cases, some other obfuscation methods should be used.

Average Over Popular [48] is a technique used to obfuscate the Average Attacks. Here, the filler items are chosen from the top X% of the most popular items with equal probability. This method is much more effective than randomly choosing from the entire collection of items. The choice of X influences the detectability of the attack.

Mixed Attack [49] is done by using the random, average, bandwagon, and segmented attacks in equal proportions, simultaneously. The detection technique should have the ability to detect all of the standard attacks to be successful. The different attack methods are used to push/nuke the same target item. It helps in evading multiple detection techniques.

Power Item Attack [36], [50] utilizes the power items which are chosen based on three methods. Power items are defined as the set of items that can influence the largest group of items. These items effectively alter the recommendations made for other users. In PIA-AS, the top-N items with the highest aggregate similarity are chosen to be the power items. Such similarity is possible only when a considerable number of users have rated the same two items. In PIA-ID, the In-Degree centrality is the criteria for choosing the power items. The similarity of each pair of items is calculated using weighted significance and the top-N of each item is selected. PIA-NR chooses the items with the highest number of users as the power items.

Power User Attack [36], [50], similar to PIA, chooses the set of users who have the maximum influence on the broadest group of users. In PUA-AS, the top X users with the highest Aggregate Similarity are chosen as the power users. In PUA-ID, the users who participate in the highest number of neighborhoods are selected as power users, based on the In-Degree centrality concept. The power users in PUA-NR are the users with the highest number of ratings in their profile.

SAShA [34] is an attack strategy that uses the semantic features extracted from a knowledge graph to improve the performance of standard CF attack models. A knowledge graph is a structured repository of factual, categorical, and ontological information [51]. This attack works by computing the semantic similarity between the knowledge graph derived features of the target item and all other items in the system. This information is leveraged to generate the most efficient set of filler items.

In [35], Chen *et al.* describe a method to use both rated item correlation and item popularity to generate malicious users with strong attack ability and similarity to real users. In their approach, each malicious user profile is generated individually. The rated items of a profile are selected based on a matrix of real user profiles.

As soon as the vulnerability of Collaborative Filtering to shilling attacks was discovered, various detection techniques were also constructed. We can broadly classify these techniques into supervised and unsupervised detection techniques. In literature, there is an array of detection attributes that govern these methods.

TABLE 2. Symbol definitions.

Symbols	Definitions
N_u	Number of ratings from user u
NR_i	Number of ratings for item i
$r_{u,i}$	User u's rating for item i
$\overline{r_i}$	Mean rating for item i
U	Total number of users
$\overline{u}$	Average length of a profile
P_u	Profile of user U
$P_{u,T}$	Target items in the profile
$P_{u,F}$	Filler items in the profile
I_u	Set of items rated by user u
U_u	Partition of the profiles of user u
$\|U_u\|$	Number of profiles of user u

IV. DETECTION ATTRIBUTES

The attributes that differentiate the shilling profiles from the authentic profiles are considered as the detection attributes. The detection attributes that are designed to work irrespective of the type of attack model are known as Generic attributes.

A. GENERIC ATTRIBUTES

The attributes that are not tailored for specific attack models fall under this category. The efficiency of these attributes alters with the different attack models used. Table. 2 gives the definitions for the symbols used in the explanations below.

Rating Deviation from Mean Agreement (RDMA) is the measure of rating deviation of a user on a set of target items with respect to other users, combined with the inverse rating frequency of these items [37].

$$RDMA = \frac{\sum_{i=0}^{N_u} \frac{|r_{u,i} - \overline{r_i}|}{NR_i}}{N_u} \tag{1}$$

Weighted Deviation from Mean Agreement (WDMA) is firmly based on the RDMA attribute. The significant difference of this attribute is that it places high weight for rating deviations for sparse items. WDMA was experimentally found out to give higher information gain [38].

$$WDMA = \frac{\sum_{i=0}^{N_u} \frac{|r_{u,i} - \overline{r_i}|}{NR_i^2}}{N_u} \tag{2}$$

Weighted Degree of Agreement (WDA) captures the cumulative differences of a user's rating of an item from the item's average rating, divided by the number of ratings for the item. WDA is empirically the same as the numerator of the RDMA [38].

$$WDA = \sum_{i=0}^{N_u} \frac{|r_{u,i} - \overline{r_i}|}{NR_i^2} \tag{3}$$

Length Variance (LengthVar) measures the difference in the length of a user's profile from the average length of a profile. Here, length denotes the number of items rated by a given user profile. Some attack profiles tend to have too many rated items, deviating substantially from an average user's length [38].

$$LengthVar = \frac{|N_u - \overline{n}|}{\sum_{k \in U} (n_k - \overline{n})^2} \tag{4}$$

B. MODEL SPECIFIC ATTRIBUTES

The problem with using only the generic attributes is that sometimes it is unable to distinguish malicious profiles from the authentic users, especially when the authentic user exhibit unusual behavior. Attack specific attributes were constructed to overcome these shortcomings. These detection attributes discover the partitions in user profiles so that their behaviors exhibit similarity to one particular attack model.

Mean Variance (MeanVar) is used to detect average attacks. It partitions the attack profiles into three parts: the items with extreme ratings (target items), all other rated items in profiles (filler items), and unrated items. This attribute works by computing the mean-variance between all the filler items and the overall average. A low variance would indicate the possibility of an average attack [38].

$$MeanVar = \frac{\sum_{j \in P_{u,F}} (r_{u,j} \overline{r_u})^2}{|P_{u,F}|} \tag{5}$$

Filler Mean Target Difference Model (FMTD) targets the segmented attack model. This attribute relies on the difference between ratings of the items in target partition and the items in filler partition [38].

$$FMTD = \left| \frac{\sum_{i \in P_{u,T}} r_{u,i}}{|P_{u,T}|} - \frac{\sum_{k \in P_{u,F}} r_{u,k}}{|P_{u,F}|} \right| \tag{6}$$

Filler Average Correlation (FAC) focuses on detecting the random attack model. When a random attack is executed, then the ratings given to the items are chosen at random. This attribute calculates the correlation between the ratings in the profile and the average ratings of the items. The correlation is expected to be low for random attacks [39].

$$FAC = \frac{\sum_{i \in I_u} (r_{u,i} - \overline{r_i})}{\sqrt{\sum_{i \in I_u} (r_{u,i} - \overline{r_i})^2}} \tag{7}$$

Filler Mean Difference (FMD) utilizes the fact that the filler items have a mean rating similar to the overall system average in the random attack model. If the mean ratings are similar, then the user profile could potentially be a random attack profile [39].

$$FMD = \frac{1}{U_u} \sum_{i=1}^{|U|} |r_{u,i} - \overline{r_i}| \tag{8}$$

V. DETECTION ALGORITHMS

The detection algorithms can be broadly classified into two: Supervised detection methods and Unsupervised detection methods. The supervised techniques require the data to be labeled during the training process, whereas the unsupervised approaches do not. The availability of labeled ground truth is minimal in the recommender system datasets. This downside has led to unsupervised approaches being adopted more than supervised in recent times.

A. TARGETED TRAIT

Most of the detection algorithms work by targeting a particular trait observed in the shilling attacks. Though obfuscation manages to evade detection to some extent, some innate qualities need to be present in an attack, to be effective. Such qualities are usually targeted by the detection algorithms, both in the supervised classification and the unsupervised clustering methods. We briefly discuss what some of those qualities are in this section.

1) USER-BASED TRAITS

The basic division of such detection traits comes from whether the detection algorithm is focusing on finding the attack user profiles or the items. In the user-based trait, the user's behavior is checked for anomalies, which can imply that the profile is fake.

1) *Similarity:* The similarity of a user profile to a large number of its neighbors is exhibited by most attack profiles.
2) *Size:* The size of an attack, the number of attack profiles injected, is relatively much smaller than the entire user set. This size difference, combined with the high similarity among them, prove to be useful resources in detection.

2) ITEM-BASED TRAITS

Most of the detection methods rely on the set of items rated by each profile to check if it is a fake profile or not. From a detection point of view, we can categorize the items in an attack profile into 2.

1) *Rated Items:* Rated items are the items that are used for supporting the push/nuke of the target profile. Both the selected and filler items fall into this category from a detection front.

   Length: The length of an attack profile, the number of items rated by an attack profile, is usually much higher than an ordinary profile. An attacker usually tries to increase the similarity between the attack profile and many other profiles by rating several filler items.

   Rating: The rating given to an item is maintained closer to the average rating of the item to ensure maximum similarity. Detection algorithms usually target such anomalous rating behaviors.
2) *Target Item:* The target item is the item that is promoted or demoted in an attack.

   Crowding: The concentration of users rating a target item will be abnormally high when an attack is executed. Such abnormalities have a sizeable effect on the overall rating of the item.

   Rating: The primary reason behind an attack is to modify the opinion about the target item among users. The opinion cannot be altered without giving the target item a high rating in the case of a push attack and the least possible rating in the case of a nuke attack. Usually, such ratings widely deviate from the authentic ratings given to the item.

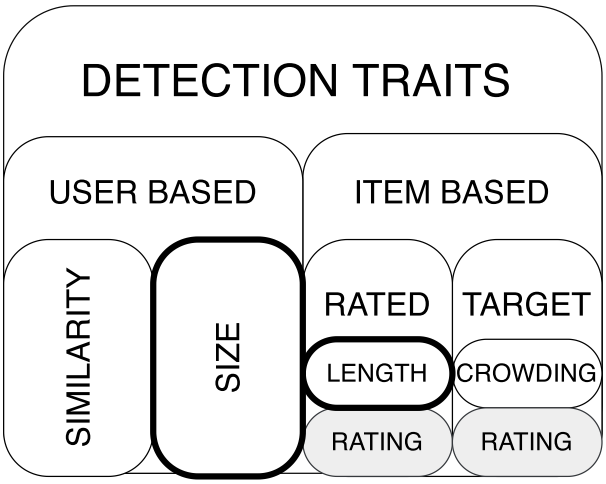


FIGURE 5. Characteristic traits of an attack which are exploited during the detection process.

Fig. 5 showcases the different types of traits used in detection. Here, both the attack size and filler length indicate the numerical differences in their behaviors and are detected using similar techniques. Likewise, the rating behavior is one of the most important features used in identifying an attack. Detection algorithms use the rating behavior differences in one form or the other in their algorithms.

B. SUPERVISED APPROACHES

The shilling attack problem was treated as a classification problem by Chirita *et al.* [37], used the RDMA and DegSim as the feature metrics for detecting malicious profiles. The method was developed to detect random and bandwagon attacks. Later on, two more generic metrics, namely WDMA and WDA, were added by Burke *et al.* [38] to improve the classifier's performance. SVM, kNN, and C4.5 were the most commonly used classifiers for the detection of fake injected profiles. The problem with using the generic attributes was that many authentic users who had extreme behaviors were misclassified as shilling profiles. To overcome this problem, as well as to improve the accuracy of the classifications, attack specific attributes were formulated by [38], [39]. Different attack specific attributes were formed for average, random, segment, and bandwagon attacks.

Williams *et al.* [52] utilized three strategies to increase the accuracy of detection in the supervised approaches: similarity to reverse-engineered attacks, target concentration, and rating anomaly detection. This detection technique is effective because of the added robustness to the system, but it is highly reliant on the classifier's choice. Their study shows that combining various attributes improves the classifier's performance, especially the support vector machine, and significantly reduces the impact of the most potent attack models. The attributes used in their method are RDMA, WDMA, DegSim, LengthVar, MeanVar, FMD, FAC, and FMTD.

The use of meta-learning was introduced by [53] to improve the precision of the detection. This algorithm can be

TABLE 3. Supervised classification based detection algorithms.

Detection Algorithms	Assumptions	Targeted Trait	Effective Against	Weak Against	Downside
Chirita *et al.* [37]	Attack profiles have high similarity with its neighbors	Rating Behavior	Random, Segmented	Average, Obfuscated	Authentic user misclassification
Burke *et al.* [38]	Items with fewer ratings have higher importance	Rating Behavior	Random, Segmented	Average, Obfuscated	Authentic user misclassification
Mobasher *et al.* [39]	Attack type is known	Filler rating variance, Profile length	Bandwagon, Random	Obfuscated	Being attack specific misses other attacks
Williams *et al.* [52]	Attack profiles have rating anomaly	Rating Behavior, Profile Similarity	Random, Bandwagon, Segmented	Obfuscated	Highly reliant on choice of classifier
Zhang *et al.* [53]	Multi-level classifiers improve efficiency	Profile Similarity and length	Random, Bandwagon, Average	Hybrid, Obfuscated	Two level classification leads to longer training time
Zhou *et al.* [54]	There are equal number of authentic and attack profiles	Target Analysis, Rating Behavior	Random, Average, Bandwagon	Hybrid, Obfuscated	Attack and authentic profiles need to be balanced to get good results
Yang *et al.* [41]	Weighted observations simulate balanced dataset	Filler Rating Variance and Length	PIA, PUA, Bandwagon	Probe	The process needs to be executed repetitively

considered a two-step process where the base-level training is done on attack profiles and available ratings. The second step is to combine the base-level output with the meta-level input for final attack detection. This algorithm had higher precision than previous methods. The diversity of the classifiers reduces the correlation of misclassification, positively impacting the meta-level prediction. They tested their approach against single SVM and voting SVM and experimentally proved to be more effective. The attributes used in their method are WDMA, RDMA, WDA, LengthVar, DegSim, MeanVar, FMD, and FAC.

SVM-TIA [54] had supervised, unsupervised, and semi-supervised detection approaches. The pitfall with using the supervised approach was that it needs a balanced data; it means that there should be an equal number of authentic profiles and attack profiles. The accuracy of the supervised approach was lower than their unsupervised approach which involved clustering and statistical methods. It is a two-phase process where rough detecting results are obtained in the first phase by alleviating class imbalance. In the second phase, the potential attack profiles are analyzed to discover the target profiles. Model-specific attributes like FMTD, MeanVar, FAC, and FMD are used in this method.

As mentioned earlier, the imbalance in the data available skewed the outcome of the supervised learning classifiers. AdaBoost was incorporated in [41] to diminish the perturbation caused by the imbalance. The authors first ease the hard classification task by using well designed features for the user profiles. It was achieved by applying weights to the various observations to accentuate the poorly modeled samples. This process was done repetitively to strengthen the correction of misclassification. The attributes used are RDMA, WDMA, WDA, LengthVar, MeanVar, FMTD, and FAC. In addition, they also use attributes that detect filler size with unpopular items.

Hao *et al.* [55] employed an ensemble detection method on features extracted from ratings, item popularity, and user-user graph. The feature extraction is performed by using Stacked Denoising AutoEncoders and PCA. It automatically extracts user features with different corruption rates. It used a three-stage process involving data preprocessing, feature extraction, and detection using weak classifiers. The novelty of items- the degree of difference between various items- was also used as a feature.

Table. 3 explains the different traits used for detection in some of the algorithms. It also discusses the various assumptions based on which the algorithms are built.

C. UNSUPERVISED APPROACH

The initial unsupervised approach introduced by Mehta *et al.* [56] applied Principle Component Analysis to the profile detection problem. Four factors led to this problem being suitable for PCA: spam users are highly correlated, low deviation from mean rating value, a high similarity with a large number of users, and the assumption that spam users work together. All the user profiles in the recommender system were projected onto a hyperplane formed from the user-item matrix. The user profiles which were clustered closer to the origin of the hyperplane were the attack profiles. The sparsity of the user-item matrix makes it harder for these predictions to be reliable. RDMA and WDMA are also used as detection attributes.

Bryan *et al.* [57] formulated a generic attribute aiding in the detection of attack profiles in an unsupervised manner. Their approach treats the attack profiles detection problem as an anomalous structure detection problem. The metric used is a variation of the Hv-score metric which was initially used in gene data analysis to aid in locating biclusters. This algorithm, called the UnRAP, seems useful in detecting both standard and obfuscated attacks. Their approach has better

chances of catching future novel attack strategies that may escape supervised methods.

Based on the assumption that attack profiles are lesser in number and exhibit high similarity, [49] applied an attribute-based k-means clustering technique. The users were divided into two clusters, and the smaller cluster was identified as attack profiles. This method showcased a higher accuracy and lesser misclassification of genuine users. Irrespective of the attack strategy used, this work claims to have fewer authentic user misclassifications than previous methods. The attributes used include RDMA, WDMA, WDA, and LengthVar, along with the Hv-score metric used in [57].

Chung *et al.* [58] applied the Beta distribution algorithm to detect attacks. This method detected as many attacks as possible without penalizing the authentic users. Most of the problems associated with this method were inherited from Beta probability distribution itself. The upsides of using this method are its low alarm rate and high detection rate. This method claims to work with sparse data and an unbalanced attack-normal profile ratio. This approach exhibits high performance even with a small attack size and has a low false alarm rate.

Another clustering approach relying on the attack profile similarities was [59], which used k-means clustering to move the fake profiles to the leaf nodes of a binary tree. With the user-item matrix and an optimal number of neighbors N, it recursively uses k-means clustering to cluster the users into two distinct groups. The indexed-cluster centers and intra-cluster correlation of the binary tree are used for attack profile detection. This approach's success rate is particularly high in the average, segment, and bandwagon attack models.

Yang *et al.* [60] developed an algorithm that focused on analyzing target users and items. It was a two-phase method. First, a density-based clustering method is applied to the dataset based on some selection features to identify malicious users. DBSCAN is used to determine the suspected users based on user features. Second, it spots suspicious items based on adaptive structure learning on selected features and further uses it to capture the attackers. The second phase helps in further scrutinizing the users from the first phase.

Zhang *et al.* [61] built a clustering approach based on the hidden Markov model (HMM) and hierarchical clustering. The users' rating behaviors are modeled using HMM. Based on the users' preference sequence and modeled rating behavior, each user's suspicion degree is calculated. Then, a hierarchical clustering method is used to group these users based on their suspicion degree into genuine and attack user clusters. They also apply their method on sampled Amazon review dataset to show its effectiveness.

Zhang *et al.* [62] proposed a method to improve the PCA approach in shilling profile detection. PCA is initially used to separate the profiles into two classes, positive labels for the detected and negative labels for all other users. Then they use the detection features - RDMA, WDMA, WDA, and LenVar - as data complexity features to calculate the CCMeasure of the dataset. CCMeasure is the classification complexity, a quantitative estimate on how difficult it is to classify the dataset. If the measure is high, it indicates that a significant number of authentic users are mislabeled, and the labels are flipped to reduce the data complexity.

Table. 4 shows the assumptions, traits, and the downsides of using some of these algorithms.

Having discussed detection techniques, other privacy risks that come with attack detection methods are also studied. Luo and Liang [63] discuss the impact of an insider attack on shilling attack detection for recommendation systems. They consider a possible scenario where an attacker poses as an examiner who is kept from individual rating profiles by secure computations. Their attack model can infer the target rating profile with little prior knowledge and the output of the secure computations. Such an insider attack would pose a serious threat to the privacy of users.

D. DEFENSE AGAINST SHILLING ATTACKS

Parallel to the works focusing on shilling attack detection, there is a line of research intended to create robust algorithms that are immune to shilling attacks. These algorithms do not have a mechanism to find and remove the shilling profiles but can reduce the attack's effectiveness. We briefly discuss some of the recent robust algorithms in this subsection.

Yang *et al.* [64] combined the soft co-clustering algorithm with the user propensity similarity method to enhance the robustness of the recommender system and detect shilling attacks. It uses Bayesian co-clustering, a soft co-clustering algorithm that allows mixed membership of row and column, highly suitable for real data. This model combines RDMA with soft co-clustering to reduce the influence of shilling attacks. All the attack profiles are clustered into the same cluster, limiting the shilling influence amongst the attack profiles.

Turk and Bilge [65] developed a robust multi-criteria collaborative filtering algorithm. A multi-criteria CF has multiple categories in which the user can rate each item. MCCF helps in better understanding the likes and dislikes of a customer. The robustness in their method is achieved by eliminating suspicious ratings based on the degree of uncertainty. The users are also categorized into different groups based on preference similarities to restrict authentic users from mixing with attack profiles.

Deng *et al.* [66] integrated entropy scaling into the collaborative filtering process to reduce the impact of over positive and negative users. They also used a minimum threshold to invert the entropy further assisting in the prevention of random attacks.

Alonso *et al.* [67] calculated a reliability value for each prediction of a user to an item. When an unusual change is observed in the item prediction's reliability value, it indicates a possible shilling attack. They use the Matrix Factorization method to neutralize the impact of a shilling attack. Promoting such shilling predictions can be avoided to reduce the extent of the attack and neutralize the presence

TABLE 4. Unsupervised clustering based detection algorithms.

Detection Algorithms	Assumptions	Targeted Trait	Effective Against	Weak Against	Downside
Mehta *et al.* [56]	Attack profiles are highly correlated	Rating Behavior, Profile Similarity	Random, Average	Segmented, Obfuscated	Sparse matrix reduces reliability
Bryan *et al.* [57]	Attack profiles exhibit anomalous structure	Rating Behavior	Bandwagon, Segmented	AoP, PIA, PUA	Misclassification of authentic users is higher in obfuscated attacks
Bhaumik *et al.* [49]	Attack profiles are smaller in number and exhibit high similarity with each other	Attack Size, Profile Similarity	Average, Segmented	Obfuscated	Effectiveness relies on attack size
Chung *et al.* [58]	Lower rating values have minimal contribution to the system	Rating Behavior	Random, Obfuscated	Probe, Average	All the downsides of using Beta probability apply to this method
Bilge *et al.* [59]	Attack profiles are similar to each other	Profile Similarity	Average, Segmented, Bandwagon	Obfuscated	Dependent on profile similarity, making it weaker against obfuscation
Yang *et al.* [60]	Most attack profiles exhibit detection features	Profile Similarity, Rating Behavior	Random, Bandwagon	Obfuscated	If attack profiles evade initial clustering, adaptive structure learning ineffective
Zhang *et al.* [61]	Authentic users do not exhibit extreme behavior	Rating Behavior	Random, Bandwagon, Segmented	Obfuscated	Authentic users with extreme behavior get misclassified.

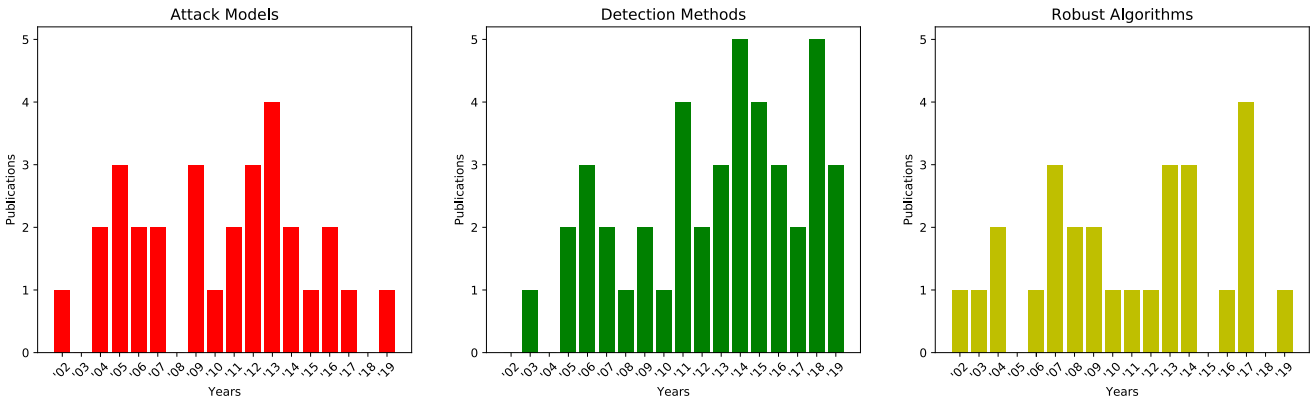


FIGURE 6. Number of publications each year.

of shilling profiles. This method's performance drops with a decrease in the size of the attack, but it is claimed that such a small attack size has a negligible impact.

Current Trends in Shilling Attack Research: Fig. 6 shows the number of publications that came out each year related to shilling attacks. This figure represents both single and multi-criteria rating systems from top conferences and journals covering both supervised and unsupervised methods. The initial stages in shilling attack research focused on creating new attack models to estimate the impact of different attacks on the recommender system. The standard attacks were created in the early 2000s, but the increase in detection techniques during these initial stages led to the research focusing more on obfuscated attacks. The number of papers related to attack models has declined in recent years, and the focus is shifted more towards detection techniques and robust algorithms. This Fig. 6 also shows the gradual growth in detection methods over the years. It is important to note that some of the detection related papers introduce a modified version of a known attack strategy which has a slightly better significant attack impact on the system.

VI. SHILLING ATTACKS IN MULTI-CRITERIA AND IMPLICIT FEEDBACK SYSTEMS

The multi-criteria system aims to find the reason behind a user's opinion about a product [68]. In such recommender systems, the user is asked to rate the same item on multiple categories, such as durability, service, etc. This feature helps in understanding the various aspects of a product. For instance, take the example of a user purchasing an item from an e-commerce website. Assume that the user likes the product but did not like the seller's service. In such cases,

having a single-criteria rating system does not efficiently capture the user's thoughts. This issue can be rectified by using a multi-criteria system. In [42], Turk *et al.* conduct a shilling attack on a multi-criteria based system by modifying random, average, bandwagon, reverse bandwagon, and love/hate attacks to fit the multi-criteria condition. They also introduce an attack method where the most repeated rating is assigned to the filler items, instead of average ratings. They experimentally show that the effect of using such a technique is superior to other methods. The existing literature on multi-criteria shilling attacks is limited.

Implicit feedback based recommender systems rely on a user's behavior, such as click, view, or purchase, to determine the likes and dislikes of the user [69]. The downside of using an explicit rating system is its intrusiveness. Most of the users end up not giving an explicit rating in e-commerce and other recommendation websites. Such cases lead to sparse data, subsequently leading to subpar performance of the recommender. With implicit feedback, the users' interaction with the website can be used to collect relevant data about a user ensuring consistent data collection, which eventually translates into good recommendations. Many sites employ a combination of explicit rating (single and multi-criteria) and implicit feedback systems. Shilling attack is possible in explicit rating systems because of the ease of attack implementation which is not valid with implicit feedback systems. So, a profile injection attack on implicit feedback systems is a possible future direction.

VII. CONCLUSION AND FUTURE WORK

In this survey, first we discuss the different shilling attack types and describe them briefly. Second, we analyze how some of the obfuscated attack models are derived from the standard attacks. Third, we define the various detection attributes which are widely used in multiple detection techniques. Fourth, we interpret and categorize the characteristic traits that are used in the detection process. We then concisely examine the various detection and robust algorithms available. Finally, we also briefly address the impact of shilling attacks on multi-criteria rating systems and implicit rating systems.

In the future, we plan to work on attack possibilities and detection methods for multi-criteria collaborative filtering. Although many shilling attacks and prevention techniques exist for collaborative filtering, there is not enough research related to attacks on the graph-based recommendation system and implicit feedback systems. We will explore the feasibility of extending shilling attacks on these recommender systems.

REFERENCES

[1] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. supported Cooperat. Work (CSCW)*, 1994, pp. 175–186.

[2] T. Osadchiy, I. Poliakov, P. Olivier, M. Rowland, and E. Foster, "Recommender system based on pairwise association rules," *Expert Syst. Appl.*, vol. 115, pp. 535–542, Jan. 2019.

[3] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi, "Top-K off-policy correction for a REINFORCE recommender system," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 456–464.

[4] F. Amato, V. Moscato, A. Picariello, and F. Piccialli, "SOS: A multimedia recommender system for online social networks," *Future Gener. Comput. Syst.*, vol. 93, pp. 914–923, Apr. 2019.

[5] Y. Qian, Y. Zhang, X. Ma, H. Yu, and L. Peng, "EARS: Emotion-aware recommender system based on hybrid information fusion," *Inf. Fusion*, vol. 46, pp. 141–146, Mar. 2019.

[6] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Syst. Appl.*, vol. 92, pp. 507–520, Feb. 2018.

[7] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention networks," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2018, pp. 3926–3932.

[8] M. Cerna, "Modified recommender system model for the utilized eLearning platform," *J. Comput. Educ.*, vol. 7, no. 1, pp. 105–129, Mar. 2020.

[9] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proc. Mach. Learn. New Inf. Age, MLnet/ECML Workshop*, vol. 30, 2000, pp. 47–56.

[10] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," *Knowl.-Based Syst.*, vol. 157, pp. 1–9, Oct. 2018.

[11] N. A. Albatayneh, K. I. Ghauth, and F.-F. Chua, "Utilizing learners' negative ratings in semantic content-based recommender system for e-learning forum," *J. Educ. Technol. Soc.*, vol. 21, no. 1, pp. 112–125, 2018.

[12] B. R. Cami, H. Hassanpour, and H. Mashayekhi, "User preferences modeling using Dirichlet process mixture model for a content-based recommender system," *Knowl.-Based Syst.*, vol. 163, pp. 644–655, Jan. 2019.

[13] M. M. Islam, M. J. Hossain, and Z. B. Zahir, "Content-based health recommender system for ICU patient," in *Proc. Multi-Disciplinary Trends Artif. Intell., 13th Int. Conf. (MIWAI)*, Kuala Lumpur, Malaysia: Springer, Nov. 11909, p. 229.

[14] D. Mittal, S. Shandilya, D. Khirwar, and A. Bhise, "Smart billing using content-based recommender systems based on fingerprint," in *Proc. ICT Anal. Appl.* Singapore: Springer, 2020, pp. 85–93.

[15] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992.

[16] R. Logesh, V. Subramaniyaswamy, D. Malathi, N. Sivaramakrishnan, and V. Vijayakumar, "Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2141–2164, Apr. 2020.

[17] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data," *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113248.

[18] T. Mohammadpour, A. M. Bidgoli, R. Enayatifar, and H. H. S. Javadi, "Efficient clustering in collaborative filtering recommender system: Hybrid method based on genetic algorithm and gravitational emulation local search algorithm," *Genomics*, vol. 111, no. 6, pp. 1902–1912, Dec. 2019.

[19] M. Azizi and H. Do, "A collaborative filtering recommender system for test case prioritization in Web applications," in *Proc. 33rd Annu. ACM Symp. Appl. Comput.*, 2018, pp. 1560–1567.

[20] Q. Han, I. M. de Rituerto de Troya, M. Ji, M. Gaur, and L. Zejnilovic, "A collaborative filtering recommender system in primary care: Towards a trusting patient-doctor relationship," in *Proc. IEEE Int. Conf. Healthcare Informat. (ICHI)*, Jun. 2018, pp. 377–379.

[21] S. K. Lam and J. Riedl, "Shilling recommender systems for fun and profit," in *Proc. 13th Conf. World Wide Web*, 2004, pp. 393–402.

[22] M. Si and Q. Li, "Shilling attacks against collaborative recommender systems: A review," *Artif. Intell. Rev.*, vol. 53, no. 1, pp. 291–319, Jan. 2020.

[23] K. Patel, A. Thakkar, C. Shah, and K. Makvana, "A state of art survey on shilling attack in collaborative filtering based recommendation system," in *Proc. 1st Int. Conf. Inf. Commun. Technol. Intell. Syst.*, vol. 1. Cham, Switzerland: Springer, 2016, pp. 377–385.

[24] R. A. Zayed, L. F. Ibrahim, H. A. Hefny, and H. A. Salman, "Shilling attacks detection in collaborative recommender system: Challenges and promise," in *Proc. Workshops Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, 2020, pp. 429–439.
[25] M. E. K. Badran, W. Jurdi, and J. B. Abdo, "Survey on shilling attacks and their detection algorithms in recommender systems," in *Proc. Int. Conf. Secur. Manage. (SAM) Steering Committee World Congr. Comput. Sci., Comput.*, 2019, pp. 141–146.
[26] B. Mehta and T. Hofmann, "A survey of attack-resistant collaborative filtering algorithms," *IEEE Data Eng. Bull.*, vol. 31, no. 2, pp. 14–22, Jun. 2008.
[27] F. Zhang, "A survey of shilling attacks in collaborative filtering recommender systems," in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, Dec. 2009, pp. 1–4.
[28] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, "Shilling attacks against recommender systems: A comprehensive survey," *Artif. Intell. Rev.*, vol. 42, no. 4, pp. 767–799, Dec. 2014.
[29] P. Kaur and S. Goel, "Shilling attack models in recommender system," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, vol. 2, Aug. 2016, pp. 1–5.
[30] R. Burke, M. P. O'Mahony, and N. J. Hurley, "Robust collaborative recommendation," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2015, pp. 961–995.
[31] M. P. O'Mahony, N. J. Hurley, and G. C. Silvestre, "Recommender systems: Attack types and strategies," in *Proc. AAAI*, 2005, pp. 334–339.
[32] B. Mobasher, R. Burke, R. Bhaumik, and J. J. Sandvig, "Attacks and remedies in collaborative recommendation," *IEEE Intell. Syst.*, vol. 22, no. 3, pp. 56–63, May 2007.
[33] Z. Yang, Z. Cai, and X. Guan, "Estimating user behavior toward detecting anomalous ratings in rating systems," *Knowl.-Based Syst.*, vol. 111, pp. 144–158, Nov. 2016.
[34] V. W. Anelli, Y. Deldjoo, T. Di Noia, E. Di Sciascio, and F. A. Merra, "SAShA: Semantic-aware shilling attacks on recommender systems exploiting knowledge graphs," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, 2020, pp. 307–323.
[35] K. Chen, P. P. K. Chan, F. Zhang, and Q. Li, "Shilling attack based on item popularity and rated item correlation against collaborative filtering," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 7, pp. 1833–1845, Jul. 2019.
[36] C. E. Seminario and D. C. Wilson, "Nuking item-based collaborative recommenders with power items and multiple targets," in *Proc. 9th Int. Flairs Conf.*, 2016, pp. 560–565.
[37] P.-A. Chirita, W. Nejdl, and C. Zamfir, "Preventing shilling attacks in online recommender systems," in *Proc. 7th ACM Int. Workshop Web Inf. Data Manage.*, 2005, pp. 67–74.
[38] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, "Classification features for attack detection in collaborative recommender systems," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 542–547.
[39] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, "Detecting profile injection attacks in collaborative recommender systems," in *Proc. 8th IEEE Int. Conf. E-Commerce Technol. 3rd IEEE Int. Conf. Enterprise Comput., E-Commerce, E-Services (CEC/EEE)*, Jun. 2006, p. 23.
[40] Y. Hao, P. Zhang, and F. Zhang, "Multiview ensemble method for detecting shilling attacks in collaborative recommender systems," *Secur. Commun. Netw.*, vol. 2018, pp. 1–33, Oct. 2018.
[41] Z. Yang, L. Xu, Z. Cai, and Z. Xu, "Re-scale AdaBoost for attack detection in collaborative filtering recommender systems," *Knowl.-Based Syst.*, vol. 100, pp. 74–88, May 2016.
[42] A. M. Turk and A. Bilge, "Robustness analysis of multi-criteria collaborative filtering algorithms against shilling attacks," *Expert Syst. Appl.*, vol. 115, pp. 386–402, Jan. 2019.
[43] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre, "Collaborative recommendation: A robustness analysis," *ACM Trans. Internet Technol.*, vol. 4, no. 4, pp. 344–377, 2004.
[44] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, Jan. 2016.
[45] R. Burke, B. Mobasher, and R. Bhaumik, "Limited knowledge shilling attacks in collaborative filtering systems," in *Proc. 3rd Int. Workshop Intell. Techn. Web Pers. (ITWP), 19th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2005, pp. 17–24.
[46] R. Burke, B. Mobasher, R. Zabicki, and R. Bhaumik, "Identifying attack models for secure recommendation," *Beyond Pers.*, vol. 2005, pp. 19–25, Jan. 2005.
[47] C. Williams, B. Mobasher, R. Burke, J. Sandvig, and R. Bhaumik, "Detection of obfuscated attacks in collaborative recommender systems," in *Proc. Workshop Recommender Syst. (ECAI)*, vol. 94, 2006, pp. 19–23.
[48] N. Hurley, Z. Cheng, and M. Zhang, "Statistical attack detection," in *Proc. 3rd ACM Conf. Recommender Syst.*, 2009, pp. 149–156.
[49] R. Bhaumik, B. Mobasher, and R. Burke, "A clustering approach to unsupervised attack detection in collaborative recommender systems," in *Proc. Int. Conf. Data Mining (DMIN)*. Riva del Garda, Italy: Citeseer, 2011, p. 1.
[50] P. Adamopoulos and A. Tuzhilin, "On unexpectedness in recommender systems: Or how to better expect the unexpected," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, pp. 1–32, Dec. 2014.
[51] V. W. Anelli, Y. Deldjoo, T. Di Noia, and F. Antonio, "Knowledge-enhanced shilling attacks for recommendation," in *Proc. 28th Italian Symp. Adv. Database Syst. CEUR Workshop*, Villasimius, Italy, vol. 2646, Jun. 2020, pp. 310–317.
[52] C. A. Williams, B. Mobasher, and R. Burke, "Defending recommender systems: Detection of profile injection attacks," *Service Oriented Comput. Appl.*, vol. 1, no. 3, pp. 157–170, Oct. 2007.
[53] F. Zhang and Q. Zhou, "A Meta-learning-based approach for detecting profile injection attacks in collaborative recommender systems," *J. Comput.*, vol. 7, no. 1, pp. 226–234, Jan. 2012.
[54] W. Zhou, J. Wen, Q. Xiong, M. Gao, and J. Zeng, "SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems," *Neurocomputing*, vol. 210, pp. 197–205, Oct. 2016.
[55] Y. Hao, F. Zhang, J. Wang, Q. Zhao, and J. Cao, "Detecting shilling attacks with automatic features from multiple views," *Secur. Commun. Netw.*, vol. 2019, pp. 1–13, Aug. 2019.
[56] B. Mehta, T. Hofmann, and P. Fankhauser, "Lies and propaganda: Detecting spam users in collaborative filtering," in *Proc. 12th Int. Conf. Intell. User Interface (IUI)*, 2007, pp. 14–21.
[57] K. Bryan, M. O'Mahony, and P. Cunningham, "Unsupervised retrieval of attack profiles in collaborative recommender systems," in *Proc. ACM Conf. Recommender Syst. (RecSys)*, 2008, pp. 155–162.
[58] C.-Y. Chung, P.-Y. Hsu, and S.-H. Huang, "βP: A novel approach to filter out malicious rating profiles from recommender systems," *Decis. Support Syst.*, vol. 55, no. 1, pp. 314–325, Apr. 2013.
[59] A. Bilge, Z. Ozdemir, and H. Polat, "A novel shilling attack detection method," *Procedia Comput. Sci.*, vol. 31, pp. 165–174, Jan. 2014.
[60] Z. Yang, Z. Cai, and Y. Yang, "Spotting anomalous ratings for rating systems by analyzing target users and items," *Neurocomputing*, vol. 240, pp. 25–46, May 2017.
[61] F. Zhang, Z. Zhang, P. Zhang, and S. Wang, "UD-HMM: An unsupervised method for shilling attack detection based on hidden Markov model and hierarchical clustering," *Knowl.-Based Syst.*, vol. 148, pp. 146–166, May 2018.
[62] F. Zhang, Z.-J. Deng, Z.-M. He, X.-C. Lin, and L.-L. Sun, "Detection of shilling attack in collaborative filtering recommender system by PCA and data complexity," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, vol. 2, Jul. 2018, pp. 673–678.
[63] Z. Luo and C. Liang, "An insider attack on shilling attack detection for recommendation systems," in *Proc. 7th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Aug. 2016, pp. 277–280.
[64] L. Yang, W. Huang, and X. Niu, "Defending shilling attacks in recommender systems using soft co-clustering," *IET Inf. Secur.*, vol. 11, no. 6, pp. 319–325, Nov. 2017.
[65] A. M. Turk and A. Bilge, "A robust multi-criteria collaborative filtering algorithm," in *Proc. Innov. Intell. Syst. Appl. (INISTA)*, Jul. 2018, pp. 1–6.
[66] D. Deng, J. J. Mai, C. K. Leung, and A. Cuzzocrea, "Cognitive-based hybrid collaborative filtering with rating scaling on entropy to defend shilling influence," in *Proc. 8th Int. Conf. Netw., Commun. Comput.*, Dec. 2019, pp. 176–185.
[67] S. Alonso, J. Bobadilla, F. Ortega, and R. Moya, "Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems," *IEEE Access*, vol. 7, pp. 41782–41798, 2019.
[68] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
[69] D. W. Oard and J. Kim, "Implicit feedback for recommender systems," in *Proc. AAAI Workshop recommender Syst.*, Wollongong, NSW, Australia, vol. 83, 1998, pp. 81–83.

AGNIDEVEN PALANISAMY SUNDAR received the B.E. degree from the Department of Electronics and Communication, Anna University, Chennai, India. He is currently pursuing the Ph.D. degree with the Department of Computer and Information Sciences, School of Science, Indiana University–Purdue University Indianapolis, Indianapolis, IN, USA. His research interests include security and privacy issues in graph networks, and vulnerabilities in recommender systems.

FENG LI received the Ph.D. degree in computer science from Florida Atlantic University, Boca Raton, FL, USA, in 2009. His Ph.D. Advisor was Dr. J. Wu. He is currently an Associate Professor in computer and information technology with Indiana University–Purdue University Indianapolis, Indianapolis, IN, USA. He has authored or coauthored more than 60 papers in top conferences, including the INFOCOM and the ICDCS. His current research interests include cybersecurity and trust issues, and cloud and mobile computing.

XUKAI ZOU is a Professor with the Department of Computer and Information Sciences, School of Science, Indiana University–Purdue University Indianapolis, and a member of the Center for Education and Research in Information Assurance and Security (CERIAS), Purdue University, and a Fellow of the Indiana University Center of Applied Cybersecurity Research. He has published two books *Secure Group Communication Over Data Networks* (Springer, 2004) and *Trust and Security in Collaborative Computing* (World Scientific, 2008), and over 100 peer-reviewed articles in premier conferences and journals. His research has been supported by the U.S. NSF, the Department of Veterans Affairs, NASA, and industry such as Cisco and Northrop Grumman. His current research interests include applied cryptography, cybersecurity, and deep-learning. He has served as the U.S. NSF Panelist, an NIH Program External Reviewer, a program chair, a member of technical program committees, on editorial boards, and a Reviewer for a number of international journals and conferences.

TIANCHONG GAO (Member, IEEE) received the Ph.D. degree in computer engineering from Purdue University, in December 2019. He is a Lecturer with the School of Cyber Science and Engineering, Southeast University, China. His Ph.D. advisor was Dr. F. Li. He joined the School of Cyber Science and Engineering, Southeast University, in April 2020. He has worked on problems on security, privacy, and social networks. He has published research articles in top-tier conferences and journals. His research vision is to explore the privacy issues in computing and networking.

EVAN D. RUSSOMANNO is currently pursuing the B.Sc. degree in computer information systems with a minor in computer security with Ball State University, Muncie, IN, USA. He has participated in the NSF Research Experiences for Undergraduates (REU) Programa in the areas of mobile cloud and data security at Indiana University–Purdue University Indianapolis, Indianapolis, IN, USA, under the supervision of Dr. F. Li.