Hindawi
Security and Communication Networks
Volume 2021, Article ID 2907691, 18 pages
https://doi.org/10.1155/2021/2907691

Research Article

An Unsupervised Approach for Detecting Group Shilling Attacks in Recommender Systems Based on Topological Potential and Group Behaviour Features

Hongyun Cai[1,2,3,4] **and Fuzhi Zhang**[1,3,4]

[1] *School of Information Science and Engineering, Yanshan University, Qinhuangdao, China*
[2] *School of Cyber Security and Computer, Hebei University, Baoding, China*
[3] *The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao, China*
[4] *The Key Laboratory for Software Engineering of Hebei Province, Qinhuangdao, China*

Correspondence should be addressed to Fuzhi Zhang; xjzfz@ysu.edu.cn

Received 28 May 2021; Revised 6 August 2021; Accepted 7 September 2021; Published 27 September 2021

Academic Editor: Rutvij Jhaveri

To protect recommender systems against shilling attacks, a variety of detection methods have been proposed over the past decade. However, these methods focus mainly on individual features and rarely consider the lockstep behaviours among attack users, which suffer from low precision in detecting group shilling attacks. In this work, we propose a three-stage detection method based on strong lockstep behaviours among group members and group behaviour features for detecting group shilling attacks. First, we construct a weighted user relationship graph by combining direct and indirect collusive degrees between users. Second, we find all dense subgraphs in the user relationship graph to generate a set of suspicious groups by introducing a topological potential method. Finally, we use a clustering method to detect shilling groups by extracting group behaviour features. Extensive experiments on the Netflix and sampled Amazon review datasets show that the proposed approach is effective for detecting group shilling attacks in recommender systems, and the $F1$-measure on two datasets can reach over 99 percent and 76 percent, respectively.

1. Introduction

Information resources are growing explosively with the rapid development of Internet, which causes information overload. To deal with this problem, recommender systems have been widely used in e-commerce, social network, cloud computing, etc. [1]. However, due to the openness, malicious users can register accounts and provide ratings for the items with the intention of promoting or demoting the recommendation of target items. This behaviour has been termed as shilling attacks or profile injection attacks [2, 3]. Previous researches have shown that a large number of injected fake profiles can bias the output of collaborative filtering recommender systems, indicating that the collaborative filtering recommender systems are vulnerable to shilling attacks [4, 5]. Over the past decade, traditional attack models have been well studied, such as random attack, average attack, bandwagon attack, AoP attack, Love/hate attack, etc. In these attacks, attack users try to promote or demote the same target item but select filler items separately. In fact, attack users can easily collude to attack a set of target items in the recommender system. The behaviour wherein a group of attack users collude to promote or demote the recommendation of a set of target items has been termed group shilling attacks [6, 7]. As attack users in the group work together and strategically generate attack profiles, the detection methods proposed for detecting traditional attacks become ineffective for group shilling attacks. Therefore, how to improve the detection performance for group shilling attacks has become a key issue in the recommender systems.

To reduce the impact of shilling attacks on recommender systems, researchers have developed a variety of shilling attack detection methods [8–31]. Most methods focus mainly on the features of individual profiles and are used to detect attack profiles for traditional attacks, e.g., random attack and bandwagon attack. These methods rarely consider the collusive behaviours among attack users, which cause low precision in detecting group shilling attacks. Although a few detection methods [27–31] have been proposed for detecting group shilling attacks in recent years, they usually need prior knowledge of attacks (e.g., the number of shilling groups or attackers). Otherwise, they suffer from low detection precision. While some approaches [32, 33] have been presented to spot the collusive groups in e-commerce and social network, the behaviour features of shilling groups vary greatly in different fields. Therefore, the approaches developed for identifying collusive groups in other fields are not suitable for detecting shilling groups in recommender systems.

To address the above limitations, we propose an unsupervised method for detecting group shilling attacks in recommender systems based on topological potential theory and group behaviour features, which is named as TP-GBF. The proposed approach focuses on the collusive behaviours among attack users and can accurately detect attack users in group shilling attacks without knowing the number of shilling groups in advance. Particularly, we first construct a weighted user relationship graph by analysing the lockstep behaviours between users. In the graph, vertices and edges represent the users and their relations, respectively. The weight of each edge is calculated by combining direct and indirect collusive degrees. Then, we analyse the network structure of shilling groups and introduce a topological potential method to spot suspicious groups. Finally, we use the clustering method to detect shilling groups by extracting group behaviour features.

The main contributions of this paper are summarized as follows:

(1) Unlike existing user relationship graph construction methods that focus mainly on direct relationship between users, we propose a graph construction method by combining direct and indirect collusive degrees between users. This method not only analyses the direct collusive relationship between two different users but also considers transitivity of direct relationship between them and, therefore, can highlight the collusive relationship between users in the same shilling group whether the group is a tightly coupled shilling group or a loosely coupled shilling group.

(2) Existing methods for generating suspicious groups are mainly based on frequent item mining (FIM) or graph clustering. The FIM-based methods are suitable for finding tightly coupled groups and need to set an appropriate minimum support, while the graph-clustering-based methods usually require prior knowledge of group size or number of groups. Differently, our method finds suspicious shilling groups by introducing a topological potential method, which can automatically decide the number of suspicious groups and find all groups with strong relationships among group members.

(3) Different from existing detection methods for group shilling attacks in recommender systems that are presented based on graph structure or individual behaviour features, our method is based on both the graph structure and group behaviour features of shilling groups, which can detect attack users more accurately.

The rest of the paper is organized as follows. Section 2 introduces the background and related work. Section 3 describes the proposed detection method for group shilling attacks in detail, which includes the construction of the weighted user relationship graph, the generation of the set of suspicious shilling groups, and the detection of shilling groups. The experimental results are reported and discussed in Section 4. The last section draws a conclusion of our work and discusses the future work.

2. Background and Related Work

2.1. Group Shilling Attacks. In Ref. [6], Su et al. first proposed the concept of group shilling attacks in recommender systems and mentioned two attack scenarios. In these scenarios, multiple attackers are well organized to conceal their intentions. The attackers in a shilling group work together to promote or demote a set of target items in scenario 1 or only attack some items in the target set in scenario 2. In order to achieve the attack intention, a shilling group must contain a certain number of attackers and each attacker in the group should give some normal ratings on nontarget items. Moreover, many shilling groups may coexist in a recommender system.

To generate the effective group shilling profiles and avoid the detection of the existing methods, Wang et al. [7] proposed a tricky group shilling attack model which includes a strict version denoted as GSAGen$_s$ and a loose version denoted as GSAGen$_l$. In their model, some shilling profiles generated by standard attack models, i.e., random attacks or average attacks, are used as the input. Then, they construct high diversity shilling profiles based on the input according to the strict condition or loose condition. Of the two versions, GSAGen$_l$ can generate a shilling group with more shilling profiles than GSAGen$_s$ because GSAGen$_l$ has less strict conditions in generating group shilling attack profiles. Table 1 summarizes the group shilling attack models.

2.2. Topology Potential Field. The concept of field was first proposed by Faraday in 1837, which is used to measure the interactions of protons in physics. The physical quantity representing the field is termed as potential and the corresponding mathematical function is called potential function. In the field of physics, the potential of the node is proportional to the strength of field source around it, and it decreases with the increase of distance from the node to the other field source. Topology potential field [34, 35] is an

Table 1: Summary of group shilling attack models.

Version	Type	Filler items	Filler item rating	Target item rating
GSAGen$_s$	GSAGen$_s$ Ran	Randomly chosen and only rated by one attacker in the shilling group	System mean	$r_{\max}/r_{\min}$
	GSAGen$_s$ Avg	Randomly chosen and only rated by one attacker in the shilling group	Item mean	$r_{\max}/r_{\min}$
GSAGen$_l$	GSAGen$_l$ Ran	Randomly chosen and at least one item is rated by no more than two attackers in the shilling group	System mean	$r_{\max}/r_{\min}$
	GSAGen$_l$ Avg	Randomly chosen and at least one item is rated by no more than two attackers in the shilling group	Item mean	$r_{\max}/r_{\min}$

application of field in a complex network. Nodes in a complex network are not isolated but interact with each other. Therefore, the topology potential of one node represents the sum of the energy radiated by other nodes.

Let $G = (V, E)$ be a graph, where $V = \{v_1, v_2, \ldots, v_m\}$ represents the set of nodes and $E \subseteq V \times V$is the set of edges. According to the definition of potential function, the topology potential of any node $v_i \in V$is denoted as $\varphi(v_i)$and defined as follows:

$$\varphi(v_i) = \sum_{j=1}^{|V|} \left(m_j \times e^{-\left(\mathrm{dis}_{i,j}/\sigma\right)^2} \right), \tag{1}$$

where $m_j \geq 0$ denotes the mass of neighbour v_j, , $1 \leq j \leq m$; $dis_{i,j}$ represents the distance between nodes $v_i$and v_j; and σ is a parameter that is used to control the influence scope of a node.

2.3. Related Work. Over the past decade, a large number of detection methods have been presented for detecting shilling attacks in recommender systems, which try to identify and eliminate the attack profiles from the perspective of ratings, time, and item popularity degree. These methods can be generally divided into two categories: supervised detection methods and unsupervised detection methods.

As for supervised detection methods of shilling attacks in recommender systems, Chirita et al. [8] first proposed the features named rating deviation from mean agreement (RDMA) and degree of similarity with top neighbours (DegSim). The effectiveness of two features depends on the attack type, attack size, and filler size. Burke et al. [9] trained a classifier based on some generic and model-specific features, which were extracted by utilizing statistics properties of ratings given by attack users. Although these detection features are helpful for detecting standard attacks, they become ineffective under obfuscated attacks. Wu et al. [10] presented an effective feature selection method for detecting five attacks and proposed the detection algorithm based on supervised learning. However, the proposed method needs to know the type of attack in advance. Yang et al. [11] proposed an ensemble shilling attack detection method based on 18 statistical features, which could obtain better detection performance than the baseline methods using a single classifier. However, it needs to consider the intensive numeric calculation on these features. Tong et al. [12] applied convolutional neural network to extract detection features from user-item rating matrix. This method can detect shilling profiles generated by the single attack method, but it is not suitable for detecting hybrid attacks. Hao et al. [13] first extracted multidimensional detection features from different angles, including ratings, rating time, and item popularity degree, and then they trained a SVM-based classifier for detecting shilling attacks. This method illustrates higher effectiveness than the baseline methods, but the detection performance is not good when the attack size is less than 10%. Zhou et al. [14] presented a deep-learning-based method to detect traditional shilling attacks in recommender systems. This method does not use the handcrafted features, but it requires a validation set to find the optimal hyper parameters. Zhang et al. [15] proposed a supervised classifier to detect attack profiles, which learned the representations of users by using graph convolutional network and calculated the probability of a user being an attacker. This method can improve the detection performance by exploiting and integrating the rich side information of users, but it requires setting up of some hyper parameters. In [16], a hybrid convolutional neural network and LSTM-based deep learning model were proposed to detect attack profiles in recommender systems. This method can identify attack profiles generated by six different attack models, but its $F1$-measure decreases with the increase of filler size. Wu et al. [17] also proposed a hybrid method for detecting shilling profiles in recommender system, which trained the classifier based on labelled profiles and unlabelled ones. This method is effective for detecting hybrid shilling attacks, but its detection performance is inferior to that of C4.5 decision tree method. The aforementioned supervised detection methods need to train a classifier based on the labelled profiles in the training set, which are only effective for detecting known types of shilling attacks.

As for unsupervised detection methods of shilling attacks in recommender systems, Mehta et al. [18] analysed the similarity structure in attack profiles and proposed the PCA-VarSelect model. This method is effective in detecting some standard shilling attacks including random attack, average attack, etc. However, it needs the prior knowledge of attack size (i.e., number of attack profiles). A detection method based on spectral clustering is presented in Ref. [19]. This method can detect unknown types of shilling attacks, but it also assumes to know the number of attack profiles in advance. In Ref. [20], Xia et al. described an identification method of the target item, which can find the target item by using dynamic time segment. Based on the identified target item, attack profiles can be spotted. This method is suitable for identifying the target item rated by a large number of genuine users before attacks, but it cannot catch the cold-start target item. Zhang et al. [21] designed a ranking algorithm for detecting attack profiles in recommender systems, which calculates the suspiciousness of each user and each item iteratively. This approach can detect unknown types of attacks effectively, but it needs to label some candidate users as seeds and to know the number of attack profiles. In Ref. [22], Zhang et al. presented an unsupervised detection approach for detecting various attacks by analysing diversity between genuine and attack users in rating behaviours. This method can determine the critical point of rating behaviour suspiciousness degrees dynamically, but the critical point is hard to determine if the attack size is low. Yang et al. [23] proposed a unified detection approach for profile injection attacks and covisitation injection attacks, which used the coupled factor graph and label propagation algorithm to determine attack profiles. The approach can detect various types of attacks, but it requires a set of spam user seeds and some empirical thresholds of parameters. The aforementioned unsupervised detection approaches can detect unknown types of shilling attacks. However, these methods usually need some prior knowledge of attacks (i.e., attack size) to ensure the detection performance. Aiming at these limitations, Zhang et al. [24] analysed the behaviour difference in item popularity between genuine profiles and attack ones, and designed a novel detection framework based on hidden Markov model and hierarchical clustering. This method can achieve excellent detection performance in detecting many types of attacks, but it becomes ineffective in detecting AoP attack. In our previous work [25], we extracted user behaviour features based on latent item relationship and detected attack users by utilizing the identified target items. The method is effective for detecting various types of attack profiles in both synthetic and real-world datasets, but it requires a reasonable threshold when the number of target items is more than one. In another work [26], we proposed an unsupervised detection method, which identified the target items with abnormal deviation of rating distributions on items, and detected attack users by performing the Density-Based Clustering of Applications with Noise (DBSCAN) clustering on the set of suspicious users. However, this method cannot identify the target items with less abnormal ratings.

The aforementioned approaches focus mainly on detecting shilling attacks based on the features of individual attack profiles in recommender systems; few of them consider the lockstep behaviours among attack users in a shilling group. There are a few methods that analyse shilling behaviours at the group level. Gao et al. [27] introduced a group-based ranking method, which grouped users and calculated their reputations based on their ratings. This approach is effective for detecting those attackers who have only rated a few items and have high rating errors, but its precision is low when there are more original distorted ratings in the dataset. Zhou et al. [28] proposed a two-stage model to detect group shilling attacks in recommender systems, which utilized features based on Rating Deviation from Mean Agreement (RDMA) and Degree of Similarity with Top Neighbours (DegSim). This approach can detect group attacks with high similarity between group members. However, it is ineffective in detecting attack profiles with low similarity in the shilling group. Gimenes et al. [29] introduced an online-Recommendation Fraud Excluder (ORFEL) which used the lockstep behaviour to detect suspicious fraud behaviour. This approach is effective in discovering potential attacks. However, it requires sufficient number of starting seeds to search the lockstep behaviours. Besides, may be not all users with lockstep behaviours are attackers. In Ref. [30], an unsupervised method for detecting shilling groups in recommender systems is presented, which generates candidate groups by dividing the rating tracks on each item and detects shilling groups by employing the bisecting K-means algorithm based on the suspicious degrees of candidate groups. This approach is very effective for detecting group attacks on the Synthesis dataset. However, its precision needs to be improved on the real-world dataset. Hao et al. [31] constructed a user-user graph based on rating behaviours similarity and determined the attacker communities with maximum filling rate of the item. This method can detect the attacker communities in which the attackers are closely connected to each other. The detection precision will decrease if the maximum filling rate of the item in the community is low.

Table 2 summarizes the above approaches for shilling attack detection in recommender systems.

Another related work is to detect review spammer groups in e-commerce websites. Existing detection approaches focus on detecting collusive review spammers using frequent pattern mining or graph mining. In Ref. [32], frequent pattern mining technique is first adopted to generate suspicious groups, and then several group shilling features are extracted at the group level. Finally, shilling groups are identified from the suspicious groups by using a PCA-based method. This approach is effective for detecting group shilling attacks with tightly coupled shilling groups. However, the calculations of features need not only ratings and timestamps but also review content. Moreover, they require the threshold of information loss rate as input. Wang et al. [33] modelled spammer groups as biconnected graphs and designed a graph-based framework to detect review spammer groups. This method uses eight group spam indicators to evaluate the suspiciousness of a candidate group, but it also requires setting up of a minimum spam threshold in detecting spammer groups. Since the shilling groups in recommender systems are different from review spammer groups in e-commerce websites, these group spam indicators presented for detecting review spammer groups are no longer applicable to detecting group shilling attacks in recommender systems. Moreover, in a real-world dataset, it is difficult to obtain a suitable minimum suspiciousness threshold or the total number of shilling groups in advance.

In this work, we focus on detecting group shilling attacks in recommender systems. Compared with our previous work in Refs. [25, 26], this work highlights the collusive

Table 2: Summary of shilling attack detection approaches.

Approaches	Category	Advantage	Disadvantage
Chirita et al. [8] and Burke et al. [9]	Supervised	Effective for specific attack strategy	Less effective for obfuscated attacks
Wu et al. [10]	Supervised	Effective for known attack strategy	Require known attack strategy
Yang et al. [11]	Supervised	Effective for specific attack strategy	Much computational effort
Tong et al. [12]	Supervised	Feature learning by deep learning	Less effective for mixed attacks
Hao et al. [13]	Supervised	Effective for various types of attack	Performance is not very high for small attack size
Zhou et al. [14]	Supervised	Automatic feature extraction	Need a validation set to set parameters
Zhang et al. [15]	Supervised	Regardless of the specific attack strategy	Require setting hyper parameters
Vivekanandan et al. [16]	Supervised	Automatic feature extraction	$F1$ decreases with the increase of filler size
Wu et al. [17]	Supervised	Effective for hybrid attacks	Performance is not high
Mehta et al. [18]	Unsupervised	Regardless of the specific attack strategy	Require the prior knowledge of attack size
Zhang et al. [19]	Unsupervised	Regardless of the specific attack strategy	Require the prior knowledge of attack size
Xia et al. [20]	Unsupervised	Effective for a target item with large number of ratings from normal users	Less effective with cold-start target items
Zhang et al. [21]	Unsupervised	Regardless of the specific attack strategy	Require some user seeds and attack size
Zhang et al. [22]	Unsupervised	Regardless of the specific attack strategy	Critical point is not always accurate
Yang et al. [23]	Unsupervised	Effective for various attacks	Require a set of spam seeds
Zhang et al. [24]	Unsupervised	Performance is high under various attacks	Less effective under AoP attack
Cai et al. [25]	Unsupervised	Effective for various attacks	Require a reasonable threshold when target items are more
Cai et al. [26]	Unsupervised	Regardless of the specific attack strategy	Less effective for identifying target items with a few ratings
Gao et al. [27]	Unsupervised	Effective if the number of distorted ratings is few	Precision decreases if the number of distorted ratings is more
Zhou et al. [28]	Unsupervised	Effective for detecting specific shilling groups	Less effective for attack profiles with low similarity
Gabriel et al. [29]	Unsupervised	Effective for detection users with collusive behaviour	Require some starting seeds
Zhang et al. [30]	Unsupervised	Effective for group shilling attacks	Require setting parameters
Hao et al. [31]	Unsupervised	Automatic feature learning	Precision depends on the maximum filling rate

relationship between attackers in the same shilling group and detects shilling attacks at the group level. Unlike the approaches in Refs. [22, 30], our method combines the graph structure and group features, which improves the detection precision for group shilling attacks. Compared with the method in Ref. [31], our method can detect attack users with loose association, which achieves excellent precision especially for detecting loosely coupled shilling groups.

3. The Framework of TP-GBF

The framework of TP-GBF is depicted in Figure 1, which consists of three stages, i.e., constructing a weighted user relationship graph, generating a set of suspicious groups, and finding shilling groups. In the first stage, we construct a weighted user relationship graph by combining direct and indirect collusive degrees between users. The aim was to highlight the collusive behaviours among attackers in the same shilling group. In the second stage, we introduce a topology potential method to generate the set of suspicious shilling groups. The number of suspicious groups is decided automatically based on the number of nodes with local maximum potential value. In the third stage, group behaviour features are extracted at the group level, and then shilling groups in the set of suspicious groups are detected by using a clustering algorithm.

To facilitate the discussion, we list the notations and their descriptions used in this paper in Table 3.

3.1. Constructing a Weighed User Relationship Graph. As described in Subsection 2.1, members in the same shilling group work collaboratively to promote or demote more than one target items. Figure 2 shows an example of a group shilling attack, where the rectangle nodes with the solid border and dotted border represent normal items and target items, and circle nodes with the solid border and dotted border represent genuine users and attack users, respectively.

As shown in Figure 2, the set of shilling group members is $\{u_2, u_4, u_5, u_6\}$and the set of target items is$\{p_2, p_3, p_4\}$. The pattern of connections between group member and target item may not form a complete bipartite graph, indicating that group members may work in a looser manner. In addition, to reach the attack intention on target items, group members are required to rate several target items in a relatively short period of time and their rating behaviours on the same target item are usually the lockstep. It is to be noted

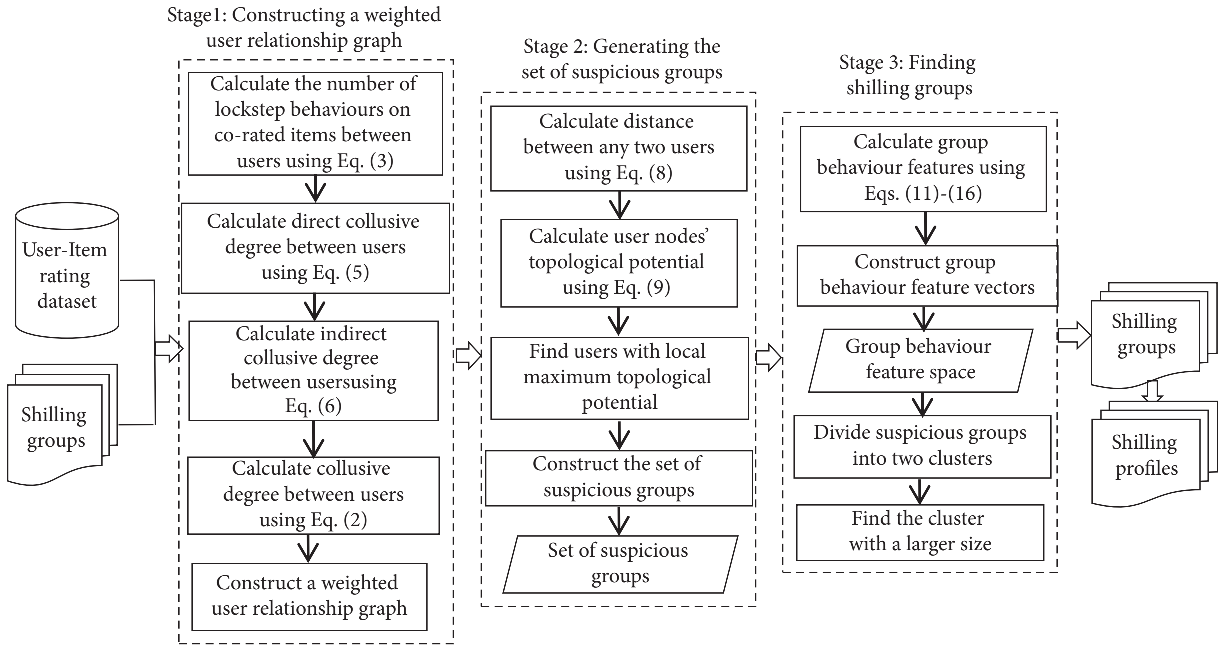

Stage1: Constructing a weighted user relationship graph
Calculate the number of lockstep behaviours on co-rated items between users using Eq. (3)
Calculate direct collusive degree between users using Eq. (5)
Calculate indirect collusive degree between usersusing Eq. (6)
Calculate collusive degree between users using Eq. (2)
Construct a weighted user relationship graph
User-Item rating dataset
Shilling groups
Stage 2: Generating the set of suspicious groups
Calculate distance between any two users using Eq. (8)
Calculate user nodes' topological potential using Eq. (9)
Find users with local maximum topological potential
Construct the set of suspicious groups
Set of suspicious groups
Stage 3: Finding shilling groups
Calculate group behaviour features using Eqs. (11)-(16)
Construct group behaviour feature vectors
Group behaviour feature space
Divide suspicious groups into two clusters
Find the cluster with a larger size
Shilling groups
Shilling profiles

Figure 1: Framework of TP-GBF.

Table 3: Notations and their descriptions.

Notation	Description
U	The set of users and $U = \{u_1, u_2, \ldots, u_m\}$
I	The set of items and $I = \{p_1, p_2, \ldots, p_n\}$
R	User-item rating matrix and $R = [ri, j]m \times n$
T	User-item rating timestamp matrix and $T = [ti, j]m \times n$
GSet	The set of suspicious groups and $\mathrm{G\,Set} = \{\mathrm{Source}_1, \mathrm{Source}_2, \ldots, \mathrm{Source}_k\}$
Source_i	Set of members in the ith suspicious group and $\lvert\mathrm{Source}_i\rvert > 2$
Target_i	Set of target items in the ith suspicious group and $\lvert\mathrm{Target}_i\rvert > 2\,(i \in \{1, 2, \ldots, k\})$

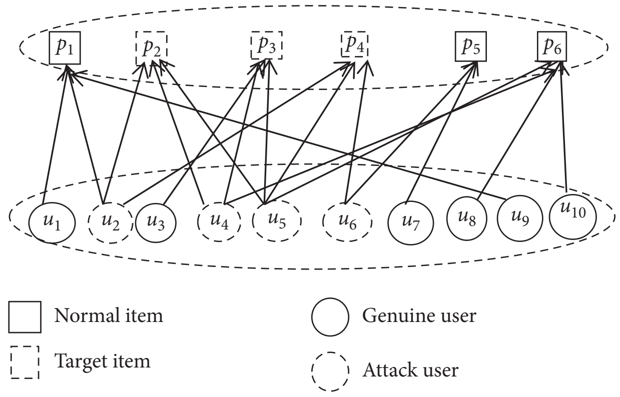


Figure 2: An example of a group shilling attack in recommender systems.

that there may coexist multiple shilling groups simultaneously in a recommender system.

Definition 1 (collusive degree between users (CDBU)). The collusive degree between $u_i \in U$ and $u_j \in U$ refers to the relationship strength based on their lockstep behaviours, which is defined as the sum of direct collusive degree $DirectC_{i,j}$ and indirect collusive degree $\mathrm{InDirect}C_{i,j}$, that is,

$$\mathrm{CDBU}_{i,j} = \mathrm{Direct\,C}_{i,j} + \mathrm{InDirect\,C}_{i,j}. \tag{2}$$

Direct collusive degree is related to the number of corated items with lockstep behaviours and indirect collusive degree is considered from the transitivity of direct relationship. For users u_i and u_j in the same shilling group, if they corate a target item, their rating deviation is usually small and the time interval is usually short. We introduce an indicator function $\Gamma(i, j, l)$ to denote the lockstep behaviour on item p_l for users u_i and u_j, and let $\mathrm{CoNum}_{i,j}$ be the number of corated items with lockstep behaviours. $\mathrm{CoNum}_{i,j}$ and $\Gamma(i, j, l)$ are calculated according to (3) and (4), respectively.

$$\mathrm{Co\,Num}_{i,j} = \sum_{l=1}^{|I|} \Gamma(i, j, l). \tag{3}$$

$$\Gamma(i, j, l) = \begin{cases} 1, & \text{if } r_{i,l} \neq 0 \wedge r_{j,l} \neq 0 \wedge \left|t_{i,l} - t_{j,l}\right| \leq \Delta \wedge \left|r_{i,l} - r_{j,l}\right| \leq \varepsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

In (4), Δrepresents the predefined time span for capturing a lockstep behaviour and ε is the predefined small nonnegative number for measuring rating deviation. In this work, ε is set to 1.

Direct collusive degree should increase with $CoNum_{i,j}$ and shows a considerable change around threshold θ($\theta$is set to 2 according to the description of group shilling attacks), which is calculated using the characteristic of the arc tangent function and is defined as

$$\text{Direct C}_{i,j} = \begin{cases} \dfrac{\arctan\left(\text{Co Num}_{i,j} - \theta\right) + \pi/2}{\pi} & \text{if Co Num}_{i,j} > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Indirect collusive degree between users is determined by their common neighbours and is defined as

$$\text{InDirect C}_{i,j} = \frac{1}{|U|} \sum_{u=1}^{|U|} \left(\text{Direct C}_{i,u} \times \text{Direct C}_{j,u}\right). \quad (6)$$

Definition 2 (weighted user relationship graph). Let $G = (V, E, W)$ be the weighted user relationship graph, where vertex set V corresponds to the user set, $E = \{(u_i, u_j) | \text{CoNum}_{i,j} > 0, i, j \in \{1, 2, \ldots, |U|\}\}$denotes the edge set and $W = [w_{ij}]_{|U| \times |U|}$ represents the weight matrix. For users $u_i \in U$and $u_j \in U$, $w_{ij} = \text{CDBU}_{i,j}$ when they have lockstep behaviours on corated items; otherwise, $w_{ij} = 0$and there is no edge between two nodes.

The algorithm of constructing the weighted user relationship graph is described as follows.

Algorithm 1 mainly includes three parts. The first part (line 1) initializes the variables. The second part (lines 2–10) generates the set of edges and the weight matrix. The third part (line 11) constructs the weighted user relationship graph.

3.2. Generating the Set of Suspicious Groups.

In the weighted user relationship graph, each vertex can be viewed as a topology potential field source and has an impact on other vertexes. However, based on the modularity and community structure characteristics in the real world, topology potential produced by one user on the other user will become very weak when they are far away from each other. According to the classical field theory, the topology potential of $u_i \in U$on $u_j \in U (i \neq j)$ can be ignored when their distance $dis_{i,j}$ is greater than $l = \lfloor 3\sigma/\sqrt{2} \rfloor$. Generally speaking, the larger the collusive degree between two users, the lesser the distance between them. However, the same collusive degree may have different effects on different users, e.g., a common user and an active user. Based on the observations, we take a local factor into consideration, which is defined as

$$\text{L Factor}_{i,j} = \begin{cases} \text{Co Num}_{i,j}/\text{meantopk} & \text{if degree}(i) > \eta, \\ 1, & \text{otherwise,} \end{cases} \quad (7)$$

where degree$(\cdot)$ means the degree of a user node in the user relationship graph, $\eta$is a threshold, and *meantopk* indicates the average number of corated items between $u_i \in U$and its top-k nearest neighbours. In the paper, $\eta$is set to 100 and the top-20 nearest neighbours are considered. Therefore, $dis_{i,j}$ is inversely proportional to the product of their local factor and collusive degree if users u_i and u_j have lockstep behaviours on corated items; otherwise, $dis_{i,j}$ refers to the minimum distance between u_i and u_j and is calculated using the Dijkstra algorithm.

In addition, to further reduce the impact of outlier distance and improve the accuracy of suspicious groups, $\text{dis}_{i,j}$ can be further modified based on k-distance and calculated by

$$\text{dis}_{i,j} = \frac{\text{dis}_{i,j}}{\text{k dis}(i)} \times \text{dis}_{i,j}, \quad (8)$$

where kdis(i) indicates the k-distance of $u_i \in U$. The value of k can be decided according to the $F1$-measure of the proposed detection method. By experiment, we set the value of k to 10 on both experimental datasets. To save space, the experimental details of this part have not been discussed.

Similar to previous research on the topology potential field, we also ignore the massive difference between vertexes and set vertex mass to 1. Therefore, the topology potential of any user u_i can be calculated by

$$\varphi(u_i) = \sum_{j=1, \text{dis}_{i,j} \leq l}^{|U|} e^{-\left(\text{dis}_{i,j}/\sigma\right)^2}. \quad (9)$$

As described above, the topology potential of one node reflects the total impact of other nodes. In the weighted user relationships graph, nodes of attackers usually show considerably larger topology potential than other user nodes. Moreover, there is a natural peak-valley structure in the topology potential field. This is because nodes belonging to the same shilling group form a local high potential area. To find out every local high potential area, we introduce the definition of local maximum topology potential node.

Definition 3 (local maximum topology potential user). Given a user vertex $u_i$and its neighbour set Neib_i, for$\forall v \in \text{Neib}_i$, if$\varphi(v) < \varphi(u_i)$, then $u_i$is a local maximum topology potential user. Let rep be the set of all local maximum topology potential users, which can be defined as:

$$\text{rep} = \{u_i | (u_i \in U) \wedge (\forall v \in \text{Neib}_i) \wedge (\varphi(u_i) > \varphi(v))\}. \quad (10)$$

In this work, a hill-climbing algorithm with multiple random restarts is used to determine all vertexes with local maximum topology potential. In the hill-climbing algorithm, we start with a random initial vertex and search along the direct neighbours with the rising direction of topology potential value until a local maximum topology potential user node is discovered or all reachable vertexes have been checked. By using the hill climbing algorithm, the users with local maximum topology potential are found and represented as $\text{rep} = \{\text{rep}_1, \text{rep}_2, \ldots, \text{rep}_k\}$, where k represents the total number of local maximum topology potential users in

```
Input: time matrix $T$, rating matrix $R$, predefined time span$\Delta$, small nonnegative number$\varepsilon$, and threshold $\theta$
Output: the weighted user relationship graph $G$
$V \leftarrow U$; $E \leftarrow, \varnothing$;$W \leftarrow 0_{|U|\times|U|}$
for $\forall i, j \in U$ do
    Calculate the value of $CoNum_{i,j}$ using (3)
    if $CoNum_{i,j} > 0$ then
        Calculate the value of $DirectC_{i,j}$ using (5)
        Calculate the value of $InDirectC_{i,j}$ using (6)
        Calculate the value of $CDBU_{i,j}$ using (2)
        $E \leftarrow E \cup \{(i,j)\}$; $wij, \leftarrow CDBU_{i,j}$
    end if
end for
Construct the graph $G$=($V$,$E$,$W$)
return $G$
```

Algorithm 1: Constructing the weighted user relationship graph.

the weighted user relationship graph. Based on $\text{rep}_i\,(1 \le i \le k)$, Source_ican be generated. First, the initial Source_i only contains rep_i. Next, a reachable vertex u_x will be included by Source_i if $d_{x,i}$ is not greater than l and the potential value is decreased from rep_i to this vertex. All reachable vertexes are considered based on breadth-first traversal.

Based on the above description, the main steps for generating the set of suspicious groups are as follows:

Step 1 Calculate the topology potential of each vertex in the weighted user relationship graph

Step 2 Find the local maximum topology potential users using hill-climbing algorithm with multiple random restarts

Step 3 Generate the set of suspicious groups based on the local maximum topology potential users

The algorithm of generating the set of suspicious groups is described as follows.

Algorithm 2 mainly includes three parts. The first part (lines 1–15) calculates the topology potential value of each user node. The second part (line 16) finds out all nodes with local maximum topology potential value by using a hill-climbing algorithm with multiple random restarts. The third part (lines 17–23) generates a set of suspicious groups, where every suspicious group expands the group size by calling Algorithm 3.

Algorithm 3 is used to judge whether the neighbour of the current node should be included. Since group members are considered one by one, Algorithm 3 is called many times during the generation of a suspicious group.

3.3. Finding Shilling Groups in the Set of Suspicious Groups.

Many features have been proposed based on the individual attack profiles over the past decade. However, the effective detection features for shilling groups are limited. Since there may be a strong lockstep relationship between some normal users, groups of these normal users may be misjudged as shilling groups. In this section, six group behaviour features are proposed at the group level to distinguish shilling groups from normal ones in the set of suspicious groups.

First, we extract four group features based on our previous work in Ref. [25]. In Ref. [25], two features, namely, SBT (Similarity of Behaviour Track) and ASTW (Average Similarity of Time window) are discussed, which can reflect the behaviour differences in user interest preference between genuine users and attack users. Inspired by these two features, we extract the following four features at the group level.

Definition 4 (group behaviour track similarity (GBTS)). For the ith candidate group in *GSet*, its group behaviour track similarity refers to the average of similarity of user behaviour tracks, which is denoted as $\text{GBTS}_i$and calculated by

$$\text{GBTS}_i = \frac{1}{\left|\text{Source}_i\right|} \sum_{u \in \text{Source}_i} \text{SBT}_u, \tag{11}$$

where $\text{Source}_i$is the user set of $g_i$and SBT_u represents the behaviour track similarity of u.

Definition 5 (group behaviour track fluctuation (GBTF)). For the ith candidate group in *GSet*, its group behaviour track fluctuation refers to the dispersion degree of behaviour track similarity of all users in the group, which is denoted as GBTF_i and calculated by

$$\text{GBTF}_i = \sqrt{\frac{1}{\left|\text{Source}_i\right| - 1} \sum_{u \in \text{Source}_i} \left(\text{SBT}_u - \text{GBTS}_i\right)^2}. \tag{12}$$

Definition 6 (group behaviour window similarity (GBWS)). For the ith candidate group in *GSet*, its group behaviour window similarity refers to the average window similarity of all users in the group, which is denoted as GBWS_i and calculated by

$$\text{GBWS}_i = \frac{1}{\left|\text{Source}_i\right|} \sum_{u \in \text{Source}_i} \text{ASTW}_u, \tag{13}$$

```
Input: the weighted user relationship graph $G$=($V$,$E$,$W$)
Output: the set of suspicious groups $\mathrm{G\,Set} = \{\mathrm{Source}_1, \mathrm{Source}_2, \ldots, \mathrm{Source}_k\}$(the value of $k$ is determined automatically)
for $\forall i, j \in V$ do
    if $wij$>0 then
        Compute $LFactor_{i,j}$ using (7)
        $dis_{i,j} \leftarrow 1/(LFactor_{i,j} \times wij)$
    else
        $dis_{i,\ j} \leftarrow$the minimum distance between $u_i$ and $u_j$ calculated by the Dijkstra algorithm
    end if
end for
for $\forall i, j \in V$ do
    modify $dis_{i,j}$ using (8)
end for
for $\forall u \in V$ do
    Compute $\varphi(u)$ using (9)
end for
Find out all local maximum topology potential vertexes $\mathrm{rep}_1, \mathrm{rep}_2, \ldots, \mathrm{rep}_k$using a hill-climbing algorithm
for $\forall \mathrm{rep}_i \in \{\mathrm{rep}_1, \mathrm{rep}_2, \ldots, \mathrm{rep}_k\}$ do
    $\mathrm{Source}_i \leftarrow \{\mathrm{rep}_i\}$
    $\mathrm{pnode} \leftarrow \mathrm{rep}_i$
    $\mathrm{cnode} \leftarrow \mathrm{rep}_i$
    call Algorithm 3
end for
return $\mathrm{G\,Set} = \{\mathrm{Source}_1, \mathrm{Source}_2, \ldots, \mathrm{Source}_k\}$
```

Algorithm 2: Generating the set of suspicious groups.

where ASTW_u denotes the average of time window similarity of user u inSource_i.

Definition 7 (group behaviour window fluctuation (GBWF)). For the ith candidate group in *GSet*, its group behaviour window fluctuation refers to the dispersion degree of window fluctuation of all users in the group, which is denoted as GBWF_i and calculated by

$$\mathrm{GBWF}_i = \sqrt{\frac{1}{\left|\mathrm{Source}_i\right| - 1} \sum_{u \in \mathrm{Source}_i} \left(\mathrm{ASTW}_u - \mathrm{GBWS}_i\right)^2}. \tag{14}$$

Next, we further extract two new features of shilling groups based on attack intention. In recommender systems, members of a shilling group work together to promote or demote the recommendation of a few target items. In order to influence as many genuine users as possible, each member in the shilling group randomly selects some items as filler items. Moreover, in view of attack cost and attack intention, target items for push attacks are usually nonpopular items, while attackers mounting nuke attacks prefer to attack popular items with a large number of high ratings.

Definition 8 (group attack intention (GAI)). For the ith candidate group in *GSet*, its group attack intention refers to the average suspicious degree of all target items, which is denoted as GAI_i and calculated by

$$\mathrm{GAI}_i = \frac{1}{\left|\mathrm{Target}_i\right|} \sum_{p \in \mathrm{Target}_i} \frac{\sum_{u \in \mathrm{Source}_i} F(u, p)}{\sum_{u' \in U} F(u', p)}, \tag{15}$$

where $F(u, p)$denotes an indicator function. $F(u, p) = 1$ if user u in $Source_i$ has rated item p with an abnormal rating; otherwise, $F(u, p) = 0$.

Definition 9. (group behaviour dissimilarity on nontarget items (GBDN)). For the ith candidate group in GSet, its group behaviour dissimilarity on nontarget items refers to the average dissimilarity degree on all nontarget items between members in Source_i, which is denoted as GBDN_i and calculated by

$$\mathrm{GBDN}_i = 1 - \frac{1}{\left|\mathrm{Source}_i\right|^2} \sum_{u,v \in \mathrm{Source}_i} \mathrm{NTSim}_{u,v}, \tag{16}$$

where $\mathrm{NTSim}_{u,v}$ denotes the similarity degree on all nontarget items between users u and v. In this work, we use cosine similarity formula to calculate the similarity degree.

To illustrate the effectiveness of the extracted group features, we use $\mathrm{GSAGen}_l\mathrm{Ran}$ and $\mathrm{GSAGen}_l\mathrm{Avg}$ to generate 10 shilling groups on the Netflix dataset, which contains 128 attack users. Next, we find out all suspicious groups via Algorithm 2. More details of the suspicious groups are shown in Table 4. It can be seen from Table 4 that 17 suspicious groups are discovered by Algorithm 2. In these groups, the first seven groups contain many genuine users and few attackers while the other groups only consist of attack users. As shown in Table 4, the difference between genuine groups and shilling groups in each extracted feature is significant.

Based on the above discussion, for the ith group in GSet, a vector of group features can be constructed and denoted by gfv_i= ($GBTS_i$, $GBTF_i$, $GBWS_i$, $GBWF_i$, GAI_i, $GBDN_i$). As

```
Input: Dis = $[\mathrm{dis}_{i,j}]_{|U|\times|U|}$, $\{\varphi(u), u \in \{1, 2, \ldots, |U|\}$, $Source_i$, $cnode$, $pnode$ and $\sigma$
Output: $Source_i$
tem←cnode
$TSet$←all neighbours of $tem$
for $\forall x \in T$Set do
    if $x \notin Source_i$ and $\varphi(x) \le \varphi(\mathrm{tem})$ and $\mathrm{disx}_{\mathrm{pnode}} \le \lfloor 3\sigma/\sqrt{2} \rfloor$ then
        $Source_i \leftarrow Source_i \cup \{x\}$
        cnode←$x$
        call Algorithm 3
    end if
end for
```

Algorithm 3: Local expanding

there is high similarity in group features between shilling groups, we can distinguish shilling groups from normal groups in the set of suspicious groups by using the k-means clustering algorithm.

The algorithm for detecting shilling groups is described as Algorithm 4, which mainly includes two parts. The first part (Lines 1–8) calculates six group features. The second part (Lines 9–14) divides all suspicious groups into two parts by using the k-means clustering algorithm. The suspicious groups in the larger cluster are regarded as shilling ones.

4. Experiments and Evaluation

4.1. Experimental Datasets and Settings. To evaluate the effectiveness of the proposed method, we use the following two datasets as the experimental data:

(1) The Netflix dataset (this dataset was published to support participants in the Netflix prize (http://netflixprize.com)): this dataset is a contest dataset including 103,297,638 rating records from 480,186 users on 17,770 items. Each rating record is a four-tuple of user id, item id, rating, and timestamp. All ratings are integers between 1 and 5, where 5 and 1 denote most liked and most disliked, respectively. We randomly sampled 215,884 records on 4,000 items by 2000 users between January 2000 and December 2005. Similar to the previous work of shilling attack detection in recommender systems, all profiles on this dataset are viewed as genuine ones. Shilling groups are generated by GSAGen$_l$Ran (or GSAGen$_l$Avg) and injected into the sampled dataset to construct a synthetic experimental dataset. The size of a shilling group is affected by the filler size and the input attack size in random (or average) attack, where the profile size in a shilling group equals the filler size and the number of profiles in a shilling group is usually much lower than the attack size. In the experiments, the input attack size is 20%, and the filler size is 2% and 3% in random (or average) attack, respectively. For each experiment, we generate 10 shilling groups with the same filler size. Each shilling group randomly selects five target items and each group member at least attacks 3 of 5 target items. The timestamp of each injected attack profile is randomly generated within a short period (e.g., 30 days in this paper). All attackers in each shilling group aim to push the target items. To make the results more accurate, we repeat the experiment 10 times. All experimental results are the average results of 10 experiments. More details of shilling groups injected in the experiments are shown in Table 5.

For ease of description, shilling groups generated under the condition of filler size 3% and 2% are denoted as loosely coupled shilling groups and tightly coupled shilling groups, respectively. As shown in Table 4, the group size of a tightly coupled shilling group is larger than that of a loosely coupled shilling group.

(2) The sampled Amazon review dataset: this dataset contains 1,205,125 rating records on 136,785 items by 645,072 users. Every rating is an integer and on a scale of one to five, where 1 and 5 represent the most disliked and liked, respectively. For the purpose of experimental evaluation, a sampled Amazon review dataset is obtained based on those labelled reviewers [36], which includes 53,777 rating records from 5055 users on 17,610 items. In the experiment, we filter out those users whose number of ratings is less than 5.

4.2. Evaluation Metrics. For the Netflix dataset, we have the ground truth information about the shilling groups because the injected shilling groups are generated by using group attack models proposed in Ref. [7]. For the sampled Amazon dataset, we have no ground truth information of shilling groups, e.g., the number of shilling groups, the size, the relationship between attack users and shilling groups, and so on. To evaluate the detection performance of TP-GBF and compare it with the existing detection methods, we first identify shilling groups and then calculate the number of attack users correctly detected and the number of genuine users misclassified as attack ones. The evaluation metrics are precision, recall, and $F1$-measure, which are defined as follows:

```
Input: time matrix $T$, rating matrix $R$, and the set of suspicious groups GSet = $\{g_1, g_2, \ldots, g_k\}$
Output: the set of shilling groups
for $\forall g_i \in$ GSet do
    Calculate the value of GBTS$_i$ using (11)
    Calculate the value of GBTF$_i$ using (12)
    Calculate the value of GBWS$_i$ using (13)
    Calculate the value of GBWF$_i$ using (14)
    Calculate the value of GAI$_i$ using (15)
    Calculate the value of GBDN$_i$ using (16)
end for
$[C_1, C_2]$ ←divide the set of suspicious groups into two clusters by using the k-means algorithm
if $|C_1| > |C_2|$ then
    return $C_1$
else
    return $C_2$
end if
```

Algorithm 4: Detecting shilling groups.

Table 4: Group features of each group in the set of suspicious groups on the Netflix dataset.

Group ID	Are all members attackers in the group?	Group size	GBTS	GBTF	GBWS	GBWF	GAI	GBDN
1	No	3	0.2813	0.0220	0.7995	0.0438	0.0098	0.6433
2	No	202	0.2655	0.0204	0.6062	0.1057	0.0102	0.9421
3	No	4	0.4097	0.0779	0.6239	0.1730	0.0237	0.6697
4	No	189	0.2979	0.0501	0.5487	0.1486	0.1835	0.7671
5	No	4	0.2790	0.0128	0.7970	0.0631	0.0388	0.6671
6	No	4	0.3157	0.0335	0.7459	0.1036	0.0078	0.6710
7	No	3	0.3587	0.0851	0.8243	0.0341	0.0087	0.6197
8	Yes	12	0.2716	0.0076	0.3352	0.0443	0.6750	0.8931
9	Yes	13	0.2723	0.0092	0.3421	0.0362	0.6350	0.9011
10	Yes	14	0.2716	0.0104	0.3183	0.0339	0.7062	0.9079
11	Yes	13	0.2719	0.0077	0.3105	0.0310	0.7336	0.9015
12	Yes	13	0.2726	0.0147	0.3259	0.0298	0.6231	0.8991
13	Yes	13	0.2688	0.0084	0.3215	0.0340	0.7155	0.8994
14	Yes	13	0.2696	0.0078	0.3364	0.0301	0.7361	0.9002
15	Yes	12	0.2720	0.0097	0.3460	0.0307	0.8231	0.8942
16	Yes	12	0.2727	0.0099	0.3337	0.0414	0.6864	0.8916
17	Yes	13	0.2775	0.0103	0.3258	0.0290	0.7256	0.9013

Table 5: Total size of ten shilling groups generated and injected into the Netflix dataset in ten experiments.

Profile size in a shilling group (%)	Total size of ten shilling groups in ten experiments									
3	122	130	118	120	123	123	130	125	133	123
2	190	197	189	210	200	195	195	205	204	195

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$
$$\text{recall} = \frac{\text{TP}}{P}, \tag{17}$$
$$F1 - \text{measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}},$$

where P denotes the total number of attack users, TP represents the number of attack users correctly detected, and FP refers to the number of genuine users misclassified as attack ones.

4.3. Experimental Results and Analysis. To verify the detection performance of the proposed method, we take the following four methods as baselines.

(1) DeR-TIA [28]: an unsupervised method for detecting group attack profiles in recommender systems, which detects attack users by using clustering

algorithm and target item analysis. In our experiments, the threshold for determining a target item is set to 6.

(2) CBS (catch the black sheep: unified framework for shilling attack detection based on fraudulent action propagation, CBS for short) [21]: an unsupervised method for detecting attack profiles in recommender systems, which calculates the spam probability of each user and each item iteratively, and detects attackers with the ranking of spam probability. In the experiments on two datasets, the number of attack users is assumed to be known in advance. In the experiments on the Netflix dataset, the detection result is the average result of top-$k_1(k_1 = k - 20)$, top-k, and top-$k_2(k_2 = k + 20)$, where k denotes the total number of attackers and 10% attack users are randomly selected as the candidate seeds. In the experiments on the Amazon dataset, the number of candidate seeds is set to 50 and the detection result is the average result of top-$k_1(k_1 = k - 50)$, top-k, and top-$k_2(k_2 = k + 50)$, where k denotes the total number of attackers. Considering the randomness of seed selection, each experiment is repeated ten times.

(3) GD-BKM [30]: an unsupervised method for detecting shilling groups in recommender systems, which generates candidate groups by dividing the rating tracks on each item and detects shilling groups by employing the bisecting K-means algorithm based on the suspicious degrees of candidate groups. In the experiments, the parameters K and TIL are set to 4 and 30, respectively.

(4) GSBC [33]: an unsupervised method for detecting spammer groups in e-commerce websites, which models spammer groups as biconnected graphs and identifies those groups with their spamicity scores. In the experiments on the Netflix dataset, coreview time window, similarity threshold, MP, and $MINSPAM$ are set to 30, 0.1, 1000, and 0, respectively. In the experiment on the Amazon dataset, coreview time window, similarity threshold, MP, and $MINSPAM$ are set to 30, 0.1, 1000, and 0.49, respectively.

(5) DSA-URB [22]: an unsupervised method based on MTD model and rank algorithm, which calculates the sum of differences of user suspicious degrees in each sliding window to determine the critical point of distinguishing genuine users from attack users. In the experiment, the hyper parameters α and β are set to 0.5 and 0.1, respectively. The number of topics is set to 20 on the Netflix dataset and is set to 10 on the Amazon dataset.

(6) DSA-AURB [25]: an unsupervised method proposed in our previous work, which detects attack profiles by identifying target items with abnormal rating distribution and analysing user behaviour differences between genuine users and attack users. In the experiment, the number of latent topics is set to 20 on the Netflix dataset and is set to 10 on the sampled Amazon dataset.

(7) DCEDM [31]: an unsupervised method for detecting shilling attacks, which extracted the robust graph features by using the stacked denoising autoencoders. In the experiment, the parameter ρ is set to 0.05 and β is set to 0.9. The noise level α is set to 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, and 0.1, respectively.

4.4. Parameters' Selection.

The parameter in TP-GBF is the impact factor σ. In this work, we select the optimal impact factor based on potential entropy [37], which reflects the uncertainty of topological potential and is defined as follows:

$$H = -\sum_{i=1}^{|U|} \frac{\varphi(u_i)}{Z} \log\left(\frac{\varphi(u_i)}{Z}\right), \tag{18}$$

where $Z = \sum_{i=1}^{|U|} \varphi(u_i)$is the normalization factor.

The smaller the potential entropy is, the smaller the uncertainty of topological potential is. Therefore, the optimal impact factor is calculated based on the minimum potential entropy. Let σ_{op} be the optimal impact factor and $H_{\min}$ be the minimum potential entropy, then the formula is:

$$\sigma_{op} = \arg\min H. \tag{19}$$

Figure 3 shows the relationship between $\sigma$and potential entropy on two datasets.

It can be seen from Figure 3 that the values of potential entropy initially decrease and then increase with the increasing of σ. On the Netflix dataset, the value of potential entropy reaches the minimum when σ is around 1. On the sampled Amazon dataset, the value of potential entropy reaches the minimum when σ is around 0.47 ($\sqrt{2}/3$). Therefore, we set σ to 1 and 0.47 on the Netflix and sample Amazon datasets, respectively.

4.5. Ratio of Remaining Genuine Users and Attack Users on Each Stage.

The TP-GBF method consists of three stages for detecting shilling groups. To further illustrate the importance of these stages, we conduct experiments to compare the ratio of the number of remaining genuine (attack) users to the total number of genuine (attack) users before and after the execution of each stage on the Netflix and Amazon datasets. The results are shown in Figure 4.

It can be seen from Figures 4(a) and 4(b), on the Netflix dataset, that the ratio of the number of remaining genuine users to the total number of genuine users decreases by about 10% after the first stage with all attack users retained, indicating that some genuine users can be filtered out in this stage but each attack user can be reserved because of having multiple lockstep behaviours. After the execution of stage 2, the ratio of the number of remaining attack users to the total number of attack users slightly decreases but only 20% of genuine users are kept. The reason is that majority of the local maximum potential nodes are attackers and most

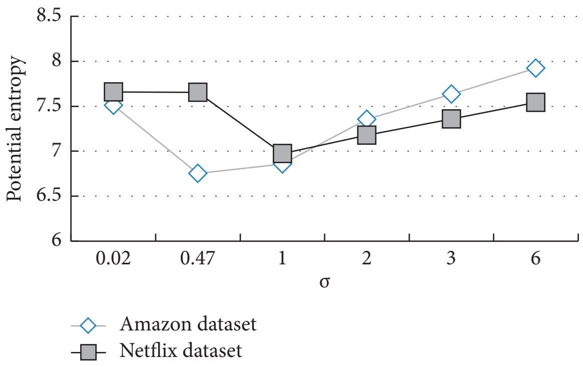
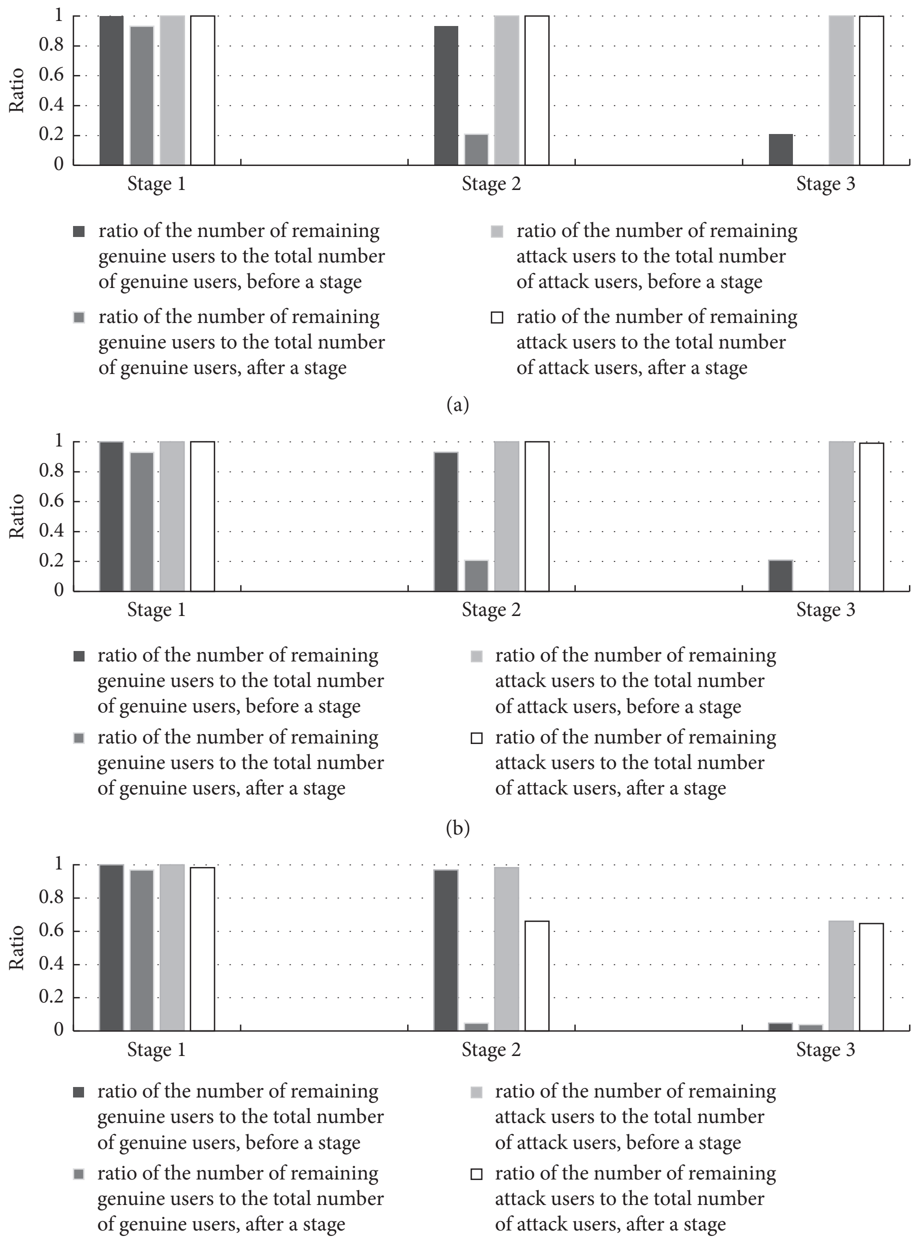


FIGURE 3: Relationship between parameter σ and potential entropy on Netflix and Amazon datasets.



(a)

(b)

(c)

FIGURE 4: Ratio of the number of remaining genuine (attack) users to the total number of genuine (attack) users before and after each stage. (a) Loosely coupled shilling groups on the Netflix dataset. (b) Tightly coupled shilling groups on the Netflix dataset. (c) Shilling groups on the sampled Amazon review dataset.

Table 6: Comparison of detection performance for five methods in detecting group attacks on the Netflix dataset.

Injected shilling groups	Method	Precision	Recall	$F1$-measure
Loosely coupled shilling groups	DeR-TIA	0	0	0
	CBS	0.4221	0.3758	0.3976
	GD-BKM	0.9734	0.9888	0.9810
	GSBC	0	0	0
	DSA-URB	0.8516	0.8195	0.8352
	DSA-AURB	0.9800	0.8033	0.8829
	DCEDM	0.5970	0.9231	0.7251
	TP-GBF	0.9903	0.9983	0.9943
Tightly coupled shilling groups	DeR-TIA	0	0	0
	CBS	0.4899	0.4888	0.4893
	GD-BKM	0.9823	0.9961	0.9892
	GSBC	0	0	0
	DSA-URB	0.9492	0.8578	0.9012
	DSA-AURB	0.9595	0.9456	0.9504
	DCEDM	0.8784	0.9898	0.9308
	TP-GBF	0.9968	0.9908	0.9938

genuine users are beyond the impact scale of these nodes in the weighted user relationship graph. After the execution of stage 3, the ratio of the number of remaining genuine users to the total number of genuine users reaches zero and that of attack users is about 100%. These results indicate that each stage of TP-GBF plays an important part in detecting group shilling attacks on the Netflix dataset.

As shown in Figure 4(c), on the sampled Amazon dataset, a small number of genuine users and attack users may be filtered in the first stage. This is because the sampled Amazon review dataset is very sparse and some users (both genuine and attack users) only rate a few items. After the execution of stage 2, the ratio of the number of remaining genuine users to the total number of genuine users is below 20% and the ratio of the number of remaining attack users to the total number of attack users is over 80%, indicating that most genuine users and some attackers are filtered out. The reason is that the group size of some shilling groups in this dataset is too small. After the execution of stage 3, the ratio of the number of remaining genuine users to the total number of genuine users drops below 5% and that of attack users remains over 60%. These results show that each stage of TP-GBF contributes to the detection of group shilling attacks on the sampled Amazon dataset.

4.6. Comparison of Detection Results on the Netflix Dataset. Table 6 shows the detection results of eight methods on the Netflix dataset. As listed in Table 6, on the Netflix dataset, both the precision and the recall of DeR-TIA are zero, indicating that DeR-TIA cannot detect shilling groups in this dataset. This is because DeR-TIA is designed to detect shilling groups with high similarity. However, the shilling group generated by the group shilling attack model in Ref. [7] shows low similarity between group members, which causes DeR-TIA to fail. When detecting loosely coupled shilling groups on the Netflix dataset, the precision, recall, and $F1$-measure of CBS are 0.4221, 0.3758, and 0.3976, respectively. The precision, recall, and $F1$-measure of CBS for detecting tightly coupled shilling groups are 0.4899, 0.4888, and 0.4893, respectively. In the experiments, the ratio of candidate seeds to the total number of attackers is 10%. As the number of candidate seeds is limited, the detection performance of CBS is not good. Moreover, these results also indicate that CBS can obtain better detection performance when there exists a stronger relation between group members. For GSBS, it is ineffective in detecting both loosely coupled and tightly coupled shilling groups on the Netflix dataset. The reason is that the number of target items for each shilling group is small on the Netflix dataset. For DSA-URB, under loosely coupled shilling groups, the precision, recall, and $F1$-measure are 0.8516, 0.8195, and 0.8352, respectively. For tightly coupled shilling groups, the precision, recall, and $F1$-measure of DSA-URB are 0.9492, 0.8578, and 0.9012, respectively. The precision of DSA-AURB is better than that of DeR-TIA, CBS, DSA-URB, and DCEDM for detecting two types of shilling groups, but its recall is only 0.8033 when detecting loosely coupled shilling groups. That is because DSA-AURB may omit some target items with a few ratings. For DCEDM, the precision, recall, and $F1$-measure are 0.8784, 0.9898, and 0.9308, respectively, indicating that DCEDM has very high recall in detecting tightly coupled shilling groups. However, the precision of DCEDM is low when detecting loosely coupled shilling groups. The reason is that DCEDM may divide some genuine users and attack users into a community when detecting loosely coupled shilling groups. The precision, recall, and $F1$-measure of GD-BKM for detecting loosely coupled shilling groups are 0.9734, 0.9888, and 0.9810, respectively. Under tightly coupled shilling groups, the precision, recall, and $F1$-measure of GD-BKM are 0.9823, 0.9961, and 0.9892, respectively. These results indicate that GD-BKM can achieve excellent detection performance on the Netflix dataset under group shilling attacks. The reason is that GD-BKM can detect various attackers based on abnormal items with larger number of ratings even if the attackers have a few corated target items. However, the detection precision and $F1$-measure are still

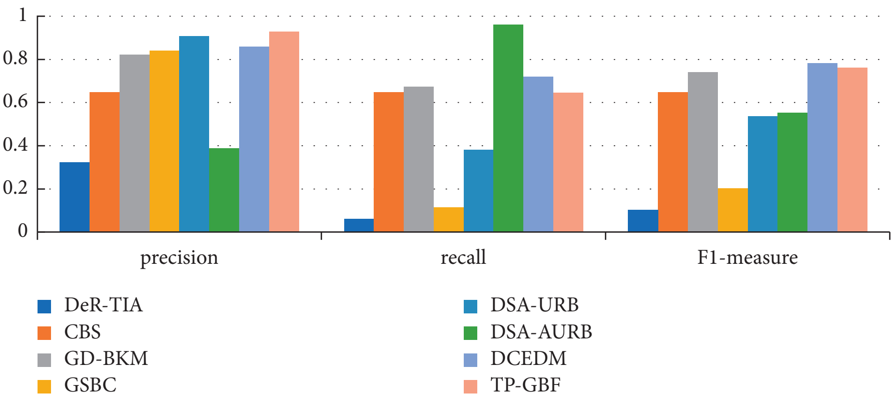


Figure 5: Comparison of the detection performance for eight methods on the sampled Amazon dataset.

inferior to that of TP-GBF. The precision, recall, and $F1$-measure of TP-GBF for detecting loosely coupled shilling groups are 0.9903, 0.9983, and 0.9943, respectively. The precision, recall, and $F1$-measure of TP-GBF for detecting tightly coupled shilling groups are 0.9968, 0.9908, and 0.9938, respectively. These results show that TP-GBF is effective and outperforms seven baseline methods in precision and $F1$-measure when detecting group shilling attacks on the Netflix dataset.

4.7. Comparison of Detection Results on the Sampled Amazon Dataset. Figure 5 shows the detection results of eight methods on the sampled Amazon dataset. As seen from Figure 5, on the sampled Amazon dataset, the detection performance of TP-GBF is the best among five methods in detecting group shilling attacks. The precision, recall, and $F1$-measure of DeR-TIA are 0.3251, 0.0607, and 0.1023, respectively. These results indicate that DeR-TIA can detect a few of the shilling groups correctly, but its recall is too low. The reason is that the sampled Amazon dataset is very sparse and behaviours of shilling groups on this dataset are varied. The precision, recall, and $F1$-measure of CBS are 0.6473, 0.6471, and 0.6472, respectively, indicating that CBS is effective for detecting group shilling attacks on the sampled Amazon dataset. Although 50 attackers are selected as the seeds, the detection performance of CBS is still inferior to that of TP-GBF. The precision, recall, and $F1$-measure of GD-BKM are 0.823, 0.673, and 0.74, respectively. Its recall is a little higher than that of TP-GBF, but its precision is much lower than that of TP-GBF, indicating that GD-BKM may misclassify some genuine groups. The precision of GSBC is 0.8409 but its recall is only 0.1146, indicating that many shilling groups cannot be detected by GSBC on the sampled Amazon dataset. This is because shilling groups with a few target items are usually filtered out by GSBC. The precision, recall, and $F1$-measure of DSA-URB are 0.9069, 0.3817, and 0.5373, respectively, indicating that this method is not effective for detecting group attacks on the sampled Amazon dataset. The reason is that the dataset is too sparse and a number of users only have a few ratings. The recall of DSA-AURB is very high but its precision is less than 0.4, indicating that a number of genuine users are identified as attackers on the Amazon dataset by DSA-AURB. On the sampled Amazon dataset, the precision and recall of DCEDM are 0.8601 and 0.7177, respectively, indicating that DCEDM is very effective for detecting attack groups. The reason is that the attackers in a shilling group usually have a tight relationship with each other on this dataset. The precision, recall, and $F1$-measure of TP-GBF are 0.9283, 0.6467, and 0.7623, respectively. This means TP-GBF not only can detect shilling groups accurately but also can also discover most of the shilling groups on the sampled Amazon dataset. Although the $F1$-measure of TP-GBF is slightly lower than the maximum $F1$-measure of all methods, its precision outperforms that of all baseline methods on the sampled Amazon dataset. These results indicate that TP-GBF is effective for detecting group shilling attacks on the sampled Amazon dataset.

4.8. Comparison of the Prediction Shift before and after Using TP-GBF. To illustrate the practical application of TP-GBF, we use the average prediction shift (ps) to measure the effects of using TP-GBF on the Netflix dataset, which is defined as follows [38]:

$$ps = \frac{1}{|\mathrm{Tar}|}\cdot\frac{1}{|U|}\cdot\sum_{i\in\mathrm{Tar}}\sum_{u\in U}\left|p'_{u,i} - p_{u,i}\right|, \tag{20}$$

where Tar and U indicate target items set and user set,$p'_{u,i}$ and $p_{u,i}$ are the pre- and post-detection predictions of user u on target item i, respectively.

We calculate predictions by using basic matrix factorization algorithm and divide the experimental data on a scale of 80% a training set and 20% a test set. All target items are pushed items with a few number of low ratings from genuine users. Figure 6 shows the comparision of ps before and after using TP-GBF. As seen in Figure 6, the ps values are 1.0196 and 0.0798 before and after using TP-GBF under attack with loosely coupled shilling groups, and the ps values are 2.1551 and 0.089 before and after using TP-GBF under attack with tightly coupled shilling groups. These results indicate that TP-GBF is very helpful to reduce the influence of group shilling attacks on the Netflix dataset.

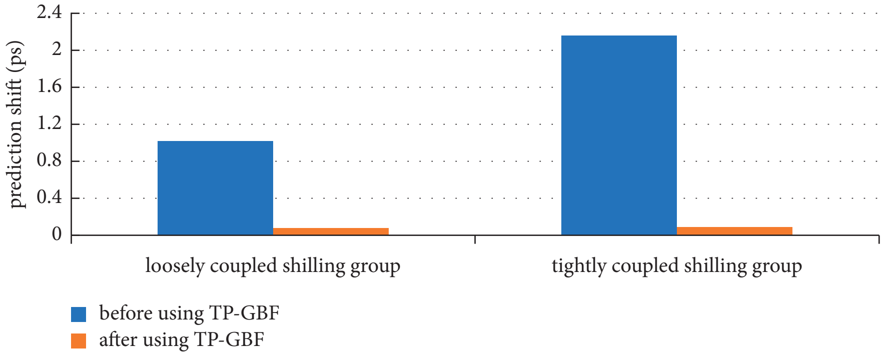


Figure 6: Comparison of the prediction shift before and after using TP-GBF.

Table 7: Recommendation lists of ten users before and after group shilling attack detection.

Users	Recommendation lists	
	Before group shilling attack detection	After using TP-GBF
u_{10}	$\mathbf{i}_{2925}, i_{857}, \mathbf{i}_{3794}, \mathbf{i}_{2726}, i_{2772}, i_{64}, \mathbf{i}_{3680}, i_{2907}, \mathbf{i}_{2826}, i_{3501}$	$i_{857}, i_{2907}, i_{64}, i_{1322}, i_{1440}, i_{1870}, i_{2289}, i_{3389}, i_{3784}, i_{1799}$
u_{205}	$\mathbf{i}_{2925}, \mathbf{i}_{2726}, i_{3514}, i_{857}, i_{960}, i_{445}, \mathbf{i}_{1001}, i_{3328}, i_{885}, i_{1322}$	$i_{2541}, i_{1065}, i_{445}, i_{2648}, i_{3631}, i_{3328}, i_{1263}, i_{862}, i_{1075}, i_{2843}$
u_{385}	$\mathbf{i}_{2925}, i_{857}, i_{1597}, \mathbf{i}_{3794}, \mathbf{i}_{2726}, i_{3241}, i_{1398}, i_{3501}, \mathbf{i}_{1001}, i_{1263}$	$i_{1597}, i_{1381}, i_{2436}, i_{3389}, i_{1263}, i_{1065}, i_{2748}, i_{12}, i_{188}, i_{2607}$
u_{764}	$\mathbf{i}_{2925}, i_{1263}, i_{1234}, i_{3939}, \mathbf{i}_{3794}, i_{3631}, i_{3687}, i_{1163}, i_{2843}, i_{1075}$	$i_{2362}, i_{1263}, i_{2436}, i_{2474}, i_{1157}, i_{1883}, i_{3039}, i_{1373}, i_{2926}, i_{2772}$
u_{853}	$\mathbf{i}_{2925}, i_{857}, \mathbf{i}_{3680}, i_{2772}, \mathbf{i}_{3794}, \mathbf{i}_{2726}, i_{2907}, i_{1398}, i_{1860}, i_{3235}$	$i_{2590}, i_{2165}, i_{3715}, i_{2218}, i_{1163}, i_{1014}, i_{1860}, i_{3803}, i_{2648}, i_{575}$
u_{1093}	$\mathbf{i}_{2925}, i_{857}, i_{3328}, \mathbf{i}_{2726}, i_{1398}, i_{2907}, \mathbf{i}_{3794}, \mathbf{i}_{3680}, i_{3501}, i_{3687}$	$i_{1263}, i_{2436}, i_{2772}, i_{2648}, i_{2176}, i_{3328}, i_{2617}, i_{1322}, i_{1932}, i_{3124}$
u_{1649}	$\mathbf{i}_{2925}, i_{857}, \mathbf{i}_{2726}, i_{2590}, \mathbf{i}_{3680}, i_{2772}, \mathbf{i}_{3794}, i_{3687}, i_{2907}, i_{1263}$	$i_{2436}, i_{2617}, i_{445}, i_{1065}, i_{2362}, i_{3328}, i_{1014}, i_{1622}, i_{1234}, i_{2343}$
u_{1673}	$\mathbf{i}_{2925}, i_{857}, \mathbf{i}_{2726}, i_{1398}, \mathbf{i}_{3680}, i_{1831}, i_{2907}, \mathbf{i}_{3794}, i_{3241}, i_{2436}$	$i_{3278}, i_{3374}, i_{1102}, i_{2244}, i_{2610}, i_{3427}, i_{2744}, i_{518}, i_{3505}, i_{2042}$
u_{1708}	$i_{857}, \mathbf{i}_{2925}, i_{2590}, i_{1263}, \mathbf{i}_{2826}, \mathbf{i}_{2726}, i_{1014}, i_{2772}, \mathbf{i}_{1001}, i_{1404}$	$i_{857}, i_{1322}, i_{2772}, i_{2436}, i_{1263}, i_{3241}, i_{1597}, i_{3278}, i_{2907}, i_{3015}$
u_{1991}	$i_{3720}, i_{2362}, i_{1263}, \mathbf{i}_{2925}, i_{3784}, i_{2218}, i_{1014}, i_{857}, i_{3687}, \mathbf{i}_{2726}$	$i_{2436}, i_{3720}, i_{2474}, i_{2772}, i_{2218}, i_{3631}, i_{1263}, i_{857}, i_{2362}, i_{2617}$

4.9. A Case Study on User Satisfaction with Item Recommendation Lists. To illustrate whether TP-GBF can help to improve user satisfaction, we compare users' recommendation lists before and after group shilling attack detection on the Netflix dataset. We use $GSAGen_l$ Ran to generate ten loosely coupled shilling groups that include 119 attackers and inject into the Netflix dataset. The profile size in a shilling group is 3%. Each shilling group randomly selects five target items and each group member at least rates 3 of 5 target items with the maximum rating. The timestamp of each injected attack profile is randomly generated within 30 days. We use a matrix factorization-based collaborative filtering recommendation algorithm to generate an item recommendation list for each user. Table 7 presents the recommendation lists of ten users, where the bold items indicate target items.

As listed in Table 7, the recommendation lists generated for the same user are quite different before and after group shilling attack detection. Before group shilling attack detection, each user's recommendation list contains at least two target items, which means that the group shilling attacks have a strong influence on the recommendation lists of ten users. After detecting group shilling attacks using TP-GBF, no target items appear in the recommendation lists of ten users, indicating that the recommendation algorithm does not recommend target items to the ten users. By analysing the recommendation lists of the ten users after detecting group shilling attacks using TP-GBF, we found that these users always give high score (i.e., 4 or 5) to the corresponding recommended items, which indicates that the users like the recommended items. Therefore, we can conclude that TP-GBF can help to improve user satisfaction when facing group shilling attacks in recommender systems.

5. Conclusions and Future Work

Compared with traditional shilling attacks in recommender systems, group shilling attacks are more harmful and much harder to detect. To improve the detection performance and minimize the impact of group shilling attacks on recommender systems, we present a three-stage group shilling attack detection method based on lockstep behaviour and group behaviour features. We propose a method of calculating collusive degree between users based on lockstep behaviour and transitivity of direct relations, based on which a weighted user relationship graph is constructed. We calculate the topology potential for each user node in the weighted user relationship graph and find out all suspicious groups by local maximum potential nodes. We propose six group behaviour features to characterize the differences between shilling groups and normal ones in the set of suspicious groups, and devise a k-means clustering-based algorithm to detect the shilling groups. The experimental results on the Netflix and Amazon datasets show that TP-GBF is effective for detecting group shilling attacks in recommender systems and outperforms three baseline methods in precision, recall, and $F1$-measure.

The proposed TP-GBF detection method uses the k-means clustering algorithm to partition the set of suspicious groups. As the k-means clustering algorithm is a hard classification method, each suspicious group is regarded as a whole and members of the same group are identified as malicious or genuine users. This hard classification method lacks flexibility, which may misclassify a few attackers who have strong correlation with some genuine users.

In our future research, we will improve our method by using the fuzzy soft clustering algorithm and dimensionality reduction techniques on big data [39]. By further analysing the rating behaviours of genuine and attack users under group shilling attacks, we measure the membership degree of each user in the corresponding suspicious group to further improve the detection performance of group shilling attacks in recommender systems.

Data Availability

The data used to support the findings of this study are available from the first author (chy_hbu@126.com) upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61772452), the Natural Science Foundation of Hebei Province, China (nos. F2015203046, F2019205303, and F2020201023), and the Hebei Provincial Department of Education, China (no. ZD2018023).

References

[1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, Springer, Berlin, Germany, 2015.

[2] M. Si and Q. Li, "Shilling attacks against collaborative recommender systems: a review," *Artificial Intelligence Review*, vol. 53, no. 1, pp. 291–319, 2020.

[3] Y. Wang, L. Qian, F. Li, and L. Zhang, "A Comparative study on shilling detection methods for trustworthy recommendations," *Journal of Systems Science and Systems Engineering*, vol. 27, no. 4, pp. 458–478, 2018.

[4] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, "Shilling attacks against recommender systems: a comprehensive survey," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 767–799, 2014.

[5] A. M. Turk and A. Bilge, "Robustness analysis of multi-criteria collaborative filtering algorithms against shilling attacks," *Expert Systems with Applications*, vol. 115, pp. 386–402, 2019.

[6] X. Su, H. Zeng, and Z. Chen, "Finding group shilling in recommendation system," in *Proceedings of the Special Interest Tracks & Posters of the International Conference on World Wide Web*, pp. 960-961, New York, NY, USA, May 2005.

[7] Y. Wang, Z. Wu, J. Cao, and C. Fang, "Towards a tricksy group shilling attack model against recommender systems," *Advanced Data Mining and Applications*, vol. 7713, pp. 675–688, 2012.

[8] P. Chirita, l. W. Nejd, and C. Zamfir, "Preventing shilling attacks in online recommender systems," in *Proceedings of the 7th annual ACM international workshop on web information and data management*, pp. 67–74, Bremen Germany, November 2005.

[9] R. Burke, B. Mobasher, and C. Williams, "Classification features for attack detection in collaborative recommendation systems," in *Proceedifgs of the 12th International Conference on Knowledge Discovery and Data Mining*, pp. 542–547, Philadelphia, PA, USA, August 2006.

[10] Z. Wu, Y. Zhuang, and Y. Wang, "Shilling attack detection based on feature selection for recommendation system," *Journal.Acta Electronica Sinica*, vol. 40, no. 8, pp. 1687–1693, 2012, in Chinese with English abstract.

[11] Z. Yang, L. Xu, Z. Cai, and Z. Xu, "Re-scale AdaBoost for attack detection in collaborative filtering recommender systems," *Knowledge-Based Systems*, vol. 100, pp. 74–88, 2015.

[12] C. Tong, X. Yin, J. Li et al., "A shilling attack detector based on convolutional neural network for collaborative recommender system in social aware network," *The Computer Journal*, vol. 61, no. 7, pp. 949–958, 2018.

[13] Y. Hao, P. Zhang, and F. Zhang, "Multiview ensemble method for detecting shilling attacks in collaborative recommender systems," *Security and Communication Networks*, vol. 2018, no. 4, 33 pages, Article ID 8174603, 2018.

[14] Q. Zhou, J. Wu, and L. Duan, "Recommendation attack detection based on deep learning," *Journal of Information Security and Applications*, vol. 52, Article ID 102493, 2020.

[15] S. Zhang, H. Yin, and T. Chen, "GCN-Based representation learning for unifying robust recommendation and fraudster detection," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 689–698, Xi'an, China, July 2020.

[16] K. Vivekanandan and N. Praveena, "Hybrid convolutional neural network (CNN) and long-short term memory (LSTM) based deep learning model for detecting shilling attack in the social-aware network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1197–1210, 2021.

[17] Z. Wu, J. Cao, Y. Wang, Y. Wang, L. Zhang, and J. Wu, "hPSD: a hybrid PU-Learning-Based spammer detection model for product reviews," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1595–1606, 2020.

[18] B. Mehta and W. Nejdl, "Unsupervised strategies for shilling detection and robust collaborative filtering," *User Modeling and User-Adapted Interaction*, vol. 19, no. 1-2, pp. 65–97, 2009.

[19] Z. Zhang and S. Kulkarni, "Detection of shilling attacks in recommender systems via spectral clustering," in *Proceedings of the 17th International Conference on Information Fusion*, pp. 1–8, IEEE, Salamanca, Spain, July 2014.

[20] H. Xia, B. Fang, M. Gao, H. Ma, Y. Tang, and J. Wen, "A novel item anomaly detection approach against shilling attacks in collaborative recommendation systems using the dynamic time interval segmentation technique," *Information Sciences*, vol. 306, pp. 150–165, 2015.

[21] Y. Zhang, Y. Tan, M. Zhang, and Y. Liu, "Catch the black sheep: unified framework for shilling attack detection based on fraudulent action propagation," in *Proceedings of the International Conference on Artificial Intelligence*, pp. 2408–2414, Buenos Aires Argentina, July 2015.

[22] F. Zhang, Z. Ling, and S. Wang, "Unsupervised approach for detecting shilling attacks in collaborative recommender systems based on user rating behaviours," *IET Information Security*, vol. 13, no. 3, pp. 174–187, 2019.

[23] Z. Yang, Q. Sun, and Y. Zhang, “Probabilistic inference and trustworthiness evaluation of associative links toward malicious attack detection for online recommendations,” *Transactions on Dependable and Secure Computing*, vol. 1, p. 1, 2020.

[24] 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,” *Knowledge-Based Systems*, vol. 148, pp. 146–166, 2018.

[25] H. Cai and F. Zhang, “An unsupervised method for detecting shilling attacks in recommender systems by mining item relationship and identifying target items,” *The Computer Journal*, vol. 62, no. 4, pp. 579–597, 2019.

[26] H. Cai and F. Zhang, “Detecting shilling attacks in recommender systems based on analysis of user rating behavior,” *Knowledge-Based Systems*, vol. 177, pp. 22–43, 2019.

[27] J. Gao, Y. Dong, M. Shang, S. Cai, and T. Zhou, “Group-based ranking method for online rating systems with spamming attacks,” *EPL (Europhysics Letters)*, vol. 110, no. 2, Article ID 28003, 2015.

[28] W. Zhou, Y. Koh, J. Wen, S. Alam, and G. Dobbie, “Detection of abnormal profiles on group attacks in recommender systems,” in *Proceedings of the 37th international ACM SIGIR conference on Research & Development in information retrieval*, pp. 955–958, Gold Coast Queensland Australia, July 2014.

[29] G. Gabriel, L. Robson, and F. Jose, “ORFEL: efficient detection of defamation or illegitimate promotion in online recommendation,” *Information Sciences*, vol. 379, pp. 274–287, 2017.

[30] F. Zhang and S. Wang, “Detecting group shilling attacks in online recommender systems based on bisecting K-means clustering,” *IEEE Transactions On Computational Social Systems*, vol. 7, no. 5, pp. 1189–1199, 2020.

[31] Y. Hao and F. Zhang, “An unsupervised detection method for shilling attacks based on deep learning and community detection,” *Soft Computing*, vol. 25, no. 1, pp. 477–494, 2020.

[32] Y. Wang, Z. Wu, Z. Bu, J. Cao, and D. Yang, “Discovering shilling groups in a real e-commerce platform,” *Online Information Review*, vol. 40, no. 1, pp. 62–78, 2016.

[33] Z. Wang, S. Gu, X. Zhao, and X. Xu, “Graph-based review spammer group detection,” *Knowledge and Information Systems*, vol. 55, no. 3, pp. 571–597, 2018.

[34] Z. Wang, Z. Li, G. Yuan, Y. Sun, X. Rui, and X. Xiang, “Tracking the evolution of overlapping communities in dynamic social networks,” *Knowledge-Based Systems*, vol. 157, pp. 81–97, 2018.

[35] Z. Wang, Z. Li, X. Ding, and T. Jin, “Overlapping community detection based on node location analysis,” *Knowledge-Based Systems*, vol. 105, pp. 225–235, 2016.

[36] C. Xu, J. Zhang, K. Chang, and C. Long, “Uncovering collusive spammers in Chinese review websites,” in *Proceedings of the 22nd ACM International Conference on conference on Information & Knowledge Management*, pp. 979–988, San Francisco, CA, USA, October 2013.

[37] W.-Y. Gan, N. He, D.-Y. Li, and J.-M. Wang, “Community discovery method in networks based on topological potential,” *Journal of Software*, vol. 20, no. 8, pp. 2241–2254, 2009, in Chinese with English abstract.

[38] Z. Yang, Q. Sun, and B. Zhang, “Evaluating prediction error for anomaly detection by exploiting matrix factorization in rating systems,” *IEEE Access*, vol. 6, Article ID 50014, 2018.

[39] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna et al., “Analysis of dimensionality reduction techniques on big data,” *IEEE Access*, vol. 8, Article ID 54776, 2020.