Detecting shilling groups in recommender systems based on hierarchical topic model

Shilei Wang, Hui Wang, Hongtao Yu, Fuzhi Zhang
School of Information Science and Engineering, Yanshan University
Qinhuangdao, Hebei Province, China
Shileiwang546@163.com, 756085343@qq.com, yu5771@163.com, xjzfz@ysu.edu.cn

2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA) | 978-1-6654-1867-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICAICA52286.2021.9498071

***Abstract*—In a group shilling attack, attackers work collaboratively to inject fake profiles aiming to obtain desired recommendation result. This type of attacks is more harmful to recommender systems than individual shilling attacks. Previous studies pay much attention to detect individual attackers, and little work has been done on the detection of shilling groups. In this work, we introduce a topic modeling method of natural language processing into shilling attack detection and propose a shilling group detection method on the basis of hierarchical topic model. First, we model the given dataset to a series of user rating documents and use the hierarchical topic model to learn the specific topic distributions of each user from these rating documents to describe user rating behaviors. Second, we divide candidate groups based on rating value and rating time which are not involved in the hierarchical topic model. Lastly, we calculate group suspicious degrees in accordance with several indicators calculated through the analysis of user rating distributions, and use the k-means clustering algorithm to distinguish shilling groups. The experimental results on the Netflix and Amazon datasets show that the proposed approach performs better than baseline methods.**

Keywords—Recommender systems, Group shilling attacks, Shilling group detection, Hierarchical topic model, Clustering

I. INTRODUCTION

Nowadays, recommender systems have been widely used in many fields such as e-commerce websites and social web applications. However, recommender systems encounter shilling attacks due to their openness. Previous studies focus mainly on detecting individual shilling attacks in which malicious users work separately to inject shilling profiles in an attempt to promote or demote a particular item being recommended [1]. With the evolving of shilling attacks, a group of attackers work collaboratively to inject shilling profiles to bias the output of the recommender system. The group of collaborative attackers is usually called a shilling group, and the behavior that a group of attackers cooperate to bias the output of the recommender system is called a group shilling attack [2]. Compared with individual shilling attacks, group shilling attacks are more threatening to recommender systems. Therefore, how to effectively detect group shilling attacks has become important for the provision of reliable recommendations.

Over the past decade, the problem of detecting individual shilling attacks has been well studied [3-7], but little work focuses on detecting group shilling attacks. In this work, we propose an approach for detecting shilling groups in recommender systems, which is based on the hierarchical Dirichlet process (HDP) and the k-means clustering algorithm (called HDP-KM). Particularly, we first use the HDP model to extract item latent topics and thus construct user vectors. Then we find candidate groups by analyzing rating target and rating time interval. Lastly, we conduct the anomaly degrees of candidate groups and utilize the k-means clustering algorithm to obtain shilling groups.

The main contributions of this study are as follows:

(1) We propose a hierarchical modeling method to describe user rating behaviors. This method models the given dataset to a series of user rating documents and employs the HDP model to find item latent topics without specifying the number of topics. Based on which, the low-dimensional user vectors are constructed to describe user rating behaviors.

(2) To analyze group member structure, we utilize ordering points to identify the clustering structure (OPTICS) to summarize user rating patterns. Then we measure the distance between users and divide users into cliques in accordance with density.

(3) We propose several indicators based on user vectors and group member structure and calculate group suspicious degrees through a comprehensive combination of these indicators. Then, we identify shilling groups by performing the k-means algorithm on group suspicious degrees.

The organization of the remaining paper rest of this paper is: the related work is introduced in Section 2; in Section 3, the proposed detection method is described; HDP-KM is evaluated in Section 4; Section 5 draws conclusions.

II. RELATED WORK

Great progress has been made in detecting individual shilling attacks. However, group shilling attacks have a strict strategy, and attackers cooperate closely; thus, traditional shilling attack detection methods hardly detect them. Presently, the detection of shilling groups has not been brought to the forefront. Zhou et al. [10] proposed a two-step method for detecting shilling groups. In the first step, they used the improved degree of similarity with top neighbors (*Degsim'*) and rating deviation from mean agreement (*RDMA*) to divide the group profiles into two parts. Then, they extracted the interaction of two higher parts as a result of step 1. In the second step, the target item analysis was used to determine the attack profiles. However, this approach can only detect attackers with enough co-rated items. When attackers have low similarity, this method does not perform ideally. Wang et al. [11] proposed five features and utilized principal component analysis to obtain the

major features. Based on the major features, the group anomaly degrees are calculated, and the candidate groups are ranked. However, the result of frequent pattern mining which is used in this method is easily influenced by the setting of support threshold. Many groups are filtered out with a high threshold, which leads to a low recall of the method. Otherwise, many genuine users are mixed in, and the method suffers from high detection false rate. Zhang et al. [12] proposed a method based on graph embedding. They constructed a weighted user graph and used nod2vec to generate user representations. Then, k-means++ is used to divide candidate groups, and group suspicious degrees are calculated. Lastly, they used Ward's hierarchical clustering algorithm to identify attack groups.

III. THE PROPOSED SHILLING GROUP DETECTION APPROACH

The proposed HDP-KM includes three procedures. User vectors are constructed in the first procedure. In the second step, candidate groups are divided based on a comprehensive consideration of different aspects of shilling groups. In the last step, group anomaly degrees are computed, and shilling groups are identified by k-means.

To facilitate discussion, notations and corresponding descriptions are list in Table I.

TABLE I. NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
U	set of users in the recommender system
I	set of items in the recommender system
T	topic pool
K	the number of latent topics
x_{ji}	the i-th item rated by the j-th user
x_j	set of items rated by user j
E_j	set of latent topics corresponding to items rated by user j
E	set of E_j
r_{ji}	rating of user j for item i
G	set of candidate groups
g_i	a candidate group divided under item i
$\lvert\bullet\rvert$	the number of elements in a set

A. Describing user rating distributions

We model the given dataset to a series of user rating documents and use HDP to reveal topic distribution from the user level, which helps to distinguish the differences between users. The specific process of describing user rating distributions is introduced below.

(1) Extract item latent topics

In order to use the HDP model to extract item latent topics, we treat an item rated by a user as a word and model the rating dataset to a series of documents according to users. Each document consists of a set of items rated by a user.

The rating document of user $j \in U$ can be represented with a set of items rated by user j, that is:

$$x_j = \{x_{j1}, x_{j2}, ..., x_{jn}\}$$

where $x_{j1}, x_{j2}, ..., x_{jn}$ denote n items rated by user j.

All rating documents of users in the dataset consist of a multi-document structure, which is represented as

$$X = \{x_1, x_2, ..., x_{|U|}\}$$

where x_j $(j = 1, 2, ..., |U|)$ denotes the rating document of user j.

Based on the multi-document structure X, we can use the HDP model to extract item latent topics.

The steps for constructing the HDP model are as follows.

First, for each item x_{ji} , a topic t is chosen from topic pool T .

The conditional probability for item choosing topic t is

$$p\left(t_{ji} = t \mid T^{-ji}, K\right) \propto \begin{cases} n_{jt\bullet}^{-j,i} f_{k_j}^{-x_j}\left(x_{ji}\right), t \text{ is used} \\ \alpha_0 p\left(x_{ji} \mid T^{-ji}, t_{ji} = t^{\text{new}}, k\right), t = t^{\text{new}} \end{cases} \quad (1)$$

where t^{new} represents that item x_{ji} chooses a new topic; then,

$$\begin{aligned} & p\left(x_{ji} \mid T^{-ji}, t_{ji} = t^{\text{new}}, K\right) \\ & = \sum_{k=1}^{K} \frac{m_{\bullet k}}{\gamma + m_{\bullet\bullet}} f_k^{-x_s}\left(x_{ji}\right) + \frac{\gamma}{\gamma + m_{\bullet\bullet}} f_{k^{newji}}^{-x_i}\left(x_{ji}\right) \end{aligned} \quad (2)$$

Second, topic ids are selected by topics.

The probability distribution for the topic id k chosen by a topic is:

$$p\left(k_{jt} = k \mid T, K^{-jt}\right) \propto \begin{cases} m_{\cdot k}^{-jt} f_k^{-x_{ji}}\left(x_{jt}\right), k \text{ is used} \\ \gamma f_{k^{newji}}^{-x_k}\left(x_{jt}\right), k = k^{\text{new}} \end{cases} \quad (3)$$

In Eqs. (1)-(3), superscript means the variable is not counted, such as $K^{-jt} = K / k_{jt}$. In addition, $n_{jt\bullet}$ represents the number of items rated by user j who chose topic t, $m_{\bullet k}$ represents the count of topics choosing topic id k, $m_{\bullet\bullet}$ represents topic amount.

(2) Construct user vectors

After using the HDP model to extract latent topics for all items, we can replace items in each x_j with the numbers of corresponding topics to construct user vectors.

For any user $j \in U$, let $tn_1^j, tn_2^j, ..., tn_K^j$ represent the numbers of topics respectively, the vector of user j can be represented as $UV_j = (tn_1^j, tn_2^j, ..., tn_K^j)$. The algorithm of constructing user vectors is described below.

Algorithm 1: Constructing user vectors

Input: rating dataset

Output: the set of user vectors UV

Begin

$X \leftarrow \varnothing$, $UV \leftarrow \varnothing$
for each user $j \in U$ **do**
  $x_j \leftarrow \varnothing$
  for each item $i \in I$ **do**
    if $r_{ji} \neq 0$ **then**
      $x_j \leftarrow x_j \cup \{i\}$
    end if
  end for
  $X \leftarrow X \cup \{x_j\}$
end for
$E \leftarrow HDP(X)$
for each user $j \in U$ **do**
  for each topic id k =1 to K **do**
    count topic number tn_k^j in E_j
  end for
  construct user vector $UV_j = (tn_1^j, tn_2^j, ..., tn_K^j)$
  $UV \leftarrow UV \cup \{UV_j\}$
end for
return UV

End

B. Dividing candidate groups

While the HDP model can be used to describe user rating behaviors based on user rating items, the rating time and rating value are not used in the sampling process. Considering that the attackers in a shilling group rate the same target item(s) with extreme ratings and finish rating within a short time, these two attributes should be taken into account when dividing candidate groups.

In the datasets used in this study, a rating record contains four parts: user, item, rating value, and rating time. For each item $i \in I$, information about it can form a sequence in chronological order. This sequence is called an item rating sequence and denoted by:

$$IRS_i = \left\{(u_1^i, r_1^i, t_1^i), (u_2^i, r_2^i, t_2^i), ..., (u_s^i, r_s^i, t_s^i)\right\}$$

where $u_1^i, u_2^i, ..., u_s^i$ represent the users who rate item i with ratings $r_1^i, r_2^i, ..., r_s^i$ at time $t_1^i, t_2^i, ..., t_s^i$, respectively.

The average time interval length (*AvgTIL*) of item i's rating sequence can be calculated by:

$$AvgTIL_i = \frac{t_s^i - t_1^i}{|IRS_i|} \tag{4}$$

The process of candidate group division are as follows.

(1) For each item $i \in I$, construct item rating sequence and calculate *AvgTIL* of the rating sequence;

(2) Remove users who do not give the maximum rating from the item rating sequence;

(3) In the remaining item rating sequence, set the first user as current user and calculate the rating time interval length *TIL* between the current user and the next user. If *TIL* is less than the average time interval length, the two users will be divided into the same group and the next user will be set as the current user; otherwise, the next user will be divided into a new group and set as the current user. This operation ends if all the users are in candidate groups.

(4) Repeat steps (2) and (3) until all the item rating sequences are divided.

C. Detecting shilling groups

In this section, we first compute group suspicious degree o, and then use a clustering method to detect shilling groups.

Definition 1 (*user intent similarity, UIS*). For any user $u \in U$ and user $u \in U$, the intent similarity of users u and v is the cosine similarity of their vectors, which is calculated as follows:

$$UIS_{uv} = \frac{UV_u \bullet UV_v}{\|UV_u\| \|UV_v\|} = \frac{\sum_{k=1}^{K} tn_k^u \times tn_k^v}{\sqrt{\sum_{k=1}^{K} (tn_k^u)^2} \times \sqrt{\sum_{k=1}^{K} (tn_k^v)^2}} \tag{5}$$

UIS_{uv} ranges from 0 to 1. The bigger the value, the more similar the intents of the two users.

Definition 2 (*group intent similarity, GIS*). For any group $g \in G$, the intent similarity of group g refers to the average value of intent similarity between users in group g, which is calculated as follows:

$$GIS_g = \frac{\sum_{u \in g} \sum_{v \in g, v \neq u} UIS_{uv}}{\eta_g} \tag{6}$$

$$\eta_g = \frac{|g| \times (|g| - 1)}{2} \tag{7}$$

where η_g is the number of user pairs formed by all users in group g. In addition, we think a shilling group should have at least two users; thus, we do not consider groups having one user, thereby ensuring $\eta_g \neq 0$.

To further analyze shilling groups, we perform the OPTICS algorithm [8] on user rating distributions and obtain user rating patterns.

Before clustering, we use term frequency–inverse document frequency (TF-IDF) to normalize user rating vectors.

Definition 3 (*topic weight, TW*). For any topic id $k \in \{1, 2, ..., K\}$, the topic weight in each $E_j \in E$ is the product of term frequency and inverse documentation frequency, which is calculated by:

$$tw_k^j = tf_k^j \times idf_k \tag{8}$$

$$idf_k = \log \frac{|U|}{1+\sum_{j \in U} tn_k^j} \tag{9}$$

$$tf_k^j = \frac{tn_k^j}{\sum_{k=1}^{K} tn_k^j} \tag{10}$$

We add one to the denominator to avoid it being zero in Eq. (9).

Definition 4 (*user normalization vector, UNV*). For any user $j \in U$, the normalization vector of user *j* refers to the vector formed by the weights of all topics and is denoted as

$$UNV_j = (tw_1^j, tw_2^j, ..., tw_K^j) \tag{11}$$

Based on user normalization vectors, we use the OPTICS algorithm to divide user rating behaviors into *n* patterns $p_1, p_2, ..., p_n$.

Definition 5 (*group user concentration degree, GUCD*). For any group $g \in G$, the user concentration degree of group *g* refers to the concentration degree of rating patterns of all users in group *g*, which is calculated as follows:

$$GUCD_g = \frac{\max(|p_m^g|)_{m=1,2,...,n}}{|g| \times (1 - \sum_{m=1}^{n} \frac{|p_m^g|}{|g|} \times \log_2 \frac{|p_m^g|}{|g|})} \tag{12}$$

where p_m^g denotes the set of users whose rating behaviors belong to pattern *m* in group *g*.

Attackers in a shilling group have the same rating patterns; thus, the higher the *GUCD*, the more suspicious the group is.

To bias the recommendation result of the target item, the number of ratings provided by group attackers for the target item should be much more than the count of ratings provided by normal users.

Definition 6 (*group item ratio, GIR*). For any group $g \in G$, the group *g*'s item ratio refers to the proportion the size of group *g* to the amount of users who have rated item *i*, which is calculated by

$$GIR_g = \frac{|g|}{|IRS_i|} \tag{13}$$

Definition 7 (*group suspicious degree, GSD*). The suspicious degree of group $g \in G$ is calculated as follows:

$$GSD_g = GIR_g \times \frac{GIS_g + GUCD_g}{2} \tag{14}$$

According to Eq. (14), we can obtain the group suspicious degrees. Then we perform the k-means algorithm to obtain clusters of shilling groups and normal user groups. The detecting algorithm is described as follows:

Algorithm 2: Detecting shilling groups

Input: candidate groups *G*

Output: shilling groups *AG*

Begin

1: $GSD \leftarrow \varnothing, UNV \leftarrow \varnothing$
2: **for** each user $j \in U$ **do**
3: $UNV_j \leftarrow$ TF-IDF(UV_j)
4: $UNV \leftarrow UNV \cup \{UNV_j\}$
5: **end for**
6: $P \leftarrow OPTICS(UNV)$
7: **for** each group $g \in G$ **do**
8: calculate GSD_g according to Eqs. (12)-(14)
9: $GSD \leftarrow GSD \cup \{GSD_g\}$
10: **end for**
11: $\{D_1, D_2\} \leftarrow doKmeans(GSD, ClusterNumber = 2)$
12: **if** the average suspicious degree of D_1 is greater than that of D_2 **then**
13: $AG \leftarrow D_1$
14: **else**
15: $AG \leftarrow D_2$
16: **end if**
17: return *AG*

End

IV. EXPERIMENTAL EVALUATION

A. Experimental datasets and setting

To evaluate HDP-KM, Netflix [13] and Amazon [14] datasets are chosen for experiments. The information of datasets is list in Table II.

TABLE II. DATASETS INFORMATION

Dataset	User amount	Item amount	Rating amount	Rating range
Netflix	2000	3985	215,884	1 to 5, integer
Amazon	5055	17610	53777	1 to 5, integer

In the process of experiment, We first use the group attack models GSAGenl Ran and GSAGenl Avg [9] to generate shilling group profiles that are injected into Netflix dataset. The attack size is set to 10%, and the filler size to 2.5%, under which the attack model can generate additional attackers to ensure the attack effect.

Then we identify the shilling groups in the Amazon dataset and compare the performance of HDP-KM with baseline methods.

B. Evaluation metrics

We use three metrics to evaluate the performance of HDP-KM. The metrics are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$F1-measure=\frac{2\times Precison\times Recall}{Precison+Recall} \quad (17)$$

where *TP* is the number of attack profiles correctly detected, *FN* is the number of attack profiles misjudged, and *FP* is the number of genuine profiles misjudged.

C. Experimental results and analysis

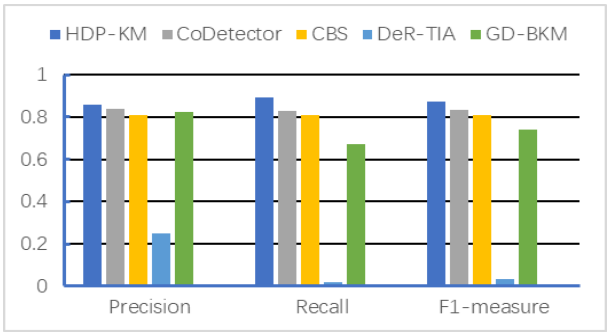
To show the effectiveness of HDP-KM, we compare it with the following four baseline methods:

(1) Catch the Black Sheep (CBS) [7] is a shilling attack detection approach, which ranks users in accordance with their spam probability values. In our experiments, we randomly selected 5% spam users from the labeled attackers in the two datasets as the seed users of CBS.

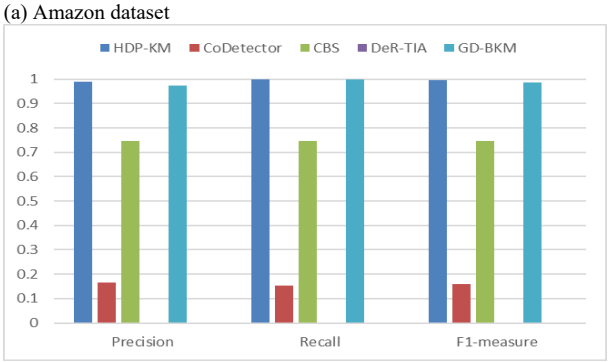
(2) DeR-TIA [10] is a two-step unsupervised method for identifying shilling group profiles. In the experiment, if the highest rating count for an item is greater than 6, then the item is regarded as a target item.

(3) GD-BKM [15] is an unsupervised detection method for shilling groups in recommender systems. This method divided candidate groups based on rating time and identify shilling group using the Bisecting K-Means clustering algorithm.

(4) CoDetector [16] is a supervised method for detecting shilling attacks. During experiments, the dataset is divided into training set and test set according to the ratio of 8 to 2. The latent factors dimension *d* is 10, the negative samples count *k* is 25, the number of iterations is 200, and the learning rate is 0.01.



(a) Amazon dataset



(b) Netflix dataset

Fig. 1. Comparison of detection performance for five methods on the Amazon and Netflix datasets.

Fig. 1 shows the comparison of detection performance for five methods on the two datasets.

As shown in Fig. 1, the precision, recall, and F1-measure of HDP-KM are approximately 0.86, 0.89, and 0.87 on the Amazon dataset. On the Netflix dataset, the three metrics are approximately 0.99, 1, and 0.99. HDP-KM performs more excellently than five baseline methods, indicating that our method can detect shilling groups whether in real or synthetic datasets. The satisfying performance of HDP-KM can be attributed to two reasons. One is that the HDP model can reveal real relationships, which may be hidden by attackers; thus, it can guarantee the effectiveness of the proposed detection features. The other is that HDP-KM uses features of different aspects of shilling groups synthetically. To avoid being detected, attackers camouflage themselves as genuine users; hence, attackers have minimal difference from genuine users in many aspects. Putting these differences together has an amplifying effect, which makes attackers evident.

For CoDetector, the precision, recall, and F1-measure of HDP-KM are approximately 0.84, 0.83, and 0.84 on the Amazon dataset. On the Netflix dataset, the three metrics are approximately 0.17, 0.15, and 0.16. CoDetector performs much better on the Amazon dataset than on the Netflix dataset. This phenomenon is mainly because of the division of the training and test sets. As a supervised method, CoDetector needs sample data to train the classification model. However, the Netflix dataset has 2,000 genuine users and 137 attackers. When the training set is divided in accordance with the method proposed in [16], the numbers of genuine users and attackers are 1,600 and 100 in the training set, respectively; such distribution is imbalanced. Meanwhile, genuine users are less than two times that of attackers. The result also indicates that CoDetector performs poorly when there are few attackers.

CBS is a classic shilling attack detection method, which needs few attackers given as seeds in advance to find the remaining attackers. The detection performance is decided by the seed user amount. In Netflix dataset, 6 seed users are given while 96 seed users are given in the Amazon dataset, so CBS performs better on the latter. The main limitation of CBS is to obtain enough seeds to drive the following process.

DeR-TIA uses *RDMA* and *DegSim*' to detect group shilling profiles. Each metric can determine a set of attackers, and the two sets take an intersection to obtain a final result. In the Netflix dataset, the Pearson similarity of genuine users is greater than that of attackers; thus, users in the set generated by the metric *DegSim*' are all genuine users. DeR-TIA fails to detect shilling groups in the Netflix dataset. Moreover, the two metrics have minimal difference between genuine users and attackers in the Amazon dataset; thus, DeR-TIA does not perform well on the Amazon dataset. The precision, recall, and F1-measure of DeR-TIA are only 0.25, 0.018, and 0.033, respectively, on the Amazon dataset.

GD-BKM performs more excellent on the Netflix dataset than on the Amazon dataset. The reason for this phenomenon is that shilling groups in Amazon dataset have different attack strategies. GD-BKM can only detect on or several kinds of them.

Based on the above analyses, a conclusion that HDP-KM performs better than baseline methods when identifying shilling groups in the two chosen datasets can be drawn.

V. Conclusions

The detection of shilling groups is very important for the recommender system to generate trustworthy recommendations. To detect shilling groups, we have proposed a detection approach based on the hierarchical topic model. We have proposed a hierarchical modeling method to describe user rating behaviors, which models a given dataset to a series of user rating documents and employs the HDP model to find item latent topics. We have proposed a novel method to divide candidate groups, which uses the OPTICS clustering algorithm to summarize user rating patterns and divides users into cliques in accordance with density. We have proposed several indicators to calculate group suspicious degrees and use the k-means clustering algorithm to identify shilling groups. The experimental results on the Netflix and Amazon datasets have demonstrated the effectiveness of the proposed shilling group detection approach.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61772452).

References

[1] M. Si, Q. Li, "Shilling attacks against collaborative recommender systems: a review," Artif. Intell. Rev., vol.53, pp. 291-319, Jan. 2020.

[2] X. F. Su, H. J. Zeng, Z. Chen, "Finding group shilling in recommendation system," Special interest tracks and posters of the 14th Int. Conf. World Wide Web, Japan, 2005, pp. 960–961.

[3] F. Yang, M. Gao, J. Yu, et al., "Detection of shilling attack based on Bayesian model and user embedding," Proc. Int. Conf. Tools Artif. Intell. ICTAI, 2018, pp. 639-646.

[4] 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 Comput. J., vol. 61, no. 7, pp. 949-958, July 2018.

[5] B. Mobasher, R. Burke, R. Bhaumik, et al., "Toward trustworthy recommender systems: an analysis of attack models and algorithm robustness," ACM Trans. Internet. Technol., Vol.7, no.4, 2007.

[6] B. Mehta, W. Nejdl, "Unsupervised strategies for shilling detection and robust collaborative filtering," User Model. User-Adap. Inter., vol. 19, no. 1-2, pp. 65-97, Feb. 2009.

[7] Y. Zhang, Y. Tan, M. Zhang, and et al., "Catch the black sheep: unified framework for shilling attack detection based on fraudulent action propagation," in Proc. 24th Int. Conf. Artif. Intell., Buenos Aires, Argentina, 2015, pp. 2408-2414.

[8] Y. Wang, Z. Wu, J. Cao, Fang. C, "Towards a tricksy group shilling attack model against recommender systems," in Proc. 8th Int. Conf. Adv. Data Min. Appl., Nanjing, China, 2012, pp. 675–688..

[9] M. Ankerst, M. M. Breunig, H.-P. Kriegel, et al., "OPTICS: ordering points to identify the clustering structure," ACM SIGMOD Rec., vol.28, no.2, pp.49-60. 1999.

[10] W. Zhou, Y. S. Koh, J. Wen, S. Alam, G. Dobbie, "Detection of abnormal profiles on group attacks in recommender systems," in Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., Australia, 2014, pp. 955–958.

[11] Y. Wang, Z. Wu, Z. Bu, et al., "Discovering shilling groups in a real e-commerce platform, "Online Inf. Rev., vol. 40, pp.62-78, Feb 2016.

[12] F. Zhang, Y. Qu, Y. Xu, S. Wang, "Graph embedding-based approach for detecting group shilling attacks in collaborative recommender systems," Knowl.-Based Syst., vol.199, July 2020.

[13] Y. Koren, "Collaborative filtering with temporal dynamics," Commun. ACM, vol.53, no.4, pp. 89-97, Apr. 2010.

[14] C. Xu, J. Zhang, K. Chang, C. Long, "Uncovering collusive spammers in Chinese review websites," in Proc. 22nd ACM Int. Conf. Inf. Know. Man., USA, 2013, pp. 979–988.

[15] F. Zhang, S. Wang, "Detecting group shilling attacks in online recommender systems based on Bisecting K-Means clustering," IEEE Trans. Computat. Soc. Syst., vol.7, no.5, pp.1189-1199, Oct. 2020.

[16] T. Dou, J. Yu, Q. Xiong, et al., "Collaborative Shilling Detection Bridging Factorization and User Embedding," Lect. Notes Inst. Comput. Sci. Soc. Informatics Telecommun. Eng., 2018, pp. 459–469.