Detection of Malicious Attacks in Mobile Operator Traffic Mobile Operator Traffic Recommendation System Based on Graph Convolutional Neural Networks

Abstracts: Mobile operators' traffic recommendation system is open to all Internet users, which leads to illegal manipulation of rating data through malicious interference and intentional attacks by unscrupulous users using system design flaws, thus affecting the recommendation results and seriously jeopardizing the security of recommendation services. Most of the existing detection methods are based on manually constructed features extracted from the rating data for TO attack detection, which is difficult to adapt to more complex common access injection attacks, and manually constructed features are time-consuming and lack of differentiation ability, while the scale of the attack behavior is much smaller than that of the normal behavior, which brings unbalanced data problems to the traditional detection methods. Therefore, in this paper, we propose a stacked multilayer graph convolutional neural network to learn the multi-order interaction behavior information between users and items end-to-end to obtain user embeddings and item embeddings, which are used as the attack detection features, and a convolutional neural network is used as the base classifier to realize the deep behavioral feature extraction, and combined with the integrated method to detect the attacks. The experimental results on real datasets show that compared with the popular malicious attack detection methods for recommender systems, the proposed method has a better detection effect on the co-access injection attack and overcomes the problem of unbalanced data to some extent.

Keywords: Attack Detection, Common Access Injection Attack, Recommender Systems, Graph Convolutional Neural Networks, Convolutional Neural Networks, Integration Methods.

1 Introduction

In today's digital era, mobile operators' traffic mobile operators' traffic recommender system plays an extremely important role in enhancing user experience, optimizing resource allocation and increasing business revenue[1-2]. With the wide application and in-depth development of traffic mobile operator traffic recommendation system, its security problems are gradually highlighted, and the threat of malicious attacks is becoming increasingly severe, which has become a key challenge that needs to be solved urgently[3].

Traffic Mobile operators' traffic recommendation systems use complex algorithmic models to accurately push personalized traffic packages and service contents for users based on multi-dimensional data such as users' historical behavioral data, preference information, and network usage patterns[4]. However, this data-driven operation mechanism also makes it a target for malicious attackers. Attackers may launch attacks on the traffic recommender system of mobile operators through a variety of means, such as injecting false user behavior data, tampering with user preference information or interfering with the normal operation logic of the recommendation algorithms[5-7], aiming at misleading the mobile operator's traffic recommender system to make wrong decisions and recommending irrelevant or irrational traffic packages for the subscribers. This will not only seriously damage the user's trust in the mobile operator, leading to a significant drop in user satisfaction, but may also damage the mobile operator's business ecosystem, causing economic losses and affecting market competitiveness. However, due to the openness and system vulnerabilities of mobile operators' traffic recommendation systems [8-10], although these inputs from users enrich the database of mobile operators' traffic recommendation systems, they also make the systems vulnerable to many types of malicious attacks. In order to gain more platform traffic exposure and present their projects to more consumers, malicious users either inject a sufficient number of carefully crafted fake user profiles (e.g., ratings, reviews) into the scoring system by implementing a TO attack and empirically rate the target project higher (promotion attack) or lower (devaluation attack) [11]; or inject fake common access into the system by implementing a common access injection attack [12]; or inject fake common access into the system by implementing a common access injection attack [13]. system to inject false co-accesses to spoof the mobile operator traffic recommender system [8]. Therefore, malicious attacks on mobile operators' traffic recommender systems not only harm consumers' interests and disturb the fairness of the platform, but also shake the confidence of customers and users in the virtual market.

In order to solve the above problem, researchers have proposed many methods such as clustering techniques, statistical techniques, classification techniques, etc [12]. Although these methods are effective in detecting malicious attacks, their limitation is that most of them are based on ratings data and manually constructed well-designed features for TO attack detection, which calculates ratings features for all users, such as Top-N Near Neighbors User Similarity (Degree of Similarity With Top Neighbors, DegSim), Deviation between the number of user ratings and the average number of ratings in the database (Length Variance, LengthVar), etc. The detection performance depends heavily on the quality of the manually constructed features and most of the features are effective only for certain types of TO attacks [11,13]. When facing large-scale real data, the detection performance is still limited due to the high computational cost [14]. Therefore, features constructed in a manual way have the drawbacks of being small in nonlinearity, usually difficult to extract, having low discriminative ability, requiring high knowledge cost, and are insufficient to handle the impact of complex attacks, such as co-access injection attacks [7]. In addition, the amount of malicious attack data in real-world mobile operator traffic recommender systems is much less than the amount of normal data, so the detection of malicious attacks in mobile operator traffic recommender systems can actually be described as an unbalanced classification problem. Traditional detection methods are insensitive to a small number of classes and cannot effectively detect relevant attacks. The introduction of deep learning algorithms can make up for the lack of manually constructed features.

In summary, the paper proposes a malicious attack detection method for mobile operators' traffic recommendation system by combining graph convolutional neural network and integration methods. We believe that user-item interactions contain rich potential behavioral features, so we construct a user-item bipartite graph by analyzing the user click behavior data in the dataset, and use Graph Convolutional Neural Networks (GCN) to display the encoded multilevel interaction information to construct user features and item features, and automatically extract the features to classify the clicks. behavior to classify and deal with co-access injection attacks. The main contributions of this paper are as follows:

(1) The GCN-based embedding representation learning module propagates the embedding display to encode multi-order user item interaction information through graph learning on a two-part graph, forming an embedding vector representation to capture the behavioral features of user item interactions and the implicit interactions embedded in attack behaviors. Its learning features from click behavior through graph learning instead of manually designing features makes up for the shortcomings of manually constructed features that are difficult to extract, require high knowledge cost and have weak distinguishing ability.

(2) Combining the attack behavior classifier based on Convolutional Neural Networks (CNN) with the integrated method of Bootstrap Aggregating (Bagging) algorithm, it realizes the automatic extraction of fine-grained interactive behavioral features between the user and the item, and well solves the unbalanced classification problem.

(3) Experimental results on real datasets show that the proposed algorithm outperforms the popular attack detection algorithms, and can effectively detect co-access injection attacks.

2 Related work

2.1 Malicious Attacks on Mobile Operators' Traffic Recommendation Systems

Malicious users launch different types of attacks on mobile operators' traffic recommendation systems, one of which is the TO attack [11]. Collaborative filtering techniques are able to use information such as a user's historical ratings to find nearest neighbors that are similar to him or her, and generate recommendations for the target user based on information from multiple nearest neighbors. The TO attack takes advantage of this by injecting a sufficient number of well-designed fake user profiles into the rating system to generate recommendations in favor of the malicious user. The form of the rating information for the TO attack is shown in Fig. 1.

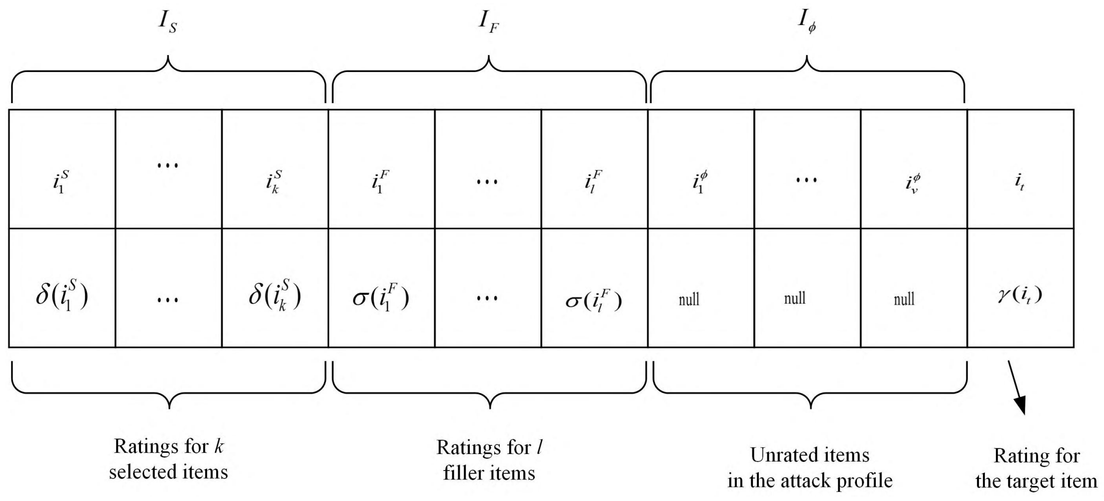


Fig.1 Form of shillin gattack profile

The unscrupulous merchant first registers a large number of fake accounts with the service, and then, each fake account gives a carefully selected subset of items, i.e., a specific rating for the selection rating item IS and the filler rating item IF, such that the ratings of IS and IF mimic the ratings of normal users as much as possible and thus masquerade as normal users, and then attempts to become a close neighbor of multiple users by giving the target item IT a rating that deviates from the true score, thus So that the target item is recommended to more users to achieve the purpose of malicious attack and affect the recommendation results. Therefore, the purpose of TO attack detection is to detect malicious users. Several studies have shown [16-17],that mobile operator traffic mobile operator traffic recommender systems that utilize user item rating data are susceptible to TO attacks. For example, some unscrupulous merchants will commission organizations that provide specialized services to rate their products with the highest score or rate their competitors' products with the lowest score in order to enhance or reduce the frequency of target products being recommended by the system.

Another attack is the common access injection attack [7]. In mobile operator traffic recommender system, if a user accesses two items, then these two items are co-accessed by that user. The key idea of mobile operator traffic recommender systems is that two items that have been frequently co-visited in the past are likely to be co-visited in the future [18-19],Intuitively, two items are more similar if they are frequently co-visited. Specifically, two popular recommendation tasks are I2I recommendation and U2I recommendation.In I2I recommendation, when a user visits item *i*, the system displays the top N recommended items that are similar to item *i*. The top *N* recommended items that are similar to item *i* are displayed. In U2I recommendation, the system recommends the top *N* items to the user by considering the user's access history. Malicious users take advantage of this by injecting a large number of fake co-visits into the system in order to cheat the mobile operator traffic recommendation system.

Malicious attacks on mobile operator traffic recommender systems not only harm the interests of consumers and reduce the shopping experience of users, but also seriously disrupt the fairness of the platform and affect the credibility of the platform and other merchants. Therefore, the detection of malicious attacks is very important to ensure the robustness of mobile operator traffic recommendation system.

2.2 Detection of Malicious Attacks on Mobile Operators' Traffic Recommendation Systems

In recent years, a number of studies on the detection of malicious attacks (especially TO attacks) on mobile operators' traffic recommender systems have emerged. Many previous detection methods have been designed based on the representation of rating behaviors extracted from rating matrices, which can be simply categorized into supervised learning-based, unsupervised learning-based, and semi-supervised learning-based [12].

Supervised learning based attack detection is considered as a classification problem. Initially, Burke et al [13] developed several rating attributes to detect TO attacks using trained K-nearest neighbor classifiers. In recent years scholars have applied deep learning techniques to the field of malicious attack detection for mobile operators' traffic recommender systems.Li et al. proposed the DegreeSAD [20]method, which extracts features from the popularity attributes of items and detects the attackers through a plain Bayesian algorithm.Dou et al. [21]proposed a precedence attack detection model, CoDetector, that jointly decomposes the user-item interaction matrix and user-user co-occurrence matrix, and the learned user latent factors containing network embedding information are used as features to detect attackers.Yang et al [22]proposed a unified detection framework for mining association graphs to identify anomalies in order to deal with co-access injection attacks as much as possible.Wang et al [23]proposed a swarm precedence attack detection method based on graph convolutional neural networks and target item identification Zhang et al [24] proposed GraphRfi, a GCN based user representation learning framework to perform robust recommendation and attack detection in a unified way.GCN accurately captures user preferences and node information, and Random Forest exploits user representation and rating prediction errors well for attack detection. Unsupervised learning based methods do not require a training process, Metha et al [15] proposed the PCAVarSel algorithm, which predicts users' ratings through low-rank matrix decomposition and performs principal component analysis on the low-rank matrix to select a batch of users with the minimum of t principal components to be determined as malicious users.Zhang et al [25]investigated a Hidden Markov Model and Hierarchical Clustering-based approach to reveal attacks. Semi-supervised learning based methods can fully utilize some of the labeled data to achieve attack detection.Wu et al [3] proposed HySAD method based on commonly used statistical features, which can effectively detect TO attacks by using great likelihood to estimate the parameter values and iteratively solving using a similar maximum expectation algorithm.Cao et al [26]proposed SemiSAD method, which firstly trains a simple Bayesian classifier on a small number of labeled users to identify the malicious users. training a plain Bayesian classifier, and then fusing the unlabeled users with EM-λ to improve the initial plain Bayesian classifier to detect attackers.

Literature [22] shows that most of the existing detection methods are based on the detection of TO attacks based on artificially constructed features extracted from rating data, which is rich in information to characterize the basic rating behavior of users. However, the detection performance depends heavily on the representation of the extracted features. In addition, the manually constructed features lack generality, thus limiting the application of these methods in real-world scenarios.

3 Definition of the problem

Suppose that P is used to denote the set of attribute information corresponding to click behaviors in the mobile operator's traffic recommender system, and p_x is a sample of click behavior in P . Set a label for p_x, $y_x \in \{0,1\}$,where 1 represents that the click behavior is an abnormal click behavior generated by the malicious attack of the mobile operator's traffic recommendation system, and 0 represents a normal click behavior, and (p_x,y_x) is a training sample. According to the above definition, n training samples form the training dataset as shown in equation (1):

$$D_{train} = ((p_1, \mathrm{y}_1),(\mathrm{p}_2, \mathrm{y}_2),\cdots(p_n,y_n)) \qquad (1)$$

In this paper, we use the dataset Dtrain to construct the model *f*. We define a loss function to optimize the model,and determine whether the current clicking behavior $p_{x'}$, is the label $y_{x'}$ of the attacking behavior.

4 Malicious Attack Detection in Mobile Operator Traffic Recommendation Systems Combining Graph Convolutional Neural Networks and Integrated Methods

In this chapter, we will introduce the malicious attack detection method for mobile operator's traffic recommendation system combined with graph convolutional neural network and integration method proposed in this paper, and its framework diagram is shown in Fig. 2. It adopts GCN, CNN-based attack behavior classifier and integrated method Bagging as the building blocks, because GCN can make full use of the node information and local structure information in the user-item interaction graph, and can learn both node features and node-to-node correlation relationships, capture the deep interaction behavior information and user preferences as well as implicit interactions embedded in the attack behaviors, and CNN can automatically perform feature extraction, and can be used in the classification task. CNN is capable of automatic feature extraction and has achieved excellent performance in classification tasks. Bagging algorithms are often combined with statistical classification and regression algorithms to improve the accuracy and stability of the algorithm, and here we combine it with deep learning models.

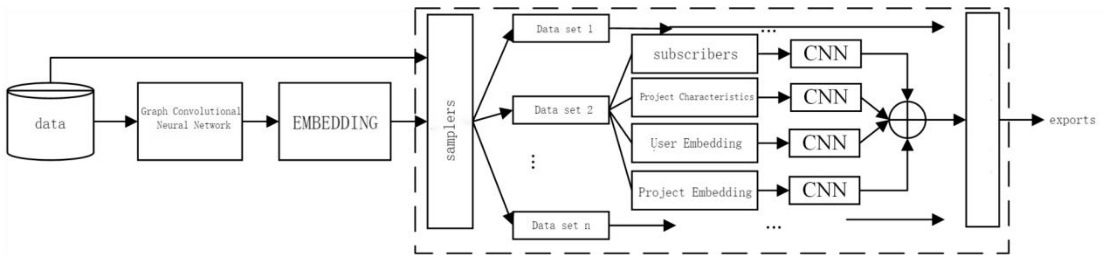


Fig.2 Overall architecture of the proposed algorithm

4.1 GCN-based embedded representation learning

We model the given dataset as a graph by considering users and items as nodes and interactions between users and items as edges, respectively, which results in a user-item bipartite graph (interaction graph), as shown in Fig. 3(a). Fig.3(b) gives a higher-order connectivity representation extended from u_2, which contains rich semantics of interaction information, such as behavioral features. For example, from the second-order interaction, both u_2 and u_3 have interacted with i4, so u_2 and u_3 have behavioral similarity; from the third-order interaction, compared to i_1,u_2 may be more interested in i_3 because there are two paths to i_3 and only one path to i_1. It can be seen that the two-part graph contains rich behavioral information and user preference information.

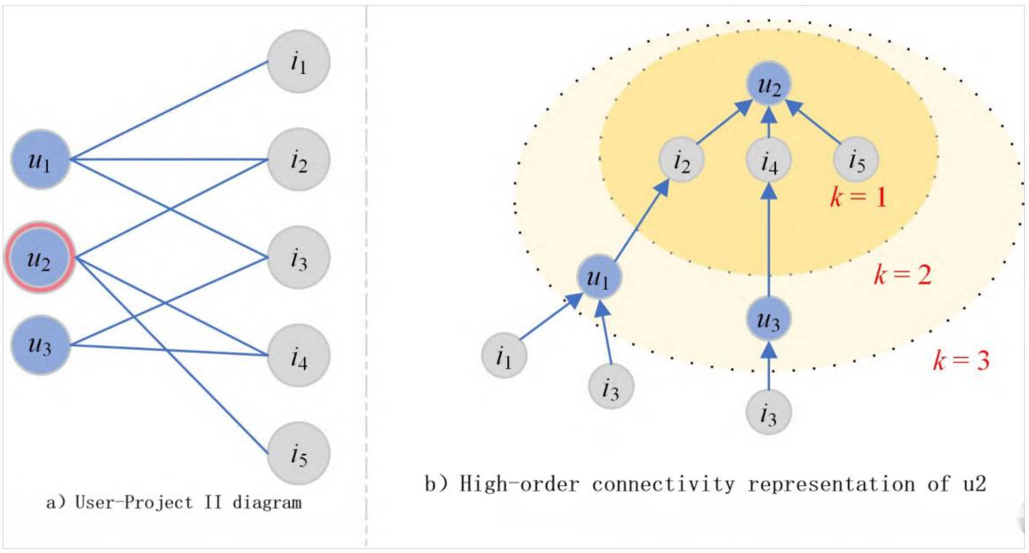


Fig.3 Illustration of user-item bipartite graph and high-orderconnectivity of u2

In order to capture the above higher-order interaction information, inspired by graph convolutional neural network[23]and graph-based collaborative filtering algorithm[30-32], we utilize the interaction information to generate user and item embedded representations on two-part graphs, and realize the aggregation of higher-order interaction information by stacking multilayers of GCNs to produce the final user and item embedded representations, and its architecture is shown in Fig. 4.

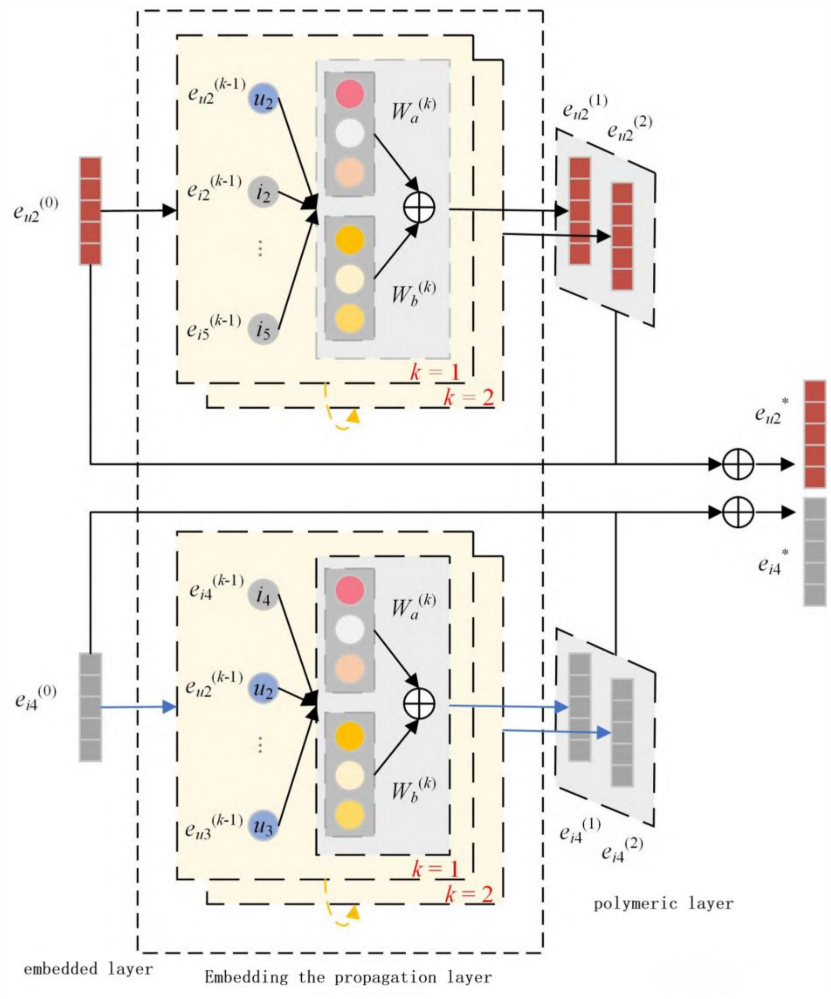


Fig.4 GCN-based framework for embedding representations learning

4.1.1 Embedding layer

This section mainly initializes the embedding vector representations of users and items. We describe user u and item i with embedding vectors $e_u \in R^d$ and $e_i \in R^d$, respectively, where d denotes the embedding size. The embedding lookup table is shown in Eq. (2), N and M denote the number of users and items respectively,each user and item can be mapped into an initialized embedding representation by this embedding lookup table.

$$E = [e_{u1}^{(0)}, \dots e_{uN}^{(0)}, \ \ e_{i1}^{(0)}, \dots, e_{iM}^{(0)}] \qquad (2)$$

4.1.2 Embedding the propagation layer

Stacked multi-layer GCN captures user-item multi-order interaction behavior information along the interaction graph structure, and refines user and item embedding.

Message construction: Stacking k embedded propagation layers, a user or item is able to receive messages propagated from its k-hop neighbors.At the k-th hop, the message delivered from i to u is defined as shown in equation (3):

$$m_{u\leftarrow i}^{(k)} = pui(W_a^{(k)}\mathrm{e}_i^{(k\text{-}1)} + W_b^{(k)}(\mathrm{e}_i^{(k\text{-}1)} \otimes \mathrm{e}_{\mathrm{u}}^{(k\text{-}1)})) \qquad (3)$$

Considering the characteristics of the node itself,we add the self-connection of the node and the self-connection message of u is shown in equation (4) as follows：

$$m_{u\leftarrow u}^{(k)} = W_a^{(k)}\mathrm{e}_u^{(k\text{-}1)} \qquad (4)$$

Where, $W_a^{(k)}$, $W_b^{(k)}$ are the kth layer trainable weight matrices, $\mathrm{e}_i^{(k\text{-}1)}$ and $\mathrm{e}_u^{(k\text{-}1)}$ are the item and user embedded representations generated from the previous message propagation step, storing information about their (k-1) hop neighbors. The interaction between i and u is encoded by the element product $\otimes$. pui is set to Laplace's paradigm,as shown in equation (5), where N_u and N_i denote the *k-th*-order neighbors of user u and item i. The introduction of *pui* is able to normalize the adjacency matrix and prevent the bias generated by the large information values after multilayer convolutional aggregation.

$$pui = \frac{1}{\sqrt{|N_u||N_i|}} \qquad (5)$$

Message aggregation: aggregates the messages propagated from the k-th order neighbors of u to refine the representation of u. The aggregation function is defined as shown in equation (6):

$$e_u^{(k)} = \mathrm{Leaky}ReLU(m_{u\leftarrow u}^{(k)} + \sum_{i\in N_\mathrm{u}} m_{u\leftarrow i}^{(k)}) \qquad (6)$$

We adopt LeakyReLU as the activation function to increase the nonlinearity of the model, and realize the embedding propagation of the higher order interaction behavior information through the stacking of *k* layers of GCN, and obtain the kth order user representation eu(k). Similarly,we can get the kth order user representation ei(k) as shown in equation (7):

$$e_u^{(k)} = LeakyReLU(\mathrm{m}_{u\leftarrow u}^{(k)} + \sum_{i\in N_\mathrm{u}} \mathrm{m}_{u\leftarrow i}^{(k)}) \qquad (7)$$

After *k*-layer propagation, we can obtain the user representation and item representation at each layer, i.e., $\{e_u^{(0)},\ldots,e_u^{(k)}\}and\{e_i^{(0)},\ldots,e_i^{(k)}\}$.

4.1.3 Aggregation layer

Since the representations of different layers emphasize different interaction information, we stitch together the representations of all layers to form the end-user embedded representation e_u^* and item embedded representation e_u^* that incorporate multi-order fine-grained interaction information , as shown in Eqs. (8) and (9).

$$e_u^* = e_u^{(0)} \| \cdots \| e_u^{(k)} \qquad (8)$$

$$e_i^* = e_i^{(0)} \| \cdots \| e_i^{(k)} \qquad (9)$$

4.2 CNN-based Attack Behavior Classifier

In this section, a CNN-based network is designed to detect the attack behaviors and its framework diagram is shown in Fig. 5. The classifier is capable of automatic feature extraction of user embeddings and item embeddings as well as user features and item features that incorporate higher-order interaction behavior information, capturing local context information and further optimizing the vector representation to extract deeper behavioral features.

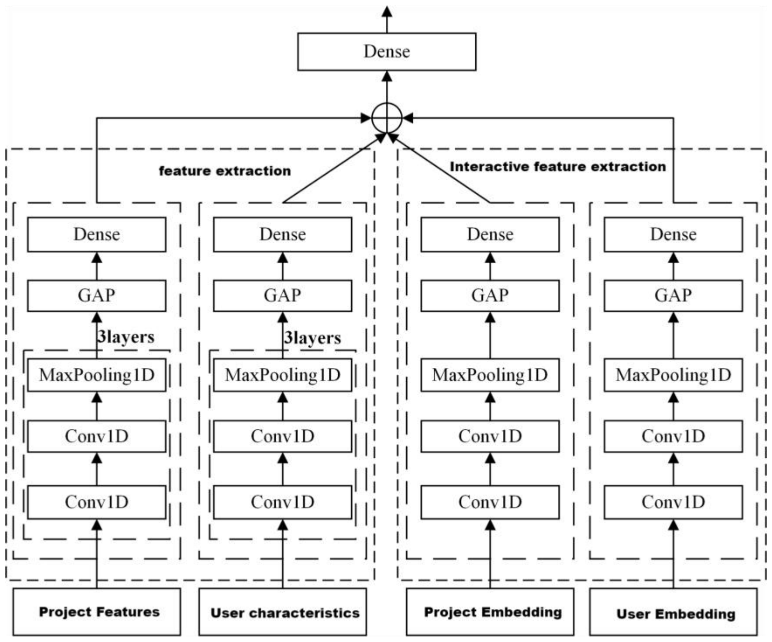


Fig.5 Frame work of CNN-based classifier form alicious attacks

The network is a parallel network, respectively, through the convolution and pooling operations to capture fine-grained interactive behavioral features, and pooling results for global average pooling to prevent overfitting, through the fully connected layer to get the user, project high-level features, and then the high-level feature vectors are spliced, through the fully connected layer to output the predicted value of the results by using the Sigmoid as an activation function, which is shown in equation (10) Shown:

$$S(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{10}$$

4.3 Model fusion based on the integration method Bagging

The problem of detecting malicious attacks in mobile operators' traffic recommender systems is an unbalanced classification problem because attacks in the real world account for only a small fraction of the total number of attacks. Integration methods have been proved to be effective in facing the unbalanced classification problem[33].The integration algorithm Bag-ging is often combined with other classification and regression algorithms to improve its accuracy and stability and reduce the variance of the results. Here we combine it with a deep learning model to solve the unbalanced classification problem.

The flow of the method is shown in Fig. 4. Here we take no-putback sampling for the normal samples of the original training set, divide the normal samples into *N* subsets, and the number of samples in each subset is the same as the number of attack samples. By repeatedly combining the minority attack samples and the same number of majority normal samples to obtain a number of new training sets, such as dataset 1 and dataset 2 in Fig. 6, a new training set corresponds to a CNN-based attack behavior classifier, because of the different distribution of samples in the training set, the classifier obtained from the training has a variety of classifiers, such as the classifiers in Fig. 6, classifiers C_1, classifiers C_2 and so on, and finally take a combination of strategy classifier weights soft voting algorithm to integrate the predicted values of each classifier.

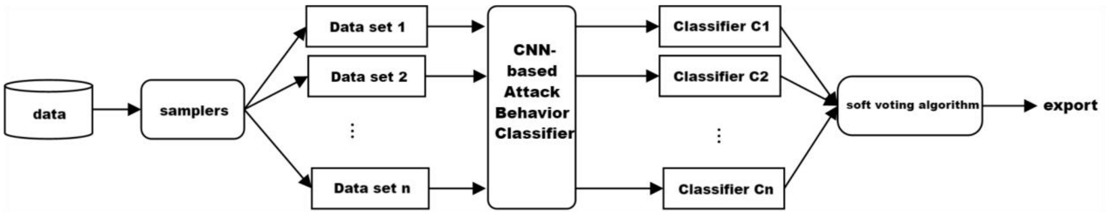


Fig.6 Ensemble architecture

Classifier weights soft voting algorithm:A specific weight w_i is assigned to each classifier C_i,T is the number of classifiers, C_i^j represents the probability of belonging to category j predicted by the ith classifier， $H^j(x)$ denotes the probability that x belongs to category j. The final output is shown in Eq. (11),Eq. (12) is the predicted label $\hat{y}$ for the click behavior.

$$H_j(x) = \frac{1}{T}\sum_{i=1}^{T} w_i C_i^j(x) \qquad (11)$$

$$\hat{y} = argmax[H^0(x), H^1(x)] \qquad (12)$$

5 Analysis of Experimental Results

5.1 Data sets

The dataset in this paper is derived from a real dataset from AliCloud Tianchi Labs, which is a dataset in which a malicious user implements a co-access injection attack by collaboratively clicking on targeted items and pop-up items. A piece of data represents the attribute information corresponding to one click behavior. The original dataset includes attribute information such as user id, item id, user characteristics, item characteristics, and labels corresponding to each piece of data.Table 1 lists the basic statistical information of the dataset.

Table 1 Basic statistics of datasets

data set	Positive sample size	Negative sample size	number of users	Number of projects
training set	450000	50000	357133	210369
test set	45000	5000	45899	37964

We randomly selected 80% of the 450,000 data from the original training set to constitute the training set, and used the remaining as the validation set, and the test set was the 50,000 data provided by the dataset itself.

5.2 Baseline methodology

This section compares the performance of the algorithm proposed in this paper with the following five benchmark TO attack detection algorithms.PCAVarSel [15]: an unsupervised learning approach that uses principal component analysis to compute the principal component coefficient scores of the profiles to detect attacks.

SemiSAD [26]: a semi-supervised learning approach,to detect attackers by improving the plain Bayesian classifier.

DegreeSAD [20]: Supervised learning approach that utilizes item popularity attributes to detect attackers via a plain Bayesian algorithm.

CoDetector [21]: a supervised learning approach that utilizes user latent factors containing network embedding information as features to detect attackers.

GraphRfi[24]: a supervised learning approach that performs robust recommendation and attack detection in a unified way using a GCN-based user representation learning framework.

5.3 Evaluation indicators

In order to evaluate the performance of the method in this paper, we used four common indexes, namely, ACC (Accuracy), P (Precision), R (Recall) and $F1$ (F-Measure), with the following formulas.

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \qquad (13)$$

$$P = \frac{TP}{TP + FP} \qquad (14)$$

$$R = \frac{TP}{TP + FN} \qquad (15)$$

$$F1 = \frac{2 \times P \times R}{P + R} \qquad (16)$$

where ,*TP* denotes the proportion of true malicious attack behaviors in the set of behaviors judged as malicious attack behaviors, *FP* denotes the proportion of true normal behaviors in the set of behaviors judged as malicious attack behaviors, *TN* denotes the proportion of true normal behaviors in the set of behaviors judged as normal behaviors, and *FN* denotes the proportion of true malicious behaviors in the set of behaviors judged as normal behaviors.

5.4 Parameterization

In the GCN-based embedding representation learning part,all the embedding sizes are fixed to 64,we use Xavier initializer to initialize the user and commodity embedding representation module model parameters,and set the embedding propagation depth k to 2,the batch size to 1024 and epoch to 400.In the CNN-based attack behavior classifier part,the weights of each of the soft-voting strategies are set to the accuracy of that classifier ACC,and the Adam optimizer is used during the experiment,the batch size is set to 256,epoch to 60,and the learning rate is set to 0. In the CNN-based attack behavior classifier part,the soft voting strategy in each of the classifier's weight is set to that classifier's accuracy ACC,and the Adam optimizer is used in the experimental process,the batch size is set to 256,epoch is set to 60,and the learning rate is set to 0.001.

5.5 Experimental results

In order to verify the effectiveness of the method in this paper, we conduct experiments on real datasets, and the experimental results are shown in Table 2.

Table 2 Comparison of multiple methods

means	aggression			normal behavior		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
PCAVarSel	0.0930	0.0994	0.0961	0.9115	0.9054	0.9085
SemiSAD	0.0712	0.9873	0.1328	0.4167	0.0007	0.0014
DegreeSAD	0.5904	0.3410	0.4323	0.9505	0.9816	0.9658
CoDetector	0.5423	0.4898	0.5147	0.9614	0.9685	0.9650
GraphRfi	0.9966	0.4008	0.5717	0.6771	0.9989	0.8071
Ours	0.7941	0.7158	0.7529	0.9688	0.9794	0.9740

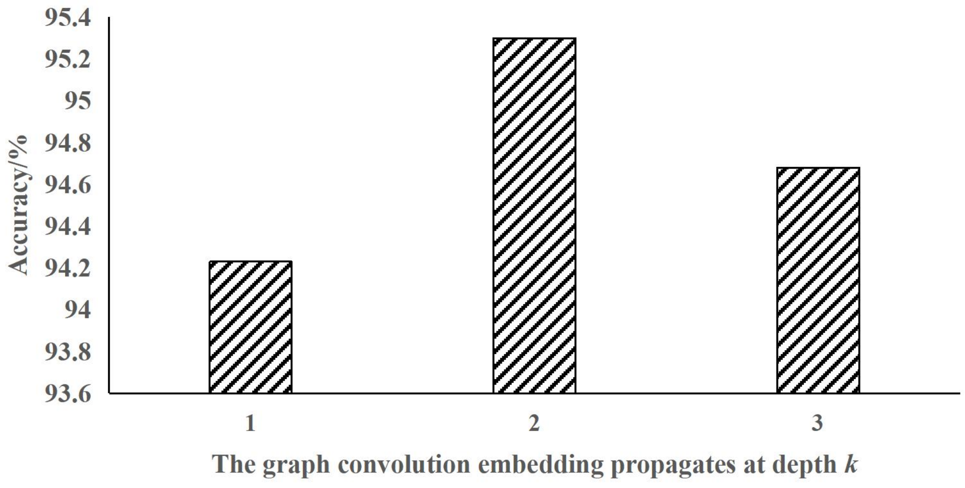
The comparison shows that:PCAVarSel algorithm is the least effective,it can detect the attacker only when the same unrated items among malicious users are more than normal users.Common access injection attack does not depend on scoring,so the condition does not hold.SemiSAD depends on user profiles and manually constructed features.Due to data imbalance problem,there are fewer profiles of malicious users,and its manually constructed features depend on ratings, which is not effective.DegreeSAD algorithm is significantly better than PCA, but the detection performance of this algorithm largely depends on the representation quality of manually constructed statistical features. In contrast, CoDetector and GraphRfi algorithms are better than the above algorithms. CoDetector can significantly improve the performance by fusing the rating and structural feature information, and verifies the validity of the structural features, however, it still relies on the rating data.

GraphRfi's attack detection module combines the user representations obtained from the GCN and the prediction error using the random forest. is carried out, and its effect verifies the effectiveness of user representation learning, but because the attack data only accounts for a small part of the dataset, there is a serious unbalanced classification problem, resulting in a larger Precision and smaller Recall and FGMeasure of the attack behaviors, and the overall performance is not good.

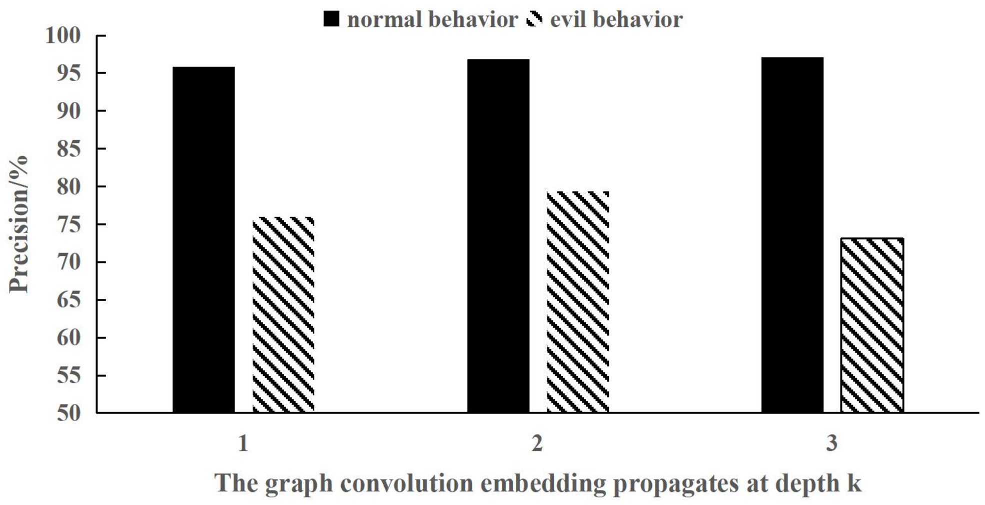
Overall, this paper's method outperforms SemiSAD and DegreeSAD methods based on manually constructed features, which verifies that the automatic extraction of multilayer features fusing interaction behaviors and structural information from user behaviors through graph learning can make up for the shortcomings of manually constructed features that are weak in differentiation ability, difficult to extract, and require high knowledge cost. In addition, this paper's algorithm outperforms other benchmark methods in detecting co-access injection attacks, which indicates that popular tor attack detection methods are not applicable to the detection of co-access injection attacks, and verifies the superior performance of this paper's method in co-access injection attack scenarios.

5.6 Hyperparametric analysis

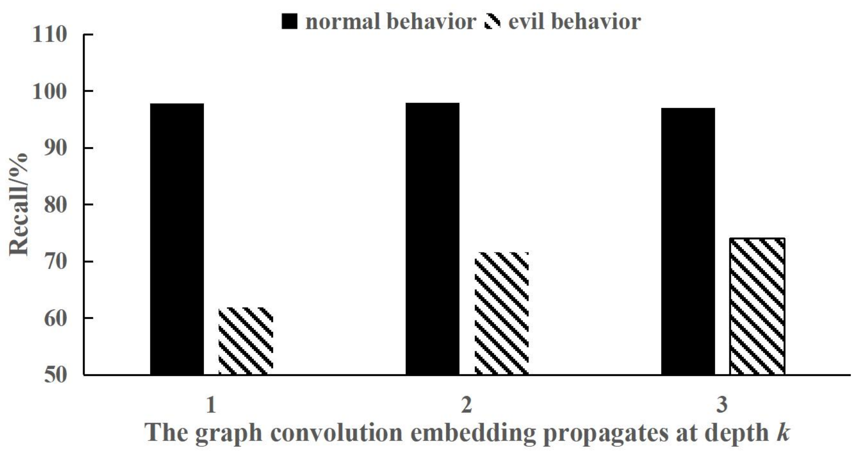
The key hyperparameter of the method in this paper is the graph convolution embedding propagation depth k. In order to understand the effect of this hyperparameter on the experimental results, this paper conducts a tuning experiment. Keeping the other parameters fixed and setting the embedding propagation depth k as 1,2,3 respectively, the experimental results are shown in Fig. 7.



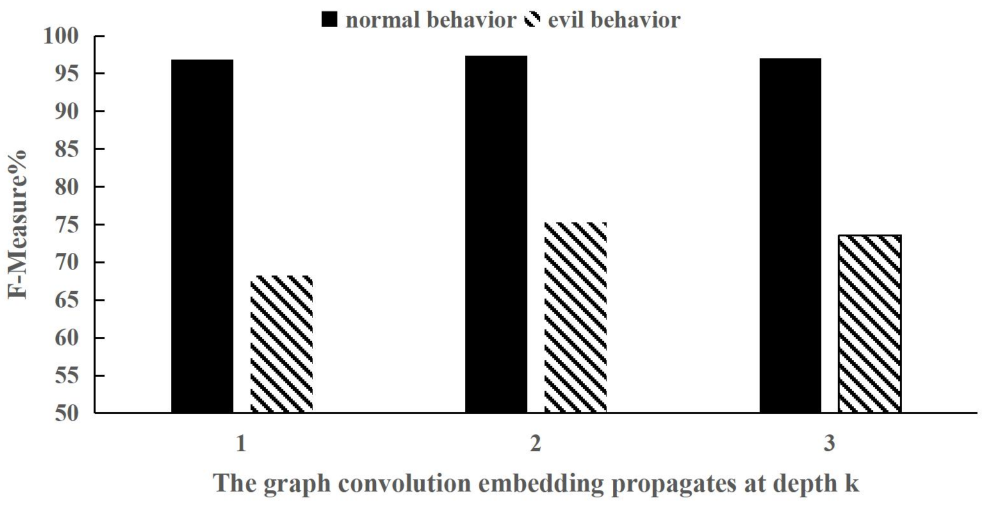
（a）Accuracy



（b）Precision



（c）Recall



（d）F-Measure

Figure 7 Effect of embedding propagation depth *k*

Figure 7 shows the effect of different embedding propagation depth k on different classification metrics. It is obvious that increasing the embedding propagation depth can greatly improve the model performance, the embedding propagation depth of *k* is 2 improves in all aspects compared to *k* is 1. The embedding propagation depth of *k* is 3 improves in FGMeasure and Accuracy compared to *k* is 1, which suggests that the information about the interactive behavior of the user and the goods is carried by the second-order or third-order embedding propagation. However, we found a certain degree of degradation in the evaluation indexes for an embedding propagation depth of 3 compared to *k* of 2. We believe that this is caused by the fact that too deep an architecture may introduce noise in the representation learning leading to overfitting. Overall, the two embedding propagation layers are sufficient to capture information about the interaction behaviors between user items and the invisible interactions embedded in the attack behaviors.

5.7 Elimination analysis

In order to verify the effect of different modules of the method in this paper on the detection performance, we designed four variants of the model to eliminate and analyze the algorithm.

(1) Removing the GCN-based embedding representation learning module (M1). Remove the user embeddings and item embeddings obtained from the GCN-based embedding representation learning module that incorporate higher-order interaction behavior information, and directly input the user features and item features into the CNN-based attack behavior classifier.

(2) Remove the user and item feature vectors (M2). Directly input the user embedding and item embedding into the CNN-based attack behavior classifier.

(3) Remove the CNN-based attack behavior classifier (M3) and directly biclassify the user-item features and their embedding representations without further feature extraction.

(4) Remove the model fusion module (M4) based on the integration method Bagging. Instead of sampling and model fusion on the dataset, the CNN-based attack behavior classifier is directly used for binary classification.

Table 3 AblationStudy

means	Accuracy	aggression			normal behavior		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
M1	0.9251	0.6178	0.6584	0.6374	0.9618	0.9547	0.9582
M2	0.8847	0.4316	0.4824	0.4556	0.9417	0.9294	0.9355
M3	0.9929	0.9823	0.2336	0.3774	0.9215	0.9995	0.9589
M4	0.9240	0.9878	0.2426	0.3895	0.9224	0.9997	0.9595
Ours	0.9530	0.7941	0.7158	0.7529	0.9688	0.9794	0.9740

Table 3 lists the experimental results of the elimination analysis, and we can get three conclusions: (1) Removing any part, the classification evaluation index FGMeasure of the algorithm shows a certain degree of decline, which indicates the effectiveness of each element of the algorithm. (2) Removing the embedding representation learning module, the classification index decreases significantly, reflecting the importance of the multi-order interaction behavior information of user items for attack behavior detection, and reflecting the superior performance of extracting behavioral features through automatic learning of GCN embedding representation. (3) Removing the model fusion module, due to the existence of a serious unbalanced classification problem, resulting in a larger Precision but smaller Recall and FGMeasure of the attack behaviors, the overall performance is poor, reflecting that the model fusion is able to solve the unbalanced classification problem well, proving its effectiveness.

6 Conclusion

With the wide application of mobile operators' traffic recommendation system, accurately identifying the malicious attacks of mobile operators' traffic recommendation system is an important issue related to the credibility of mobile operators' traffic recommendation system. Considering that most of the current detection methods are based on manually constructed features and statistical methods, this paper proposes a malicious attack detection method for mobile operators' traffic recommender system by combining graph convolutional neural network (GCN) and integration methods to address the inadequacy of manually constructed features. The method uses GCN to learn the multi-order interaction information of user items, automatically extracts behavioral features and user preferences and integrates them into user embedding and item embedding, combines the features of user items themselves, and detects the malicious attacks of mobile operators' traffic recommender system by using CNN-based attack behavior classifier and Bagging algorithm. Experiments on real datasets show that the method in this paper outperforms the popular algorithms for detecting malicious attacks on mobile operators' traffic recommendation systems, and has a better detection effect on co-access injection attacks. In our future work, we plan to use the probability that a click behavior is recognized as a normal behavior as a weight to determine the contribution of the click behavior in the mobile operator traffic recommender system, and to construct a robust mobile operator traffic recommender system that can produce stable recommendations even in the presence of a malicious attack on the mobile operator traffic recommender system, which is of great practical significance.

Reference

[1] Xin L ,MengChu Z ,Yunni X , et al.Generating Highly Accurate Predictions for Missing QoS Data via Aggregating Nonnegative Latent Factor Models.[J].IEEE transactions on neural networks and learning systems,2016,27(3):524-37.

[2] Xin L ,MengChu Z ,Shuai L , et al.An Inherently Nonnegative Latent Factor Model for High-Dimensional and Sparse Matrices from Industrial Applications[J].IEEE Transactions on Industrial Informatics,2018,14(5):2011-2022.

[3] WU Z,WU J,CAO J,et al.HySAD:A semi-supervised hybrid shilling attack detector for trustworthy product recommendation[C]//Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.2012:985-993.

[4] Grating evolution[C]//Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.2014:841-850.

[5] GÜNNEMANN N,GÜNNEMANN S,FALOUTSOS C.Robustmultivariate autoregression for anomaly detection in dynamic product ratings[C]//Proceedings of the 23rd International Conference on World Wide Web.2014:361-372.

[6] LU J,WU D,MAO M,et al.Recommender system application developments:a survey[J].Decision Support Systems,2015,74:12-32.

[7] YANG G,GONG N Z,CAI Y.Fake Co-visitation Injection Attacks to

Recommender Systems[C]//NDSS.2017.

[8] XING X,MENG W,DOOZAN D,et al.Take This Personally:Pollution Attacks on Personalized Services[C]//USENIX Security Symposium.2013:671-686.

[9] CALANDRINO J A,KILZER A,NARAYANAN A,et al.You might also like:Privacy risks of collaborative filtering[C]∥2011 IEEE Symposium on Security and Privacy.IEEE,2011:231-246.

[10] FANG M,YANG G,GONG N Z,et al.Poisoning attacks tograph-based recommender systems[C]//Proceedings of the 34th Annual Computer Security Applications Conference.2018:381-392.

[11] GUNES I,KALELI C,BILGE A,et al.Shilling attacks against recommender systems:A comprehensive survey[J].Artificial Intelligence Review,2014,42(4):767-799.

[12] SI M,LI Q.Shilling attacks against collaborative recommender systems:a review[J].Artificial Intelligence Review,2020,53:291-319.

[13] BURKE R,MOBASHER B,WILLIAMS C,et al.Classification features for attack detection in collaborative recommender systems[C]//Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.2006:542-547.

[14] YANG Z,CAI Z,GUAN X.Estimating user behavior towarddetecting anomalous ratings in rating systems[J].Knowledge-Based Systems,2016,111:144-158.

[15] MEHTA B,NEJDL W.Unsupervised strategies for shilling detection and robust collaborative filtering[J].User Modeling and User-Adapted Interaction,2009,19:65-97.

[16]O′ MAHONY M,HURLEY N,KUSHMERICK N,et al.Colla-borative recommendation:A robustness analysis[J].ACM Transactions on Internet Technology(TOIT),2004,4(4):344-377.
[17]MOBASHER B,BURKE R,BHAUMIK R,et al.Toward trustworthy recommender systems:An analysis of attack models and algorithm robustness[J].ACM Transactions on Internet Technology(TOIT),2007,7(4):23.
[18]LIU B,KONG D,CEN L,et al.Personalized mobile app recommendation:Reconciling app functionality and user privacy pre-ference[C]//Proceedings of the Eighth ACM International Conference on Web Search and Data Mining.2015:315-324.
[19]KOREN Y,BELL R,VOLINSKY C.Matrix factorization techniques for recommender systems[J].Computer,2009,42(8):30-37.
[20]LI W,GAO M,LI H,et al.Shilling attack detection in recommender systems via selecting patterns analysis[J].IEICE Transactions on Information and Systems,2016,99(10):2600-2611.
[21]DOU T,YU J,XIONG Q,et al.Collaborative shilling detection bridging factorization and user embedding[C]//Collaborative Computing:Networking,Applications and Worksharing.Sprin-ger International Publishing,2018:459-469.
[22]YANG Z,SUN Q,ZHANG Y,et al.Inference of suspicious co-visitation and co-rating behaviors and abnormality forensics for recommender systems[J].IEEE Transactions on Information Forensics and Security,2020,15:2766-2781.

[23]WANG S,ZHANG P,WANG H,et al.Detecting shilling groups in online recommender systems based on graph convolutional network[J].Information Processing & Management,2022,59(5):103031.

[24]ZHANG S,YIN H,CHEN T,et al.Gcn-based user representation learning for unifying robust recommendation and fraudster detection[C]//Proceedings of the 43rd international ACM SIGIR Conference on Research and Development in Information Retrieval.2020:689-698.

[25]ZHANG F,ZHANG Z,ZHANG P,et al.UD-HMM:An unsupervised method for shilling attack detection based on hidden Markov model and hierarchical clustering[J].Knowledge-Based Systems,2018,148:146-166.

[26]CAO J,WU Z,MAO B,et al.Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system[J].World Wide Web,2013,16:729-748.

[27]DEFFERRARD M,BRESSON X,VANDERGHEYNST P.Convolutional neural networks on graphs with fast localized spectral filtering[J/OL].Advances in Neural Information Processing Systems,2016,29.https://proceedings.neurips.cc/paper_files/paper/2016.

[28]TAN M,LE Q.Efficientnet:Rethinking model scaling for convolutional neural networks[C]//International Conference on Machine Learning.PMLR,2019:6105-6114.

[29]BREIMAN L.Bagging predictors[J].Machine Learning,1996,24:123-140.

[30]WANG X,HE X,WANG M,et al.Neural graph collaborative filtering[C]//Proceedings of the 42nd international ACM SIGIR Conference on

Research and Development in Information Retrieval.2019:165-174.

[31]HE X,DENG K,WANG X,et al.Lightgcn:Simplifying andpowering graph convolution network for recommendation[C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.2020:639-648.

[32]PENG S,SUGIYAMA K,MINE T.Less is More:Reweighting Important Spectral Graph Features for Recommendation[C]//Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.2022:1273-1282.

[33]LIU X Y,WU J,ZHOU Z H.Exploratory undersampling forclass-imbalance learning[J].IEEE Transactions on Systems,Man,and Cybernetics,Part B(Cybernetics),2008,39(2):539-550.