Crowdturfing Detection in Online Review System: A Graph-Based Modeling

Qilong Feng, Yue Zhang, and Li Kuang(✉)

School of Computer Science and Engineering, Central South University, Changsha 410075, HN, China
kuangli@csu.edu.cn

Abstract. With the widespread popularity of online reviews and crowdsourcing, people may publish fake comments on online review system and get paid for crowdsourcing tasks. In order to identify these reviewers, machine learning methods are commonly used in traditional strategies and it is difficult to guarantee the accuracy of detection. In this work, we adopt a modeling method based on the graph structure and propose a novel aggregation method called CrowdDet. Therefore, two clear diagrams of Reviewer-to-Product and Co-Reviewer are constructed. Specifically, we first extract the node features and structure information in the graph, gaining the reviewers' features and neighborhood relations features. Secondly, we use an elaborate attention-based mechanism to aggregate the factors of reviewers in Review-space and Sociality-space, which comprehensively combines the representation of the reviewer factors from multiple dimensions. Thirdly, we get the classification results and optimize the original loss function by Focal loss to alleviate the impact of class imbalance. In the experiment, we verify the proposed scheme on a real dataset and compare it with other methods. The results show that our scheme has a significant effect under the real dataset, with a recall rate of 0.85+. Our research also provides a relevant foundation for resisting the malicious behavior from crowdsourcing.

Keywords: Crowdturfing detection · Graph-based modeling · Class imbalance solution · Online review system

1 Introduction

For e-commerce platforms or review sites such as Amazon, Yelp, or Taobao, user reviews can affect people's consumption choices and play an important role in people's decisions for purchasing. However, with the continuous development of crowdsourcing platforms such as Amazon Mechanical Turk and CrowdFlower, a large number of malicious reviews can be organized by merchants. Sellers post review tasks on the crowdsourcing platform, hiring crowdsourcing workers to promote products, or slander competitors [1]. Spam workers from crowdsourcing platforms are called crowdturfing. Crowdturfing is a word combines the "crowdsourcing" and "astroturfing". Unlike the original fake reviewers, crowdturfing can perform tasks alone, so it lacks concentrated

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 407, pp. 352–369, 2021.
https://doi.org/10.1007/978-3-030-92638-0_21

behavior features and is more difficult to be detected. The fakers are divided into the following three categories [2] according to the different motivations of reviewers:

- Camouflage. They add links to ordinary reviews by adding special symbols, which creates confusion among normal reviews and pretends to be normal reviewers in this way.
- Spammer. They post irrelevant comments (e.g., advertisements, commodity promotion) in the review system, or even carry out illegal activities (e.g., selling contraband, sending sensitive words), and maliciously increase the popularity of the product through the above methods.
- Crowdturfing. The review task is published by the organizer on the crowdsourcing platforms (e.g., Rapidworkers, Amazon Mechanical Turk), and the crowdsourcing workers perform the task by reviewing for a fee.

Such behavior can easily mislead consumers' true judgments on product quality and cause unfair competition among merchants. On the other hand, it will reduce the credibility of product review content and violate the original intention of the crowdsourcing strategy. We found that there are commonalities between crowdturfings, so how to identify them in an automated way is the goal of this research. In order to detect fake reviews or scoring behavior like this, many explorations have been carried out by researchers. The feature exploited in detection can be divided into review content features and reviewer behavior features.

1. Review Content Features. It refers to the description of the features of the review content, including both the analysis of the review text and the comment rating, such as review length, the proportion of special symbols, similarity of comments, review sentiment classification, etc.
2. Reviewer Behavior Features. It refers to the statistical features of the reviewer's comments, including the reviewer's attributes, relationships, and behavior, such as the number of reviews, product categories reviewed, the number of devices, or whether they are verified, etc.

Nitin and Liu [3] were the first to propose fake reviews or reviewer detection tasks. They regarded repeated or similar reviews as positive case, and the rest as negative case, and a logistic regression classifier is trained to distinguish fake reviewers from real reviewers. After that, the heterogeneous graph was used to model the relation among reviewers, reviews, and products, so as to explore the relationship among the credibility of the reviewer, the credibility of the review, and the credibility of the product [4, 5]. Recently, some researchers have constructed a homogeneous graph model based on the relationship among reviewers and obtained the possible labels of the current node [2, 6, 7] based on the state of neighboring nodes.

Although the association among reviewers, products and product reviews is considered in the above methods, the analysis of review text is limited to the statistical features of the review text, so it has great limitations in semantic understanding. In addition, taking into account that the data number of different classes is uneven, that is, the long tail problem still exists in reality. Oversampling, undersampling [19], and cost-sensitive learning [20, 21] are widely used to solve it. However, the solution of data imbalance in existing methods have not been applied in crowdsourced fake reviewer detection.

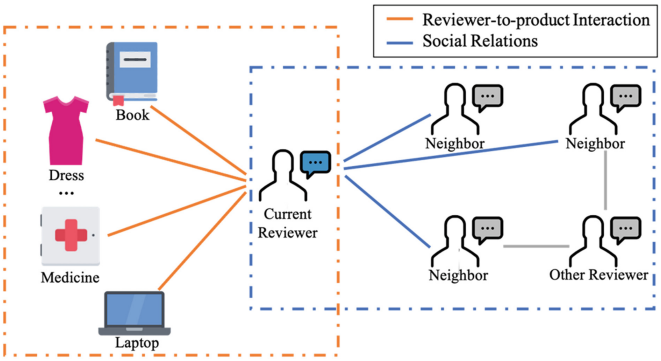

Fig. 1. Reviewer-to-product graph and co-reviewer graph.

In order to solve the above problems, inspired by the recommendation system method with graph modeling [22], we attempted to extract the common features of the crowdsourced fake reviewers from a small amount of category. Therefore, we modeled the current reviewer (see Fig. 1) and extracted the review content features and behavior features from the comment records. We assume that a user has only one comment record for a product, so there will be no conflicts and inconsistencies when modeling.

The representation of the reviewer in the review-space is obtained based on the attention aggregation, and the representation in the sociality-space is the aggregation of its neighbor nodes. Finally, based on the reviewer vector, the final predicted reviewer label is obtained through a three-layer multilayer perceptron (MLP). In reality, the number of normal reviewers is far greater than the number of fake reviewers, this paper optimizes the loss function of the detection model by specifying the weight of the cost-sensitive matrix or the weight of the cost-sensitive vector.

In the experiment, we used a high-quality online reviews dataset from the Amazon platform and compared our method with logistic regression (LR), random forest (RF), Deepwalk, and FdGars [2]. The results prove that our method is effective and has reached a high accuracy.

In this paper, we studied how to identify fake reviewers from crowdsourcing in online review platforms. Taking into account the particularity of reviewers from the crowdsourcing platform, we have established an effective detection model that is sensitive to cost and can aggregate multiple features. The main contributions of this article are summarized as follows:

1. This paper integrates the review content features and the reviewer behavior features, using attention to obtain factors of the reviewer in both review-space and sociality-space.
2. Because of the imbalance of reviewer classes in real-world scenarios, we apply a cost-sensitive method to optimize the loss function of the detection model.
3. We verified our model on a real Amazon dataset and proved the feasibility and effectiveness by comparing other methods.

2 Related Work

Crowdturfing is a new type of fake reviewer which has emerged with the rise of crowdsourcing services. Many researchers have accumulated a certain amount of research on identifying fake reviewers in social networks or comment systems. These methods can be roughly divided into three categories [3]: Content-based Detection Methods, Behavior-based Detection Methods, and Graph-based Detection Methods.

Content-Based Detection Methods. Researchers found that the content of the reviews may contain some advertising information or deceptive instructions. A lot of work is devoted to distinguishing fake reviews from numerous reviews by using text analysis or Natural Language Processing (NLP) models. For example, Mukherjee et al. [8] crawl 64,000 user reviews on the Yelp platform, they first extract the N-gram and part-of-speech features of the review text, and then use a Support Vector Machine (SVM)classifier for reviewer classification; Yao et al. [17] use recurrent neural network (RNN) to generated fake online reviews for products and services automatically, and then use the loss features in RNN training to distinguish real reviews from fake reviews; Ott et al. [9, 11], Shojaee et al. [12], and Li et al. [10] use N-gram and other NLP methods to detect fake reviewers from the perspective of text analysis in the dataset based on a crowdsourcing platform.

Behavior-Based Detection Methods. Since fake reviewers can organize their language normally and answer questions like normal users, the reviews of fake reviewers have similar attribute values to those of normal users. Therefore, in addition to the review text, some researchers mine the behavior characteristics of the reviewers to predict fake reviewers. For example, Fei et al. [13] believe that reviewers that emerge explosively are more likely to be fake commenters. Through time series analysis and loopy belief propagation (LBP) methods, it can be inferred whether the reviewers in the graph are fake reviewers; Jiang et al. [14] find that there is a clear difference between spammers and ordinary users in the attributes of their tracking networks, but the structure of the tracking networks among spammers is similar, so they provide an unsupervised method to model the link features, and then calculate the degree of suspiciousness by sending the features to downstream tasks.

Graph-Based Detection Methods. In order to further analyze the interrelationships among objects in real-world problems, the researchers model the interacting objects into a graph structure, and then identify fake reviewer nodes through matrix calculation or graph embedding. For example, Wang et al. [4, 15] use a method which is similar to hyperlink-induced topic search (HITS) to iteratively calculate the reliability of the reviewer, the authenticity of the review and the credibility of the product by constructing a heterogeneous graph model; Akoglu et al. [16] construct a bipartite graph for reviewers and products, and propose a FraudEagle model based on the hidden Markov model, and use the sIA algorithm based on the LBP cyclic belief propagation to calculate the user's credibility; Parisa [6] believes that reviewers have similar behaviors if they reviewed the same product. Therefore, a reviewer-reviewer graph is constructed, and then the graph propagation algorithm is used to calculate the suspiciousness of the reviewers are from the crowdsourcing task according to the reviewers

with fake tags as seeds; Wang et al. [2] propose the FdGars that comprehensively considers the content of reviews and the behavior of reviewers. The model obtains the classification results through a two-layer graph convolutional networks (GCN). Then they propose a co-trained model to train the behavior-based model and the content-based model respectively for collaborative training [7]. Dou et al. [18, 23] designed the GNN-based model to against camouflages and solve the inconsistent problem in fraud detection. Besides, reinforcement learning was utilized to detect spammers [24].

All in all, the graph-based detection method takes into account the network topology among reviewers, reviewers and review products. However, in reality, there is a small proportion of critics from crowdsourcing in the comment area, and the existing methods rarely consider the problem of data imbalance. In addition, semantic feature in reviews is an important indicator for analyzing the characteristics of reviewers, while it has rarely been used in current methods. Therefore, it is worth studying that how to extract the common features of crowdturfing from a small number of categories and how to learn an unbiased model by using graph structure modeling.

3 Crowdturfing Detection Model

In the method, the strategy based on graph modeling is first introduced. In order to represent the node vector more accurately, the extraction and selection of review features are used. Then introduced a detection framework based on attention to aggregate reviewer vectors. Finally, considering the impact of unbalanced data on the detection, an improved cost-sensitive loss function method is introduced.

3.1 Modeling and Initialization

According to the previous content, one person can review multiple products, so there is a reviewer-to-product relationship diagram as shown in Fig. 2 between the reviewer and the product.

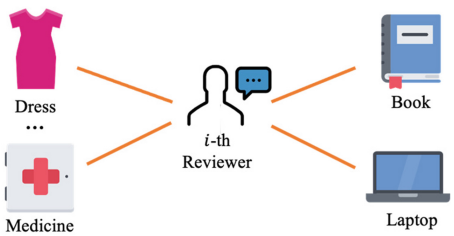

Fig. 2. The reviewer-to-product graph of the *i*-th reviewer.

Given that the crowdsourcing task is commodity-centric, users who have reviewed the same product may have similar characteristics. We assume that people who have reviewed the same product are neighbors, so a co-reviewer homogeneous sociality-space network can be built, as shown in Fig. 3.

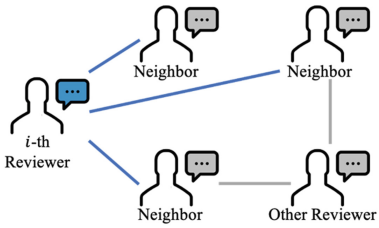

Fig. 3. Co-reviewer graph of all reviewers.

However, we found that the review information contains many features that can be mined, which can help distinguish fake reviewers from all reviewers. Therefore, the eigenvalues are used to replace the random vector representation when initializing the reviewers' embedding. Therefore, we extract and select the features from both the content of the reviews and the behavior of the reviewers.

Advertising information or deceptive indications may be included in the content of the review. Such features are called review content features. The detailed content features and description are shown in Table 1.

Table 1. The review content features and description.

Feature	Description
RL	Review Length
RSN	Review Symbol Number
RSR	Review Symbol Ratio
MCS	Max Content Similarity
RE	Review Emotion

- RL indicates the length of the review, represents the number of words in a review. It is found that the length of the fake reviews from the crowdsourced task is generally less than the normal reviews, and it is about half of the normal review. Therefore, the length of reviews can be used as an indicator to distinguish fake reviews from normal reviews.
- RSN indicates the number of special symbols in a review. There is existing that fake reviews will add special symbols to the text so as to avoid being recognized the same content or advertising information by the machine.
- RSR represents the proportion of special characters in the reviews. This value is the quotient of RSN and RL. RSR can explain the existence of special characters better compared to RSN and RSR.
- MCS represents the maximum content similarity of reviews, refers to the maximum similarity between the current review and the previous review of the reviewer. When a fake commenter writes a fake comment, it is very likely that he will copy the comment he has written before. So this indicator can be used for crowdturfing detection.

- RE indicates the emotional group of reviews. By dividing comment sentiment into multiple categories from positive to negative, the semantic information in the comment text can be reflected. Richer semantic information like this helps to detect fake reviewers.

In addition to analyzing the review text, the behavior of reviewers can also be used as a factor to distinguish between fake reviewers and normal reviewers. Table 2 lists the behavior features and descriptions of the reviewers.

Table 2. The reviewer behavior features and description.

Feature	Description
RN	Review Number
PN	Product Number
PCN	Product Category Number
PR	Praise Ratio
RD	Rate Deviation
WRN	Whether Real Name

- RN represents the number of reviews made by a user. Previous research has shown that fake reviewers generally post more reviews than ordinary reviewers, so the number of reviews can be used as one of the behavior features.
- PN represents the number of products that a user has reviewed.
- PCN represents the type's number of products a user has reviewed. Like RN and PN, it reflects user review behavior as a digital feature.
- PR represents good ratings. It's the number of five-star reviews a user rates as a percentage of the total. It is found that since the fake reviewers are mostly for the crowdsourced task of checking out positive or negative reviews, a user with either too high or too low a rating is likely to be a fake reviewer.
- RD represents the bias of rate. It is obtained by calculating the gap between the current reviewer and other reviewers for the same product. Because in general, there should not be a big gap in the ratings of the same product.
- WRN refers to whether a reviewer has verified his or her real name when registering for an account. Generally speaking, reviewers who have verified their real names are more credible and less likely to be fake reviewers.

The above is a summary of the features of the existing methods that are applicable to the current task. However, the applicable features may not be exactly the same, taking into account the differences between different datasets. Therefore, it is necessary to use a feature selection strategy to select the above features. By removing "Redundant" and "Irrelevant" features, the performance of the model can be improved.

Commonly used feature selection methods include filtering and packaging (e.g., Chi-Squared Test, Mutual Information). The selected features are used to initialize the embedding representations of reviewers and product reviews and contain more information than the random vectors.

3.2 CrowdDet Framework

The fake reviewer features extracted above can be used to initialize the reviewer node in Fig. 1. In order to integrate all the information to get the latent factor of the current reviewer, we propose an effective aggregation mechanism called CrowdDet.

The CrowdDet Framework. The overall structure is shown in Fig. 4. We can learn the latent vector of the reviewer from the review-space and the sociality-space by using this framework, and then combine the two to get the final classification prediction result. This scheme comprehensively considers the review content features and the reviewer behavior features, and the correlation among neighbors in the homogeneous graph is used to discover the spread of fake reviewer tags.

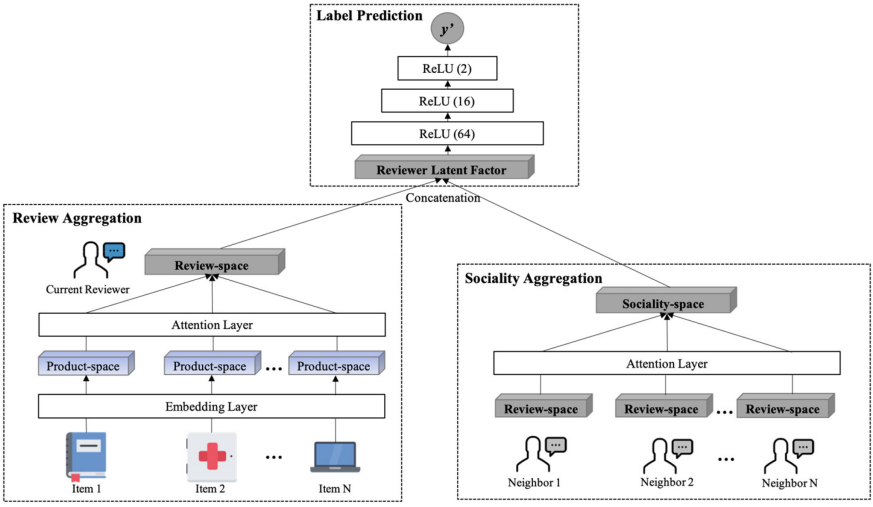

Fig. 4. The structure of the attention-based feature aggregation model CrowdDet.

It can be found from the figure that the CrowdDet framework contains three parts: review aggregation, sociality aggregation and label prediction. In review aggregation, the user's reviews are divided into products to obtain the user's embedding representation in the product-space, and the embedding of multiple product-spaces is aggregated through an attention layer to obtain the embedding in the review-space. Neighbors embedding in sociality aggregation is obtained from review aggregation, and through an attention layer to obtain sociality-space reviewer embedding. Then aggregating the output of the two aggregations to get the reviewer latent factor, and then input it into a three-layer MLP to get the prediction result. These are described in detail as follows.

Review Aggregation. It can be found from Fig. 2 that for the i-th reviewer, multiple products can be reviewed, and the edge of the graph is the corresponding comment. Therefore, we can formulate an aggregation rule to extract the potential factor $h_i^R \in R^d$

of the reviewer in the review-space from the reviewer-to-product graph, where d represents the potential dimension of the reviewer.

The general method is to obtain the review-space factor by learning the information in the product-space. The embedding in product-space is the aggregation of product embedding and comment information embedding, and the following function is needed, it is defined as Eq. (1) shows:

$$h_i^R = \sigma(W \cdot (\sum\nolimits_{p \in P(i)} \alpha_i x_{ip}) + b) \tag{1}$$

Among them, $P(i)$ represents the set of products that are connected to the i-th reviewer in the reviewer-to-product graph. x_{ip} is the aggregated representation of commodity a in product-space, x_{ip} can be calculated by Eq. (2):

$$x_{ip} = RELU(u_i \oplus v_p \oplus e_p) \tag{2}$$

u_i represents the embedding of user i, v_p can be understood as product review embedding, and e_p is product embedding. The reviewer behavior feature and review content feature extracted in the previous article are used to initialize u_i and v_p here, respectively. These three parts are connected by channel merging $\oplus$, and finally, the rectified linear unit (ReLU)is used for linear transformation, which aims to obtain the vector of the target dimension.

α_i represents the weight of each product-space vector aggregated by the i-th reviewer. Since it is an average strategy, α_i is expressed as Eq. (3) shows:

$$\alpha_i = \frac{1}{|P(i)|} \tag{3}$$

The mean aggregation strategy assumes that each product has the same impact on the current reviewer's embedding. But in practice, when reviewers comment, there may be deviations in their attention to each product. For example, a person's comment may come from completing a crowdsourcing task, or it may be written spontaneously. Since the mean aggregator exhibits limitations, we hope to comprehensively consider the impact of different product reviews on the vector representation of reviewers. Inspired by the attention mechanism, limited attention can be focused on key products.

The calculation of the product-space vector is as shown as Eq. (2), the h_i^R function is shown as Eq. (4) shows:

$$h_i^R = \sigma(W \cdot (\sum\nolimits_{p \in P(i)} \alpha_{ip} x_{ip}) + b) \tag{4}$$

In this case, α_{ip} can be adjusted according to different products, it is defined as Eq. (5) shows:

$$\alpha_{ip} = \frac{\exp\left(s\left(x_{ip}, u_i\right)\right)}{\sum_{p \in P(i)} \exp\left(s\left(x_{ip}, u_i\right)\right)} \tag{5}$$

Finally, the softmax function is used to normalize the above attention score to get the final attention weight. The attention scoring function $s(x_{ip}, u_i)$ is defined as Eq. (6) shows:

$$s(x_{ip}, u_i) = w_2^T \cdot \sigma(W_1 \cdot [x_{ip} \oplus u_i] + b_1) + b_2 \tag{6}$$

Social Aggregation. A reviewer will have neighbors. If they have reviewed the same product, the neighbor relationship is shown in Fig. 3. Due to the influence of neighbor nodes on the current reviewer, the potential representation $h_i^S \in R^d$ of the reviewer in sociality-space can be obtained by aggregating the embeddings of neighbor nodes.

For the i-th reviewer, the embedding of its neighbor node is the representation of his neighbor in review-space. The degree of attention to different neighbors is different during aggregation. For example, a product review may be written by crowdturfing or by ordinary users. Therefore, the correlation between a reviewer and its neighbors in the co-reviewer graph is different. The same attention mechanism is used to integrate the neighbor node vector with the attention weight of β_{in} to obtain h_i^S. From Eqs. (7)–(9), the specific operation is described:

$$h_i^S = \sigma(W \cdot (\sum\nolimits_{n \in N(i)} \beta_{in} h_n^R) + b) \tag{7}$$

$$\beta_{in} = \frac{\exp(s(h_n^R, u_i))}{\sum_{n \in N(i)} \exp(s(h_n^R, u_i))} \tag{8}$$

$$s(h_n^R, u_i) = w_2^T \cdot \sigma(W_1 \cdot [h_n^R \oplus u_i] + b_1) + b_2 \tag{9}$$

In the formula, h_n^R represents neighbors' vector of the current reviewer.

Label Prediction. When getting the node embedding of the product and review content, the potential representation h_i^R of the reviewer in the review-space is obtained by aggregating them. At the same time, by aggregating neighbors who have an interactive relationship with the current reviewer, the potential representation h_i^S of the reviewer in the sociality-space is obtained. At last, the potential expression of the reviewer needs to consider both the latent factor in review-space and it in sociality-space. Thus, after fusing the features, the two latent factors are combined into the final reviewer latent factor through a mapping. The definition of the reviewer latent factor h_i are represented as Eqs. (10) and (11) show:

$$c = [h_i^R \oplus h_i^S] \tag{10}$$

$$h_i = \sigma(\mathrm{W} \cdot \mathrm{c} + \mathrm{b}) \tag{11}$$

After calculating the latent representation h_i of the user, the final prediction result y' is obtained through a standard MLP. In a mathematical method, the operation are described from Eqs. (12)–(14):

$$g_1 = \sigma(W_1 \cdot h_i + b_1) \quad \ldots \tag{12}$$

$$g_l = \sigma(W_l \cdot g_{l-1} + b_l) \tag{13}$$

$$y' = w^T \cdot g_l \tag{14}$$

In the formula, l represents the number of hidden layers of the MLP.

Our research problem can be regarded as a binary classification problem, so the output y' of the model is a two-dimensional result. The two values represent the probability of ordinary and crowdsourced. Then we calculate the loss based on the above probability and train the parameters in the model through the gradient descent method.

However, the positive and negative data is unbalanced in the real scenario. In order to achieve a better effect, we provide an optimization solution to this problem.

3.3 Improved Cost-Sensitive Loss Function

In order to deal with the sample imbalance problem from the algorithm level, we adopt a direct cost-sensitive learning method to alleviate the impact of class imbalance by specifying the misclassification weight of the cost-sensitive matrix or the cost-sensitive vector. Thus, this study intends to use cost-sensitive methods to optimize the loss function of the detection model and to train the model supervised. The cost information is embedded in the objective function of the learning model, and a cost-sensitive learning algorithm f is obtained by minimizing the expected loss. The general form of the training process is shown as Eq. (15) shows:

$$\min_f : loss(f, X, C) \tag{15}$$

X represents the training sample set, C is the misclassification cost matrix.

For the traditional cross-entropy loss, even if it is easy to classify (the probability of correct classification is greater than 0.5), there is a higher loss. In this study, a modulation factor $(1 - p_t)^{\gamma}$ is added to the original cross-entropy loss. We reduce the weight of easy-to-classify samples so that the model can focus more on difficult-to-classify samples during training. The definition of Focal loss and p_t are shown as Eqs. (16) and (17) show:

$$FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t) \tag{16}$$

$$p_t = \begin{cases} p & if \ \ y' = 1 \\ 1 - p & otherwise \end{cases} \tag{17}$$

p_t represents the probability that the user is estimated to be positive or negative. γ is the focus parameter, and $\gamma > 0.p$ represents the probability that user is regarded as a fake reviewer.

Considering the large gap in the number of positive and negative samples, adding weight to the positive and negative samples is a common strategy. The frequency of negative samples is high, so we reduce the weight of negative samples. Since the number of positive samples is small, the weight of positive samples should be relatively increased. The final focal loss function is shown as Eq. (18) shows:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{18}$$

$\alpha \in [0,1]$ is the weight used to control the balance. We can control the shared weight to the total loss by setting the α value and take a relatively small value to reduce the weight of the negative sample (the one with more samples). Then take a smaller value of α to reduce the weight of negative samples (the one with more samples).

By adjusting the value of the weight α and the focus parameter γ, the model will be more concerned about the few samples and set a higher loss for the samples that are difficult to classify. So we can get a classifier that is sensitive to the cost of a small number of samples by optimizing the loss of the model.

4 Experiment

In this section, we use a real dataset from the Amazon platform for verification. Evaluation indicators, experimental settings and experimental results are also introduced.

4.1 Dataset and Evaluation Metrics

We use real datasets to conduct experiments, aims to verify the effectiveness of the model in identifying fake reviews created by crowdturfing platform organizations, and check the performance of the model in unbalanced data. Yao et al. [1] collected more than 110,000 high-quality online review data on the Amazon platform in 2017. The reviews of this dataset come from different categories and different reviewers. The reviewers with fake tags are all from the tasks posted by the crowdsourcing platform. The number of reviews, reviewers, and products in the dataset is shown in Table 3. Every review, reviewer and product have truthful and deceptive labels. The selected products are distributed in different product categories such as Books, Health, Electronics, Movies or Other, and the Other category is a mixture of multiple categories.

Table 3. Statistical data of reviews, reviewers and products.

	Deceptive	Truthful
Reviews	10114	101226
Reviewers	1524	18162
Products	994	72266

For the 19,686 reviewers in the dataset, we divided them into 11,812 as the training set, 1969 as the validation set, and 5,909 as the test set. The validation set was used to train the hyperparameters of the model.

Crowdturfing detection is essentially a binary classification task, so we can get a confusion matrix, as shown in Table 4. The positive cases are fake reviewers from crowdsourcing, and the negative cases are regular commenters. The horizontal header of the table represents the prediction result, and the vertical header represents the actual label.

Table 4. Confusion matrix of crowdturfing detection.

	Fake	Normal
Fake	TP (True Positive)	FP (False Positive)
Normal	FN (False Negative)	TN (True Negative)

Since the purpose of detection is to detect as many fake reviewers as possible, in the performance evaluation, recall is used as the main evaluation index. Its calculation process is defined as Eq. (19) shows:

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

In addition, precision, F1, and accuracy are also important references that can be used to measure the detection effect, and the calculation process are defined as follows from Eqs. (20)–(22):

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{21}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

4.2 Baseline Methods

In order to verify the feasibility of the above theory, we verified the proposed method on an actual dataset. We selected several different types of algorithms that are commonly used as a classification task as the experimental baseline methods, including machine learning methods and graph-based methods.

- Logistic Regression. Based on the features of 19,686 reviewers in the Amazon dataset, an LR classifier is trained supervised. 70% of the data is used as the training set and 30% as the test set. Through this classifier, the reviewers in the test set are divided into real reviewers and fake reviewers.

- Random Forest. Similarly, the RF model is trained based on the features of the reviewer. When the tree is set to 1, the classifier is the decision tree (DT) model.
- Deepwalk. The nodes embedding is learned from the co-reviewer homogeneous graph. When training the DeepWalk model, the number of walks for each node is 80, the length of random walk is 10, and the length of node embedding is 128. Finally, the final result is obtained through an LR classifier.
- FdGars. The FdGars model is proposed by Wang et al. [2]. It is actually a two-layer GCN model. When initializing node embedding, we use review content features and reviewer behavior features in stand of random features. The length of node embedding is 50. The dataset is divided into the training set, validation set, and test set.

4.3 Data Preparation and Experiment Settings

According to the review records from Amazon, we constructed a reviewer-to-product diagram for each reviewer and a co-reviewer diagram that includes all reviewers. We extracted the node and the structural features of the graph based on the features of the comment data, then obtained a feature matrix containing the information in Table 5.

Table 5. Amazon dataset characteristics and description.

Feature	Description
RL	Review Length
RR	Review Rate
RE	Review Emotion
RP	Review Product
WP	Whether Purchase
WRN	Whether Real Name
NI	Neighbor's ID

Among them, RL, RR, and RE belong to the characteristics of the comment content. RP, WP, WRN, and NI belong to the characteristics of the commenter's comment behavior. This paper adopts filtering or encapsulation methods such as deleting Variance Threshold, Chi-Square Detection, L1 Regularization, and Decision Tree Strategies to filter the extracted features. The filtered features are integrated to initialize the embedding representation of reviewers, products, and product reviews instead of the random representation vector in the original strategy.

In order to explore the influence of the parameters of the focal loss function on the evaluation indicators, we fixed other influencing factors and modified γ values to obtain the experimental results shown in Table 6.

Table 6. The impact of changes in the parameters of the focal loss function on the classification results.

γ	Recall	Precision	F1	Accuracy
0.5	0.83	0.82	0.81	0.970
0.75	0.85	0.84	0.84	0.975
2	0.88	0.83	0.78	0.971

Recall, precision, F1 and accuracy of the entire test set classification are used as evaluation indicators. It can be found that the recall increases with the change of the γ. However, only when the value of γ is 0.75, the comprehensive index of the model is relatively high.

4.4 Performance Evaluation

To explore the effectiveness of the model for extracting review content and reviewer relations, we conducted ablation study on the two modules of the review aggregation and sociality aggregation. The results are shown in Table 7.

Table 7. Performance evaluation using different modules.

Method	Recall	Precision	F1	Accuracy
Review-only	0.47	0.71	0.56	0.94
Sociality-only	0.14	0.74	0.24	0.92
Review+Sociality	0.85	0.84	0.84	0.975

- Review-only. Only product and review features are considered in this method, without the neighbor relationship between user nodes.
- Sociality-only. This method only considers the features of the user's neighbors, without the features of products and reviews.
- Review + Sociality. This is the strategy we propose, taking into account the impact of products, comment features and neighbor relationships on user embedding.

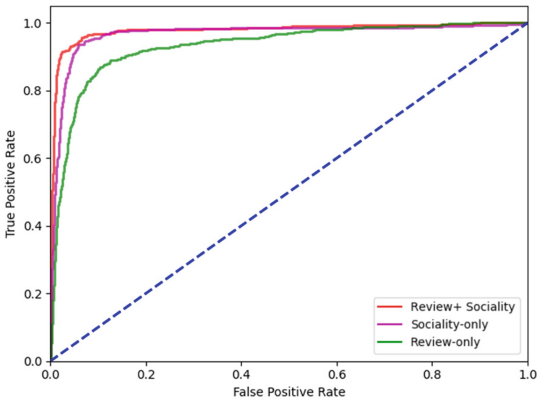

Fig. 5. Receiver operating characteristic (ROC) curve with different modules.

Figure 5 shows the ROC curve using different modules. The results show that the combined effect of the two aggregators is the best in terms of recall, accuracy, F1, accuracy, and area under curve (AUC) value, which is better than one aggregator. This is because the two dimensions of reviewer features are considered comprehensively, and richer information is integrated. It confirms that the model does not have redundancy and integrates the features of the reviewer effectively.

In addition, the proposed method needs to be compared with baseline methods. Table 8 shows the classification results of different methods for the Amazon dataset.

Table 8. The impact of the focal loss function's parameters on the classification results.

Method	Recall	Precision	F1	Accuracy
Logistic regression	0.13	0.73	0.22	0.93
Decision tree	0.41	0.55	0.47	0.93
Random forest	0.42	0.58	0.49	0.93
DeepWalk	0.82	0.80	0.81	0.968
FdGars	0.31	–	–	0.925
CrowdDet	0.85	0.84	0.84	0.975

From the experimental results, it can be found that the machine learning classification method has reached a high accuracy rate, which is caused by the imbalance amounts of positive and negative categories. In fact, the rate of recall is relatively low. When the method of graph embedding is adopted, the classification effect has been significantly improved, which shows that the structural information of the graph has a positive influence on the classification of reviewers. However, the solution of the FdGars model has not achieved good results in the current problem. The method we proposed obtained a good detection effect by extracting multiple features and using the CrowdDet method for fusion. The recall rate is 0.85, the precision is 0.84, the F1 is 0.84 and the accuracy is 0.975. Compared with the DeepWalk method, recall, precision, F1 and accuracy are increased by 0.03, 0.04, 0.03, 0.007, respectively.

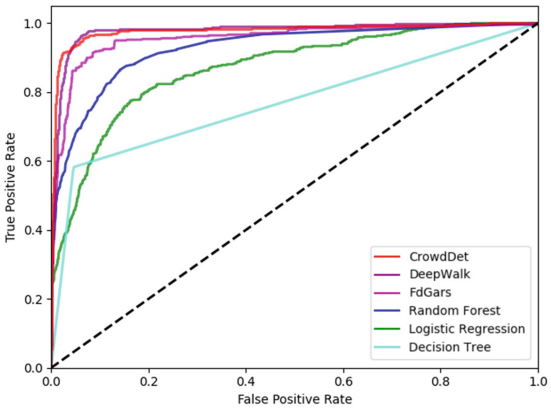

Fig. 6. ROC curve of different detection strategies.

Figure 6 shows the ROC curve of different strategies, the AUC values are 0.9784 for CrowdDet, 0.9783 for DeepWalk, 0.9557 for GCN, 0.9268 for RF, 0.8695 for LR, 0.7681 for DT. Although the curve of DeepWalk is close to the proposed method, its other indicators have not surpassed CrowdDet. It can be seen that we have verified the effectiveness of the proposed method on real datasets and achieved better classification results than other strategies.

5 Conclusion

In this paper, we studied a method for detecting fake reviewers from crowdsourcing. Considering the existing graph-based methods that have not achieved significant results. An attention-based strategy was used to integrate the representation of the review-space and the sociality-space. Firstly, we constructed two graphs and extracted the structural information and node features in the graphs. Not only the review features in the product review-space but also the impact of social interaction among reviewers was considered. Secondly, we calculated the reviewer latent factors through the neural network aggregation mechanism, using the focal loss to optimize the classifier. Finally, we validated the proposed method on the Amazon dataset. It is found that the proposed method can basically cover the fake reviewers among all reviewers, and the classification effect has been evaluated in the experiment. In the future, we plan to utilize the information of review groups and product clusters to further expand the detection methods of crowdturfing.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (grant no. 61772560), and the Fundamental Research Funds for the Central Universities of Central South University (grant no. 2021zzts0747).

References

1. Yao, W., Dai, Z., Huang, R., et al.: Online deception detection refueled by real world data collection. arXiv preprint arXiv:1707.09406 (2017)
2. Wang, J., Wen, R., Wu, C., et al.: FdGars: fraudster detection via graph convolutional networks in online app review system. In: Companion Proceedings of the 2019 World Wide Web Conference, pp. 310–316 (2019)
3. Jindal, N., Liu, B.: Opinion spam and analysis. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 219–230 (2008)
4. Wang, G., Xie, S., Liu, B., et al.: Review graph based online store review spammer detection. In: 2011 IEEE 11th International Conference on Data Mining, pp. 1242–1247. IEEE (2011)
5. Rayana, S., Akoglu, L.: Collective opinion spam detection: Bridging review networks and metadata. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 985–994 (2015)
6. Kaghazgaran, P., Caverlee, J., Squicciarini, A.: Combating crowdsourced review manipulators: a neighborhood-based approach. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 306–314 (2018)

7. Wang, J., Wen, R., Wu, C., et al.: Analyzing and detecting adversarial spam on a large-scale online APP review system. In: Companion Proceedings of the Web Conference, pp. 409–417 (2020)
8. Mukherjee, A., Venkataraman, V., Liu, B., et al.: What yelp fake review filter might be doing? In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 7(1) (2013)
9. Ott, M., Choi, Y., Cardie, C., et al.: Finding deceptive opinion spam by any stretch of the imagination. arXiv preprint arXiv:1107.4557 (2011)
10. Li, J., Ott, M., Cardie, C., et al.: Towards a general rule for identifying deceptive opinion spam. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1566–1576 (2014)
11. Ott, M., Cardie, C., Hancock, J.T.: Negative deceptive opinion spam. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 497–501 (2013)
12. Shojaee, S., Murad, M.A.A., Azman, A.B., et al.: Detecting deceptive reviews using lexical and syntactic features. In: 2013 13th International Conference on Intelligent Systems Design and Applications, pp. 53–58. IEEE (2013)
13. Fei, G., Mukherjee, A., Liu, B., et al.: Exploiting burstiness in reviews for review spammer detection. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 7(1), pp. 175–184 (2013)
14. Jiang, M., Cui, P., Beutel, A., et al.: Catching synchronized behaviors in large networks: a graph mining approach. ACM Trans. Knowl. Discovery Data (TKDD) **10**(4), 1–27 (2016)
15. Wang, G., Xie, S., Liu, B., et al.: Identify online store review spammers via social review graph. ACM Trans. Intell. Syst. Technol. (TIST) **3**(4), 1–21 (2012)
16. Akoglu, L., Chandy, R., Faloutsos, C.: Opinion fraud detection in online reviews by network effects. ICWSM **13**(2–11), 29 (2013)
17. Yao, Y., Viswanath, B., Cryan, J., et al.: Automated crowdturfing attacks and defenses in online review systems. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, pp. 1143–1158 (2017)
18. Liu, Z., Dou, Y., Yu, P.S., et al.: Alleviating the inconsistency problem of applying graph neural network to fraud detection. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1569–1572 (2020)
19. Moreno-Torres, J.G., Herrera, F.: A preliminary study on overlapping and data fracture in imbalanced domains by means of genetic programming-based feature extraction. In: 2010 10th International Conference on Intelligent Systems Design and Applications, pp. 501–506. IEEE (2010)
20. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans. Knowl. Data Eng. **18**(1), 63–77 (2005)
21. Lin, T.Y., Goyal, P., Girshick, R., et al.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
22. Fan, W., Ma, Y., Li, Q., et al.: Graph neural networks for social recommendation. In: The World Wide Web Conference, pp. 417–426 (2019)
23. Dou, Y., Liu, Z., Sun, L., et al.: Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 315–324 (2020)
24. Dou, Y., Ma, G., Yu, P.S., et al.: Robust spammer detection by Nash reinforcement learning. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 924–933 (2020)