Camouflage is NOT easy: Uncovering adversarial fraudsters in large online app review platform

Jianyu Wang and Chunming Wu

Abstract

Given users and products that he/she reviews, can we recognize fake reviews just using the text information, or determine whether a reviewer is a fraud or not? Automatically detecting fake reviews and reviewers is an urgent problem and lots of work attempts for discovering linguistics, behaviors and graph patterns. However, in reality, there are new kinds of fraudsters who can change their behaviors to camouflage as genuine reviewers to avoid detection systems. With the fraudsters become distributed, dynamic, and adversarial, anti-spam tasks face a new challenge. In this paper, we tackle the challenge of adversarial fraudsters in online app review platform and propose a system called DDF (Detect, Defense, and Forecast) to uncover camouflage accounts. Firstly, we select a small set of seed with high-precision based on text and behavior features; Secondly, we build our graph-based detection model for uncovering hidden (distant) users who serve structurally similar to the seed by utilizing Graph Convolutional Network (GCN) algorithm. Thirdly, we evaluate DDF using real-world data set from Tencent APP Store and analyze the potential fraudsters detected by DDF. It is worth mentioning that precision can achieve 0.95. Finally, we validate the efficiency and scalability of DDF and show that it can be well transferred to other anti-spam tasks.

Keywords

Adversarial Fraudsters, Graph Convolutional Network, Detection System

Introduction

User review plays an important role in making decisions since it provides valuable information. With the openness of the online review platform, many attackers get profit and monetary rewards by doing some irregular activities, such as writing fake reviews and posting advertisements to interfere with the rank of products.

The task of detecting fraud reviews and reviewers exists for a long period. Some works focus on the characteristics of content and build text-based classifiers to get high accuracy. Some works focus on analyzing abnormal behaviors (e.g. timestamps, footprints, and distributions) to discover suspicious patterns. Graph-based methods are popular since they leverage the relationship ties between users, which makes considerable signs of progress in spotting malicious accounts. At the same time, these methods are gradually adopted by many organizations for deploying their risk-control systems. In particular, deep learning-based models could be applied for generating fake reviews. Even worse, all of these attack methods are practical and effective.

As fraudsters become adversarial, distributed, and dynamic, anti-spam tasks face a huge challenge. Figure 1 shows three types of fraud users in online app review platforms. A(I) is an attacker who wants to promote the rank of certain products and post many positive reviews for it. It is easy to be detected by behavior features. To avoid the detection systems, A(II) is an updated strategy with posting positive reviews to different products, which means the attacker presents like a normal user. A(III) is an attacker who hires workers to take part in spam activities by the crowdsourcing platforms (e.g. RapidWorkers and Microworkers). In particular, deep learning-based models could be applied for generating fake reviews. Even worse, all of these attack methods are practical and effective.

Unfortunately, attackers begin to avoid detection systems by changing their measures. Some of them try to look normal, which add links to popular entities. Some of them hire workers to take part in spam activities by the crowdsourcing platforms (e.g. RapidWorkers and Microworkers). In particular, deep learning-based models could be applied for generating fake reviews. Even worse, all of these attack methods are practical and effective.

As fraudsters become adversarial, distributed, and dynamic, anti-spam tasks face a huge challenge. Figure 1 shows three types of fraud users in online app review platforms. A(I) is an attacker who wants to promote the rank of certain products and post many positive reviews for it. It is easy to be detected by behavior features. To avoid the detection systems, A(II) is an updated strategy with posting positive reviews to different products, which means the attacker presents like a normal user. A(III) is an attacker who hires workers to take part in spam activities by the crowdsourcing platforms (e.g. RapidWorkers and Microworkers). In particular, deep learning-based models could be applied for generating fake reviews. Even worse, all of these attack methods are practical and effective.

Corresponding authors:
Chunming Wu, College of Computer Science and Technology, Zhejiang University, 38 Zheda Road, Hangzhou 310027, China. Email: wuchunming@zju.edu.cn
Jianyu Wang, College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang China. Email: xiaoyu_paopao@zju.edu.cn
normal user by camouflage. A(III) is more sophisticated, reviewers take part in crowdsourcing activities and post reviews to products with the distributed time slots and different devices, which brings a challenge to fraud user detection.

In this paper, we draw attention to the camouflage users and propose our DDF system by utilizing Graph Convolutional Network (GCN) algorithm. By studying the patterns of data set and analyzing the behaviors of fraud users, we firstly identify obvious abnormal users of train set as positive samples (seed candidates) and others as negative samples. Secondly, we construct a graph with 82,542 nodes and 42,433,134 edges and extract the text and behavior characteristics for each node. Then, we train a GCN model to find suspicious users from the test set.

Finally, we measure the precision and recall rates of the detected suspicious users.

Considering some adversaries present like normal users in the early days and then do illegal activities to avoid detection systems, it is hard to verify the accuracy of suspicious users. Therefore, we evaluate these suspicious users by expert experience with text and behavior characteristics for 30 days. Surprisingly, DDF is able to spot almost 50% potential adversaries and the precision remains nearly 95%.

We summarize the contributions of this work as follows:

1. We shed light on the importance of detecting adversarial fraudsters in anti-spam tasks and firstly build our DDF system for discovering more potential abnormal users.

2. Our system is an efficient and scalable system by using Graph Convolutional Networks. Additionally, it can be transferred to other anti-spam applications and platforms.

3. We validate the performance of our system in real-world datasets by deploying it on the Tencent Venus Computation Platform, which shows that our DDF system is quite competitive and achieves a high precision in real anomaly detection industry tasks.

Related work

Camouflage problems has received significant attention in recent years since fraudsters begin to change their methods to avoid detection systems, which means that they may look “normal” when doing illegal activities.1 focuses on fake followers on Twitter. They find some fraudsters evade detection systems by adding reviews or follows with honest targets.6 conducts the problems on Sybil and show this new advanced scheme in Dianping platform. They discover the Sybils become increasingly sophisticated by providing fake content among little pieces of their information, regardless of their accuracy.2 target on crowdsourcing problem, and find some normal users could engage in posting fake reviews by getting a reward from the crowdsourcing platform (RapidWorkers, ShortTask, Microworkers). Even worse,4 demonstrate a new class of attacks based on RNN models for fake review generation and customization, which is practical and effective to automate the fake online reviews and avoid detection.
Defense Methods in Anti-Spam Tasks have been proposed by researchers and industrial organizations for many years, which can be classified into three types: content-based,7 behavior-based8 and graph-based.1–3,9–12 Since text and behavior features cannot capture relationship between entities, graph-based methods are widely used in anti-spam works.1 spot camouflage or hijacked accounts by finding dense subgraphs.9 employs the Belief Propagation mechanism to detect suspicious patterns.2 exploits neighborhood-based characteristics to uncover hidden (distant) users by mapping users into a low-dimensional embedding space.3 proposes a risky account detection system based on local graph clustering algorithms. Moreover, some recent works13,14 apply GCN (graph convolution network)5 models in their risk-control systems to learn functions for each account with good performance. However, these approaches may have difficulty in discovering potential adversarial fraudsters in the future.

The framework

Our goal is to detect adversarial fraudsters in the large online app review platform. In this work, we propose a Smart DDF system by combining content, behavior, and graph characteristics and deploy it in real-world scenarios.

Overview

As Figure 2 shows, DDF mainly consists of three components: pre-processor, seed collector, and detector. The pre-processor is used for raw data processing and building a graph for the GCN module’s training and predicting. The seed collector is designed based on the feature extraction after the preprocessor. After characterizing and modeling fraud users in online app review systems, we can identify a small number of precision-focused seeds. With many iterations of training a GCN model, our seed set will expand dynamically and uncover new types of potential adversarial fraudsters. The graph-based detection module focuses on the structural characteristics of fraud user, it is designed by leveraging neighbors information via the GCN algorithm.

Firstly, we collect user review data set and divide it into two parts: (1) raw review logs for text and behavior feature extraction and (2) a user graph for obtaining structural characteristics. Secondly, we assume that fraud users can be identified by utilizing content (similarity, special symbol, semantic) and behavior (review time, device update, frequency) features. By setting different thresholds, fraud users can be identified by a seed collector with high precision. Then, we train our GCN models by leveraging user features and labeled information means it is from seed set and the corresponding user is a positive sample). After that, users who have high similarities with fraud users will be recognized. Especially, the hidden (distant) users can be discovered via a propagation function. Finally, we evaluate the detected users and expand our seed set to uncover adversarial fraudsters as many as possible.

Feature extraction

In this section, we discuss the process of feature extraction, which leverages content and behavior characteristics for each user.

In online app review platform, user review (comment) is regarded as an obvious feedback signal to products. In addition, the rating of a review scaled from 1 star (worst) to 5 stars (best) represents a user’s attitude. Table 1 shows 11 content-based features we extracted for each user. The SRN (Similar Review Number) is calculated by using Simhash15 algorithm and the WF/BF is labeled by using our own black dictionary for online app review.

In reality, user behavior contains much information to describe different users such as review times, the frequency of posting reviews, and the number of review devices. Table 2 shows three behavior features we extracted for each user.

Apart from that, temporal action reflects the sequential behavior of users. Table 3 lists the temporal action features we extract for each user, which are utilized as the attributes of nodes in our graph models.
TQD is a 24-dimensional vector, each dimension represents the number of reviews of each user during 24 hour.

SQD is the vector of user rating from 1 to 5.

Seed identification

Aiming at modeling different types of users in online app review platform, we categorize all users into three types: fraud user, normal user and suspicious user. We see that normal user is hard to identify since adversarial fraudsters could camouflage to look like normal users. For example, given a certain review not having illegal information, we hardly to determine if it is honest. If a person write three reviews for same products, we also cannot make sure that he/she is a fraud user just by leveraging his/her behavior feature. Through our investigation, we find that fraud users always have the same distribution in some characteristics inevitably, so we define fraud user firstly. Fraud user is defined by using content and behavior features mentioned in Tables 1 and 2. For instance, if a person who posts advertisement, phone number, irrelevant content or has obvious attentions to promote products with a large number of fake reviews, we regarded he/she as a fraud user. Other undetected users are labeled as normal user. However, most adversarial fraudsters are not detected and labeled as a normal user in this process. To distinguish normal user and adversarial fraudsters, we increase a new type of user—suspicious user. Based on labeled data of fraud users and normal users, we build GCN-based model to discover users who have a strong relationship with fraud users, even they camouflage as normal users in content and behavior characteristics. Particularly, we define suspicious users with the help of domain experts.

Graph construction

In this work, we consider a two-layer GCN for fraudster detection. As Figure 3 shows, nodes in G represent users and edges represent their relationships. A is the symmetric adjacency matrix of the undirected graph G, and  is a symmetric normalization of A with self-loops. \[ \hat{A} = \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}, \quad A = \hat{D}^{-1} \hat{A}, \] where  is the row-normalized adjacency matrix.  is the diagonal node degree matrix of A and X represents user feature vectors.

Here, we define layer-wise propagation rule:

\[ H^{(0)} = X \] \[ H^{(1)} = \sigma \left( \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(0)} W^{(0)} \right) \] \[ H^{(2)} = \sigma \left( \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(1)} W^{(1)} \right) \] Note that  denotes the t-th hidden layer with  = X,  and  = the matrix of activations in the 1st and 2nd layer. \( W^{(i)} \) is the layer-specific parameter.

\[ D = \begin{bmatrix} \sum_{j=0}^{N-1} \hat{A}_{0,j} & 0 & \cdots & 0 \\ 1 & \sum_{j=0}^{N-1} \hat{A}_{1,j} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \sum_{j=0}^{N-1} \hat{A}_{N-1,j} \\ 0 & 0 & \cdots & 1 \end{bmatrix} \] (4)
Here, the softmax activation function is defined as

$$
\text{Softmax} = \frac{1}{Z} \exp(x_i)
$$

where $Z = \sum_i \exp(x_i)$.

Figure 4 shows the two-layer GCN we constructed in our DDF system. Propagation of feature information from neighboring nodes in every layer improves classification. We show that GCN model can detect more suspicious fraudsters than other classification models.

**Deployment**

Finally, we deploy our risk-control system DDF on Venus Computation Platform provided by Tencent Inc. Specifically, we implement the content and behavior features extraction module using Hive SQL, graph model training using Python and store graph and model information in PCG S3 which is an object storage system in Tencent.

**Experiment**

In this section, we evaluate DDF on real-world data set with baseline methods to verify the performance of fraudster detection.

**Data set**

The review data set provided by Tencent includes 85,025 users, 302,097 reviews and 7,584 apps. Based on this data set, we extract features for each user as expressed in Section 3.2 and construct a graph structure as introduced in Section 3.4. Table 4 shows the number of nodes and edges. It’s worth mentioning that we have excluded the isolated nodes from the graph. Because isolated nodes have no neighbors, they cannot be affected by their neighborhood during network learning.

We divide the data set into train set and test set to verify the detection performance of our DDF system. As Table 4 shows, the number of train set is 31,450 and the number of test set is 51,092. In real industry work, it is hard to manually label each log since adversarial users can change their attack methods frequently and human labor is expensive. As introduced in Section 3.3, we filter out obvious abnormal users (also called seed) as positive samples by experience rules in our train set. For test set in our work, we label each user whether he/she is a fraudster according to the text and behaviors in the next 30 days. Fraud users are regarded as positive samples and others are negative samples. The labeled result is listed in Table 4.

**Seed selection**

In this work, two rules are utilized to select obvious abnormal users as seed. Fraud users usually review apps in continuous days, or use lots of devices to publish their comments in order to seek profit. We choose two attributes to label users, expressing as

$$
\text{Label}(i) = \begin{cases} 
1 & \text{if } C_i \geq \theta_c \land d_i \geq \theta_d \\
0 & \text{otherwise}
\end{cases}
$$

where, $c_i$ is the continuous days that user has posted reviews in a designated period; $d_i$ records the number of device that user has used in the same period; and are corresponding thresholds. If $C_i \geq \theta_c$ and $d_i \geq \theta_d$, we consider user as a fraud user because this user is obvious abnormal in some behaviors; otherwise, user is a normal user.

By setting $\theta_c = 7$ and $\theta_d = 20$, Figure 5 shows the distributions of fraud users and normal users with content-based features of RQ and RRR as described in Section 3.2. Obviously, normal users gather at the bottom-left of the coordinate. Based on our analysis, most normal users’ RQs are less than 30. Comparatively, fraudsters gather at the bottom of the coordinate. Most fraudsters’ RRRs are less than 0.2. Consequently, we think attributes and can be used to recognize fraud users and label train set in the following experiment.

Combining the two attributes above, Table 5 lists the quantity of fraud users by using different thresholds of $\theta_c$ and $\theta_d$. Obviously, the quantity has a linear relationship with the thresholds. Then, we hope to find fraud users from the test set as much as possible according to the train set.

**Table 4. Graph statistics.**

| Graph | Train set | Test set |
|-------|-----------|----------|
| Nodes | 82,542    | 31,450   | 1.7824   |
| Edges | 42,433,134| 0.43,268 |          |

Wang and Wu 2141
Baseline methods

We compare the detection results with the following state-of-the-art baselines. In this work, two widely-used methods for classification and two famous graph structure based methods for node embedding are utilized as our baseline methods.

- **Logistic Regression**: is used to binary classification and to explain the relationship between one dependent binary variables.
- **Random Forest**: is a classification algorithm consisting of many decisions trees.
- **DeepWalk**: learns latent representations of nodes. We set the number of random walks with 10, the length of random walk is 80, and the embedding length is 128.
- **LINE**: learns latent representations of nodes with 1-step and 2-step neighbors. We set negative ratio with five and the embedding length with 128.

Detection results

We evaluate our experiment with the precision and recall metrics. Table 6 lists the detected results of DDF.

| Login device number | Continuous day |
|---------------------|----------------|
|                     | 5  | 7  | 10 | 15 | 20 | 25 | 31 |
| 10                  | 5388 | 4288 | 2924 | 1469 | 874 | 299 | 46 |
| 20                  | 5190 | 4234 | 2922 | 1469 | 874 | 299 | 46 |
| 50                  | 2469 | 2424 | 2097 | 1346 | 843 | 299 | 46 |
| 100                 | 825  | 825  | 791  | 700  | 509  | 221 | 45 |
| 150                 | 224  | 224  | 224  | 217  | 182  | 88  | 27 |
| 200                 | 40   | 50   | 40   | 39   | 39   | 22  | 3  |

Table 5. Suspicious user statistics of train set by using different threshold.

| Threshold : $\theta$ | Train set | Detected users | Precision |
|-----------------------|-----------|----------------|-----------|
| d-10                  | 1:9142 0:73,400 | 5059 | 0.87 |
| d-20                  | 1:5830 0:76,712 | 4385 | 0.93 |
| d-50                  | 1:2474 0:80,068 | 2652 | 0.95 |
| c-5                   | 1:4811 0:77,731 | 4049 | 0.94 |
| c-7                   | 1:4330 0:78,212 | 3435 | 0.95 |
| c,d-5,10              | 1:5388 0:77,154 | 4155 | 0.94 |
| c,d-7,10              | 1:4288 0:78,254 | 3434 | 0.95 |
| c,d-5,20              | 1:5190 0:77,352 | 4012 | 0.94 |
| c,d-7,20              | 1:4234 0:78,308 | 3444 | 0.95 |

Table 6. Detected results of DDF by using different threshold.

Users of train set are labeled through thresholds and as introduced in Section 4.2. In Table 6, $(d-n)$ means $\theta_d = n$, $(c,-n,m)$ represents $\theta_c = n$ and $\theta_d = m$. Different compositions of $\theta_c$ and $\theta_d$ will generate different ground truths, further detecting different numbers of suspicious users.
Figure 6 clearly shows the detection details with different thresholds. The black histograms represent fraud users of train set with the corresponding thresholds. The red histograms represent suspicious users of test set who have been detected according to the corresponding trained model. We conclude the results from two aspects:

**Quantity:** The number of detected users is linear to the number of positive samples. For example, the train set with d-10 has the largest positive samples which is 9142 (the highest black histogram in Figure 6), and the number of detected users are 5059 (the highest red histogram in Figure 6) which is larger than the other groups. More positive samples can help GCN to learn more information about fraud users.

**Precision:** Most precision rates in Table 6 are nearly 0.95, reflecting the stability of DDF. However, precision of d-10 is the lowest of all and is just 0.87. By analyzing, even though more positive samples can give more information, rich information may mislead GCN to a certain extent.

Accordingly, seed selection seems important for adversarial fraudster detection when it is impossible to label a ground truth for each user of the train set.

**Attribute:** Attributes for seed selection should reflect obvious fraud behavior. For example, continuous days and a number of devices to publish comments are suspicious behaviors in-app review systems. Because a normal user would not write comments every day and change lots of devices to publish their comments according to our investigation, only if there exist some profits.

**Threshold:** Thresholds of attributes should filter out obvious fraud users. As listed in Table 6, the detected result of d = 10 has a lower comparing with d = 20 and d = 50. This indicates that for attribute, thresholds 20 and 50 can recognize fraud users more accurately than a threshold 10.

### Table 7. Detected users of baseline methods with different thresholds.

| Threshold : θ | LR | RF | Deepwalk | Line |
|--------------|----|----|----------|------|
| d-10         | 7566 | 7564 | 4747    | 4091 |
| d-20         | 3800 | 3758 | 4014    | 3343 |
| d-50         | 284  | 278 | 998     | 472  |
| c-5          | 3074 | 3074 | 3696    | 2771 |
| c-7          | 1820 | 1821 | 2758    | 1765 |
| c,d-5,10     | 3001 | 2991 | 3644 | 2726 |
| c,d-7,10     | 1804 | 1797 | 2717 | 1753 |
| c,d-5,20     | 2655 | 2801 | 3504 | 2558 |
| c,d-7,20     | 1792 | 1728 | 2719 | 1726 |

### Table 8. Time costs of training and predicting procedures with different thresholds.

| Threshold : θ | Training time (minutes) | Prediction time (minutes) |
|--------------|--------------------------|---------------------------|
| d-10         | 233.08                   | 13.77                     |
| d-20         | 212.03                   | 14.93                     |
| d-50         | 230.95                   | 14.9                     |
| c-5          | 230.88                   | 12.22                     |
| c-7          | 231.38                   | 14.8                     |
| c,d-5,10     | 222.07                   | 12.22                     |
| c,d-7,10     | 228.75                   | 13.67                     |
| c,d-5,20     | 237.37                   | 14.4                     |
| c,d-7,20     | 226.37                   | 15.55                     |

**Performance comparisons**

Detected users of baseline methods are listed in Table 7. Performance comparisons of DDF and baseline methods are shown in Figure 7. It can be seen from 7 that recall of DDF (the red line) remains stable under different thresholds. Comparatively, recall rates of LR and RF are high in (d-10) but reduce a lot under other thresholds such as (d-50) and (c, d-7,20). This indicates the uncertain detection performance of LR and RF. What’s more, recall rates of DeepWalk and LINE remain much more stable compared with that of LR and RF, but their number of detected users are less than DDF. In conclusion, DDF can find more suspicious users from unlabeled users and has stable performance.

FdGars14 is our previous work to apply GCN on fraud detection. In that work, we mainly focus on network construction without considering the influence of seed selection. To control the complexity of our experiments, we just choose one group of thresholds to select fraud users as our initial seeds. In reality, seed selection is the key point of fraud detection because labeling fraud users is a difficult problem. Therefore, in this work, we design several groups of thresholds to test the stability and efficiency of the methods. The detection results demonstrate the good performance of our presented system.
Efficiency is an important factor in industry tasks. Table 8 lists the time costs in training and predicting process. The iteration number of GCN is set to 500 and we only use CPU to train a GCN model. We can see that the time costs are all in a tolerable range. As mentioned in Section 3.5, the detection system is deployed on the Tencent Venus Computation Platform and provided as a fraud detection service on the Tencent Beacon Platform.

Conclusion

In this paper, we study adversarial fraudsters in online app review platform and categorize them into three types by their motivations. To tackle the problem of finding new abnormal users in a large-scale network, we present DDF (Detect, Defense, and Forecast) system by combining content, behavior, and temporal action features and build a GCN model to capture structural information between users. After that, we evaluate the DDF system by utilizing real-world review data set and comparing the detection results with baseline methods. Finally, we demonstrate its good performance on adversarial fraudster detection and provide this system on the Tencent Beacon Platform.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Chunming Wu https://orcid.org/0000-0002-5184-409X

References

1. Hooi B, Song HA, Beutel A, et al. Fraudar: bounding graph fraud in the face of camouflage. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA, 13–17 August 2016, pp.895–904. New York: ACM.
2. Ma J, Zhang D, Wang Y, et al. GraphRAD: a graph-based risky account detection system. In: Proceedings of ACM SIGKDD conference, London, UK, August 2018, p. 9. United Kingdom: ACM.
3. Kaghazgaran P, Caverlee J and Squicciarini A. Combating crowdsourced review manipulators: a neighborhood-based approach. In: Proceedings of the Eleventh ACM international conference on web search and data mining, Marina Del Rey, CA, USA, 5–9 February 2018, pp.306–314. New York: ACM.
4. Yao Y, Viswanath B, Cryan J, et al. Automated crowd-turfing attacks and defenses in online review systems. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, Dallas, TX, USA, 30 October–3 November 2017, pp.1143–1158. New York: ACM.
5. Kipf TN and Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th international conference on learning representations (ICLR-17), 2016. Toulon, France.
6. Zheng H, Xue M, Lu H, et al. Smoke screen or straight shooter: detecting elite sybil attacks in user-review social networks. In: Proceedings of the network and distributed system security symposium 2018, San Diego, CA, USA, 18–21 February 2018. San Diego: ISOC.
7. Peng Q and Zhong M. 2014. Detecting spam review through sentiment analysis. JSW 2014; 9(8): 2065–2072.
8. Li H, Fei G, Wang S, et al. Bimodal distribution and co-bursting in review spam detection. In: Proceedings of the 26th international conference on World Wide Web, Perth, Australia, 3–7 April 2017, pp.1063–1072. New York: ACM.
9. Pandit S, Chau DH, Wang S, et al. Netprobe: a fast and scalable system for fraud detection in online auction networks. In: Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, 8–12 May 2007, pp.201–210. New York: ACM.
10. Rayana S and 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, Sydney, NSW, Australia, 10 August 2015, pp.985–994. New York: ACM.

11. Shah N, Beutel A, Gallagher B, et al. Spotting suspicious link behavior with fBox: an adversarial perspective. In: 2014 IEEE international conference on data mining, Shenzhen, China, 14–17 December 2014, pp.959–964. IEEE.

12. Wang G, Xie S, Bing Liu B, et al. Review graph based online store review spammer detection. In: 2011 IEEE 11th international conference on data mining, Vancouver, BC, Canada, 11–14 December 2011, pp.1242–1247. IEEE.

13. Liu Z, Chen C, Yang X, et al. Heterogeneous graph neural networks for malicious account detection. In: Proceedings of the 27th ACM international conference on information and knowledge management, Torino, Italy, 22–26 October 2018, pp.2077–2085. New York: ACM.

14. 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, San Francisco, CA, USA, 13–17 May 2019, pp.310–316. New York: ACM.

15. Charikar MS. Similarity estimation techniques from rounding algorithms. In: Proceedings of the thirty-fourth ACM symposium on theory of computing, Montreal, Quebec, Canada, 19–21 May 2002, pp.380–388. New York: ACM.