FdGars: Fraudster Detection via Graph Convolutional Networks in Online App Review System

Jianyu Wang*
xiaoyu_paopao@zju.edu.cn
College of Computer Science,
Zhejiang University
Hangzhou, China

Rui Wen*
rachelwen@tencent.com
Platform and Content Group, Tencent
Shenzhen, China

Chunming Wu
wuchunming@zju.edu.cn
College of Computer Science,
Zhejiang University
Hangzhou, China

Yu Huang
neilyuhuang@tencent.com
Platform and Content Group, Tencent
Shenzhen, China

Jian Xiong
janexiong@tencent.com
Platform and Content Group, Tencent
Shenzhen, China

ABSTRACT

Online review system enables users to submit reviews about the products. However, the openness of Internet and monetary rewards for crowdsourcing tasks stimulates a large number of fraudulent users to write fake reviews and post advertisements to interfere the rank of apps. Existing methods for detecting spam reviews have been successful but they usually aims at e-commerce (e.g. Amazon, eBay) and recommendation (e.g. Yelp, Dianping) systems. Since the behaviors of fraudulent users are complex and varying across different review platforms, existing methods are not suitable for fraudster detection in the online app review system.

To shed light on this question, we are among the first to analyze the intentions of fraudulent users from different review platforms and categorize them by utilizing characteristics of content (similarity, special symbols) and behaviors (timestamps, device, login status). With a comprehensive analysis of spamming activities and relationships between normal and malicious users, we design and present FdGars, the first graph convolutional network approach for fraudster detection in the online app review system. Then we evaluate FdGars on a real-world large-scale dataset (with 82,542 nodes and 42,433,134 edges) from Tencent App Store. The result demonstrates that F_1-score of FdGars can achieve 0.938+, which outperforms several baselines and state-of-the-art fraudsters detecting methods. Moreover, we deploy FdGars on Tencent Beacon Anti-Fraud Platform to show its effectiveness and scalability. To the best of our knowledge, this is the first work to use graph convolutional networks for fraudster detection in the large-scale online app review system. It is worth to mention that FdGars can uncover malicious accounts even the data are lack of labels in anti-spam tasks.

*Both authors contributed equally to this research. The student author finished this work during intern.

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316586

CCS CONCEPTS

• **Information systems** → **Web indexing**; **Spam detection**; • **Hardware** → Dynamic memory.

KEYWORDS

Fraud Detection, Graph Convolutional Networks, Online App Review System

ACM Reference Format:
Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xiong. 2019. FdGars: Fraudster Detection via Graph Convolutional Networks in Online App Review System. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308560.3316586

1 INTRODUCTION

Online review systems provide services for people to share their opinions and make decisions, such as which clothes to buy and where to eat. It is usually regarded as a type of explicit feedback signal for products. Since users can generate content containing rich information, online review system has become an attractive target for fraudsters.

Detecting fake reviews and reviewers is a non-trivial problem. Prior works [11, 13, 14, 17, 19, 20] attempt to solve this problem by using language models, analyzing abnormal behaviors and building graphs to discover suspicious patterns. However, they mostly focus on detecting spammers in e-commerce platform. Few research has been done to detect multi-class fraudsters in different review systems. Figure 1 shows the different review patterns between Yelp and Myapp Store. Fraudsters in Yelp aim to influence consumers' decisions by writing masses of reviews with high quality, always camouflage like normal reviewers. In contrast, fraudsters in application stores, also called spammers, aim to post advertisements (e.g. telephone number, Wechat, URL) to promote the rank of apps (e.g. flushing, crowdturfing) and so on.

In this paper, we investigate the characteristics of spam reviews from different platforms, including online marketplace (eBay, Amazon), app store (Myapp, Google Play) and content service (Yelp,

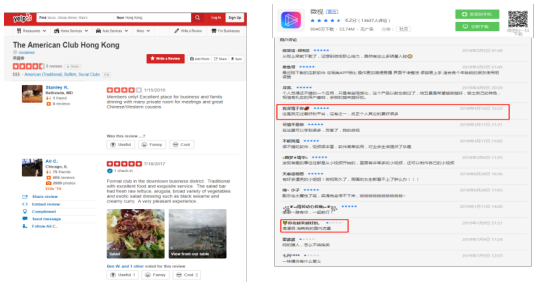


Figure 1: Comparison Between Different Review Platform.

Dianping). In addition, we also categorize fraudsters by their motivations such as spreading fake information, promoting product ranking and advertising into 3 types:

- Camouflage [6]: fraudsters pretend themselves as normal reviewers by adding links to popular items or famous people;
- Crowdturfing [8]: fraudsters can easily hire web workers to take part in particular spam activities with monetary award from crowdsourcing platforms (e.g. Rapidworkers, Microworkers).
- Spammer: fraudsters publish irrelevant reviews in open review systems to increase popularity of their products (e.g. advertisements, product information) or do some illegal activities (e.g. selling drugs, sensitive words).

As fraudsters in review platforms are adversarial, irregular and distributed, anti-spam tasks face a huge challenge. In the field of fraud detection, many works focus on the characteristics of content. They develop text or Nature Language Processing (NLP) models to distinguish fake reviews from legitimate reviews [13]. Since users' behavioral attributes (e.g. timestamps, footprints) can yield clues as to which are fraudulent, behavior-based approaches also attract many researchers and are widely applied in many industrial works [11]. Recently, graph-based detection methods [1, 2, 5, 6, 19, 20, 20] have been studied with inspiration from network embedding approaches, which leverage the relationship between users, reviews and items, and made considerable progress in spotting malicious accounts.

In order to detect fraudsters in large-scale app stores, we propose a method named FdGars by combining reviewers' features and Graph Convolutional Network (GCN). Firstly, we analyze review logs and extract content and behavior features for each users. Secondly, a graph structure is constructed to express the characteristics of reviewers and relationships between reviewers. Then, reviewers are classified into fraudsters and normal users through a predefined labeling method. Based on the limited labeled reviewers, a two-layer GCN is developed to detect more fraudsters from unlabeled reviewers.

Through experiments, we evaluate FdGars over the sample of 82,542 reviewers and 302,097 reviews. As shown in Figure 2, the constructed graph includes 82,542 nodes and 42,433,134 edges. We label the reviewers on 1 Aug, 2018 as training set and detect more fraudsters from 2 Aug, 2018 and 3 Aug, 2018. Finally, FdGars's recall reaches to 0.958+. To make the result more convincing, LR, RF and DeepWalk are used and their recalls are 0.516+, 0.828+ and 0.911+ respectively, demonstrating the effectiveness of the presented method.

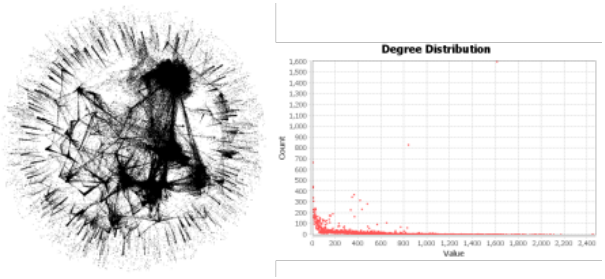


Figure 2: User graph in online App review platform.

We summarize the contributions of this work as follows:

1. *Fraudsters Analysis*. We shed light on detecting different types of fraudster in online review systems and conclude 3 popular patterns of generating fake/spam reviews by fraudsters.
2. *FdGars Implementation*. We develop an efficient and scalable methods system by using graph convolutional networks for anti-spam tasks in a large-scale online app review system. FdGars performs better than other state-of-the-art methods for detecting malicious accounts.
3. *Deployment and Evaluation*. We validate the performance of FdGars in a real word dataset by deploying it on the Tencent's Beacon Anti-Fraud Platform. Our method achieves both high precision and high recall.

2 METHODOLOGY

In this section, we first give an overview of FdGars, followed by the fraudster measurement and GCN.

2.1 FdGars Framework

The procedure of FdGars is illustrated in Figure 3.

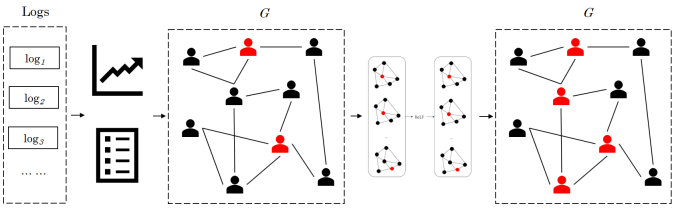


Figure 3: Diagram of FdGars Framework.

First of all, we extract content features and behavior features for each reviewer based on their review logs. Content feature concentrates on a reviewer's text content, such as a review's symbol ratio and the quantity of similar reviews. Behavior feature focuses on a reviewer's behavior in a designated period, including the review quantity, the 24-hours distribution of review quantity and the score distribution.

Second, we construct a graph structure G according to the following rules:

1. represent reviewers as nodes in graph G;
2. construct an edge to connect two nodes if their corresponding reviewers have reviewed the same app.

Consequently, the review logs are transformed into a graph structure. The graph G can be expressed as

$$G = \{\mathbf{N}, \mathbf{E}, \mathbf{A}\} \tag{1}$$

Where, $\mathbf{N} = \{n_1, n_2, n_3, ...\}$ is the reviewer set, $\mathbf{E} = \{e_1, e_2, e_3, ...\}$ is the edge set and $\mathbf{A} = \{a_1, a_2, a_3, ...\}$ is the reviewer attribute set. An edge between nodes n_i and n_j indicates that reviewers n_i and n_j have reviewed the same app, denoted as

$$e_{ij} = Edge(n_i, n_j) = (n_i, a_i, n_j, a_j) \tag{2}$$

After the construction, graph G clearly expresses the characteristics of reviewers and relationships between reviewers.

Third, we utilize some rules to label high suspicious fraudsters. As described in Section 2.2, reviewers are divided into fraudsters and normal users. We design a labeling method to label high suspicious fraudsters and normal users. According to our data analysis, we believe the labeling method can recognize high suspicious fraudsters to some extent.

Then, we train a GCN model to learn node feature and graph structure according to graph G and a small account of labeled reviewers. The GCN's detail is introduced in Section 2.3. When the training is finished, we utilize the learned model to find more fraudsters from unlabeled reviewers in app store. Experiments demonstrate the performance of the proposed FdGars method.

2.2 Fraudsters Measurement

In reality, accessing fraudsters is a hard work since malicious accounts always change their strategies to avoid detection systems. Traditional methods for labeling fraudsters usually relay on researcher opinion or a team of human labors, but they have obvious limitations by just utilizing single characteristics such as language feature, posting times and the frequency of posting reviews to make evaluations. In contrast to previous efforts, we aims to collect fraudsters as seeds who have strong evidence to anti-spam tasks at first by leveraging reviews' content features and behavior distributions. Then we propagate these seeds to find more malicious accounts, who have strong connections with seed users.

Review Classification. Firstly, we use content features to classify the reviews. Table 1 shows the features we extract from each review.

Table 1: Content Features and Description

Feature	Description
SRN	Similar Review Number
RSN	Review Symbol Number
RL	Review Length
PRR	Proportion of RSN to RL
REB	Regular Expression Blacklist

- **SRN** is the number of similar reviews in a designate period, which calculated by *SimHash* algorithm ;
- **RSN** is the number of special character in each review, such as emoji expression and Mars symbols. In general, spam review contains more than 50% symbols;
- **RL** is the length of each review;
- **PRR** is the ratio of symbol number and review length;
- **REB** is the collection of regular expression for reviews posting by spammers with obvious intentions including telephone number ($\hat{}1(3|4|5|7|8)\backslash d\{9\}\$$), ambiguous word ($[WwVv][Xx]$), URLs and so on.

Reviewer Analysis. Secondly, we classify reviewers by utilizing two behavior attributes: the continuous days and the number of login device.

- **CD** is the continuous days that reviewers have posted reviews in a designated period;
- **DN** records the device quantity that reviewers have used in the same period.

We design a labeling function expressing as

$$Label(n) = \begin{cases} 1 & if\ CD > \theta_{CD}\ \&\ DN > \theta_{DN} \\ 0 & otherwise \end{cases} \tag{3}$$

where, θ_{CD} and θ_{DN} are thresholds. If an reviewer's CD > θ_{CD} or DN > θ_{DN}, we consider the reviewer as a fraudster; otherwise, the reviewer is a normal user.

Through the multiple rounds of data analysis, we believe $\theta_{CD} = 7$ and $\theta_{DN} = 20$ can recognize high suspicious fraudsters in the scenario of online app store. Table 2 lists the features we extract for each reviewer:

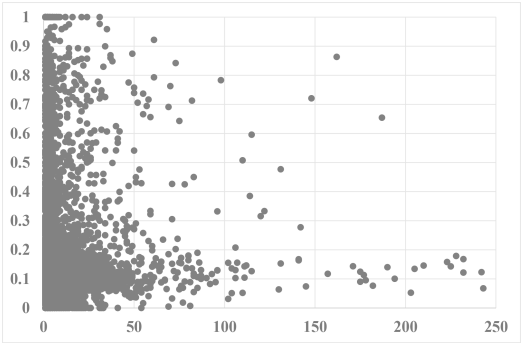
Table 2: Behavior Features and Descriptions

Feature	Description	Latitude
RQ	Review Quantity	1
TQD	Time-based Quantity Distribution	24
SQD	Score-based Quantity Distribution	5

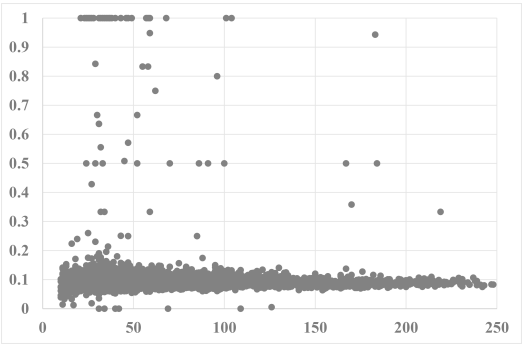
- **RQ** records the quantity of an reviewer's reviews in a designated period.
- **TQD** is the 24-hours quantity distribution of an reviewer's reviews.
- **SQD** is an reviewer's rating distribution.

By labeling reviewers based on Equation 3, Figure 4 shows the fraudster and normal user distributions with RQ and PRR. Obviously, normal users gather at the bottom-left of the coordinate. Based on our analysis, The RQ of most normal users are less than 30. Comparatively, fraudsters gather at the bottom of the coordinate. Most fraudsters' PRRs are less than 0.2. Consequently, $\theta_{CD} = 7$ and $\theta_{DN} = 20$ can be used to recognize high suspicious fraudsters and label reviewers in the following experiment.

Fraudsters Identification. Since our data set is not labeled by human labors, we use review classification results to make sure that our ground truth is necessarily an approximation. We evaluate each reviewers by considering the number of spam reviews he/she has posted during a month. In this way, we can identify fraudsters with high confidence even their reviews or behaviors look normal.



(a) Normal Users



(b) Fraudsters

Figure 4: Distribution comparison between Fraudsters and Normal Users with RQ and PRR features. (Note: X-axis represents RQ and Y-axis represents PRR.)

2.3 Graph Convolutional Networks

We use a two-layer GCN for semi-supervised fraudster detection in this paper. Take a graph G in Figure 5 as an example, nodes in G represent 6 reviewers and edges represent their behavioral relationships.

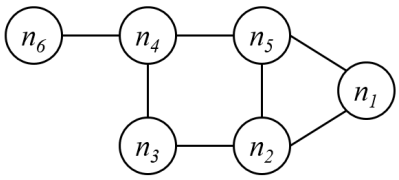


Figure 5: A Simple Example of Graph G.

First, we calculate a symmetric adjacency matrix A on graph G. According to Figure 5, the result of A is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Second, we get a matrix X to express reviewers' feature vectors which are described in Section 2.2. Then, we define a two-layer GCN as the following layer-wise propagation rule:

$$H^{(0)} = X$$

$$H^{(1)} = \sigma[\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(0)}W^{(0)}]$$

$$H^{(2)} = \sigma[\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(1)}W^{(1)}]$$

Where, $H^{(1)}$ is the matrix of activations in the $1st$ layer and $H^{(2)}$ is the matrix of activations in the $2nd$ layer. $\widetilde{A} = A + I_N$ is the adjacency matrix of the undirected graph with added self-connections. I_N is the identity matrix. D is defined as

$$D = \begin{bmatrix} \sum_{j=0}^{N-1}\widetilde{A}_{0,j} & 0 & \dots & 0 \\ 1 & \sum_{j=0}^{N-1}\widetilde{A}_{1,j} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & \sum_{j=0}^{N-1}\widetilde{A}_{N-1,j} \end{bmatrix}$$

$W^{(0)}$ is an input to hidden weight matrix for a hidden layer with H feature maps. $W^{(1)}$ is a hidden to output weight matrix. Both of them are layer-specific trainable weight matrices. $\sigma(\cdot)$ denotes an activation function, such as $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$.

Finally, we train the model $f(X, A)$ according to the steps described above and predicted results will be the outputs of $f(X, A)$, expressing as

$$f(X, A) = \mathrm{softmax}\Big(\hat{A}\,\mathrm{ReLU}\Big(\hat{A}XW^{(0)}\Big)W^{(1)}\Big)$$

Here, the softmax activation function is defined as

$$\mathrm{softmax} = \frac{1}{\mathcal{Z}}exp(x_i)$$

where $\mathcal{Z} = \sum_i \exp(x_i)$.

In conclusion, the two-layer GCN is schematically depicted in Figure 6. It is constructed based on a small amount of labeled reviewers as Section 2.2 described. After training, the learned model can detect more fraudsters who have similar behaviors with the labeled fraudsters from unlabeled reviewers.

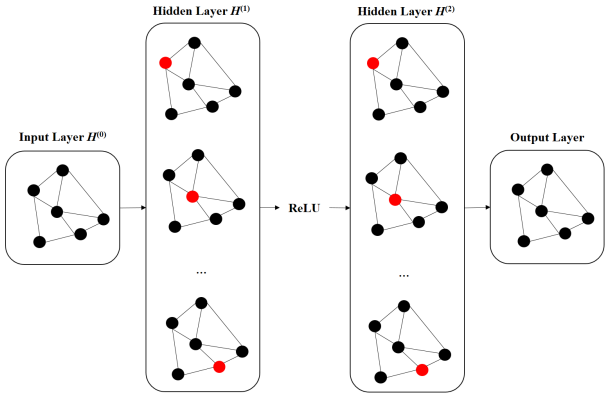


Figure 6: Graph Convolutional Network.

3 EXPERIMENTS

In this section, the presented detection method is evaluated on real world big data and deployed on Tencent's Venus Computation Platform.

3.1 Data Collection and Annotation

To validate the performance of the proposed method, large-scale review datasets are utilized from Tencent Inc. and its detail is listed in Table3. It can be seen that 82,542 users have posted 302,097 reviews from 1 Aug, 2018 to 3 Aug, 2018.

Table 3: Review Data Statistics

Date	Users	Apps	Reviews
1 Aug, 2018	31,450	4,374	94,378
2 Aug, 2018	32,540	4,328	105,958
3 Aug, 2018	35,104	4,559	101,761
All Three Days	82,542	7,584	302,097

We take the reviewers on Aug 1, 2018 as training set and the reviewers on 2 Aug and 3 Aug as testing set. To verify the diffusion performance of the presented fraudster detection, one in particular is that our testing set has excluded the reviewers who have posted reviews in training set. Table 4 shows the graph information based on our dataset. It is worth mentioning that the graph construction process costs nearly 2 minutes and the ratio of positive reviews is 38.1%.

Table 4: Graph Statistics

Nodes	Edges	Label	Quantity
82,542	42,433,134	1	5,926
		0	25,524

3.2 Baseline Methods

We compare the performance of FdGars with two widely-used methods for classification in many fields. We also add two famous graph structure based methods as our baseline methods.

- **Logistic Regression** According to the features listed in Table 2 and the labeling dataset mentioned in Table 3, we train a Logistic Regression (LR) classifier. Then, we use the classifier to divide the reviewers into fraudsters and normal users from the unlabeled reviewers.
- **Random Forest** Similarly, we train a Random Forest (RF) classification model. The learned RF model is also used to find fraudsters from the unlabeled reviewers.
- **DeepWalk** We train a DeepWalk [16] model to learn latent representations of nodes in our graph. In our experiment, the number of random walks is set to 10; the length of random walk is 80; the embedding length is 128. At last, we train a RF model to test DeepWalk's performance.
- **LINE** We train a LINE [18] model to learn latent representations of nodes in our graph. The negative ratio is set to 5 and the embedding length is 128. At last, we train a RF model to test LINE's performance.

3.3 Performance Evaluation

Table 5 lists the training and predicting time costs. It is worth mentioning that the training and predicting procedures are executed on CPU. According to our statistics, the predicting time lasts nearly 14 minutes.

Table 5: Time Costs of Training and Predicting Procedure

Epochs	Training Time	Predicting Time
100	39.37 minutes	13.65 minutes
500	150.58 minutes	12.83 minutes

Table 6 shows the predicted results with different epochs, as are True Positive (TP), False Positive (FP), True Negative (TN) and False Positive (FP). To measure the performance of the proposed method, Table 7 shows the details of metrics, as are $Accuracy$, $Recall$, $Precision$ and F_1-score.

Table 6: Evaluation Results with Different Epochs

Epochs	Date	TP	FP	TN	FN
100	2 Aug, 2018	1,335	205	23,520	572
	3 Aug, 2018	2,444	347	24,849	601
500	2 Aug, 2018	1,776	327	23,398	131
	3 Aug, 2018	2,913	587	24,609	132

Table 7: Performance Statistics with Different Epochs

Epochs	Date	$Recall$	$Precision$	F_1
100	2 Aug, 2018	0.7001	0.8669	0.7746
	3 Aug, 2018	0.8026	0.8757	0.8376
500	2 Aug, 2018	0.9588	0.9195	0.938
	3 Aug, 2018	0.9667	0.9135	0.9379

Table 8 presents the performances of fraudster detection with different methods. Obviously, LR and RF have lower recalls comparing with other methods. It is noteworthy that DeepWalk takes 170.17 minutes to calculate each node's embedding and LINE spends almost 116 hours. Comparatively, FdGars only takes 150.58 minutes as expressed in Table 5. This indicates that FdGars has a better computational efficiency in real applications.

Figure 7 shows the ROC curves with different datasets. We also test FdGars on real world review logs from 1 Sep, 2018 to 3, Sep 2018. By labeling reviewers in 1 Sep, 2018 as training set, we calculate the detection results from new reviewers in 2 Sep, 2018 and 3 Sep, 2018. The four curves indicate that FdGars can maintain a stable performance for fraudster detection.

Table 8: Performance Comparison with Baseline Methods

Method	Date	*Recall*	*Precision*	F_1
LR	2 Aug, 2018	0.5165	0.7949	0.6241
	3 Aug, 2018	0.6059	0.8094	0.6924
RF	2 Aug, 2018	0.8285	0.9695	0.8927
	3 Aug, 2018	0.8716	0.9687	0.917
DeepWalk	2 Aug, 2018	0.9114	0.8627	0.8862
	3 Aug, 2018	0.9363	0.8702	0.902
LINE	2 Aug, 2018	0.8138	0.875	0.8432
	3 Aug, 2018	0.8598	0.8761	0.8677
FdGars	2 Aug, 2018	**0.9588**	**0.9195**	**0.938**
	3 Aug, 2018	**0.9667**	**0.9135**	**0.9379**

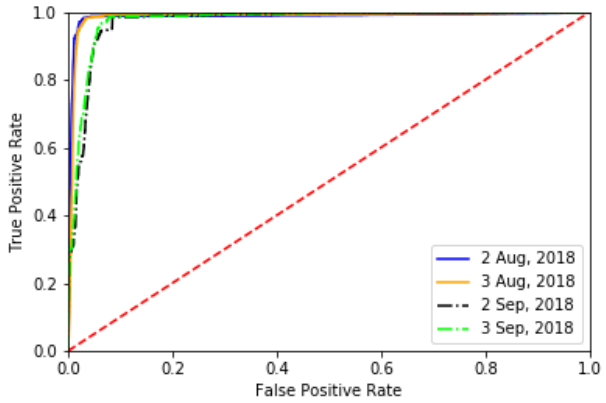


Figure 7: FdGars's ROC curves. Curves with different colors indicate the fraudster detection performance in different dates.

4 RELATED WORK

Fraudulent Users in Social Networks. Social spams became one of most popular forms due to the openness of the Internet. Fraudsters post fake/spam reviews, spread false information on blogs, forums and social media with the monetary rewards. [3, 7, 22] investigate the rumor problems, they find there exists lots of misinformation in micro-blogging and twitter platform. [3] shows that social fraudsters can even potentially alter affect the outcome of political elections.[15] analyzes the motivation of fraudsters in E-commerce (e.g. Amazon, Yelp, Alibaba), they report spammers often write fake reviews to promote their products or mislead consumers. [6] spots fraudsters in the presence of camouflage or hijacked accounts. [21] investigates content polluters, who establish links with normal users and blend the malicious information with legitimate content. Recently, fraudulent users in social networks are more adversarial, flexible and variable. A survey can be found in [10].

Graph-based Detection Methods Graph embedding methods [4, 16, 18] have been applied in lots of tasks, including node classification, link prediction, community detection, recommendation and risk control. Recently, many anomaly detection methods have focused on using graphs algorithms since it can represent and propagate the suspiciouness between objects [2]. [19] firstly captures inter-relationships among reviewers, reviews and stores based on graph model without using text information. [6] provides Fraudar for spotting fraudsters in the presence of camouflage or hijacked accounts. [23] presents NetWalK, which aims to detect structural anomalies for dynamic networks by learning network representation. [9, 17] design graph-based system (FraudEagle and FariJudge) for identifying untrustworthy users. For industrial applications, [12, 14] present graph embedding methods for detecting malicious accounts at Alipay platform.

5 CONCLUSION

In this paper, we present the first work to apply Graph Convolutional Networks on solving the problem of fraudsters detection in the online app review system. Specifically, we focus on the recall of new fraudsters. By analyzing the language styles, behaviors and relationships of reviewers, we find it is difficult to make a judgment just by utilizing single characteristic. So we propose a framework named FdGars, which combines text, behavior and relationship features of reviewers. Firstly, we extract content and behavior features for each reviewers. Secondly, we construct a graph by exploiting the relational nature of fraudsters and normal users. Thirdly, reviewers are labeled through a predefined labeling method. Based on the limited labeled reviewers, FdGards is developed to detect more fraudsters from unlabeled reviewers. Finally, we evaluate FdGars by leveraging the real-world review dataset from Tencent App Store. The results indicate that FdGars can achieve both high precision and recall. We also implement FdGars on Tencent Beacon Anti-Fraud Platform and demonstrate the effectiveness and scalability in practical scenarios. In summary, our research in this paper is expected to shed light on defending against fraudsters for large-scale online App review platforms.

ACKNOWLEDGMENTS

The review data used in this paper is courtesy of Tencent Inc. This work is supported by the National Key Research and Development Program of China (2016YFB0800102, 2016YFB0800201), the Key Research and Development Program of Zhejiang Province (2018C01088, 2018C03052), and the Major Scientific Project of Zhejiang Lab (2018FD0ZX01).

REFERENCES

[1] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. 2013. Opinion Fraud Detection in Online Reviews by Network Effects. *ICWSM* 13 (2013), 2–11.
[2] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery* 29, 3 (2015), 626–688.
[3] Alessandro Bessi and Emilio Ferrara. 2016. Social bots distort the 2016 US Presidential election online discussion. (2016).
[4] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.
[5] Bryan Hooi, Kijung Shin, Hyun Ah Song, Alex Beutel, Neil Shah, and Christos Faloutsos. 2017. Graph-based fraud detection in the face of camouflage. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11, 4 (2017), 44.
[6] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. 2016. Fraudar: Bounding graph fraud in the face of camouflage. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 895–904.

[7] Xia Hu, Jiliang Tang, Yanchao Zhang, and Huan Liu. 2013. Social Spammer Detection in Microblogging.. In *IJCAI*, Vol. 13. 2633–2639.

[8] Parisa Kaghazgaran, James Caverlee, and Anna Squicciarini. 2018. Combating Crowdsourced Review Manipulators: A Neighborhood-Based Approach. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 306–314.

[9] Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and VS Subrahmanian. 2018. Rev2: Fraudulent user prediction in rating platforms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 333–341.

[10] Srijan Kumar and Neil Shah. 2018. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559* (2018).

[11] Huayi Li, Geli Fei, Shuai Wang, Bing Liu, Weixiang Shao, Arjun Mukherjee, and Jidong Shao. 2017. Bimodal distribution and co-bursting in review spam detection. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1063–1072.

[12] Xiang Li, Wen Zhang, Jiuzhou Xi, and Hao Zhu. 2018. HGsuspector: Scalable Collective Fraud Detection in Heterogeneous Graphs. (2018).

[13] Yuming Lin, Tao Zhu, Xiaoling Wang, Jingwei Zhang, and Aoying Zhou. 2014. Towards online review spam detection. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 341–342.

[14] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. Heterogeneous Graph Neural Networks for Malicious Account Detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2077–2085.

[15] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie S Glance. 2013. What yelp fake review filter might be doing?. In *ICWSM*. 409–418.

[16] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[17] Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*. ACM, 985–994.

[18] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1067–1077.

[19] Guan Wang, Sihong Xie, Bing Liu, and S Yu Philip. 2011. Review graph based online store review spammer detection. In *Data mining (icdm), 2011 ieee 11th international conference on*. IEEE, 1242–1247.

[20] Guan Wang, Sihong Xie, Bing Liu, and Philip S Yu. 2012. Identify online store review spammers via social review graph. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 4 (2012), 61.

[21] Liang Wu, Xia Hu, Fred Morstatter, and Huan Liu. 2017. Detecting Camouflaged Content Polluters.. In *ICWSM*. 696–699.

[22] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on Sina Weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. ACM, 13.

[23] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. 2018. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2672–2681.