Uncovering Insurance Fraud Conspiracy with Network Learning

Chen Liang
Ant Financial
Hangzhou, China
lc155190@antfin.com

Ziqi Liu
Ant Financial
Hangzhou, China
ziqiliu@antfin.com

Bin Liu
Ant Financial
Hangzhou, China
lb88701@alibaba-inc.com

Jun Zhou
Ant Financial
Beijing, China
jun.zhoujun@antfin.com

Xiaolong Li
Ant Financial
Seattle, USA
xl.li@antfin.com

Shuang Yang
Ant Financial
San Francisco, USA
shuang.yang@antfin.com

Yuan Qi
Ant Financial
Hangzhou, China
yuan.qi@antfin.com

ABSTRACT

Fraudulent claim detection is one of the greatest challenges the insurance industry faces. Alibaba's return-freight insurance, providing return-shipping postage compensations over product return on the e-commerce platform, receives thousands of potentially fraudulent claims everyday. Such deliberate abuse of the insurance policy could lead to heavy financial losses. In order to detect and prevent fraudulent insurance claims, we developed a novel data-driven procedure to identify groups of organized fraudsters, one of the major contributions to financial losses, by learning network information.

In this paper, we introduce a device-sharing network among claimants, followed by developing an automated solution for fraud detection based on graph learning algorithms, to separate fraudsters from regular customers and uncover groups of organized fraudsters. This solution applied at Alibaba achieves more than 80% precision while covering 44% more suspicious accounts compared with a previously deployed rule-based classifier after human expert investigations. Our approach can easily and effectively generalizes to other types of insurance.

CCS CONCEPTS

• **Social and professional topics** → **Financial crime**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

fraud detection; graph learning; network learning; insurance fraud

ACM Reference Format:
Chen Liang, Ziqi Liu, Bin Liu, Jun Zhou, Xiaolong Li, Shuang Yang, and Yuan Qi. 2019. Uncovering Insurance Fraud Conspiracy with Network Learning. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331372

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331372

1 INTRODUCTION

What if you bought a shirt but found significant color difference between the on-screen product and the real-life product? What if you discovered a less expensive alternative after purchasing a laptop? Returning the item is likely to be the first choice when shopping online. However, returning an unused item can raise lots of disputes between buyers and sellers because of the ambiguities over which party should take responsibilities. Surprisingly, most disputes focus not on whether the undamaged item should be returned, but on who should pay for return shipping costs. It takes enormous efforts and a great deal of time to resolve such disputes, especially at Alibaba[1], a platform with millions of sellers and diverse return policies. To resolve disputes and protect buyers' right of regret, a new form of insurance has been created.

Return-freight insurance, designed to pay buyers the return shipping costs, has retained billions of dollars in revenue. However, the loss caused by fraudulent claims is non-trivial. Fraudsters receive shipping discounts from their partner express companies and file claims with the regular shipping price. According to the estimates of insurance experts at Alibaba, thousands of potentially fraudulent claims go undiscovered with the previous rule-based fraud detection system. Among these false claims, the most destructive ones are claimed by groups of organized fraudsters. The need for a more powerful and more flexible fraud detection solution is significant.

1.1 Our Fraud Detection Problem

Fraud detection in insurance claims can be viewed as a supervised binary classification problem. We classify insurance accounts into two categories: fraudulent and regular. Labels of accounts in the training set are obtained from a formerly deployed rule-based system with some, but not sufficient, confidence. We aim to discover

[1] https://www.alibaba.com

many more fraudulent accounts than the rule-based system while retaining high precision.

Networks provide straightforward information for describing and modeling complex relations among colluders (collaborating fraudsters). We build a device-sharing graph, a transaction graph, and a friendship graph to illustrate such relations, and apply two graph learning approaches, one based on node2vec [3] and another based on GeniePath [4], to mine such information. We conduct extensive experiments to compare these approaches and describe our complete fraud detection solution which implements the device-sharing graph and our workflow deployed at Alibaba.

1.2 Challenges in Fraud Detection

The challenges we face that hinder the performance of fraud detection systems include **concept drift**, **label uncertainty**, and **excessive human effort**.

Concept drift in fraud detection refers to the phenomenon that new types of fraud evolve over time and get more and more unpredictable. It's mainly caused by the use of non-stationary features in fraud detection systems. Non-stationary behaviors, such as the number of claims made in the past month, can be easily affected when fraudsters change their tactics. We address this problem by adding more stationary data. Relations between collaborating fraudsters are naturally stationary, e.g. device-sharing graphs.

Label uncertainty arises because of the usage of rule-generated labels. The formerly deployed rule-based fraud detection system outputs a risk tag for each account, say 'high risk' and 'no observable risk'. We are confident at 'high risk' accounts, but it is unclear that whether the 'no observable risk' accounts are at risk or not. In other words, the labels consist of a small amount of true positive labels and a large amount of unknown labels. To build training labels, we randomly undersample samples from the 'no observable risk' class, which is explained in the Data Preparation section.

Excessive human effort comes from the labeling tasks and evaluation tasks in traditional insurance fraud detection settings. As we focuses on automated risk control that with negligible human effort, our approach requires no human interventions besides a periodical evaluation (weekly or monthly) conducted by insurance professionals that samples and examines the classification results for loss estimation.

2 RELATED WORK

Insurance fraud detection approaches can be generally divided into supervised learning, unsupervised learning, and a mixture of both. Popular supervised algorithms, such as Bayesian networks and decision trees have been applied or combined in [5, 6]. Unsupervised approaches, such as cluster analysis and outlier detection have also been applied [1, 7]. Hybrids of supervised and unsupervised algorithms have been studied, and unsupervised approaches have been used to segment insurance data into clusters for supervised approaches in [2].

Our two approaches fall under supervised learning and hybrids of both, respectively. Our approaches differ, as they represent data with graphs, which are one of the most natural representations of data and allow for complex analysis without simplification of data.

Table 1: Graphs for Comparison

Graph	|V|	|E|	nodes	edges
device-sharing	3 M	6 M	account / UMID	device usage
transaction	2 M	2 M	account	fund exchange
friendship	8 M	11 M	account	friendship

3 GRAPH CONSTRUCTION

To address concept drift as well as to uncover organized fraudsters, we resort to the power of graphs that help reveal strong relations of accounts. In this section, we construct and compare different types of graphs including a device-sharing graph, a transaction graph, and a friendship graph. We explain which graphs fit our needs, and apply the device-sharing graph in our final fraud detection solution.

The following properties of graphs can help separate fraudulent from regular:

(1) distance aggregation: closer nodes have similar labels;
(2) structural differentiation: structures of organized fraudsters are different from structures of regular accounts.

3.1 Three Graphs

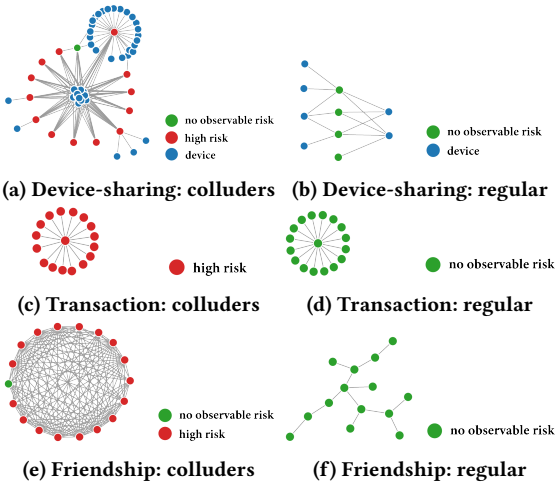
The device-sharing graph reveals the relation of accounts sharing a device. A vertex is either a device (User Machine ID, UMID[2]) or an account. Edges only exist between a device vertex and a UMID vertex, indicating log-in activities in the history. The transaction graph shows fund exchange relations between accounts. A vertex is an account, and an edge indicates the existence of established transactions between accounts. The friendship graph is built upon friendship at Alipay, a product of Ant Financial with social networking features. We preprocess these graphs to remove singleton accounts.

3.2 Graph Comparison

Typical subgraphs of organized fraudsters and regular users are visualized in Figure 1. Colluders are organized in ways that are contrasting with regular customers' as exhibited by the device-sharing graph and the friendship graph. Accounts with high risk tend to share the same group of devices and transfer funds with each other. Such patterns imply that a group of fraudsters works together to conduct frauds and split profits. The transaction graph fails to show such properties.

Besides, a proper graph needs to distinguish fraudulent accounts from regular accounts. Non-stationary features revealing online behavior patterns are selected as account node features. We assume a group of fraudsters share similar behaviors. As graph neural network methods aggregate information from the neighborhood, a graph constructed with closer nodes sharing similar labels makes the classification problem easier. We measure the ability to aggregate fraudulent accounts with node distribution with respect to the distance from fraudulent nodes. The distribution is shown in Figure 2. Fraudulent accounts gather around each other in the device-sharing graph, implying that it is more appropriate for the account classification task.

[2]Device fingerprints to uniquely identify devices.



(a) Device-sharing: colluders (b) Device-sharing: regular

(c) Transaction: colluders (d) Transaction: regular

(e) Friendship: colluders (f) Friendship: regular

Figure 1: Visualization for typical colluders and regular users in device-sharing graph, transaction graph, and friendship graph.

4 GRAPH LEARNING APPROACH

Node embedding approaches and graph neural network approaches are two major techniques to understand graph information. Here we present an inductive graph neural network algorithm for the insurance fraud detection problem. A node embedding approach is briefly introduced in Section 5.2.

4.1 Graph Neural Networks (GNNs)

GNNs are a set of deep learning architectures that aggregate information from nodes' neighbors using neural networks. A deeper layer in neural networks aggregates more distant neighbors, and the t-th layer embedding of node u is

$$h_u^{(t)} = \sigma(W_t \cdot \text{AGG}(h_v^{(t-1)}, \forall v \in \mathcal{N}(u) \cup \{u\}))$$

where the initial embedding $h_u^{(0)} = x_u \in \mathbb{R}^P$ is the account feature, $h_u^{(t)} \in \mathbb{R}^K$ denotes the intermediate embedding at t-th layer, σ is the activation non-linear function, and AGG$(\cdot)$ is an aggregation function over neighbors that differs in GNN algorithms [4].

The GNN approach we use is based on GeniePath [4], that simply stacks adaptive path layers to aggregate each node's neighborhood based on breadth and depth exploration in the graph. For breadth exploration, it iteratively aggregates neighbors for T times:

$$h_u^{(t+1)} = \tanh\left(W_t \sum_{v \in \mathcal{N}(u) \cup \{u\}} \text{softmax}_v(\mu^\top \tanh(W_s h_u^{(t)} + W_d h_v^{(t)})) \cdot h_v^{(t)}\right)$$

This breadth-search function learns the importance of neighbors with pairwise account feature patterns. Given those hidden units $(h_u^{(0)}, h_u^{(1)}, ..., h_u^{(T)})$ at various depths, a depth-search function $h_u = \text{LSTM}(h_u^{(0)}, h_u^{(1)}, ..., h_u^{(T)}; \phi)$ is added to further extract and filter the signals. The resulting embeddings h_u's are fed to the final softmax or sigmoid layers for downstream fraud account classification tasks.

4.2 Optimization with Label Uncertainty

Given the final embedding h_u's, we have to optimize parameters $\theta := \{W_t, W_s, W_d, \mu, \phi\}$. The labels used for classification are based on 'risk tags' generated by a rule-based account risk indicator. We treat 'high risk' accounts as fraudulent, and 'no observable risk' accounts as regular.

However, the dataset suffers from label uncertainty - the rule-based risk indicator is much more confident about 'high risk' accounts being fraudulent than about 'no observable risk' accounts being regular. To address this problem, 'regular' accounts are sampled randomly to reduce the uncertainty of classifying a 'no observable risk' account as fraudulent, as shown in the modified objective function:

$$\mathcal{L}(\theta) = \min_\theta \Big(\sum_{v \in \mathcal{V}_{\text{fraudulent}}} \ell(\text{GeniePath}(x_v; \theta), \text{fraudulent}) + \sum_{v' \in \text{sample}(\mathcal{V}_{\text{regular}})} \ell(\text{GeniePath}(x_{v'}; \theta), \text{regular}) \Big)$$

Our goal is to minimize the losses caused by wrong classifications. The chance of punishment of a false positive is controlled by the downsampling rate in terms of the new objective function.

5 EXPERIMENTS

We compare three approaches for fraud detection, two of which use graph learning algorithms, while the baseline uses account-level features only.

5.1 Data Preparation

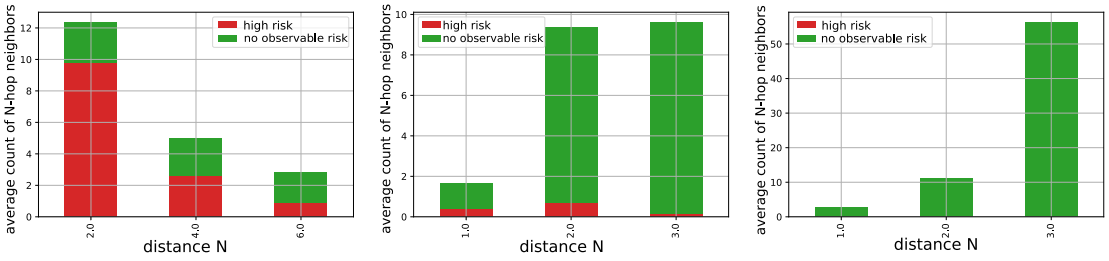
Each graph we constructed contains accounts that have filed a claim within a 30-day period. Device UMIDs used by these accounts in the past 40 days are also added as graph nodes. Edges are established between account nodes and UMID nodes with login relations. Isolated subgraphs, which contain only one account node, are removed to reduce computational effort. For initial features of each account node, we collect 50 features (e.g., number of claims submitted over a month, duration as a customer, etc.), derived from insurance claim history, shipping history, and shopping history. The resulting graph contains around three million nodes and six million edges (see Table 1).

5.2 Comparison Methods

We evaluate our GNN-based graph learning approach against a gradient boosted decision tree (GBDT) classifier, and against a node embedding approach. For all approaches, we calculate the probability of being at risk for each account in the test dataset, and then compare the F1 score.

5.2.1 The GBDT approach. The GBDT classifier uses account features as inputs without any graph structural information.

5.2.2 The Node Embedding Approach. node2vec [3] assigns a low-dimensional vector to represent a graph node. It is unsupervised and only uses graph structural information. Node2vec-generated embeddings are concatenated with account features and fed to downstream classification tasks using a GBDT [8].



(a) Device-sharing (b) Transaction (c) Friendship

Figure 2: Average number of N-hop neighbors around fraudulent accounts.

Table 2: Results based on Rule-based Labels.

	GBDT	Node Embedding	GNNs
F1	0.547	0.535	0.623
DE	1.47	1.44	1.44

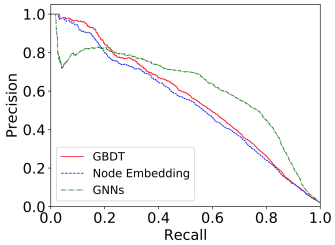


Figure 3: Model comparison with the Precision-Recall curve.

5.3 Experimental Setups

We set the same hyperparameters for all GBDT modules: 500 trees, max tree depth of 5, data sampling rate of 0.6, feature sampling rate of 0.7, and a learning rate of 0.009. We randomly sample 25% of 'no observable risk' accounts as negative samples.

5.4 Results and Discussion

Our results, summarized in Table 2 and plotted in Figure 3, show that the GNNs-based approach outperforms the others. Detection expansion (DE), defined as $\frac{FP+TP+FN}{TP+FN}$, indicates the ability to detect more fraudulent accounts. All of our approaches raise the coverage of fraudulent account detection by more than 40% while GNNs-based approach has higher precision and recall at most time.

The GBDT approach is slightly better than the node embedding one. This result implies that embeddings learned solely from graph information are not as good as account features. We find out the most valuable features come from shopping history - if a user has spent a lot over the past year, we are confident he/she is not a fraudster.

6 APPLICATION

The fraudulent claim detection system collects accounts that have filed a claim over the past months and classifies them in a batch mode that updates daily. The classification result is evaluated monthly by an insurance professional, who randomly samples and examines 300 accounts out of the reported fraudulent accounts. Daily human intervention is not necessary and human effort is enormously saved. The most recent reports show that we have achieved precision of over 80% while covering 44% more suspicious accounts compared with the former rule-based classifier. The estimated savings are over 10 thousand dollars per month.

7 CONCLUSION

This paper proposes a device-sharing graph and graph learning-based approaches to address the fraud detection problem. It is the first paper in the literature that introduces a real-world insurance fraud detection system utilizing the strong expressiveness of graphs. Graphs have proved their power in multiple online insurance areas.

We illustrate three types of graphs and show their advantages in separating fraudulent and regular using graph neural networks. We propose optimization algorithms for GNNs with only positive and unlabeled data. With proper graphs, features, and algorithms, we have achieved precision of over 80% and covered 44% more suspicious accounts in return-freight insurance fraud detection with automated solutions.

REFERENCES

[1] Patrick L Brockett, Richard A Derrig, Linda L Golden, Arnold Levine, and Mark Alpert. 2002. Fraud classification using principal component analysis of RIDITs. *Journal of Risk and Insurance* 69, 3 (2002), 341–371.

[2] Patrick L Brockett, Xiaohua Xia, and Richard A Derrig. 1998. Using Kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. *Journal of Risk and Insurance* (1998), 245–274.

[3] 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.

[4] Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, and Le Song. 2018. GeniePath: Graph Neural Networks with Adaptive Receptive Paths. *arXiv preprint arXiv:1802.00910* (2018).

[5] Jesús M Pérez, Javier Muguerza, Olatz Arbelaitz, Ibai Gurrutxaga, and José I Martín. 2005. Consolidated tree classifier learning in a car insurance fraud detection domain with class imbalance. In *International Conference on Pattern Recognition and Image Analysis*. Springer, 381–389.

[6] Stijn Viaene, Richard A Derrig, and Guido Dedene. 2004. A case study of applying boosting Naive Bayes to claim fraud diagnosis. *IEEE Transactions on Knowledge and Data Engineering* 16, 5 (2004), 612–620.

[7] Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams, and Peter Milne. 2004. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery* 8, 3 (2004), 275–300.

[8] Jun Zhou, Qing Cui, Xiaolong Li, Peilin Zhao, Shenquan Qu, and Jun Huang. 2017. PSMART: parameter server based multiple additive regression trees system. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 879–880.