Who Stole the Postage? Fraud Detection in Return-Freight Insurance Claims

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

ABSTRACT

The emerging online shopping has led to the creation of a new type of insurance called return-freight insurance. It provides return-shipping postage compensations for settling disputes between buyers and sellers over product return on e-commerce platforms. However, 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 fraudulent accounts that could help prevent fund losses at the claim stage.

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 former deployed rule-based classifier after human expert investigations. Our approach can easily generalize to other types of insurance.

KEYWORDS

fraud detection, graph learning, network learning, insurance fraud

1 INTRODUCTION

What if you bought a dress 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? What if you spent too much and regretted your impulse purchases? When shopping online, returning an unused item can raise lots of disputes between buyers and sellers because of the ambiguity 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. According to the estimates of insurance experts at Alibaba, thousands of potentially fraudulent claims go undiscovered with the previous rule-based fraud detection system. The need for a smarter 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 previously deployed rule-based system with some, but not sufficient, confidence. We aim to discover many more fraudulent accounts than the rule-based system while retaining high precision.

Networks[2] 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 [8] and another based on GeniePath [9], 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 GeniePath.

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 for the fraud detection system [1]. Non-stationary behaviors of accounts, extracted from insurance claim history, shipping history, and shopping history, are the main causes of concept drift. Some systems solve the concept drift problem by modeling such non-stationary behaviors with adaptive learning algorithms [21]; we address this problem by adding more stationary relations instead. Relations between collaborating fraudsters are naturally illustrated through a device-sharing graph and modeled with graph learning algorithms.

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

[2] To avoid ambiguity between neural networks and relational networks, we use graph to represent relational networks in the rest of the paper.

Label uncertainty arises because of the usage of rule-generated labels. The former deployed rule-based fraud detection system outputs a risk level for each account, say 'high risk', 'low 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 we have consist of a small amount of true positive labels and a large amount of unknown labels. To build our training labels, we randomly undersample samples from the 'no observable risk' class, which will be explained in the Data Preparation section.

Excessive human effort comes from the labeling tasks and evaluation tasks in traditional insurance fraud detection settings. As a fintech company, Ant Financial[3] focuses on automated risk control that with negligible human effort. Our approach requires no human interventions besides a periodical evaluation (every week, or even every month) 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 neural networks, support vector machine, regression, Bayesian networks, and decision trees have been applied or combined in [2–4, 7, 10–16, 19]. Unsupervised approaches, such as cluster analysis, outlier detection, and spike detection have also been applied [5, 18]. 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 [6, 17].

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.

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 what makes good graphs and fits our needs, and integrate the device-sharing graph finally in our fraud detection solution.

3.1 Good Graphs

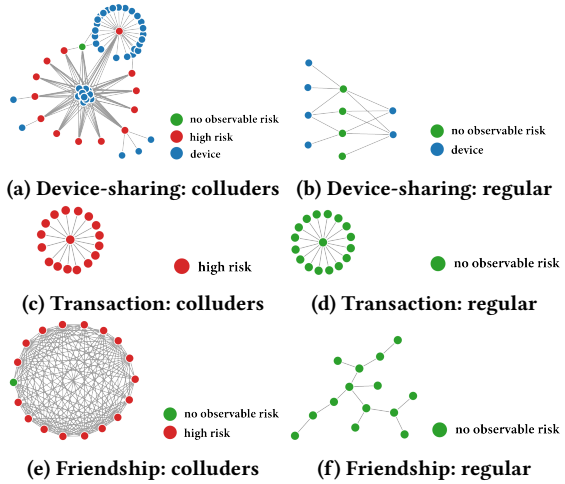
Our desired graphs should be able to separate fraudulent from regular with the following properties.

(1) distance aggregation: nodes that are close to each other should have similar labels;
(2) structural difference: structures of organized fraudsters should be different from structures of regular accounts.

[3]https://www.antfin.com/

Table 1: Graphs for Comparison

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



(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.

3.2 Three Graphs

The device-sharing graph reveals the relation of accounts sharing a device. A vertex is either a device (User Machine ID, UMID[4]) 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 preprocessed these graphs to remove isolated accounts. In the transaction graph and friendship graph, nodes with zero degree (the number of edges incident to the node) are removed. In the device-sharing graph, account nodes, sharing no common UMIDs with other accounts, and their neighboring UMID nodes are removed.

3.3 Graph Comparison

Typical subgraphs of organized fraudsters and regular users are summarized and 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. But the transaction graph fails to show such properties.

Besides, a qualified graph needs to distinguish fraudulent accounts from regular accounts. In particular, nodes that are close to each other should have similar labels. We measure the ability to

[4]The fingerprint built by Alibaba to uniquely identify devices.

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.

4 GRAPH LEARNING APPROACHES

We introduce two graph learning approaches for insurance fraud detection based on node embedding and graph neural network algorithms respectively.

4.1 The Node Embedding Approach

4.1.1 Node Embedding. Node embedding is a low-dimensional vector to represent a graph node. Nodes are mapped to embeddings so that similarity in the embedding space approximates similarity in the graph.

4.1.2 Node2vec. Node2vec [8] assigns a unique embedding vector to each node. It explicitly induces random walks along edges by starting from a random node $v_0 \in \mathcal{V}$, and repeatedly sampling an edge $(v_i, v_{i+1}) \in \mathcal{E}[v_i]$. The edge sampling is biased in node2vec and it can trade off between local and global views of the graph. A subset of random walks $\mathcal{S}_i = (v_{i-c}, ..., v_i, ..., v_{i+c})$, is used to learn similar embeddings for nodes in the same context.

4.1.3 Our Approach. Node2vec is an unsupervised algorithm that only uses graph structural information. We concatenate graph embeddings and account features and feed the new feature vectors to downstream classification tasks using gradient boosted decision tree (GBDT) [20].

4.2 The Graph Neural Network Approach

4.2.1 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 kth layer embedding of node v is

$$\mathbf{h}_v^k = \sigma(\mathbf{W}_k \cdot \mathrm{AGG}(\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v) \cup \{v\}))$$

where the initial embedding $\mathbf{h}_v^0 = \mathbf{x}_v$ is the account feature, σ is a non-linear function, and AGG is an aggregation function across layers and neighbors that differs in GNN algorithms.

4.2.2 GeniePath. The fraud detection approach we use is based on GeniePath [9], that simply stacks adaptive path layers for breadth and depth exploration in the graph. For breadth exploration, it aggregates neighbors as

$$\mathrm{AGG}(\mathbf{h}_u^k) = \sum_{u \in \mathbf{N}(v) \cup \{v\}} \mathrm{softmax}(\mathbf{w}^{\mathbf{T}} \tanh(\mathbf{W}_s \mathbf{h}_v^k + \mathbf{W}_d \mathbf{h}_u^k)) \cdot \mathbf{h}_u^k$$

This breadth-search function emphasizes the importance of neighbors with similar account features.

Given those hidden units $(\mathbf{h}_v^0, \mathbf{h}_v^1, ..., \mathbf{h}_v^K)$, a depth-search function is added to further extract and filter the signals at various depths. The resulting embeddings are fed to the final softmax or sigmoid layers for downstream fraud account classification tasks.

5 EXPERIMENTS

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

5.1 Data Preparation

All of our training data and test data contain accounts that have filed a claim within the last 30 days. We train once each week, and report classification measures on the following test data.

For each account, 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. Device usage history is also collected for graph construction.

The labels used for classification are based on 'risk level' 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, the 'regular' class in the training dataset is sampled randomly. Downsampling helps to reduce the effect of classifying a 'no observable risk' account as fraudulent, as shown in the modified objective function:

$$\begin{aligned}\mathcal{L}(w) = \min_w \Big(& \sum_{v \in \mathcal{V}_{\text{fraudulent}}} \ell(f(\mathbf{x}_v; w), \text{fraudulent}) \\ & + \sum_{v \in \text{sample}(\mathcal{V}_{\text{regular}})} \ell(f(\mathbf{x}_v; w), \text{regular}) \Big)\end{aligned}$$

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

5.2 Baseline

We evaluate our two graph learning approaches against a GBDT classifier. It uses account features as inputs without any graph structural information. For all methods, we calculate the probability of being at risk for each account in the test dataset, then use them to compute the F1 score[5].

5.3 Experimental Setups

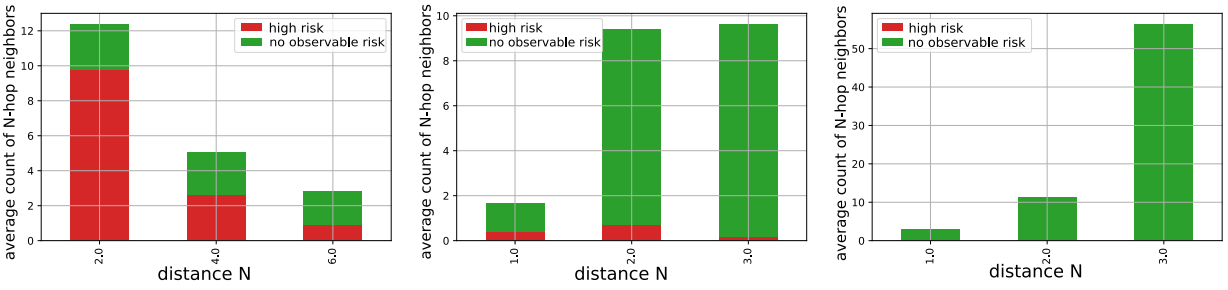
We use the implementations of GeniePath, node2vec, and GBDT (Parameter Server-based Scalable Multiple Additive Regression Tree [20]) as components implemented at Ant Financial's Platform of Artificial Intelligence (PAI).

For all the GBDT modules used in the experiments, we set the hyperparameters to be the same: 500 trees, max depth of 5 for each tree, data sampling rate of 0.6 and feature sampling rate of 0.7 to avoid overfitting, and a learning rate of 0.009. We randomly sample 25% of 'no observable risk' accounts as negative samples in the training dataset.

5.4 Results and Discussion

Our results, summarized in Table 2 and plotted in Figure 3, show that the GNNs-based graph learning approach outperforms the

[5] https://en.wikipedia.org/wiki/F1_score



(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
RE	1.47	1.44	1.44

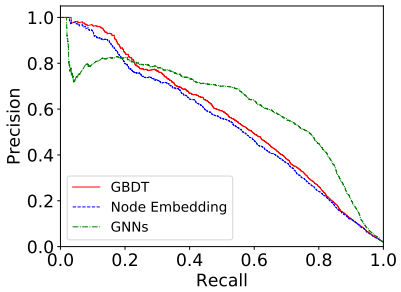


Figure 3: Model comparison with the Precision-Recall curve.

others. Report expansion (RE), defined as $\frac{FP+TP+FN}{TP+FN}$, indicates the ability to detect more fraudulent accounts. F1 scores and REs are calculated using the confusion matrix[6], whose 'ground truth labels' are based on a rule-based account risk indicator. 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 APPLICATIONS

Our workflow for the return-freight insurance fraudulent claim detection is shown in Figure 4. It 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 and monitored periodically by insurance professionals. They randomly sample our

[6]https://en.wikipedia.org/wiki/Confusion_matrix

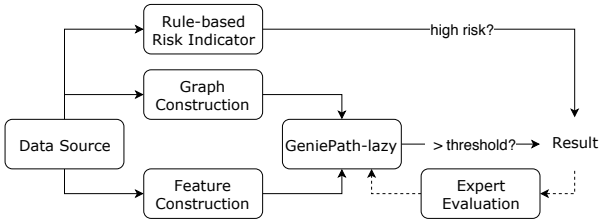


Figure 4: Workflow for fraud detection.

reported fraudulent accounts and the most recent report shows 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.

The same device-sharing graph and similar graph-learning approaches have been applied to other kinds of online insurance such as Ant Financial's health insurance. The result is promising and yet to be further refined.

7 CONCLUSION

This paper proposes device-sharing graph and graph learning-based approaches to solve the fraud detection problem in return-freight insurance claims. 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. The device-sharing graph provides better visualization and separation between good and bad. The GNNs-based GeniePath-lazy approach outperforms the others by choosing more meaningful receptive paths for information aggregation. With proper graphs, features, and algorithms, we have achieved precision of over 80% and covered 44% more suspicious accounts in one of the insurance fraud detection area with automated solutions.

REFERENCES

[1] Aisha Abdallah, Mohd Aizaini Maarof, and Anazida Zainal. 2016. Fraud detection system: A survey. *Journal of Network and Computer Applications* 68 (2016), 90–113.

[2] Manuel Artís, Mercedes Ayuso, and Montserrat Guillén. 2002. Detection of automobile insurance fraud with discrete choice models and misclassified claims. *Journal of Risk and Insurance* 69, 3 (2002), 325–340.

[3] Manuel ArtìĄs, Mercedes Ayuso, and Montserrat Guillen. 1999. Modelling different types of automobile insurance fraud behaviour in the Spanish market.

Insurance: Mathematics and Economics 24, 1-2 (1999), 67–81.

[4] El Bachir Belhadji, George Dionne, and Faouzi Tarkhani. 2000. A model for the detection of insurance fraud. *The Geneva Papers on Risk and Insurance-Issues and Practice* 25, 4 (2000), 517–538.

[5] 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.

[6] 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.

[7] Štefan Furlan and Marko Bajec. 2008. Holistic approach to fraud management in health insurance. *Journal of Information and Organizational Sciences* 32, 2 (2008), 99–114.

[8] 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.

[9] 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).

[10] Lindsay CJ Mercer. 1990. Fraud detection via regression analysis. *Computers & Security* 9, 4 (1990), 331–338.

[11] Thomas Ormerod, Nicola Morley, Linden Ball, Charles Langley, and Clive Spenser. 2003. Using ethnography to design a Mass Detection Tool (MDT) for the early discovery of insurance fraud. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*. ACM, 650–651.

[12] 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.

[13] Clifton Phua, Damminda Alahakoon, and Vincent Lee. 2004. Minority report in fraud detection: classification of skewed data. *Acm sigkdd explorations newsletter* 6, 1 (2004), 50–59.

[14] Stijn Viaene, Richard A Derrig, Bart Baesens, and Guido Dedene. 2002. A Comparison of State-of-the-Art Classification Techniques for Expert Automobile Insurance Claim Fraud Detection. *Journal of Risk and Insurance* 69, 3 (2002), 373–421.

[15] 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.

[16] Herbert I Weisberg and Richard A Derrig. 1998. Quantitative methods for detecting fraudulent automobile bodily injury claims. *Risques* 35, July–September (1998), 75–99.

[17] Graham J Williams and Zhexue Huang. 1997. Mining the knowledge mine. In *Australian Joint Conference on Artificial Intelligence*. Springer, 340–348.

[18] 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.

[19] Wan-Shiou Yang and San-Yih Hwang. 2006. A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications* 31, 1 (2006), 56–68.

[20] 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.

[21] Indrė Žliobaitė. 2010. Learning under concept drift: an overview. *arXiv preprint arXiv:1010.4784* (2010).