TitAnt: Online Real-time Transaction Fraud Detection in Ant Financial

Shaosheng Cao
AI Department (Hangzhou)
Ant Financial Services Group
556 Xixi Rd, Hangzhou, China
shaosheng.css@antfin.com

XinXing Yang
AI Department (Beijing)
Ant Financial Services Group
9F East Tower, WFC, 1 East
3rd Ring, Beijing, China
xinxing.yangxx@antfin.com

Cen Chen
AI Department (Singapore)
Ant Financial Services Group
1 Raffles Place, Singapore
chencen.cc@antfin.com

Jun Zhou
AI Department (Beijing)
Ant Financial Services Group
9F East Tower, WFC, 1 East
3rd Ring, Beijing, China
jun.zhoujun@antfin.com

Xiaolong Li
AI Department (Seattle)
Ant Financial Services Group
500 108th A. NE Bellevue,
Washington 98004, USA
xl.li@antfin.com

Yuan Qi
AI Department (Hangzhou)
Ant Financial Services Group
556 Xixi Rd, Hangzhou, China
yuan.qi@antfin.com

ABSTRACT

With the explosive growth of e-commerce and the booming of e-payment, detecting online transaction fraud in real time has become increasingly important to Fintech business. To tackle this problem, we introduce the `TitAnt`, a transaction fraud detection system deployed in Ant Financial, one of the largest Fintech companies in the world. The system is able to predict online real-time transaction fraud in mere milliseconds. We present the problem definition, feature extraction, detection methods, implementation and deployment of the system, as well as empirical effectiveness. Extensive experiments have been conducted on large real-world transaction data to show the effectiveness and the efficiency of the proposed system.

PVLDB Reference Format:
Shaosheng Cao, Xinxing Yang, Cen Chen, Jun Zhou, Xiaolong Li, and Yuan Qi. TitAnt: Online Real-time Transaction Fraud Detection in Ant Financial. *PVLDB*, 12(12): 2082 - 2093, 2019.
DOI: https://doi.org/10.14778/3352063.3352126

1. INTRODUCTION

Fraud, such as phone fraud, insurance fraud and credit card fraud, causes severe problems for government and business. However, detecting such a fraud has always been challenging. With the rapid development of the e-commerce and e-payment, the problem of online transaction fraud has become increasingly prominent. Compared with traditional areas, online transaction is facing a considerably larger volume of fund transfer.

According to the statistics [41], in the year of 2017, the number and the volume of online transaction reaches 48 billion and 2, 075 trillion yuan respectively only in China. Ant Financial[1], also known as Alipay, accounts for about 58% of China's third-part online payment transactions [30]. Specifically, on 2017's Double Eleven Shopping Festival[2] (similar to Black Friday Day in the US), a single day's transaction shot up to US$25 billion [51, 25]. With such transaction volume, it becomes thus of great significance to detect and prevent online transaction fraud.

To collect and analyze such a magnitude of transaction data, it requires a robust database component for offline storage and management. Furthermore, a large-scale distributed computing component for running algorithms is also necessary. To satisfy the low latency requirements for online serving, online prediction with efficient data accessing is of great significance. Meanwhile, feature extraction and detection methods are equally important.

Rule-based methods have been extensively studied over the years [46] for fraud detection problem. However, fraud patterns change rapidly over time, greatly deteriorating the effectiveness of rules summarized by expert experience. Subsequently, many data mining based methods have been investigated. For example, supervised learning methods, are proposed recently [40, 53]. However, transaction data usually exhibit two kinds of characteristics: 1) the labels are unbalanced, i.e., the majority of transactions are not fraudulent but normal, and 2) compared with analyzing individual transaction records, aggregated data often provides much richer information to identify fraud patterns.

To cope with the first characteristic, several unsupervised learning and anomaly detection methods are introduced [10, 35], however label information can hardly be utilized. On the other hand, some existing data aggregation strategies are also applied for detecting fraud [65, 28], nevertheless, most of the previous approaches can hardly capture the complex fraud patterns of the online transactions. It is this paper's

Proceedings of the VLDB Endowment, Vol. 12, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3352063.3352126

[1] https://en.wikipedia.org/wiki/Ant_Financial

[2] https://en.wikipedia.org/wiki/Singles%27_Day

topic to investigate how to deal with these two characteristics with our methods.

In this paper, we present a real-world task in FinTech and introduce our TitAnt[3] system, which is actively detecting fraudulent transactions. Our contributions are summarized as follows:

- We carefully analyze the task and some discoveries are excavated. Based on our observations, new feature extraction approaches for transaction fraud detection are examined, which is capable of making full use of the information from aggregated data.
- We design and develop a real-world transaction fraud detection system which is able to train offline large-scale data in hours, and predict online real-time transaction fraud within only milliseconds.
- We conduct extensive experiments on a large transaction record dataset to validate the effectiveness and efficiency of our system, including rule-based methods, anomaly detection approaches and classification models.

Our paper is organized as follows. Section 2 discusses related work of fraud detection. Section 3 presents the problem definition, feature extraction and detection methods. Section 4 describes the details of the implementation and deployment of our TitAnt system. Section 5 shows experimental results, followed by the conclusion in Section 6.

2. RELATED WORK

In this section, we investigate the related literature, including expert systems and rule-based approaches, supervised and unsupervised learning algorithms for fraud detection task, as well as recently proposed network representation learning models.

2.1 Rule-based Methods and Expert System

Quinlan [48] and Cohen [13] introduce assertion statement of IF {conditions} and THEN {a consequent} to recognize fraud records at first. By distinguishing fraudulent and normal records, Brause et al. [7] generalizes and weighs the association rules of detecting credit card fraud. Based on previous achievement, Baulier et al. [4] identifies implicit fraudulent calls by generating decision variables, Rosset et al. [52] investigates a two-stage rule-based solution to detect telephone fraud, and Wheeler and Aitken [64] adopt case-based reasoning to analyze the hardest ones of misclassified cases. Expert system based methods, on the other hand, also have been well investigated. Major and Riedinger [38] uses statistical knowledge to construct a five-layer system, Von Altrock [61], Stefano and Gisella [54], Pathak et al. [42] respectively design different fuzzy expert systems for a specific scene. Besides, Chiu and Tsai [12] proposes FPM algorithm to mine frequent patterns of credit card transactions. With the rapid evolution of fraud patterns, only hand-summarized rules or expert knowledge are not sufficient to satisfy today's online detection, the methods [47, 33] learning knowledgeable information from historical data is more worthwhile to investigate.

2.2 Supervised Learning Models

Hand [27] first uses a linear discriminative model to detect fraud, and later Foster and Stine [18] propose an improved least square regression with stepwise selection predicting. Bayesian approaches have been investigated, where Ezawa and Norton [17] employ a four-stage Bayesian network model for telephone fraud and Viaene et al. [60] adopts AdaBoosted naive Bayes for insurance fraud. Otherwise, neural network based models are applied in fraud diagnosis [21, 43, 2]. Subsequently, Syeda et al. [55] develops a parallel system of fuzzy neural networks, Barse et al. [3] leverages the memory-based neural network to capture temporal dependencies, and Maes et al. [37] combines Bayesian networks and neural networks for detecting credit card fraud. Also, Bhowmik [5] applies Bayesian classification and decision trees in insurance fraud detection task. Besides, Kim et al. [31] and Wang and Ma [63] utilize SVM-based ensemble strategy for detecting telecommunication subscription fraud and credit fraud. Besides, Halvaiee and Akbari [26] and Jia-jie [29] respectively investigate the effectiveness of the artificial immune system and particle swarm optimization algorithm in fraud detection.

2.3 Unsupervised Learning and Aggregation Strategies

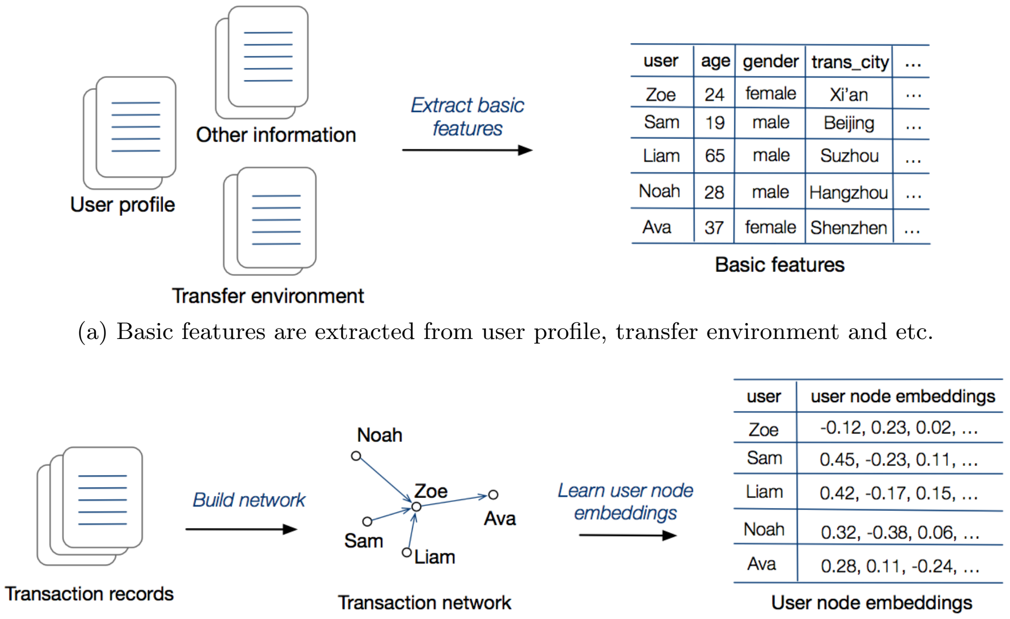
Cox et al. [15] visualizes data with the information of color, position, size and etc. to help to detect fraud. Bolton et al. [6] introduces profiling method to detect credit card fraud, Burge and Shawe-Taylor [8] uses a recurrent neural network to exploit temporal information of account behavior, and Cortes et al. [14] explores graph mining algorithms such as link analysis. Later, Yamanishi et al. [66] detects the fraud from medical insurance data by recognizing statistical outliers. Aggregated data analysis is also investigated, in which Perlich and Provost [44] propose a novel target-dependent aggregation method, Casas et al. [10] utilizes k-means to classify network security data, and Vadoodparast et al. [59] combines the results of several different clustering methods. In addition, Jha et al. [28] and Whitrow et al. [65] detect credit card fraud employing transaction aggregation. Anomaly detection methods, such as isolation forest [35], sheds light on fraud detection tasks, since fraudulent transactions are undoubtedly regarded as abnormal cases.

2.4 Network Representation Learning Models

Recently, network representation learning, also known as graph embeddings, plays an increasingly important role in network analysis. Perozzi et al. proposes DeepWalk [45], which is superior to traditional graph analysis approaches like Spectral Clustering [58], Modularity [57], and wvRN [36]. After that, many models are introduced, for example, LINE [56], GraRep [9], node2vec [24] and etc. Besides, Structure2Vec [16] is a state-of-the-art supervised fashion of generating embeddings. Although these models have been demonstrated to be effective on the public dataset, there does not exist a distributed version that is able to support real-world industrial-scale transaction records.

3. PROBLEM DEFINITION, FEATURE EXTRACTION AND DETECTION METHODS

[3] It indicates the combination of Titan and Ant, where the Titans are giant deities with incredible strength in Greek mythology and the Ant is the name of the company by the meaning of "greatness comes from micro things".



user	age	gender	trans_city	...
Zoe	24	female	Xi'an	...
Sam	19	male	Beijing	...
Liam	65	male	Suzhou	...
Noah	28	male	Hangzhou	...
Ava	37	female	Shenzhen	...

(a) Basic features are extracted from user profile, transfer environment and etc.

user	user node embeddings
Zoe	-0.12, 0.23, 0.02, ...
Sam	0.45, -0.23, 0.11, ...
Liam	0.42, -0.17, 0.15, ...
Noah	0.32, -0.38, 0.06, ...
Ava	0.28, 0.11, -0.24, ...

(b) User node embeddings are learned from historical transaction records.

Figure 1: An illustrated example of basic features extraction and user node embeddings generation.

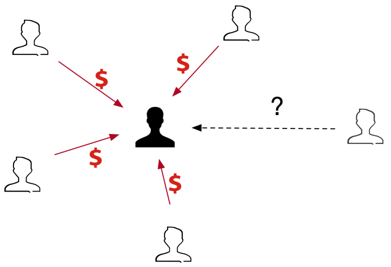


Figure 2: A simple case of aggregated data over the transaction network.

In this section, based on our analysis of the problem, feature extraction and detection methods are introduced.

3.1 Problem Definition

In general, online transaction fraud can be categorized into two different types: explicit and implicit. In an explicit case, a user is aware of the fraud afterwards. After a transaction is completed, the user could file a fraud report and upload the supporting proofs. Based on the transaction details, profiles and evidence, the authenticity of transaction fraud will be examined. If this user indeed suffers from a fraud, the fraudsters would be punished with punitive measures, such as action restrictions or account lockout, but it would be difficult to recover the losses according to the laws. This type is defined as an explicit fraud after an accident.

In an implicit case, what we are concerned is to take proactive actions to prevent the potential event of fraudulent transactions, i.e., actively detecting online transaction fraud and taking immediate steps to prevent suspicious transactions. Contrary to explicit fraud, implicit one reveals less information and requires real-time prediction of the system. In this paper, we aim to tackle the implicit online real-time transaction fraud detection task and a formal problem definition is described as follows:

Definition 1. *(Online Real-time Transaction Fraud Detection) Given historical transaction records with fraud labels, the task of online real-time transaction fraud detection is to design a system to predict whether an online real-time transaction is a fraud or not.*

3.2 Feature Extraction from Aggregated Data

In order to discover transaction fraud, user profile and transfer contextual information are often of great importance. In particular, the fraudulent rates in some specific locations are always higher than other areas. Figure 1 (a) illustrates the basic user profile features extracted, such as age, gender, and transfer city (trans_city)[4].

In addition, aggregated information on transaction records can provide much richer information. Based on our investigation, approximately 70% of the fraudsters have fraudulent behaviors more than once. It suggests that fraudsters tend to repeat their deceitful actions once successful. In Figure 2, we give a simple example to demonstrate the value of aggregated data. A directed edge reflects the transfer relationship from the corresponding transferor to the transferee. Directed red lines with a dollar sign indicate the fraudulent transactions, while a black user node stands for the fraudster. The on-going transaction, i.e., the dashed line with a question mark, is very likely to be a potentially implicit fraud. Such gathering behaviors are often observed in the real cases and manifest in more complex ways.

To extract useful information from the aggregated transaction data, a transaction network is leveraged. Formally, we define the transaction network as follows:

[4] trans_city can be inferred from transfer IP address.

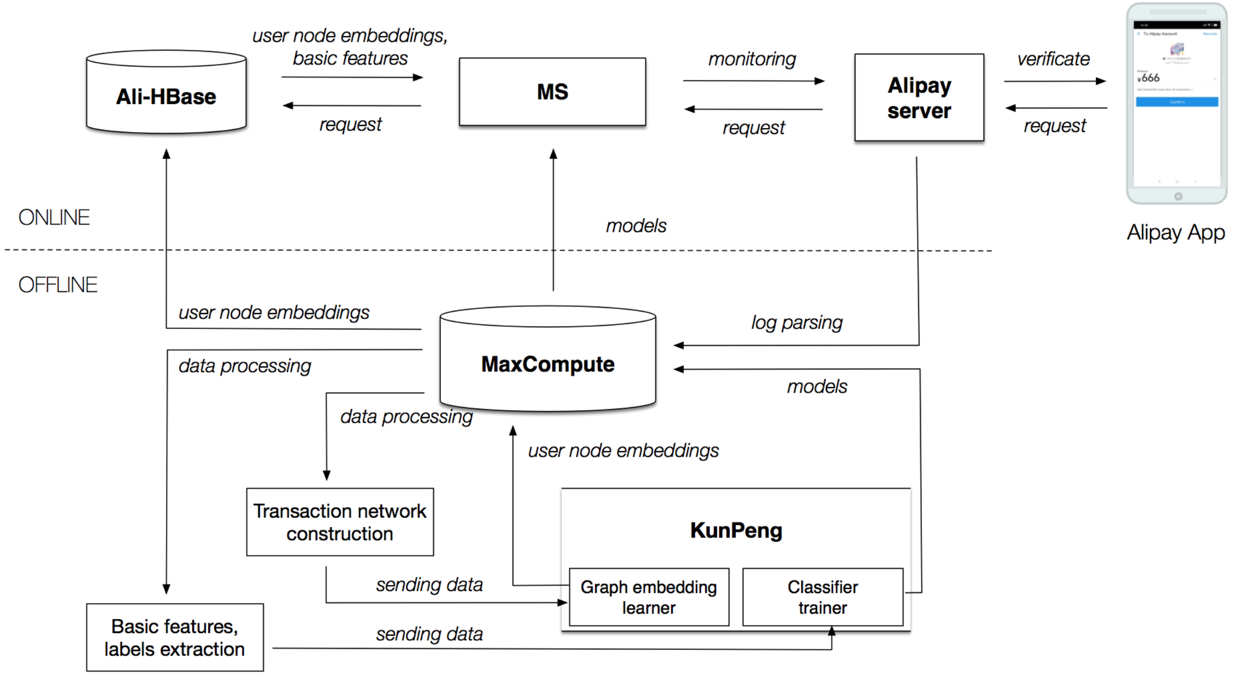


Figure 3: The architecture of TitAnt system.

Definition 2. *(Transaction Network) A transaction network is defined as* $G = (V, E)$. $V = \{v_1, v_2, \ldots, v_n\}$ *is a collection of* nodes *with each node* v *indicating a user while* $E = \{e_{i,j}\}$ *is a set of* edges *with each edge* e *indicating the transfer relationship from a transferor to a transferee, both regarded as user nodes.*

Based on historical records in a period of time, transaction network is built for analysis. Recall the simple case in Figure 2, all the victims including the potential one have a same neighbor, i.e., the fraudster. It suggests they are 2-hop neighbors to each other. Therefore, the analysis of topological relationship is worthy of well studying in the transaction network. To capture topological relationship information, Network Representation Learning (NRL) is a promising direction to be explored [67]. Given a transaction network, NRL methods aim to learn a low dimensional representation matrix $D \in \mathbb{R}^{|V| \times d}$, whose i-th row D_i is a d-dimensional vector representing the node v_i in the transaction network. In this way, the topological information can be captured by dense vectors, i.e., node embedding. Figure 1 (b) shows the procedures of generating user node embeddings. First, historical transaction records are collected to construct transaction network, and then user node embeddings are learned by NRL methods.

As most NRL implementations in the literature are limited to a single machine, we need to reimplement in a distributed learning framework, since huge amount of transaction records are being produced every day. Based on the insights that no one NRL method is the best in all cases [22], we select DeepWalk (**DW**) [45] for its efficiency, effectiveness and simplicity.

Original DW utilizes random walk to generate short node sequences which transforms the topological information from the network into the sequences. Intuitively, the neighbors of one node will often occur in its contextual position in the linear node sequences. After the linear node sequences are generated, Skip-gram with negative sampling in word2vec [39] is applied to generate user node embeddings finally.

We also reimplement Structure2Vec (**S2V**) [16] as an alternative. Such supervised method can take full advantages of label information, but the learned user node embeddings are also affected by unbalanced labels. Meanwhile, unsupervised methods like DW do not require any labels, therefore, the topological information is extracted only from transaction network without being influenced by the imbalance of labels.

3.3 Detection Methods

As the problem of fraud detection is vital to a Fintech business, efforts have been spent for years, where about fifty features are carefully engineered. We call such features as basic features, which are also treated as rules or attributes. For each user, we generate user node embeddings, i.e., aggregated features, as additional information from the aggregated transaction records. Basic features and aggregated features are then concatenated together. Labels are collected from user fraud reports, thus cannot be obtained in real-time.

In order to precisely find out fraud, we extensively investigate and validate rule-based methods, anomaly detection approaches and classification models.

Rule-based methods are widely used in many fraud detection applications. Iterative Dichotomiser 3 (**ID 3**) [47] is a traditional approach based on decision tree learning, whereas **C5.0** [33, 50] is revised version of C4.5 [49] to extract informative patterns from data with higher accuracy. In those methods, features are regards as rules and label information is utilized to do fine-tune.

Isolation Forest (**IF**) [35] is a classical anomaly detection

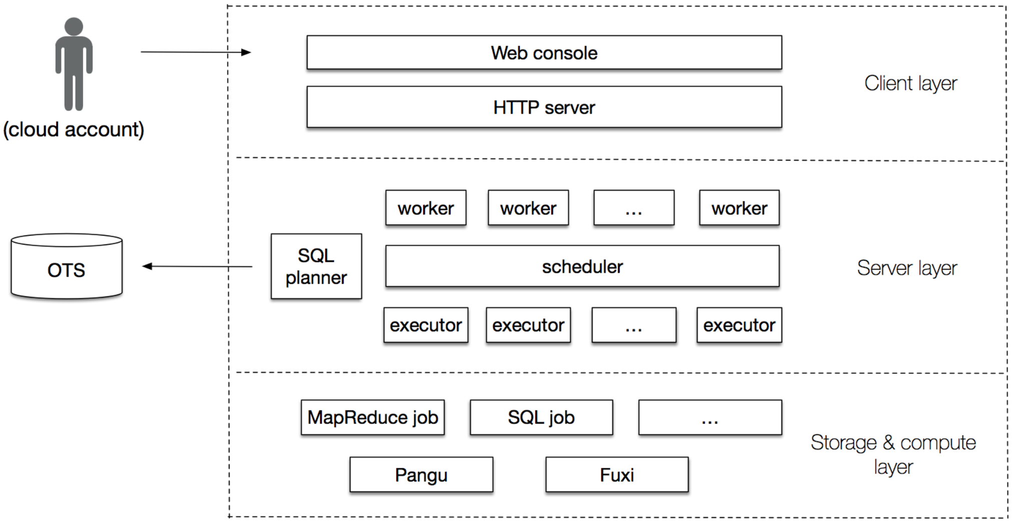


Figure 4: The architecture of MaxCompute.

approach widely used due to its effectiveness. We treat features as attributes and directly predict fraudulent transactions, since it does not require any label information. Intuitively, transaction fraud detection is similar to anomaly detection tasks, since the goal is to find out abnormal transactions, i.e., outliers that are more likely to be separated from most of the other data.

One of the most popular classification models is Logistic Regression (**LR**) [62]. Although continuous features can be used in LR, better performance can be achieved after feature discretization in most cases. Compared with LR, non-linear models such as, Gradient Boosting Decision Tree (**GBDT**) [19, 20, 1] is able to achieve better performance in a variety of industrial tasks. GBDT is a tree-based classification model, whose decision trees learn the decision boundary of the classification dataset, and gradient boosting combines several weak classifiers into a stronger one.

We will examine the effectiveness of the above detection methods in Section 5.

4. TITANT SYSTEM IMPLEMENTATION AND DEPLOYMENT

In this section, we show the details of the implementation and deployment of our TitAnt system.

4.1 The Framework of TitAnt System

To guarantee timely response on fraud detection requests, low latency predictor, robust database storage platform, and distributed algorithms ought to be carefully designed. As illustrated in Figure 3, our system mainly has two parts, i.e., offline periodical training and online real-time prediction. In the offline training part, where models are trained on a fixed time basis, and model files are uploaded to online predictor for real-time transaction monitoring.

Once users initiate transaction requests in Alipay[5], transaction logs will be periodically sent to MaxCompute[6] for offline computation. MaxCompute supports SQL and MapReduce for extracting basic features/labels and constructing transaction network. At the same time, KunPeng supports large-scale distributed NRL and classification model training[7]. The learned user node embeddings and classification models are stored in MaxCompute.

Online prediction happens at Model Server (MS), where the model files are periodically updated. Once a transaction created by a user in Alipay APP, Alipay server immediately requests the Model server (MS). MS then gets the related data from Ali-HBase and makes real-time prediction. If the transaction is detected as fraud, the on-going transaction will be interrupted and transferor will be notified. More details on each component will be elaborated in the following subsections.

4.2 MaxCompute

MaxCompute, formerly known as Open Data Processing Service (ODPS), is a database storage and management platform. It has three logical layers: client layer, server layer and storage & compute layer. As illustrated in Figure 4, developers can login with their cloud account and submit jobs by web console in client layer, where HTTP server receives the command and send message to next layer. Server layer contains workers, executors and scheduler to split jobs into subjobs for distribution. Also, heterogeneous jobs, such as mapreduce, SQL and etc., can be recognized and operated in the storage & compute layer based on Pangu and Fuxi, where Pangu is a disk storage module and Fuxi is a resource scheduling module [68].

When a SQL command is submitted by web console, the message is sent to the HTTP server, which requires the verification of cloud account information. If authentication passes, the job will be delivered to worker and the corresponding job instance will be sent to the scheduler. After that, scheduler registers the instance in Open Table Service

[5] https://itunes.apple.com/us/app/alipay-simplify-your-life/id333206289?mt=8

[6] https://www.alibabacloud.com/product/maxcompute

[7] Only classification based detection methods are reimplemented in KunPeng for better performance, as reimplementation is time-consuming. Rule-based and anomaly detection methods are not distributed.

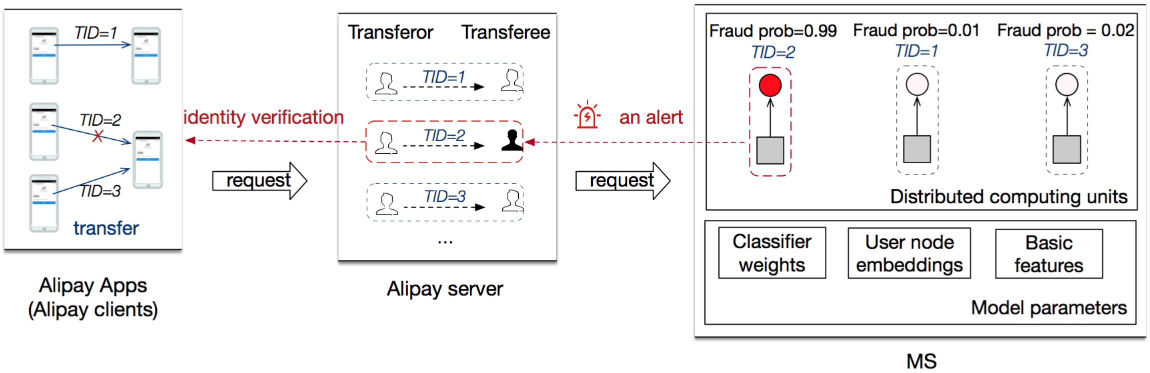


Figure 5: The architecture of the MS and its interactions with other components.

(OTS) via SQL planner and its status is set as "running" simultaneously. OTS maintains the status of all the instances. Finally, scheduler adds the instance into the queue and corresponding instance ID will be generated.

Subsequently, the scheduler will split the task of job instance into multiple subtasks, which are arranged into task pool in priority order. After that, scheduler keeps waiting for the available resource for computing. As soon as the resource conditions are satisfied, the subtasks are sent to an executor, which requests Fuxi to trigger computing resources in the compute layer. When all the subtasks are finished, the executor updates the status of the instance as "terminated" in OTS. Finally, the results will be stored in Pangu.

4.3 KunPeng

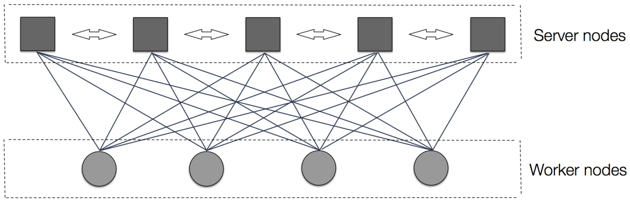


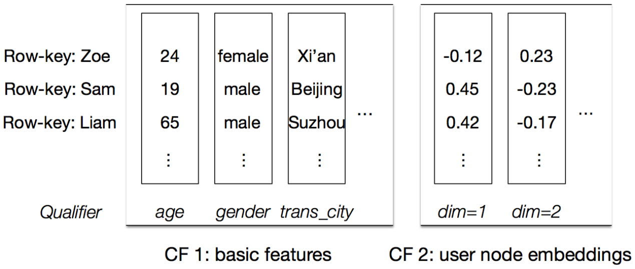
Figure 6: The system architecture of KunPeng.

As numerous transaction records wait for analysis every day, a distributed computing platform is an urgent need. Traditional frameworks, such as MPI [23], do not support good failure tolerance. However, Parameter Server (PS) [34] supports a single point of failure, i.e., the failed instance can be restarted and recovered to the previous status automatically while other instances remain not affected. KunPeng system [69] is self-developed by the company based on PS framework , where various machine learning algorithms are running simultaneously.

KunPeng supports data parallelism and model parallelism. As illustrated in Figure 6, it consists of server nodes and worker nodes, where server nodes store the model parameters while worker nodes are responsible for training. Pull and Push operations are defined between server and worker nodes for data exchange. Besides, communication also happens among server nodes.

Based on KunPeng, we redesign NLR and classification algorithms, such as DW, S2V, LR, and GBDT. As an important part of DW, our reimplemented word2vec is involved in both worker and server nodes. Worker nodes receive the node sequences by Random walk algorithm. For every iteration, each worker first read a batch of sequence data and generate negative word list. The embeddings are then pulled from server nodes and are updated by gradient descent. Subsequently, the updated embeddings are uploaded to server nodes. On the other hand, server nodes are responsible for communication with workers in order to exchange embedding data. Server nodes first randomly initialize the embeddings and wait for the push requests from worker nodes. Once the push request is received, the corresponding embeddings are sent. After the update of each worker, server nodes pull the new embeddings and aggregate them by executing the model average operation.

4.4 MS and Ali-HBase



	CF 1: basic features				CF 2: user node embeddings		
Qualifier	age	gender	trans_city		dim=1	dim=2	
Row-key: Zoe	24	female	Xi'an		-0.12	0.23	
Row-key: Sam	19	male	Beijing	...	0.45	-0.23	...
Row-key: Liam	65	male	Suzhou		0.42	-0.17	
	⋮	⋮	⋮		⋮	⋮	

Figure 7: The architecture of Ali-HBase.

Once offline training section ends, online real-time prediction works. Figure 5 shows an illustrative example of the whole real-time prediction process. When a user transfer money in Alipay App, the transfer request is sent to the Alipay server, followed by the MS for fraud monitoring. MS will access data from Ali-HBase for the latest version of user node embeddings and basic features. MS are distributed to satisfy low latency and high service load. As shown in Figure 5, the transaction TID=2 is probably a fraud with predicted fraud probability of 99%, thus MS sends an alert to the Alipay server, which will further interrupt the corresponding on-going transaction.

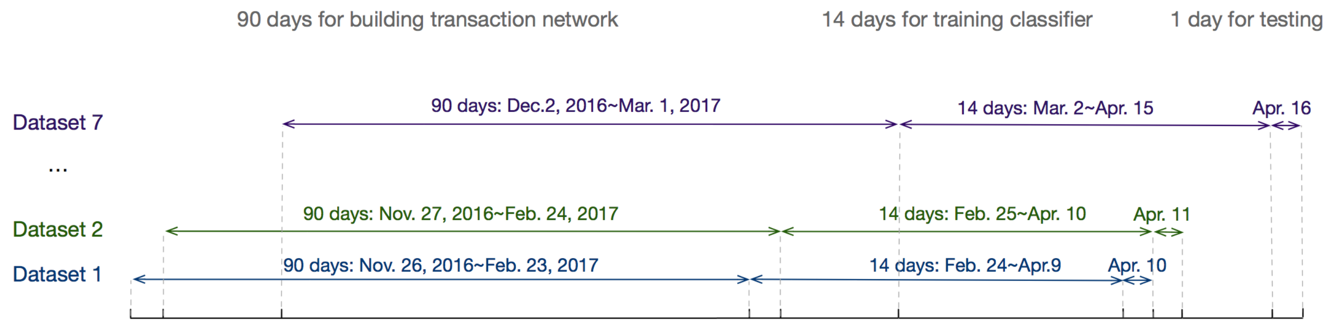

90 days for building transaction network
14 days for training classifier
1 day for testing
Dataset 7
90 days: Dec.2, 2016~Mar. 1, 2017
14 days: Mar. 2~Apr. 15
Apr. 16
...
Dataset 2
90 days: Nov. 27, 2016~Feb. 24, 2017
14 days: Feb. 25~Apr. 10
Apr. 11
Dataset 1
90 days: Nov. 26, 2016~Feb. 23, 2017
14 days: Feb. 24~Apr.9
Apr. 10

Figure 8: A graphical illustration of the datasets.

Ali-HBase is based on HBase Project[8]. HBase is first proposed as Bigtable [11], a distributed, scalable and big data store, which is suitable for our real-time data accessing scenario. The inner data is organized in the form of Column Family (CF), where qualifier is used as a marker. As shown in Figure 7, the first CF is basic features where age, gender, and trans_city are qualifiers. And the second is user node embeddings, where each dimension of value is the qualifier. In Figure 7, users like Zoe, Sam and Liam are row-keys, to index the corresponding data. Every time offline training is completed, the data is uploaded to Ali-HBase by the version of date time.

4.5 Discussion

In this section, we discuss the implementation design, deployment issues and the construction of transaction network.

First, the system has strict serving requirements, i.e., tens of milliseconds at most for online detection including computation and communication costs. However, labels are usually delayed, as they are collected through user feedbacks, where online training is impractical. Thus, we adopt periodical offline training and real-time prediction in our system.

Second, in our system, we only demonstrated the usefulness of user node embeddings learned from transaction network. One may ask what about other aggregated information, such as device and IP information? It is an interesting question to construct a heterogeneous network. We will explore this direction in future work.

5. EXPERIMENTS

To empirically quantify the benefits of each component of our TitAnt system, we conduct experiments under different configurations.

5.1 Experimental Setup

On this task, we adopt "T+1" mode to update the model, which means a model will be trained and deployed in an offline manner on a daily basis and will be used for prediction for the next day on a real-time basis. To demonstrate the effectiveness of our system, we have conducted several experiments and reported the performance of each day over a continuous week. In total, we have seven sets of data, where each one is sliced into three subsets: one for learning user node embeddings, another for training the classifier, and the last for testing. Specifically, we collect 90 days of transaction records to build the transaction network. The next 14 days of labeled records are treated as the training set and the last day of labeled records are used for the test set.

For example in Dataset 1 (illustrated in Figure 8), transaction records of April 10, 2017 are chosen as the test set, 14 days' records prior to the test set are used as the training, and the earlier 90 days of records are employed to build the transaction network. Different from other industrial scenes, such as e-commerce recommendation, online testing is hard to achieve since labels are not real-time obtained.

In our experiment, one of the goals is to investigate the effectiveness of basic features and the learned user node embeddings based on transaction network. More specifically, we compare both unsupervised DW and supervised S2V models on our task with unbalanced labels. For a fair comparison, the size of the learned embeddings is set to 32 and is concatenated with the basic features. In addition, for detection methods, we test the validity of rule-based ID3 and C5.0, anomaly detection based IF and classification based LR and GBDT.

For DW, we set the length of the random walk as 50, where each node is sampled as the first node of the sequences 100 times, i.e., the number of sampling is 100. It takes around 1.5 hours to learn the embeddings with approximate 8 million randomly selected transaction records with 20 machines equipped with 10 threads in our production environment. Aside from the transaction network, we also feed S2V with the fraud ground truth as the edge labels. Besides, there are a total of 52 basic features carefully extracted.

We set 100 trees for IF and raw basic features are fed as attributes. As rule-based ID3 and C5.0 cannot support continuous values well, we discretize the data into different bins [32]. We impose L1 regularization and assign its weight as 0.1 for LR, and set 300 iterations as the stopping criteria. For GBDT, we generate 400 trees with the depth of 3 to ensemble the results and use root mean square error as the objective. The subsampling rate of samples and features are set as 0.4 to prevent overfitting.

5.2 Empirical Results on Transaction Fraud Detection

In this section, we empirically evaluate the effectiveness of our proposed system for the transaction fraud detection task. Eleven configurations are tested in Table 1 from April 10 to April 16, where F1 score is chosen as the evaluation metric. The best results are written in bold font for each day.

First, we analyze the effectiveness of the learned user node embeddings. With the same classifier, it is obvious that in-

[8] https://hbase.apache.org/

Table 1: Performance under different eleven configurations.

Number	F1 Score	April 10	April 11	April 12	April 13	April 14	April 15	April 16
1	Basic Features/Attributes+IF	10.30%	10.38%	11.62%	11.21%	10.82%	11.00%	13.30%
2	Basic Features/Rules+ID3	42.08%	44.72%	41.21%	44.25%	42.33%	41.94%	47.69%
3	Basic Features/Rules+C5.0	44.56%	51.55%	45.94%	51.17%	50.23%	51.91%	57.07%
4	Basic Features+LR	53.08%	58.47%	55.72%	60.13%	56.87%	52.52%	64.38%
5	Basic Features+GBDT	56.80%	65.47%	59.05%	64.87%	59.19%	60.34%	68.85%
6	Basic Features+S2V+LR	55.21%	62.08%	60.78%	64.11%	61.04%	55.83%	68.86%
7	Basic Features+S2V+GBDT	60.23%	66.37%	63.24%	68.87%	64.79%	63.30%	71.10%
8	Basic Features+DW+LR	56.06%	61.15%	58.37%	61.13%	60.08%	56.00%	67.33%
9	Basic Features+DW+GBDT	**61.43%**	**66.87%**	**64.11%**	**69.93%**	**65.10%**	**64.00%**	**71.84%**
10	Basic Features+DW+S2V+LR	56.70%	61.41%	60.69%	62.78%	63.29%	57.74%	67.21%
11	Basic Features+DW+S2V+GBDT	61.37%	66.76%	**64.11%**	69.67%	64.53%	63.48%	71.40%

troducing additional features from aggregated data can consistently improve the performance of the task. For example, on April 10, the F1 score for "Basic features+GBDT" is 56.80%. Adding the embeddings generated by S2V will improve the baseline by 3.4% while adding the embeddings by DW will boost the performance by 4.6%. The similar conclusion can be obtained for the rest of the days.

We can observe that using user node embeddings learned by DW leads to better results than S2V. Although supervised S2V utilizes extra label information, it also suffers from the issue of unbalanced labels. In this case, experimental results demonstrate that the benefits from the label information are weaker than the losses suffering from the label imbalance. Moreover, in the experiments, we further concatenate the different sources of the learned user node embeddings together from DW and S2V, but the performance is not improved compared with only DW is used. This suggests the topological information has already been well extracted by DW.

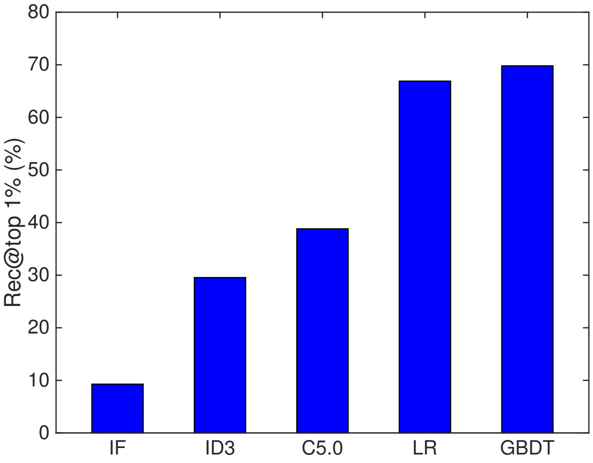


Figure 9: Recall scores for the top 1% of the most suspicious frauds under different detection methods.

In general, the performance of rule-based methods is not as good as that of classification models. C5.0 has better than ID3 by 6.9% on average, probably because it takes better data discretization and segmentation mechanisms such as Gain Ratio. LR is implemented with discretization preprocessing which tremendously improves performance. Only the best performance of LR is shown in the table, whose discretization bin size is set as 200. But still, it is obvious that GBDT can achieve better results than LR, i.e., outperforms LR by 4.5%, 2.2%, 4.5% and 4.2% for "Basic Features", "Basic Feature+S2V", "Basic Feature+DW" and "Basic Features+DW+S2V" on April 16.

Besides F1, recalls at different thresholds are also important for real-world analysis. Such recall metric can measure the ability of the classifier to find the most suspicious fraud. Figure 9 shows the recall for the top 1% of the most suspicious cases, i.e., rec@top 1%, over five different detection methods. From the results, we can see that IF performs the worst, i.e., under 10%, which is consistent with F1. Such results are intuitive as outliers found by IF are probably not caused by fraud cases but for other reasons. Rule-based ID3 and C5.0 methods achieve much higher results, i.e., 30% and 40%, respectively. GBDT slightly outperforms LR and performs the best.

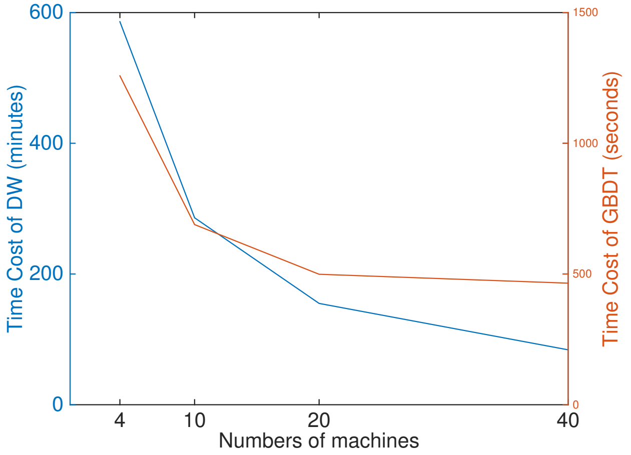


Figure 10: Time cost over the numbers of machines.

Based on the above observations, we choose DW to extract additional aggregated information. GBDT is selected as the classifier for its good performance. In order to decide the computing resources, we further test time cost versus the number of machines. For our reimplemented version on KunPeng, half of the machines are selected as server nodes, and the rest are used as worker nodes.

As shown in Figure 10, the time cost continues to decreases as the number of the machines increases for DW. However, we also notice that the time cost of GBDT does not obviously halve when the number of machines increases to 40 from 20. In fact, in real-world PS environment, IO and network communication might become the bottleneck besides computation, while more machines often indicate greater communication cost due to uneven machine traffic. Moreover, in the production environment, heterogeneous tasks execute at the same time, so resource allocation is necessary to be considered. More resources requested, more waiting time may be needed for allocation. As a compromise, we finally assign 40 machines for DW and 20 machines for GBDT.

5.3 Hyperparameter Sensitivity

We further perform a hyperparameter sensitivity analysis, where Dataset 1 shown in Figure 8 is selected for this experiment.

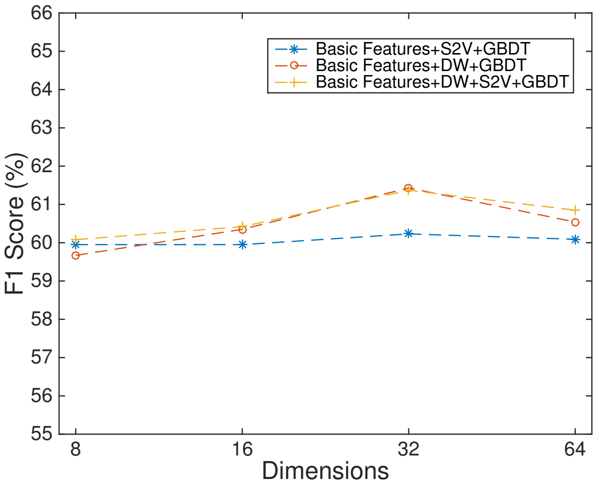


Figure 11: Performance versus the dimensions of the learned user node embeddings.

The dimension size of the learned embeddings is an important hyperparameter, which influences the amount of topological information of the transaction network extracts. As shown in Figure 11, we compare F1 score against the dimension size using different NRL methods. Obviously, 32 is the best dimension size. We believe that the topological information of the network is not well extracted when the dimension is too small, while the results probably overfit when it is too large.

In addition, we vary the tree size to examine the importance of tree size in GBDT. As illustrated in Figure 12, F1 score consistently improves as the number of trees increases to 400 and then decreases when the number of trees further increases to 800. It is intuitive that the model is not sufficiently trained when the number of trees used is too small. On the contrary, the model prone to overfitting when the number of trees is too big.

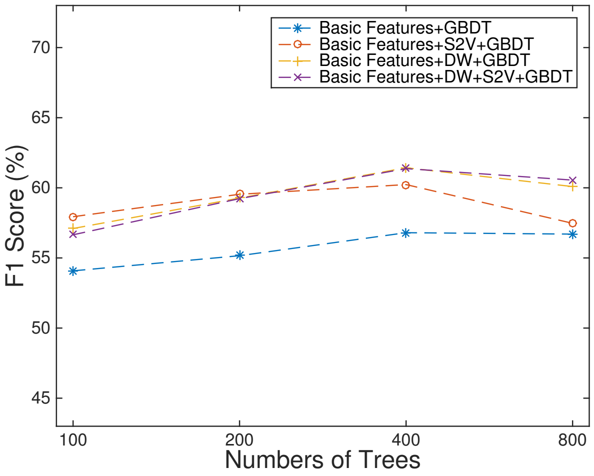


Figure 12: Performance versus the numbers of GBDT decision trees.

Table 2: Performance versus the number of node sampling.

No. of Sampling	25	50	100	200
F1 Score	59.67%	60.62%	61.43%	61.57%

Finally, we analyze the impact of the number of node sampling in DW, which controls the number of linear node sequences generated. Similar to the conclusion in [45], the performance tends to be stable as the number reaches a specific value. Table 2 suggests that the performance tends to stabilize when the number reaches 100. Although the result is slightly better as for 200, it takes about double time to generate node sequences and learn embeddings. Besides, the depth of GBDT decision trees is also worthy of exploring, we omit it here as it can be analyzed in the similar way.

6. CONCLUSION

In this paper, we first reveal the significance of online real-time transaction fraud detection task in Ant Financial and then demonstrate our feature extraction approaches, detection models and implementation details. Extensive experiments on real-world data are conducted, showing the effectiveness and performance of our proposed TitAnt system. In the future, we will investigate more possibilities for system design, explore dynamic construction and modeling of a heterogeneous network, and study the interpretability of learned embeddings by NRL models.

7. ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their constructive and valuable advice, and MaxCompute team for their suggestions on data storage and management, as well as Kai Xiao and Xiujing Lin's help on data preparation.

8. REFERENCES

[1] M. M. Ahmed and M. Abdel-Aty. Application of stochastic gradient boosting technique to enhance

reliability of real-time risk assessment: use of automatic vehicle identification and remote traffic microwave sensor data. *Transportation research record*, 2386(1):26–34, 2013.

[2] E. Aleskerov, B. Freisleben, and B. Rao. Cardwatch: A neural network based database mining system for credit card fraud detection. In *Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr)*, pages 220–226. IEEE, 1997.

[3] E. L. Barse, H. Kvarnstrom, and E. Jonsson. Synthesizing test data for fraud detection systems. In *19th Annual Computer Security Applications Conference, 2003. Proceedings.*, pages 384–394. IEEE, 2003.

[4] G. D. Baulier, M. H. Cahill, V. K. Ferrara, and D. Lambert. Automated fraud management in transaction-based networks, Dec. 19 2000. US Patent 6,163,604.

[5] R. Bhowmik. Detecting auto insurance fraud by data mining techniques. *Journal of Emerging Trends in Computing and Information Sciences*, 2(4):156–162, 2011.

[6] R. J. Bolton, D. J. Hand, et al. Unsupervised profiling methods for fraud detection. *Credit Scoring and Credit Control VII*, pages 235–255, 2001.

[7] R. Brause, T. Langsdorf, and M. Hepp. Neural data mining for credit card fraud detection. In *Proceedings 11th International Conference on Tools with Artificial Intelligence*, pages 103–106. IEEE, 1999.

[8] P. Burge and J. Shawe-Taylor. An unsupervised neural network approach to profiling the behavior of mobile phone users for use in fraud detection. *Journal of parallel and distributed computing*, 61(7):915–925, 2001.

[9] S. Cao, W. Lu, and Q. Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 891–900. ACM, 2015.

[10] P. Casas, A. D'Alconzo, G. Settanni, P. Fiadino, and F. Skopik. Poster:(semi)-supervised machine learning approaches for network security in high-dimensional network data. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1805–1807. ACM, 2016.

[11] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems*, 26(2):4, 2008.

[12] C.-C. Chiu and C.-Y. Tsai. A web services-based collaborative scheme for credit card fraud detection. In *IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004*, pages 177–181. IEEE, 2004.

[13] W. W. Cohen. Fast effective rule induction. In *Machine Learning Proceedings 1995*, pages 115–123. Elsevier, 1995.

[14] C. Cortes, D. Pregibon, and C. Volinsky. Computational methods for dynamic graphs. *Journal of Computational and Graphical Statistics*, 12(4):950–970, 2003.

[15] K. C. Cox, S. G. Eick, G. J. Wills, and R. J. Brachman. Brief application description; visual data mining: Recognizing telephone calling fraud. *Data Mining and Knowledge Discovery*, 1(2):225–231, 1997.

[16] H. Dai, B. Dai, and L. Song. Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning*, pages 2702–2711, 2016.

[17] K. J. Ezawa and S. W. Norton. Constructing bayesian networks to predict uncollectible telecommunications accounts. *IEEE Expert*, 11(5):45–51, 1996.

[18] D. P. Foster and R. A. Stine. Variable selection in data mining: Building a predictive model for bankruptcy. *Journal of the American Statistical Association*, 99(466):303–313, 2004.

[19] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[20] J. H. Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[21] S. Ghosh and D. L. Reilly. Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, volume 3, pages 621–630. IEEE, 1994.

[22] P. Goyal and E. Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[23] W. D. Gropp, W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: portable parallel programming with the message-passing interface*, volume 1. MIT press, 1999.

[24] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[25] T. Guardian. Chinese shoppers spend a record $25bn in singles day splurge. `https://www.theguardian.com/world/2017/nov/12/chinese-shoppers-spend-a-record-25bn-in-singles-day-splurge/`, 2018. Accessed May 24, 2018.

[26] N. S. Halvaiee and M. K. Akbari. A novel model for credit card fraud detection using artificial immune systems. *Applied soft computing*, 24:40–49, 2014.

[27] D. J. Hand. Discrimination and classification. *Wiley Series in Probability and Mathematical Statistics, Chichester: Wiley, 1981*, 1981.

[28] S. Jha, M. Guillen, and J. C. Westland. Employing transaction aggregation strategy to detect credit card fraud. *Expert systems with applications*, 39(16):12650–12657, 2012.

[29] S. Jia-jie. Electronic transaction fraud detection based on improved pso algorithm. In *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*, pages 2121–2125. IEEE, 2012.

[30] W. S. Journal. 5 things to know about china's ant financial. `https://blogs.wsj.com/briefly/2016/04/26/5-things-to-know-about-chinas-ant-financial/`, 2016.

Accessed May 24, 2018.

[31] J. Kim, A. Ong, and R. E. Overill. Design of an artificial immune system as a novel anomaly detector for combating financial fraud in the retail sector. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, volume 1, pages 405–412. IEEE, 2003.

[32] S. Kotsiantis and D. Kanellopoulos. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58, 2006.

[33] M. Kuhn and K. Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.

[34] M. Li, L. Zhou, Z. Yang, A. Li, F. Xia, D. G. Andersen, and A. Smola. Parameter server for distributed machine learning. In *Big Learning NIPS Workshop*, volume 6, page 2, 2013.

[35] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[36] S. A. Macskassy and F. Provost. A simple relational classifier. Technical report, NEW YORK UNIV NY STERN SCHOOL OF BUSINESS, 2003.

[37] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick. Credit card fraud detection using bayesian and neural networks. In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, pages 261–270, 2002.

[38] J. A. Major and D. R. Riedinger. Efd: A hybrid knowledge/statistical-based system for the detection of fraud. *Journal of Risk and Insurance*, 69(3):309–324, 2002.

[39] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[40] E. W. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision support systems*, 50(3):559–569, 2011.

[41] P. B. of China. The overall operation of the payment system in 2017. `http://www.pcac.org.cn/Upload/image/20180306/20180306144824\_91997.pdf/`, 2018. Accessed Feburay 19, 2019.

[42] J. Pathak, N. Vidyarthi, and S. L. Summers. A fuzzy-based algorithm for auditors to detect elements of fraud in settled insurance claims. *Managerial Auditing Journal*, 20(6):632–644, 2005.

[43] R. Patidar, L. Sharma, et al. Credit card fraud detection using neural network. *International Journal of Soft Computing and Engineering (IJSCE)*, 1(32-38), 2011.

[44] C. Perlich and F. Provost. Aggregation-based feature invention and relational concept classes. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 167–176. ACM, 2003.

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

[46] C. Phua, V. Lee, K. Smith, and R. Gayler. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*, 2010.

[47] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[48] J. R. Quinlan. Learning logical definitions from relations. *Machine learning*, 5(3):239–266, 1990.

[49] J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[50] R. Quinlan. Data mining tools see5 and c5.0. `http://www.rulequest.com/see5-info.html`. Accessed February 12, 2019.

[51] M. T. Review. Big data game-changer: Alibaba's double 11 event raises the bar for online sales. `https://www.technologyreview.com/s/602850/big-data-game-changer-alibabas-double-11-event-raises-the-bar-for-online-sales/`, 2016. Accessed May 24, 2018.

[52] S. Rosset, U. Murad, E. Neumann, Y. Idan, and G. Pinkas. Discovery of fraud rules for telecommunicationschallenges and solutions. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 409–413. ACM, 1999.

[53] B. Sagar, P. Singh, and S. Mallika. Online transaction fraud detection techniques: A review of data mining approaches. In *2016 3rd International Conference on Computing for Sustainable Global Development*, pages 3756–3761. IEEE, 2016.

[54] B. Stefano and F. Gisella. Insurance fraud evaluation: a fuzzy expert system. In *10th IEEE International Conference on Fuzzy Systems.(Cat. No. 01CH37297)*, volume 3, pages 1491–1494. IEEE, 2001.

[55] M. Syeda, Y.-Q. Zhang, and Y. Pan. Parallel granular neural networks for fast credit card fraud detection. In *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291)*, volume 1, pages 572–577. IEEE, 2002.

[56] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[57] L. Tang and H. Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826. ACM, 2009.

[58] L. Tang and H. Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478, 2011.

[59] M. Vadoodparast, A. R. Hamdan, et al. Fraudulent electronic transaction detection using dynamic kda model. *International Journal of Computer Science and Information Security*, 13(3):90, 2015.

[60] S. Viaene, R. A. Derrig, and G. Dedene. A case study of applying boosting naive bayes to claim fraud diagnosis. *IEEE Transactions on Knowledge and Data*

Engineering, 16(5):612–620, 2004.

[61] C. Von Altrock. *Fuzzy logic and neurofuzzy applications in business and finance*. Prentice-Hall, Inc., 1996.

[62] S. H. Walker and D. B. Duncan. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179, 1967.

[63] G. Wang and J. Ma. A hybrid ensemble approach for enterprise credit risk assessment based on support vector machine. *Expert Systems with Applications*, 39(5):5325–5331, 2012.

[64] R. Wheeler and S. Aitken. Multiple algorithms for fraud detection. In *Applications and Innovations in Intelligent Systems VII*, pages 219–231. Springer, 2000.

[65] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams. Transaction aggregation as a strategy for credit card fraud detection. *Data mining and knowledge discovery*, 18(1):30–55, 2009.

[66] K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, 2004.

[67] D. Zhang, J. Yin, X. Zhu, and C. Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 2018.

[68] Z. Zhang, C. Li, Y. Tao, R. Yang, H. Tang, and J. Xu. Fuxi: a fault-tolerant resource management and job scheduling system at internet scale. *PVLDB*, 7(13):1393–1404, 2014.

[69] J. Zhou, X. Li, P. Zhao, C. Chen, L. Li, X. Yang, Q. Cui, J. Yu, X. Chen, Y. Ding, et al. Kunpeng: Parameter server based distributed learning systems and its applications in alibaba and ant financial. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1693–1702. ACM, 2017.