Random Forest for Credit Card Fraud Detection

Shiyang Xuan
Department of Computer Science
Tongji University
Shanghai, China
xsyfor@tongji.edu.cn

Guanjun Liu
Department of Computer Science
Tongji University
Shanghai, China
liuguanjun@tongji.edu.cn

Zhenchuan Li
Department of Computer Science
Tongji University
Shanghai, China
1510482@tongji.edu.cn

Lutao Zheng
Department of Computer Science
Tongji University
Shanghai, China
zhenglutao103@163.com

Shuo Wang
Department of Computer Science
Tongji University
Shanghai, China
wangshuo@tongji.edu.cn

Changjun Jiang
Department of Computer Science
Tongji University
Shanghai, China
cjjiang@tongji.edu.cn

***Abstract*—Credit card fraud events take place frequently and then result in huge financial losses. Criminals can use some technologies such as Trojan or Phishing to steal the information of other people's credit cards. Therefore, an effictive fraud detection method is important since it can identify a fraud in time when a criminal uses a stolen card to consume. One method is to make full use of the historical transaction data including normal transactions and fraud ones to obtain normal/fraud behavior features based on machine learning techniques, and then utilize these features to check if a transaction is fraud or not. In this paper, two kinds of random forests are used to train the behavior features of normal and abnormal transactions. We make a comparison of the two random forests which are different in their base classifiers, and analyze their performance on credit fraud detection. The data used in our experiments come from an e-commerce company in China.**

***Index Terms*—Random forest, decision tree, credit card fraud**

I. Introduction

Credit cards are widely used due to the popularization of e-commerce and the development of mobile intelligent devices. Card-not-present transactions (i.e., online transaction without a physical card) [1] is more popular, especially all credit card operations are performed by web payment gateways, e.g., PayPal and Alipay. Credit card has made an online transaction easier and more convenient. However, there is a growing trend of transaction frauds resulting in a great losses of money every year [18] [19]. It is estimated that losses are increased yearly at double digit rates by 2020 [2]. Since the physical card is not needed in the online transaction environment and the card's information is enough to complete a payment [17], it is easier to conduct a fraud than before. Transaction fraud has become a top barrier to the development of e-commerce and has a dramatic influence on the economy. Hence, fraud detection is essential and necessary.

Fraud detection is a process of monitoring the transaction behavior of a cardholder in order to detect whether an incoming transaction is done by the cardholder or others [10]. Generally, there are two kinds of methods for fraud detection [15]: misuse detection and anomaly detection. Misuse detection uses classification methods to determine whether an incoming transaction is fraud or not. Usually, such an approach has to know about the existing types of fraud to make models by learning the various fraud patterns. Anomaly detection is to build the profile of normal transaction behavior of a cardholder based on his/her historical transaction data, and decide a newly transaction as a potential fraud if it deviates from the normal transaction behavior. However, an anomaly detection method needs enough successive sample data to characterize the normal transaction behavior of a cardholder.

This paper is about misuse method. We use random forest [20] to train the normal and fraud behavior features. Random forest is a classification algorithm based on the votes of all base classifiers.

The major contributions of this paper are summarized as follows. 1)To deal with fraud detection problem, two kinds of random forests are used to train the normal/fraud behavior features. They are Random-tree-based random forest and CART-based one, respectively. 2)By using the data from an e-commerce company in China, experiments are conducted to evaluate the effectiveness of these two methods. 3)From the result of experiments, some conclusions are made which would be helpful for future work.

The paper is organized as follows. Section II describes some related work about credit card fraud. Section III introduces the methods used in our experiment. The experiments and performance measures are discussed in Section IV. Finally, some conclusions and future work are presented.

II. Related Work

A comprehensive understanding of fraud detection technologies can be helpful for us to solve the problem of credit card fraud. The work in [16] provides a comprehensive discussion on the challenges and problems of fraud detection research. Mohammad et.al., [14] review the most popular types of credit card fraud and the existing nature-inspired detection methods that are used in detection methods. Basically, there are two types of credit card fraud: application fraud and behavior fraud

978–1–5386–5053–0/18/$31.00 © 2018 IEEE

[3]. Application fraud is that criminals get new credit cards from issuing companies by forging false information or using other legitimate cardholders information. Behavior fraud is that criminals steal the account and password of a card from the genuine cardholder and use them to spend.

Recently, a kind of fraud detection method is popular in some commercial banks which is to check behaviors of the associated cardholder [7]. Almost all the existing work about detection of credit card fraud is to capture the behavior patterns of the cardholder and to detect the fraud transactions based on these patterns. Srivastava et.al. [5] model the sequence of transaction features in credit card transaction processing using a hidden markov model (HMM) and demonstrate its effectiveness on the detection of frauds. An HMM is initially trained with the normal behavior of the cardholder. If the current transaction is not accepted by the trained HMM with a high probability, it is considered to be fraudulent. However, they only consider the transaction amount as the feature in the transaction process. Amlan et.al [8] propose a method using two-stage sequence alignment which combines both misuse detection and anomaly detection [15]. In their method, a profile analyzer is used to determine the similarity of an incoming sequence of transaction on a given credit card with the legitimate cardholder's past spending sequence. Then, the unusual transactions traced by the profile analyzer are passed to a deviation analyzer for possible alignment with the past fraudulent behavior. The final decision about the nature of a transaction is taken on the basis of the observations by the two analyzers. However, this method cannot detect frauds in real time. Elaine et.al. [9] propose a user behavior model which treats the transaction features independently. Gabriel et.al [13] propose an alternative method to prevent fraud in E-commerce applications, using a signature-based method to establish a user's behavior deviations and consequently detect the potential fraud situations in time. However they only consider the click stream as the element of the signature. We believe that instead of using only one transaction feature for a fraud detection, it is better to consider multiple transaction features.

Sahin and Duman [12] make a comparison between decision tree and support vector machine (SVM) in detecting credit card fraud. They divide a dataset into three groups which are different in the ratio between fraudulent transactions and legitimate one, and they develop seven decision tree and SVM based models and test them in these datasets. The results of experiments reveal that decision tree based model is better than SVM model. However, the accuracy of SVM based models could reach the same performance as the decision tree based models with increasing size of training dataset. Leila et.al [4] propose a method of aggregating profile which exploits the inherent patterns in time series of transactions, and the fraud detection is performed online at the end of a day or at the end of a period respectively. In their work, they evaluate and compare several techniques such as support vector machine and random forest for predicting credit card fraud, and the conclusion is that random forest has the best performance among these techniques with the process of aggregation. However, the aggregated method in this work fails to detect a fraud in real time.

III. Random Forest

In our experiment, we use random forest [20] as a classifier. The popularity of decision tree models [23] in data mining is owed to their simplification in algorithm and flexibility in handling different data attribute types. However, single-tree model is possibly sensitive to specific training data and easy to overfit [11]. Ensemble methods can solve these problems by combine a group of individual decisions in some way and are more accurate than single classifiers [21]. Random forest, one of ensemble methods, is a combination of multiple tree predictors such that each tree depends on a random independent dataset and all trees in the forest are of the same distribution [20]. The capacity of random forest not only depends on the strength of individual tree but also the correlation between different trees. The stronger the strength of single tree and the less the correlation of different tress, the better the performance of random forest. The variation of trees comes from their randomness which involves bootstrapped samples and randomly selects a subset of data attributes. Although there possibly exist some mislabeled instances in our dataset, random forest is still robust to noise and outliers. We introduce two kinds of random forests, named as random forest I and random forest II, which are different in their base classifiers (i.e., a tree in random forest).

For readability, some notations are introduced here. Considering a given dataset D with *n* examples (i.e. $|D| = n$), we denote: $D = \{(\boldsymbol{x_i}, y_i)\}$,$i = 1, ..., n$, where $\boldsymbol{x_i} \in X$ is an instance in the *m*-dimensional feature space $X = \{f_1, f_2, ..., f_m\}$ and $y_i \in Y = \{0, 1\}$ is the class label associated with instance $\boldsymbol{x_i}$.

A. Random-tree-based random forest

A base classifier of random forest I, which is a simple implement of decision tree, is called a random tree [22]. The training set of each tree is a collection of bootstrapped samples selected randomly from the standard training set with replacement. At each internal node, it randomly selects a subset of attributes and computes the centers of different classes of the data in current node. The centers of class 0 and 1 are denoted as $leftCenter$ and $rightCenter$, respectively. The kth element of a center is computed based on the following equations [22].

$$leftCenter[k] = \frac{1}{n}\sum_{i=1}^{n} x_{ik} I(y = 0) \tag{1}$$

$$rightCenter[k] = \frac{1}{n}\sum_{i=1}^{n} x_{ik} I(y = 1) \tag{2}$$

where $I(y = 0)$ and $I(y = 1)$ are the dictator functions. At the current node, each record of the dataset is allocated to the corresponding class according to the Manhattan distance between the record and the center as shown in (3).

$$\text{disance}(\text{center}, \text{record}) = \sum_{i \in sub} |center[i] - record[i]| \quad (3)$$

Note, *sub* is the subset of attributes randomly selected from X whose size is the square root of $m = |X|$. Each tree grows fully without pruning.

Algorithm I describes the process of producing a type-I random forest:

Algorithm I:

Input: Dataset D and the number of trees NT.

Output: A random forest.

For $i = 1$ to NT:

1) Draw a bootstrap sample D_i from the training set D whose size is n.
2) Construct a binary tree of the bootstrapped data recursively from root node. Repeatedly perform the following steps until all records of current node belong to a class.
   a) Randomly select a subset of $\sqrt{m}$ attributes.
   b) For $j = 1$ to $\sqrt{m}$:
      i) Compute $leftCenter[j]$ and $rightCenter[j]$.
   c) For $k = 1$ to $|D_{ic}|$:
      i) Compute the Manhattan distance dL_k and dR_k between the $record_k$ and each center.
      ii) if $dL_k \leq dR_k$
      Allocate $record_k$ to the left child of the current node.
      else
      Allocate $record_k$ to the right child of the current node.
   d) Split the node into a left child and a right child.

where D_{ic} is the subset of D_i in the current node.

A simple example of random tree is shown in Fig. 1. The internal nodes are represented by circles. The variables in a circle are attributes randomly chosen from $X = \{x_1, x_2, x_3, x_4\}$. The decisions are made according to their values. Each terminal node is represented by a rectangle and corresponds to a class. The number in a terminal node represents which class the node belongs to.

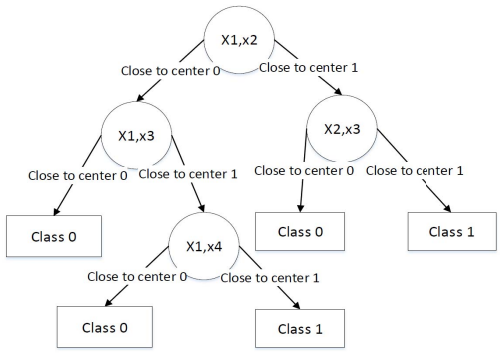

Fig. 1. Illustration of Random Tree.

B. CART-based random forest

The base classifier of random forest II is CART (Classification and Regression Trees) [24] whose training set also comes from bootstrapped samples. At each node, it splits dataset by choosing the best attribute in a subset of attributes according to Gini impurity which measures uncertainty of dataset. The subset of attributes are randomly selected from all attributes of dataset. According to the advice from Breiman, the size of the subset is set to the square root of the number of all attributes [20]. The Gini impurity is defined in (4) and is described in (5) under the condition of feature x_i.

$$Gini(Node) = 1 - \sum_{k=1}^{C} p_k^2 \quad (4)$$

Where C is the number of classes which is 2 in binary classification problem and p_k is the probability that a record belongs to class k.

$$\begin{aligned} Gini(Node, x_i) = &\frac{|Node_l|}{|Node|} Gini(Node_l) \\ &+ \frac{|Node_r|}{|Node|} Gini(Node_r) \end{aligned} \quad (5)$$

Where $Node_l$ is the left child of the current node and $|Node|$ represents the number of records in the dataset w.r.t. the current node.

A following algorithm II describes the process of producing a type-II random forest:

Algorithm II:

Input: Dataset D, the number of trees NT and the threshold T of Gini impurity

Output: A random forest

For $i = 1$ to NT:

1) Draw a bootstrap sample D_i of size n from the training set D.
2) Construct a decision tree of the bootstrapped data recursively from root node. Repeatedly perform the following steps until Gini impurity less than T.
   a) Randomly select a subset of $\sqrt{m}$ attributes.
   b) For $j = 1$ to $\sqrt{m}$:
      i) Compute Gini impurity for feature x_j.
   c) Choose the feature and its value with the minimum Gini impurity as the split attribute and split value.
   d) Split the internal node into two child nodes according to the split attribute and value.

A simple example of CART is shown in Fig. 2. The internal nodes are represented by circles. The variable and number in circles are the best splitting attribute and its value, respectively. The number labeled on the left edge from internal node means the value of this attribute greater than or equal to the splitting value, while the number labeled on the right edge means that the attribute has a less value. Each terminal node is represented by a rectangle. The number in a terminal node is the class the node belongs to.

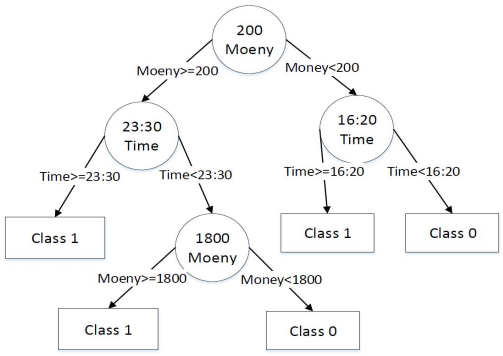

Fig. 2. Illustration of Classification and regression tree.

If we use a type-II random forest to classify a new instance for which we know only the values of independent attributes, then we drop the instance down the tree until a terminal node such that at each internal node a appropriate branch is taken according to the required condition.

The main difference between the two algorithms is the way of splitting of nodes. In type-I random forest, the data are distributed by comparing distances between records and two centers; In type-II random forest, the data are distributed according to the attribute which has minimum gini impurity. They all have their own advantages and drawbacks. In Algorithm I, it could be faster in computing centers but slower in distributed records, because it has to compute distances between centers and all records. In algorithm II, although it could be slower in computing gini impurity for attributes, it would be faster in the process of distributing data.

IV. Experiment

This section shows the details and results of experiments. Firstly, a performance comparison is made on the same subset. Then we explore the relation between a model's performance and the ratio of legal and fraud transactions in a subset. Finally, it shows the performances of models on a much bigger dataset, which is more closed to the actual result.

A. Performance measures

Before we describe the experiment, we first introduce the measures we used. Because accuracy rate is not enough to measure the performance of a random forest model when the data is significantly imbalanced. For instance, a default prediction of all instances into the majority class will also have a high value of accuracy. Therefore, we need to consider other measures. The basic measures are listed in Table I where Positive corresponds to fraud instances and Negative corresponds to normal instances. Precision rate is a measure of the result of prediction and recall rate measures the detection rate of all fraud cases. F-measure is the harmonic mean of recall and precision. Intervention rate is a measure of degree of intercept of normal instances.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

TABLE I
BASIC MEASURES

Predict \ Real	Positive	Negative
Positive	True Positives	False Positive
Negative	False Negative	True Negative

TABLE II
OVERVIEW OF THE ORIGINAL DATASET

Items \ Time	Nov. 2016 (30 days)	Dec. 2016 (31 days)	Jan. 2017 (first 11 days)
Total transactions	13607812	13397346	4749993
Legal transactions	13597637	13363584	4711539
Fraud Transactions	10175	33762	38454

$$precision = \frac{TP}{TP + FP} \tag{7}$$

$$recall = \frac{TP}{TP + FN} \tag{8}$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{9}$$

$$Intervention = \frac{FP}{TN + FP} \tag{10}$$

B. Experimental dataset

The dataset used in the paper comes from an e-commerce company of China, and consists of fraudulent and legitimate B2C transactions from November 2016 to January 2017. As shown in Table II, the total original dataset contains more than 30,000,000 individual transactions. Each transaction record consists of 62 attribute values such as transaction time, place, and amount. Each record is labled by Fraud or Legal. As required by the company, details of attributes of dataset is not permitted to be introduced. In the dataset only about 82,000 transactions were labeled as fraud, meaning the fraud ratio of 0.27% and dataset imbalance problem should be taken into consideration.

C. Experiment I

The aim of this experiment is to show which kind of random forest introduced in the previous section is more practical for identifying fraud detection. The subset used in this experiment consists of all the fraud transactions in January 2017 and 150,000 legal transactions randomly selected from all the legal transactions in January 2017 in order to balance the positive and negative samples of the training set. Then 70% transactions of the above data are as the training dataset, and the rest as the testing dataset.

Table III shows the results produced by random forest I and random forest II. Although the precision of random forest II is a little worse, the accuracy, recall and F-measure is

TABLE III
RESULTS OF TWO KINDS RANDOM FORESTS

Measure / Models	Accuracy	Precision	Recall	F-Measure
Random Forest I	91.96%	90.27%	67.89%	0.7811
Random Forest II	96.77%	89.46%	95.27%	0.9601

TABLE IV
OVERVIEW OF THE ORIGINAL DATASET

Ratio of legal and fraud transactions	Legal transactions	Fraud transactions	Under-sampling ratio of legal transactions
1:1	38000	38454	0.81%
2:1	76000	38454	1.61%
3:1	114000	38454	2.42%
4:1	152000	38454	3.23%
5:1	190000	38454	4.03%
6:1	228000	38454	4.84%
7:1	266000	38454	5.64%
8:1	304000	38454	6.45%
9:1	342000	38454	7.26%
10:1	380000	38454	8.06%

much better. Obviously, the comprehensive performance of random forest II is much more suitable for application on this experiment subset.

D. Experiment II

The original dataset has a fraud ratio of 0.27%, denoting that there is a seriously data imbalance problem. This experiment explores the relation between a model's performance and the ratio of legal and fraud transactions. We use the dataset in January 2017. As fraud transactions of the dataset are limited, this experiment adopts the random under-sampling method [6] on legal transactions. Thus, all transactions labeled by fraud are preserved as the base for regulating the under-sampling ratio. As shown in Table IV, ten subsets are extracted with the ratio of legal and fraud transactions from 1:1 to 10:1.

Considering random forest II is more practical, this experiment uses it to explore the influence of different ratio of legal and fraud transactions on fraud transactions detection. For any one of the above ten datasets, its transactions of each subset are divided into training subset and testing one according to the ratio of 7:3. The results of each group are shown in Fig. 3.

As shown in Fig. 3, the accuracy of fraud detection is increasing with the growth of legal transactions, while the recall is decreasing. However, the F-measure gets the maximum value when the ratio of legal and fraud transaction is 5:1. A perspective can be get that random forest II is suitable for the balance class problem and under-sampling method is effective to deal with class unbalance problem. The performance of fraud detection models is related to the ratio of legal and fraud transactions, and the best results should be obtained by practical testing.

E. Experiment III

This experiment is done via random forest II using a bigger and more closed to the actual application dataset in order to demonstrate fraud detection effectiveness.

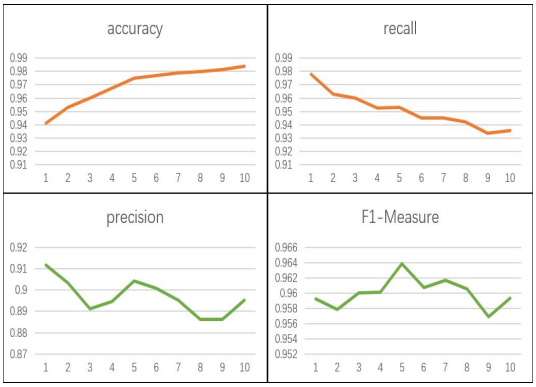

Fig. 3. Results on every group dataset.

TABLE V
RESULT OF FRAUD DETECTION MODEL

Measure / Models	Random Forest Model
Accuracy of transaction	98.67%
Precision of transaction	32.68%
Recall of transaction	59.62%
Intervention rate of transaction	1.48%
Coverage rate of customer	34.09%

The training set is extracted from the original dataset of November 2016 and December 2016. Similarly to experiment I, all the fraud transactions is used and legal transactions are randomly sampled to make the ratio of legal and fraud transactions is 5:1, which has been proved to be the best ratio for random forest II. The testing subset is all the transactions of January 2017, which has about 4.7 million legal transactions and 38 thousand fraud ones. Two performance measures, intervention rate of transaction and coverage rate of customer, are added which are defined by the company. Intervention rate is the ratio of fraud transactions signed by a model and all the tested transactions, which is an important measure to indicate the impact of fraud detection model on customers disturbance. Coverage rate of customer is to measure how many fraud customers the model can detect from all the fraud customers, which is the ration of detected fraud customers and all the fraud customers. The experimental results are shown in Table V.

We also apply other algorithms in our experiments, such as support vector machine, naive bayes an nerual network. But the results of them are worse than random forest. Due to the limited space, we don't describe them here.

V. Conclusions

This paper has examined the performance of two kinds of random forest models. A real-life B2C dataset on credit card transactions is used in our experiment. Although random forest obtains good results on small set data, there are still some problems such as imbalanced data. Our future work will focus on solving these problems. The algorithm of random forest itself should be improved. For example, the voting mechanism

assumes that each of base classifiers has equal weight, but some of them may be more important than others. Therefore, we also try to make some improvement for this algorithm.

VI. Acknowledgment

Authors would like to thank reviewers for their helpful comments. This paper is supported in part by the National Natural Science Foundation of China under grand no. 61572360 and in part by the Shanghai Shuguang Program under grant no. 15SG18. Corresponding author is G.J. Liu.

References

[1] Gupta, Shalini, and R. Johari. "A New Framework for Credit Card Transactions Involving Mutual Authentication between Cardholder and Merchant." International Conference on Communication Systems and Network Technologies IEEE, 2011:22-26.

[2] Y. Gmbh and K. G. Co, "Global online payment methods: Full year 2016," Tech. Rep., 3 2016.

[3] Bolton, Richard J., and J. H. David. "Unsupervised Profiling Methods for Fraud Detection." Proc Credit Scoring and Credit Control VII (2001):5–7.

[4] Seyedhossein, Leila, and M. R. Hashemi. "Mining information from credit card time series for timelier fraud detection." International Symposium on Telecommunications IEEE, 2011:619-624.

[5] Srivastava, A., Kundu, A., Sural, S., and Majumdar, A. (2008). Credit card fraud detection using hidden markov model. IEEE Transactions on Dependable and Secure Computing, 5(1), 37-48.

[6] Drummond, C., and Holte, R. C. (2003). C4.5, class imbalance, and cost sensitivity: why under-sampling beats oversampling. Proc of the Icml Workshop on Learning from Imbalanced Datasets II, 1–8.

[7] Quah, J. T. S., and Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. Expert Systems with Applications, 35(4), 1721-1732.

[8] Kundu, A., Panigrahi, S., Sural, S., and Majumdar, A. K. (2009). Blast-ssaha hybridization for credit card fraud detection. IEEE Transactions on Dependable and Secure Computing, 6(4), 309-315.

[9] Shi, E., Niu, Y., Jakobsson, M., and Chow, R. (2010). Implicit Authentication through Learning User Behavior. International Conference on Information Security (Vol.6531, pp.99-113). Springer-Verlag.

[10] Duman, E., and Ozcelik, M. H. (2011). Detecting credit card fraud by genetic algorithm and scatter search. Expert Systems with Applications, 38(10), 13057-13063.

[11] Bhattacharyya, S., Jha, S., Tharakunnel, K., and Westland, J. C. (2011). Data mining for credit card fraud: a comparative study. Decision Support Systems, 50(3), 602-613.

[12] Sahin, Y., and Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines. Lecture Notes in Engineering and Computer Science, 2188(1).

[13] Mota, G., Fernandes, J., and Belo, O. (2014). Usage signatures analysis an alternative method for preventing fraud in E-Commerce applications. International Conference on Data Science and Advanced Analytics (pp.203-208). IEEE.

[14] Behdad, M., Barone, L., Bennamoun, M., and French, T. (2012). Nature-inspired techniques in the context of fraud detection. IEEE Transactions on Systems Man and Cybernetics Part C, 42(6), 1273-1290.

[15] Ju, W. H., and Vardi, Y. (2001). A hybrid high-order markov chain model for computer intrusion detection. Journal of Computational and Graphical Statistics, 10(2), 277-295.

[16] Bolton, R. J., and Hand, D. J. (2002). Statistical fraud detection: a review. Statistical Science, 17(3), 235-249.

[17] Vlasselaer, V. V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., and Snoeck, M., et al. (2015). Apate : a novel approach for automated credit card transaction fraud detection using network-based extensions. Decision Support Systems, 75, 38-48.

[18] Chan, P. K., Fan, W., Prodromidis, A. L., and Stolfo, S. J. (2002). Distributed data mining in credit card fraud detection. IEEE Intelligent Systems and Their Applications, 14(6), 67-74.

[19] RONG-CHANG CHEN, TUNG-SHOU CHEN, and CHIH-CHIANG LIN. (2006). A new binary support vector system for increasing detection rate of credit card fraud. International Journal of Pattern Recognition and Artificial Intelligence, 20(02), 227-239.

[20] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[21] Dietterich, T. G. (2000). Ensemble methods in machine learning. , 1857(1), 1-15.

[22] Abeel, T., de Peer, Y. V. and Saeys, Y. Java-ML: A Machine Learning Library, Journal of Machine Learning Research, 2009, 10, 931-934

[23] Quinlan, J. R. (1986). Induction on decision tree. Machine Learning, 1(1), 81-106.

[24] Breiman, L., Friedman, J. H., Olshen, R., and Stone, C. J. (1984). Classification and regression trees. Biometrics, 40(3), 358.