International Journal of Engineering & Technology, 7 (2) (2018) 917-920

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i2.9356
Research paper

Machine Learning Approaches for Credit Card Fraud Detection

S. Venkata Suryanarayana [1] *, G. N. Balaji [1], G. Venkateswara Rao [2]

[1] Associate Professor, CVR College of Engineering, Hyderabad
[2] Associate Professor, GITAM University, Visakhapatnam
**Corresponding author E-mail: suryahcu@gmail.com*

Abstract

With the extensive use of credit cards, fraud appears as a major issue in the credit card business. It is hard to have some figures on the impact of fraud, since companies and banks do not like to disclose the amount of losses due to frauds. At the same time, public data are scarcely available for confidentiality issues, leaving unanswered many questions about what is the best strategy. Another problem in credit-card fraud loss estimation is that we can measure the loss of only those frauds that have been detected, and it is not possible to assess the size of unreported/undetected frauds. Fraud patterns are changing rapidly where fraud detection needs to be re-evaluated from a reactive to a proactive approach. In recent years, machine learning has gained lot of popularity in image analysis, natural language processing and speech recognition. In this regard, implementation of efficient fraud detection algorithms using machine-learning techniques is key for reducing these losses, and to assist fraud investigators. In this paper logistic regression, based machine learning approach is utilized to detect credit card fraud. The results show logistic regression based approaches outperforms with the highest accuracy and it can be effectively used for fraud investigators.

Keywords: *Machine Learning; Decision Tree; Neural Network; Logistic Regression; Precision and Recall.*

1. Introduction

In recent years credit card usage is predominant in modern day society and credit card fraud is keep on growing. Financial losses due to fraud affect not only merchants and banks (e.g. reimbursements), but also individual clients. If the bank loses money, customers eventually pay as well through higher interest rates, higher membership fees, etc. Fraud may also affect the reputation and image of a merchant causing non-financial losses that, though difficult to quantify in the short term, may become visible in the long period. For example, if a cardholder is victim of fraud with a certain company, he may no longer trust their business and choose a competitor. A Fraud Detection System (FDS) should not only detect fraud cases efficiently, but also be cost-effective in the sense that the cost invested in transaction screening should not be higher than the loss due to frauds [1]. Bhatla [2] shows that screening only 2% of transactions can result in reducing fraud losses accounting for 1% of the total value of transactions. However, a review of 30% of transactions could reduce the fraud losses drastically to 0.06%, but increase the costs exorbitantly. In order to minimize costs of detection it is important to use expert rules and statistical based models (e.g. Machine Learning) to make a first screen between genuine and potential fraud and ask the investigators to review only the cases with high risk. Typically, transactions are first filtered by checking some essential conditions (e.g. sufficient balance) and then scored by a predictive model as shown in Fig. 1. The predictive model scores each transaction with high or low risk of fraud and those with high risk generate alerts. Investigators check these alerts and provide a feedback for each alert, i.e. true positive (fraud) or false positive (genuine). These feedbacks can then be used to improve the model. A predictive model can be built upon experts' rules, i.e. rules based on knowledge from fraud experts, but these require manual tuning and human supervision. Alternatively, with Machine Learning (ML) techniques [3] we can efficiently discover fraudulent patterns and predict transactions that are probably to be fraudulent. ML techniques consist in inferring a prediction model on the basis of a set of examples. The model is in most cases a parametric function, which allows predicting the likelihood of a transaction to be fraud, given a set of features describing the transaction. In the domain of fraud detection, the use of learning techniques is attractive for a number of reasons. First, they allow to discovery patterns in high dimensional data streams, i.e. transactions arrive as a continuous stream and each transaction is defined by many variables. Second, fraudulent transactions are often correlated both over time and space. For example, fraudsters typically try to commit frauds in the same shop with different cards within a short time period. Third, learning techniques can be used to detect and model existing fraudulent strategies as well as identify new strategies associated to unusual behavior of the cardholders. Predictive models based on ML techniques are also able to automatically integrate investigators' feedbacks to improve the accuracy of the detection, while in the case of expert system, including investigators feedbacks requires rules revision that can be tedious and time consuming.

Copyright © 2018 S. Venkata Suryanarayana et. al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

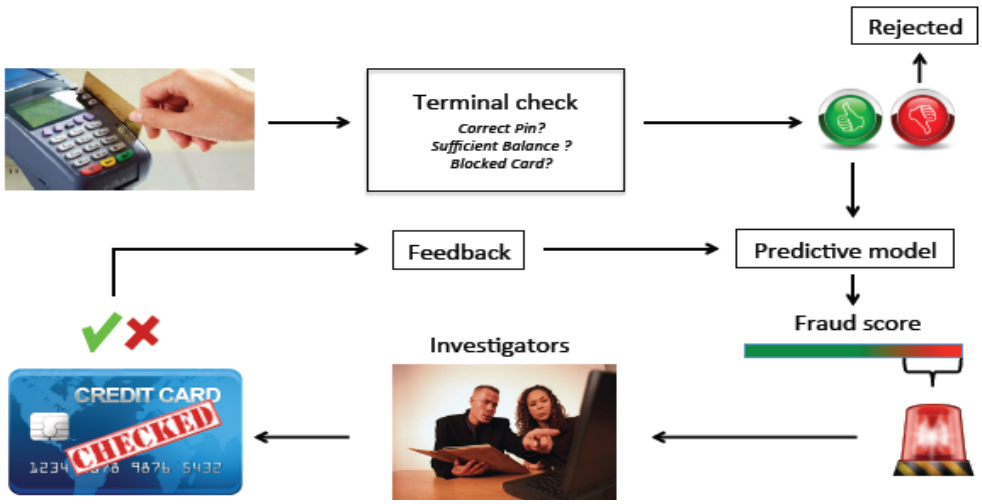


Fig. 1: The Credit Card Fraud Detection Process.

When a fraud cannot be prevented, it is desirable to identify. The number of domain constraints and characteristics exaggerate the problem of detection and prevention. Customer irritation is to be avoided. Most banks considers huge transactions, among which very few is fraudulent, often less than 0.1% [4]. Also, only a limited number of transactions can be checked by fraud investigators, i.e. we cannot ask a human person to check all transactions one by one if it is fraudulent or not.

2. Related work

Several machine learning techniques have been used in the literature to approach the credit card fraud detection problem. The authors in [5] developed models based on an individual and combined machine learning techniques for handwritten digits' recognition. The results showed that the classification accuracy of the combined classifier model outperformed the individual classifier model. In [6] the authors tried Bayesian Belief Networks (BBN) and Artificial Neural Networks (ANN) on a real dataset obtained from Europay International. Their experiment showed that the Bayesian Belief networks outperforms ANN in terms of classification accuracy and training time. It was found that ANN may need several hours for training while BNN takes only 20 minutes. However, the trained ANN was found to be faster in classifying new instances. In [7] ANN based model and decision trees model are compared, and the authors found that the ANN outperforms decision trees. In [8] decision trees and support vector machines (SVM) are applied on a dataset obtained from a real world national bank's credit card data warehouses. They found out that decision trees outperform SVM in solving the problem. The authors in [9] developed two models based on logistic regression and SVM [10]. It was found that logistic regression outperforms SVM. A fraud detection model based on the decision trees was developed in [11] and founded that decision tress suffer from under fitting problem in case of imbalanced data set (case of fraud detection dataset).

3. Methodology

Fraud detection is a binary classification task in which any transaction will be predicted and labeled as a fraud or legit. In this paper state of the art classification techniques were tried for this task and their performances were compared. The following subsections briefly explain these classification techniques, data set and metrics used for performance measure.

3.1. Classification techniques

1) Naïve Bayes Algorithm

Naïve Bayes is based on two assumptions. Firstly, all features in an entry that needs to be classified are contributing evenly in the decision (equally important). Secondly, all attributes are statistically independent, meaning that, knowing an attribute's value does not indicate anything about other attributes' values which is not always true in practice. The process of classifying an instance is done by applying the Bayes rule for each class given the instance. In the fraud detection task, the following formula is calculated for each of the two classes (fraudulent and legitimate) and the class associated with the higher probability is the predicted class for the instance.

2) Decision Tree Algorithm

There are two categories in decision trees, regression trees and classification trees. In this decision tree algorithm a decision tree is constructed using training dataset. A decision tree consists of nodes that forms a tree structure; the topmost node is called the root node. Each non-leaf node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf node holds a class label. Leaf nodes represent classes that are returned if reached as the final prediction by the model. In [7] the authors elaborated, given an instance with its features' values, the model is able to classify the instance by traversing the decision tree. There are several decision tree algorithms including C4.5, CART and ID3.

3) K-Nearest Neighbors Algorithm

The k-Nearest Neighbors (KNN) algorithm is a simple instance-based algorithm that plots all training instances and classify unlabelled instances based on their closest neighbors. In instance-based learners instances themselves are used to represent the model unlike the decision tree algorithms that use instances to develop a tree and that tree represents the model. However, it is argued that all learning algorithms are instance-based since they all use instances of the training set to construct models. In the KNN technique, an unlabelled instance is classified by calculating the distances between the instance and surrounding instances based on a determined distance metric and the majority class is assigned to the unlabelled instance.

4) Support vector machines (SVM)

SVM is introduced by Vapnik, in 1992 [12] to solve binary classification problems and then they are extended to nonlinear regression problems. SVMs are based on structural risk minimization unlike ANNs which is based on empirical risk minimization. SVM map the data to a predetermined very high- dimensional space via a kernel function and finds the hyperplane that maximizes the margin between the two classes. The solution is based only on those data points, which are at the margin. These points are called support vectors.

5) Logistic Regression

Logistic regression also does not require independent variables to be linearly related, nor does it require equal variance within each group, which also makes it a less stringent procedure for statistical analysis. As a result, logistic regression was used to predict the probability of fraudulent credit cards.

Assumptions and Limitations of Logistic Regression. Logistic regression analysis uses maximum likelihood estimation to predict group membership. However, to interpret the results of the prediction of group membership with precision and accuracy, a preliminary analysis of the cleaned dataset was conducted to observe if the assumptions of logistic regression were met.

6) Artificial Neural Networks

Artificial neural network (ANN) is a mathematical model for predicting system performance (i.e., system output) inspired by the structure and function of human biological neural networks. The ANN is developed and derived to have a function similar to the human brain by memorizing and learning various tasks and behaving accordingly. It is trained to predict specific behavior and to remember that behavior in the future like the human brain does. Its architecture also is similar to human neuron layers in the brain as far as functionality and inter-neuron connection. ANN has been successfully used in various applications.

3.2. Data set

The dataset used is cc Fraud dataset. The dataset has 1, 00,000 records and 9 attributes. They are: CustId ,Gender, State , Cardholder , Balance ,Number of transactions, Number of International transactions , Credit Line and Fraud Risk. The data set contains a real-life data of financial truncations of an e-commerce organization.

3.3. Data pre-processing

The dataset was pre-processed for the purpose of improving the performance of the classifiers and reducing their training and operating time. The pre-processing includes investigating the dataset feature space and handling the imbalanced nature of the dataset.

3.4. Performance metrics

A number of performance metrics could be used to report the performance of the fraud detection classifiers including the confusion matrix, Sensitivity, Specificity, false positive rate, balanced classification Rate and Matthews correlation coefficient.

Confusion matrix

A confusion matrix of a binary classifier is a table that shows the number of instances classified correctly/incorrectly in each class. Table 1 illustrates the confusion matrix [10] of a binary classifier. In the fraud detection problem, positive represents the legitimate transactions and negative represents fraudulent transactions.

Table 1: The Confusion Matrix of a Binary Classifier.

Actual	Predicted	
	Positive	Negative
Positive (Legit)	true positive (TP)	false positive (FP)
Negative (Fraud)	false negative (FN)	true negative (TN)

Specificity is defined as the number of fraud case predictions to the total number of fraud cases.

$$\text{Specificity} = TN / (TN + FP) \quad (1)$$

Sensitivity is defined as the number of legit predictions compared to the total number of legit transactions. In fraud detection, the most important measure is specificity or fraud detection rate, as a higher value of recall means a lowest financial loss to the company.

$$\text{Sensitivity} = TP / (TP + FN) \quad (2)$$

Accuracy gives the overall efficacy of the proposed system. It is defined as the total number of predictions to the total number of cases.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (3)$$

Accuracy of the model can be misleading in case of credit card fraud detection, where the numbers of fraudulent transactions is much lower than the legitimate transactions and the dataset is highly imbalanced.

Selecting the right performance metrics depends on the business objective because one measure can help to prevent financial losses and the other can help to gain customer satisfaction.

4. Experiments and results

The four fraud detection models were trained and tested using Weka. Weka is an abbreviation for “Waikato Environment for Knowledge Analysis”. Weka is a workbench for machine learning that implements the majority of data mining techniques and data pre-processing and filtering techniques. Weka tool was developed in Java language by University of Waikato in New Zealand.

We have used 0-fold, 5-fold, 10-fold, 15-fold and 20-fold cross validation in the process of training and testing the different models. The average performance results are then recorded. This methodological approach ensures that all data were represented once as a test data and several times as a training data producing accurate results.

Table 2 represents the performance of Decision Tree algorithm. Table 3 represents the performance of K-Nearest Neighbor algorithm. Table 4 represents the performance of Neural Network algorithm. Table 5 represents the performance of Logistic regression algorithm. It could be observed that the Logistic Regression algorithm outperformed other models in terms of the Accuracy.

Table.2: Performance of Decision Tree across Different Fraud Rates

Folds	Accuracy	Sensitivity	Specificity
0	0.94119706	94.30%	76%
5	0.94293	94.68%	71%
10	0.94476	94.70%	74%
15	0.9417	94.94%	65%
20	0.94776	94.77%	75%

Table 3: Performance of K-Nearest Neighbor across Different Fraud Rates

Folds	Accuracy	Sensitivity	Specificity
0	0.958897945	96.50%	76.90%
5	0.95838	96.70%	74.44%
10	0.95838	96.70%	74.40%
15	0.9585	96.71%	74.50%
20	0.95838	96.70%	74.44%

Table 4: Performance of Neural Network across Different Fraud Rates

Folds	Accuracy	Sensitivity	Specificity
0	0.769688	99.30%	69.70%
5	0.84369	97.70%	44.70%
10	0.93995	95.70%	68.50%
15	0.9382	96.40%	56.31%
20	0.92813	96.61%	54.33%

Table 5: Performance of Logistic Regression across Different Fraud Results

Folds	Accuracy	Sensitivity	Specificity
0	0.963198	96.90%	82.00%
5	0.96238	96.85%	80.49%
10	0.963198	96.80%	80.40%
15	0.9624	96.85%	80.62%
20	0.96238	96.85%	80.49%

We also presented the comparison of all these four algorithms as shown in Fig. 2 and Fig. 3. Fig. 2 represents the comparison using 15 folds of cross validation and Fig. 3 represents the comparison using 20 folds of cross validation. In both the figures blue color represents the accuracy metric, green color represents sensitivity metric and red color represents the specificity metric.

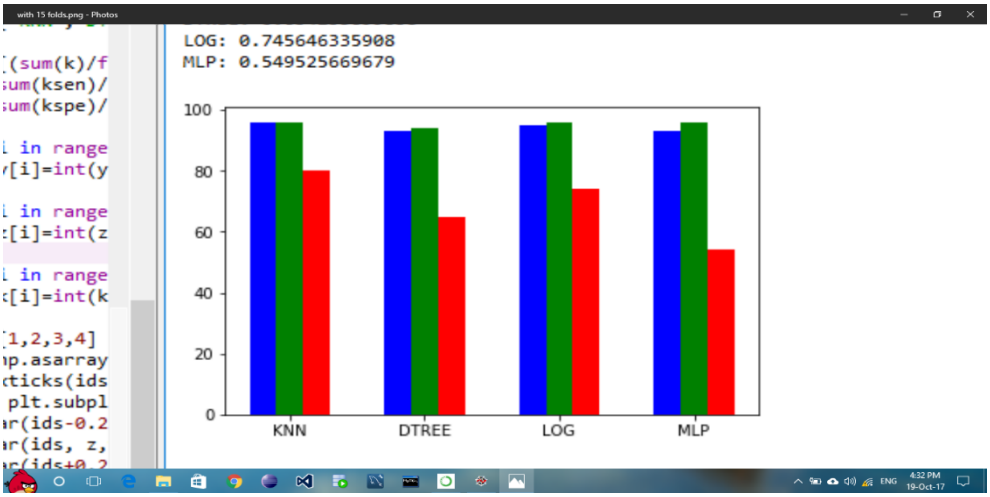


Fig. 2: Comparing the Results with 15 Folds.

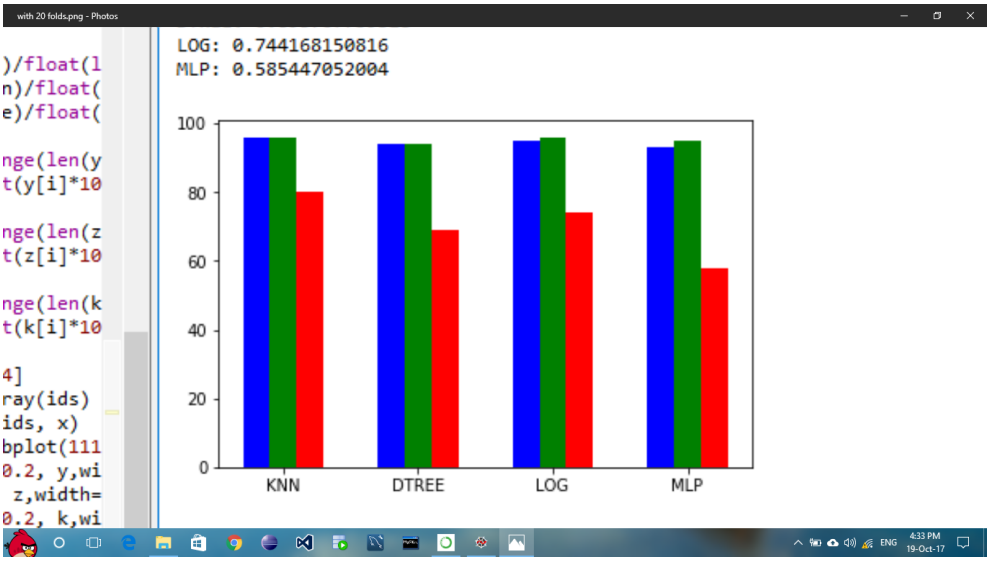


Fig. 3: Comparing the Results with 20 Folds.

5. Conclusion

Decision trees, K-Nearest Neighbors, Logistic Regression and Neural Network algorithms were used in developing four fraud detection models to classify a transaction as fraudulent or legitimate. Three metrics were used in evaluating their performances. The results showed that there is no data mining technique that is universally better than others. Performance improvement could be achieved through developing a fraud detection model using a combination of different data mining techniques (ensemble).

References

[1] Jon TS Quah and M Sriganesh. Real-time credit card fraud detection using computational
Intelligence. Expert Systems with Applications, 35(4):1721–1732, 2008. https://doi.org/10.1016/j.eswa.2007.08.093.

[2] Tej Paul Bhatla, Vikram Prabhu, and Amit Dua. Understanding credit card frauds. Cards business review, 1(6), 2003.

[3] Christopher M Bishop et al. Pattern recognition and machine learning, volume 4. Springer New York, 2006.

[4] Piotr Juszczak, Niall M Adams, David J Hand, Christopher Whitrow, and David J Weston. Off the peg and bespoke classifiers for fraud detection. Computational Statistics & Data Analysis, 52(9):4521–4532, 2008. https://doi.org/10.1016/j.csda.2008.03.014.

[5] Ng, G., & Singh, H. (1997). Democracy in pattern classifications: combinations of votes from various pattern classifiers. Nanyang, Singapore: Elsevier.

[6] Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. (2002). Credit Card Fraud Detection Using Bayesian and Neural Networks. Brussel, Belgium.

[7] Zaki, M., & Meira, W. (2014). Data Mining and Analysis: Fundamental Concepts and Algorithms. New York City, New York: Cambridge University Press. https://doi.org/10.1017/CBO9780511810114.

[8] Sahin, Y., & Duman, E. (2011). Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. Hong Kong, China: The International MultiConference of Engineers and Computer Scientists.

[9] Huang, S. (2013). Fraud Detection Model by Using Support Vector Machine Techniques. Chiayi, Taiwan: International Journal of Digital Content Technology & its Applications.

[10] Balaji, G. N., T. S. Subashini, and N. Chidambaram. "Detection of heart muscle damage from automated analysis of echocardiogram video." *IETE Journal of Research* 61.3 (2015): pp: 236-243. https://doi.org/10.1080/03772063.2015.1009403.

[11] Ehramikar, S. (2000). The Enhancement of Credit Card Fraud Detection Systems. Toronto, Canada: Master of Applied Science Thesis, University of Toronto. V. Vapnik. Statistical Learning Theory Wiley, New York, 1998.