IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2017.Doi Number*

# Credit card fraud detection using AdaBoost and majority voting

**Kuldeep Randhawa[1], Chu Kiong Loo[1], *Senior Member, IEEE*, Manjeevan Seera[2,3], *Senior Member, IEEE*, Chee Peng Lim[4], Asoke K. Nandi[5,6], *Fellow, IEEE***

[1]Faculty of Computer Science and Information Technology, University Malaya, Kuala Lumpur, Malaysia
[2]Faculty of Engineering, Tunku Abdul Rahman University College, Kuala Lumpur, Malaysia
[3]Faculty of Engineering, Computing and Science, Swinburne University of Technology (Sarawak Campus), Malaysia
[4]Institute for Intelligent Systems Research and Innovation, Deakin University, Geelong, Victoria, Australia
[5]Department of Electronic and Computer Engineering, Brunel University London, Uxbridge, UB8 3PH, United Kingdom
[6]The Key Laboratory of Embedded Systems and Service Computing, College of Electronic and Information Engineering, Tongji University, Shanghai, China

Corresponding author: Chu Kiong Loo (e-mail: ckloo.um@um.edu.my).

**ABSTRACT** Credit card fraud is a serious problem in financial services. Billions of dollars are lost due to credit card fraud every year. There is a lack of research studies on analyzing real-world credit card data owing to confidentiality issues. In this paper, machine learning algorithms are used to detect credit card fraud. Standard models are firstly used. Then, hybrid methods which use AdaBoost and majority voting methods are applied. To evaluate the model efficacy, a publicly available credit card data set is used. Then, a real-world credit card data set from a financial institution is analyzed. In addition, noise is added to the data samples to further assess the robustness of the algorithms. The experimental results positively indicate that the majority voting method achieves good accuracy rates in detecting fraud cases in credit cards.

**INDEX TERMS** AdaBoost; classification; credit card; fraud detection; predictive modelling; voting.

## I. INTRODUCTION

Fraud is a wrongful or criminal deception aimed to bring financial or personal gain [1]. In avoiding loss from fraud, two mechanisms can be used: fraud prevention and fraud detection. Fraud prevention is a proactive method, where it stops fraud from happening in the first place. On the other hand, fraud detection is needed when a fraudulent transaction is attempted by a fraudster.

Credit card fraud is concerned with the illegal use of credit card information for purchases. Credit card transactions can be accomplished either physically or digitally [2]. In physical transactions, the credit card is involved during the transactions. In digital transactions, this can happen over the telephone or the internet. Cardholders typically provide the card number, expiry date, and card verification number through telephone or website.

With the rise of e-commerce in the past decade, the use of credit cards has increased dramatically [3]. The number of credit card transactions in 2011 in Malaysia were at about 320 million, and increased in 2015 to about 360 million. Along with the rise of credit card usage, the number of fraud cases have been constantly increased. While numerous authorization techniques have been in place, credit card fraud cases have not hindered effectively. Fraudsters favour the internet as their identity and location are hidden. The rise in credit card fraud has a big impact on the financial industry. The global credit card fraud in 2015 reached to a staggering USD $21.84 billion [4].

Loss from credit card fraud affects the merchants, where they bear all costs, including card issuer fees, charges, and administrative charges [5]. Since the merchants need to bear the loss, some goods are priced higher, or discounts and incentives are reduced. Therefore, it is imperative to reduce the loss, and an effective fraud detection system to reduce or eliminate fraud cases is important. There have been various studies on credit card fraud detection. Machine learning and related methods are most commonly used, which include artificial neural networks, rule-induction techniques, decision trees, logistic regression, and support vector machines [1]. These methods are used either standalone or by combining several methods together to form hybrid models.

In this paper, a total of twelve machine learning algorithms are used for detecting credit card fraud. The algorithms range from standard neural networks to deep learning models. They are evaluated using both benchmark and real-world credit card data sets. In addition, the AdaBoost and majority voting methods are applied for forming hybrid models. To further evaluate the robustness and reliability of the models, noise is added to the real-world data set. The key contribution of this paper is the evaluation of a variety of machine learning models with a real-world credit card data set for fraud detection. While other researchers have used various methods on publicly available data sets, the data set used in this paper are extracted from actual credit card transaction information over three months.

The organization of this paper is as follows. In Section II, related studies on single and hybrid machine learning algorithms for financial applications is given. The machine learning algorithms used in this study are presented in Section III. The experiments with both benchmark and real-world credit card data sets are presented in Section IV. Concluding remarks and recommendations for further work are given in Section V.

## II. RELATED STUDIES

In this section, single and hybrid machine learning algorithms for financial applications are reviewed. Various financial applications from credit card fraud to financial statement fraud are reviewed.

### A. SINGLE MODELS

For credit card fraud detection, Random Forest (RF), Support Vector Machine, (SVM) and Logistic Regression (LOR) were examined in [6]. The data set consisted of one-year transactions. Data under-sampling was used to examine the algorithm performances, with RF demonstrating a better performance as compared with SVM and LOR [6]. An Artificial Immune Recognition System (AIRS) for credit card fraud detection was proposed in [7]. AIRS is an improvement over the standard AIS model, where negative selection was used to achieve higher precision. This resulted in an increase of accuracy by 25% and reduced system response time by 40% [7].

A credit card fraud detection system was proposed in [8], which consisted of a rule-based filter, Dumpster–Shafer adder, transaction history database, and Bayesian learner. The Dempster–Shafer theory combined various evidential information and created an initial belief, which was used to classify a transaction as normal, suspicious, or abnormal. If a transaction was suspicious, the belief was further evaluated using transaction history from Bayesian learning [8]. Simulation results indicated a 98% true positive rate [8]. A modified Fisher Discriminant function was used for credit card fraud detection in [9]. The modification made the traditional functions to become more sensitive to important instances. A weighted average was utilized to calculate variances, which allowed learning of profitable transactions. The results from the modified function confirm it can eventuate more profit [9].

Association rules are utilized for extracting behavior patterns for credit card fraud cases in [10]. The data set focused on retail companies in Chile. Data samples were de-fuzzified and processed using the Fuzzy Query 2+ data mining tool [10]. The resulting output reduced excessive number of rules, which simplified the task of fraud analysts [10]. To improve the detection of credit card fraud cases, a solution was proposed in [11]. A data set from a Turkish bank was used. Each transaction was rated as fraudulent or otherwise. The misclassification rates were reduced by using the Genetic Algorithm (GA) and scatter search. The proposed method doubled the performance, as compared with previous results [11].

Another key financial loss is related to financial statement fraud. A number of methods including SVM, LOR, Genetic Programming (GP) and Probabilistic Neural Network (PNN) were used to identify financial statement fraud [12]. A data set involving 202 Chinese companies was used. The t-statistic was used for feature subset selection, where 18 and 10 features were selected in two cases. The results indicated that the PNN performed the best, which was followed by GP [12]. Decision Trees (DT) and Bayesian Belief Networks (BNN) were used in [13] to identify financial statement fraud. The input comprised the ratios taken from financial statements of 76 Greek manufacturing firms. A total of 38 financial statements were verified to be fraud cases by auditors. The BBN achieved the best accuracy of 90.3% accuracy, while DT achieved 73.6% [13].

A computational fraud detection model (CFDM) was proposed in [14] to detect financial reporting fraud. It utilized textual data for fraud detection. Data samples from 10-K filings at Security and Exchange Commission were used. The CDFM model managed to distinguish fraudulent filings from non-fraudulent ones [14]. A fraud detection method based on user accounts visualization and threshold-type detection was proposed in [15]. The Self-Organizing Map (SOM) was used as a visualization technique. Real-world data sets related to telecommunications fraud, computer network intrusion, and credit card fraud were evaluated. The results were displayed with visual appeal to data analysts as well as non-experts, as high-dimensional data samples were projected in a simple 2-dimensional space using the SOM [15].

Fraud detection and understanding spending patterns to uncover potential fraud cases was detailed in [16]. It used the SOM to interpret, filter, and analyze fraud behaviors. Clustering was used to identify hidden patterns in the input data. Then, filters were used to reduce the total cost and processing time. By setting appropriate numbers of neurons and iteration steps, the SOM was able to converge fast. The resulting model appeared to be an efficient and a cost-effective method [16].

### B. HYBRID MODELS

Hybrid models are combination of multiple individual models. A hybrid model consisting of the Multilayer Perceptron (MLP) neural network, SVM, LOR, and Harmony Search (HS) optimization was used in [17] to detect corporate tax evasion. HS was useful for finding the best parameters for the classification models. Using data from the food and textile sectors in Iran, the MLP with HS optimization acquired the highest accuracy rates at 90.07% [17]. A hybrid clustering system with outlier detection capability was used in [18] to detect fraud in lottery and online games. The system aggregated online algorithms with statistical information from the input data to identify a number of fraud types. The training data set was compressed into the main memory while new data samples could be incrementally added into the stored data-cubes. The system achieved a high detection rate at 98%, with a 0.1% false alarm rate [18].

To tackle financial distress, clustering and classifier ensemble methods were used to form hybrid models in [19]. The SOM and k-means algorithms were used for clustering, while LOR, MLP, and DT were used for classification. Based on these methods, a total of 21 hybrid models with different combinations were created and evaluated with the data set. The SOM with the MLP classifier performed the best, yielding the highest prediction accuracy [19]. An integration of multiple models, i.e. RF, DR, Roush Set Theory (RST), and back-propagation neural network was used in [20] to build a fraud detection model for corporate financial statements. Company financial statements in period of 1998 to 2008 were used as the data set. The results showed that the hybrid model of RF and RST gave the highest classification accuracy [20].

Methods to identify automobile insurance fraud were described in [21] and [22]. A principal component analysis (PCA)-based (PCA) RF model coupled with the potential nearest neighbour method was proposed in [21]. The traditional majority voting in RF was replaced with the potential nearest neighbour method. A total of 12 different data sets were used in the experimental study. The PCA-based model produced a higher classification accuracy and a lower variance, as compared with those from RF and DT methods [21]. The GA with fuzzy c-means (FCM) was proposed in [22] for identification of automobile insurance fraud. The test records were separated into genuine, malicious or suspicious classes based on the clusters formed. By discarding the genuine and fraud records, the suspicious cases were further analyzed using DT, SVM, MLP, and a Group Method of Data Handling (GMDH). The SVM yielded the highest specificity and sensitivity rates [22].

## III. MACHINE LEARNING ALGORITHMS

A total of twelve algorithms are used in this experimental study. They are used in conjunction with the AdaBoost and majority voting methods. The details are as follows.

### A. ALGORITHMS

Naïve Bayes (NB) uses the Bayes' theorem with strong or naïve independence assumptions for classification. Certain features of a class are assumed to be not correlated to others. It requires only a small training data set for estimating the means and variances is needed for classification.

The presentation of data in form of a tree structure is useful for ease of interpretation by users. The Decision Tree (DT) is a collection of nodes that creates decision on features connected to certain classes. Every node represents a splitting rule for a feature. New nodes are established until the stopping criterion is met. The class label is determined based on the majority of samples that belong to a particular leaf. The Random Tree (RT) operates as a DT operator, with the exception that in each split, only a random subset of features is available. It learns from both nominal and numerical data samples. The subset size is defined using a subset ratio parameter.

The Random Forest (RF) creates an ensemble of random trees. The user sets the number of trees. The resulting model employs voting of all created trees to determine the final classification outcome. The Gradient Boosted Tree (GBT) is an ensemble of classification or regression models. It uses forward-learning ensemble models, which obtain predictive results using gradually improved estimations. Boosting helps improve the tree accuracy. The Decision Stump (DS) generates a decision tree with a single split only. It can be used in classifying uneven data sets.

The MLP network consists of at least three layers of nodes, i.e., input, hidden, and output. Each node uses a non-linear activation function, with the exception of the input nodes. It uses the supervised backpropagation algorithm for training. The version of MLP used in this study is able to adjust the learning rate and hidden layer size automatically during training. It uses an ensemble of networks trained in parallel with different rates and number of hidden units.

The Feed-Forward Neural Network (NN) uses the backpropagation algorithm for training as well. The connections between the units do not form a directed cycle, and information only moves forward from the input nodes to the output nodes, through the hidden nodes. Deep Learning (DL) is based on an MLP network trained using a stochastic gradient descent with backpropagation. It contains a large number of hidden layers consisting of neurons with tanh, rectifier, and maxout activation functions. Every node captures a copy of the global model parameters on local data, and contributes periodically toward the global model using model averaging.

Linear Regression (LIR) models the relationship between scalar variables by fitting a linear equation to the observed data. The relationships are modelled using linear predictor functions, with unknown model parameters estimated from the data set. The Akaike criterion, a measure of relative goodness of fit for a statistical model, is used for model selection. Logistic Regression (LOR) can handle data with both nominal

and numerical features. It estimates the probability of a binary response based on one or more predictor features.

The SVM can tackle both classification and regression data. SVM builds a model by assigning new samples to one category or another, creating a non-probabilistic binary linear classifier. It represents the data samples as points in the space mapped so such that the data samples of different categories can be separated by a margin as wide as possible. A summary of the strengths and limitations of the methods discussed earlier is given in Table I.

TABLE I
STRENGTHS AND LIMITATIONS OF MACHINE LEARNING METHODS

| Model | Strengths | Limitations |
|---|---|---|
| Bayesian | Good for binary classification problems; efficient use of computational resources; suitable for real-time operations. | Need good understanding of typical and abnormal behaviors for different types of fraud cases |
| Trees | Easy to understand and implement; the procedures require a low computational power; suitable for real-time operations. | Potential of over-fitting if the training set does not represent the underlying domain information; re-training is required for new types of fraud cases. |
| Neural Network | Suitable for binary classification problems, and widely used for fraud detection. | Need a high computational power, unsuitable for real-time operations; re-training is required for new types of fraud cases. |
| Linear Regression | Provide optimal results when the relationship between independent and dependent variables are almost linear. | Sensitive to outliers and limited to numeric values only. |
| Logistic Regression | Easy to implement, and historically used for fraud detection. | Poor classification performances as compared with other data mining methods. |
| Support Vector Machine | Able to solve non-linear classification problems; require a low computational power; suitable for real-time operations. | Not easy to process the results due to transformation of the input data. |

### B. MAJORITY VOTING

Majority voting is frequently used in data classification, which involves a combined model with at least two algorithms. Each algorithm makes its own prediction for every test sample. The final output is for the one that receives the majority of the votes, as follows.

Consider $K$ target classes (or labels), with $C_i, \forall i \in \Lambda = \{1, 2, \dots, K\}$ represents the $i$-th target class predicted by a classifier. Given an input $\boldsymbol{x}$, each classifier provides a prediction with respect to the target class, yielding a total of $K$ prediction, i.e., $P_1, \dots, P_K$. Majority voting aims to produce a combined prediction for input $x$, $P(\mathbf{x}) = j, j \in \Lambda$ from all the $K$ predictions, i.e., $p_k(\mathbf{x}) = j_k, k = 1, \dots, K$. A binary function can be used to represent the votes, *i.e.*,

$$V_k(\mathbf{x} \in C_i) = \begin{cases} 1, \text{if } p_k(\mathbf{x}) = i, i \in \Lambda \\ 0, \text{otherwise} \end{cases} \quad (1)$$

Then, sum the votes from all $K$ classifiers for each $C_i$, and the label that receives the highest vote is the final (combined) predicted class.

### C. ADABOOST

Adaptive Boosting or AdaBoost is used in conjunction with different types of algorithms to improve their performance. The outputs are combined by using a weighted sum, which represents the combined output of the boosted classifier, i.e.,

$$F_T(x) = \sum_{t=1}^{T} f_t(x) \quad (2)$$

where every $f_t$ is a classifier (weak learner) that returns the predicted class with respect to input $\boldsymbol{x}$. Each weak learner gives an output prediction, $h(x_i)$, for every training sample. In every iteration $t$, the weak learner is chosen, and is allotted a coefficient, $\alpha_t$, so that the training error sum, $E_t$, of the resulting $t$-stage boosted classifier is minimized,

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)] \quad (3)$$

where $F_{t-1}(x)$ is the boosted classifier built in the previous stage, $E(F)$ is the error function, and $f_t(x) = \alpha_t h(x)$ is weak learner taken into consideration for the final classifier.

AdaBoost tweaks weak learners in favor of misclassified data samples. It is, however, sensitive to noise and outliers. As long as the classifier performance is not random, AdaBoost is able to improve the individual results from different algorithms.

## IV. EXPERIMENTS

In this section, the experimental setup is firstly detailed. This is followed by a benchmark evaluation using a publicly available data set. The real-world credit card data set is then evaluated. All experiments have been conducted using RapidMiner Studio 7.6. The standard settings for all parameters in RapidMiner have been used. A 10-fold cross-validation (CV) has been used in the experiments as it can reduce the bias associated with random sampling in the evaluation stage [23].

### A. EXPERIMENTAL SETUP

In the credit card data set, the number of fraudulent transactions is usually a very small as compared with the total number of transactions. With a skewed data set, the resulting accuracy does not present an accurate representation of the system performance. Misclassifying a legitimate transaction causes poor customer services, and

failing to detect fraud cases causes loss to the financial institution and customers. This data imbalance problem causes performance issues in machine learning algorithms. The class with the majority samples influences the results. Under-sampling has been used by Bhattacharyya et al. [6], Duman et al. [24], and Phua et al. [25] to handle data imbalance problems. As such, under-sampling is used in this paper to handle the skewed data set.

While there is no best way of describing the true and false positives and negatives using one indicator, the best general measure is the Matthews Correlation Coefficient (MCC) [26]. MCC measures the quality of a two-class problem, which takes into account the true and false positives and negatives. It is a balanced measure, even when the classes are from different sizes. MCC can be calculated using:

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (4)$$

where the result of +1 indicates a perfect prediction, and −1 a total disagreement.

### B. BENCHMARK DATA

A publicly available data set is downloaded from [27]. It contains a total of 284,807 transactions made in September 2013 by European cardholders. The data set contains 492 fraud transactions, which is high imbalanced. Due to the confidentiality issue, a total of 28 principal components based on transformation are provided. Only the time and the amount data are not transformed, and are provided as such.

The results from various models are shown in Table II. It can be seen that the accuracy rates are high, generally around 99%. This however is not the real outcome, as the rate of fraud detection varies from 32.5% for RT up to 83% for NB. The rate of non-fraud detection is similar to the accuracy rates, i.e., the non-fraud results dominate the accuracy rates. SVM produces the highest MCC score of 0.813, while the lowest is from NB with an MCC score of 0.219.

TABLE II
RESULTS OF VARIOUS INDIVIDUAL MODELS

| Model | Accuracy | Fraud | Non-fraud | MCC |
|---|---|---|---|---|
| NB | 97.705% | 83.130% | 97.730% | 0.219 |
| DT | 99.919% | 81.098% | 99.951% | 0.775 |
| RF | 99.889% | 42.683% | 99.988% | 0.604 |
| GBT | 99.903% | 81.098% | 99.936% | 0.746 |
| DS | 99.906% | 66.870% | 99.963% | 0.711 |
| RT | 99.866% | 32.520% | 99.982% | 0.497 |
| DL | 99.924% | 81.504% | 99.956% | 0.787 |
| NN | 99.935% | 82.317% | 99.966% | 0.812 |
| MLP | 99.933% | 80.894% | 99.966% | 0.806 |
| LIR | 99.906% | 54.065% | 99.985% | 0.683 |
| LOR | 99.926% | 79.065% | 99.962% | 0.786 |
| SVM | 99.937% | 79.878% | 99.972% | 0.813 |

In addition to the standard models, AdaBoost has been used with all 12 models. The results are shown in Table III. It can be seen that the accuracy and non-fraud detection rates are similar to those without AdaBoost. However, the fraud detection rates increase from 79.8% to 82.3% for SVM. Some models suffer a minor reduction in the fraud detection rate up to 1%. The MCC rates show very minor changes, in which NB is able to improve its MCC score from 0.219 to 0.235.

TABLE III
RESULTS OF ADABOOST

| Model | Accuracy | Fraud | Non-fraud | MCC |
|---|---|---|---|---|
| NB | 98.038% | 82.520% | 98.064% | 0.235 |
| DT | 99.919% | 81.098% | 99.951% | 0.775 |
| RF | 99.889% | 42.683% | 99.988% | 0.604 |
| GBT | 99.903% | 81.707% | 99.935% | 0.747 |
| DS | 99.906% | 66.870% | 99.963% | 0.711 |
| RT | 99.866% | 32.520% | 99.982% | 0.497 |
| DL | 99.915% | 79.878% | 99.950% | 0.765 |
| NN | 99.933% | 81.301% | 99.965% | 0.807 |
| MLP | 99.933% | 80.894% | 99.966% | 0.806 |
| LIR | 99.907% | 54.472% | 99.985% | 0.686 |
| LOR | 99.926% | 79.065% | 99.962% | 0.786 |
| SVM | 99.927% | 82.317% | 99.957% | 0.796 |

Based on the models that produce good rates in Table II, the majority voting method is applied to the models. A total of 7 models are reported in Table IV. The accuracy rates are all above 99%, with DS+GBT yields a perfect non-fraud rate. The best fraud detection rate is achieved by NN+NB at 78.8%. The highest MCC score at 0.823 is yielded by NN+NB, which is higher than those form individual models.

TABLE IV
RESULTS OF MAJORITY VOTING

| Model | Accuracy | Fraud | Non-fraud | MCC |
|---|---|---|---|---|
| DS+GBT | 99.848% | 11.992% | 100.000% | 0.343 |
| DT+DS | 99.850% | 14.024% | 99.998% | 0.361 |
| DT+GBT | 99.920% | 60.366% | 99.988% | 0.737 |
| DT+NB | 99.932% | 72.967% | 99.978% | 0.788 |
| NB+GBT | 99.919% | 66.463% | 99.976% | 0.742 |
| NN+NB | 99.941% | 78.862% | 99.978% | 0.823 |
| RF+GBT | 99.865% | 23.780% | 99.996% | 0.468 |

For performance comparison, the results presented in Saia and Carta [28] are used, which used the same data set with a 10-fold CV evaluation. The results are shown in Table V. Two models were used in [28], one from the Frequency Domain (FD) and another with Random Forest (RF). The sensitivity rate as defined in [28] measures the number of transactions correctly classified as legitimate, which is the same as the non-fraud detection rate in Tables II to IV. The best accuracy and sensitivity acquired by RF are at 95% and 91%, respectively, as shown in Table V. In comparison, the best accuracy and non-fraud (sensitivity) from the experiments in this paper are above 99% for most of the individual models.

TABLE V
PERFORMANCE COMPARISON WITH RESULTS EXTRACTED FROM [28]

| Model | Accuracy | Sensitivity |
|---|---|---|
| FD | 77% | 76% |
| RF | 95% | 91% |

### C. REAL-WORLD DATA

A real credit card data set from a financial institution in Malaysia is used in the experiment. It is based on cardholders from the South-East Asia region from February to April 2017. A total of 287,224 transactions are recorded, with 102 of them classified as fraud cases. The data consist of a time series of transactions. To comply with customer confidentiality requirements, no personal identifying information is used. The features used in the experiment are given in Table VI.

TABLE VI
FEATURES IN CREDIT CARD DATA

| Code | Description |
|---|---|
| DE002 | Primary account number (PAN) |
| DE004 | Amount, transaction |
| DE006 | Amount, cardholder billing |
| DE011 | System trace audit number |
| DE012 | Time, local transaction |
| DE013 | Date, local transaction |
| DE018 | Merchant type |
| DE022 | Point of service entry mode |
| DE038 | Authorization identification response |
| DE049 | Currency code, transaction (ISO 4217) |
| DE051 | Currency code, cardholder billing (ISO 4217) |

A total of 11 features are used. The codes used are based on the standard ISO 8583 [29], while the last two codes are based on ISO 4217. As PAN is a 16-digit credit card number, a running sequence of numbers is used to mask the real numbers, in order to protect the personal information of customers. The results from various individual models are shown in Table VII. All accuracy rates are above 99%, with the exception of SVM at 95.5%. The non-fraud detection rates of NB, DT, and LIR are at 100%, while the rest are close to perfect, with the exception of SVM. The best MCC rates are from NB, DT, RF, and DS, at 0.990. The fraud detection rates vary from 7.4% for LIR up to 100% for RF, GBT, DS, NN, MLP, and LOR.

TABLE VII
RESULTS OF VARIOUS INDIVIDUAL MODELS

| Model | Accuracy | Fraud | Non-fraud | MCC |
|---|---|---|---|---|
| NB | 99.999% | 98.039% | 100.000% | 0.990 |
| DT | 99.999% | 98.039% | 100.000% | 0.990 |
| RF | 99.999% | 100.000% | 99.999% | 0.990 |
| GBT | 99.999% | 100.000% | 99.999% | 0.986 |
| DS | 99.999% | 100.000% | 99.999% | 0.990 |
| RT | 99.992% | 80.392% | 99.999% | 0.886 |
| DL | 99.985% | 93.137% | 99.987% | 0.819 |
| NN | 99.997% | 100.000% | 99.997% | 0.963 |
| MLP | 99.997% | 100.000% | 99.997% | 0.954 |
| LIR | 99.965% | 7.407% | 100.000% | 0.272 |
| LOR | 99.999% | 100.000% | 99.999% | 0.981 |
| SVM | 95.564% | 9.804% | 95.595% | 0.005 |

Similar to the benchmark experiment, AdaBoost has been used with all individual models. The results are shown in Table VIII. The accuracy and non-fraud detection rates are similar to those without AdaBoost. AdaBoost helps improve the fraud detection rates, with a noticeable difference for NB, DT, RT, which produce a perfect accuracy rate. The most significant improvement is achieved by LIR, i.e., from 7.4% to 94.1% accuracy. This clearly indicates the usefulness of AdaBoost in improvement the performance of individual classifiers. The best MCC score of 1 are achieved by NB and RF.

TABLE VIII
RESULTS OF ADABOOST

| Model | Accuracy | Fraud | Non-fraud | MCC |
|---|---|---|---|---|
| NB | 100.000% | 100.000% | 100.000% | 1.000 |
| DT | 99.999% | 100.000% | 99.999% | 0.990 |
| RF | 100.000% | 100.000% | 100.000% | 1.000 |
| GBT | 99.999% | 100.000% | 99.999% | 0.986 |
| DS | 99.999% | 100.000% | 99.999% | 0.990 |
| RT | 100.000% | 100.000% | 100.000% | 0.995 |
| DL | 99.994% | 96.078% | 99.995% | 0.917 |
| NN | 99.998% | 100.000% | 99.998% | 0.967 |
| MLP | 99.996% | 100.000% | 99.996% | 0.950 |
| LIR | 99.992% | 94.118% | 99.994% | 0.890 |
| LOR | 99.999% | 100.000% | 99.999% | 0.981 |
| SVM | 99.959% | 1.961% | 99.994% | 0.044 |

The majority voting method is then applied to the same models used in the benchmark experiment. The results are shown in Table IX. The accuracy and non-fraud detection rates are perfect, or near perfect. DS+GBT, DT+DS, DT+GBT, and RF+GBT achieve a perfect fraud detection rate. The MCC scores are close to or at 1. The results of majority voting are better than those of individual models.

TABLE IX
RESULTS OF MAJORITY VOTING

| Model | Accuracy | Fraud | Non-fraud | MCC |
|---|---|---|---|---|
| DS+GBT | 100.000% | 100.000% | 100.000% | 0.995 |
| DT+DS | 100.000% | 100.000% | 100.000% | 0.995 |
| DT+GBT | 100.000% | 100.000% | 100.000% | 1.000 |
| DT+NB | 99.999% | 98.039% | 100.000% | 0.990 |
| NB+GBT | 99.999% | 98.039% | 100.000% | 0.990 |
| NN+NB | 99.998% | 95.098% | 100.000% | 0.975 |
| RF+GBT | 99.999% | 100.000% | 99.999% | 0.990 |

To further evaluate the robustness of the machine learning algorithms, all real-world data samples are corrupted noise, at 10%, 20% and 30%. Noise is added to all data features. Figure 1 shows the fraud detection rate while Figure 2 shows the MCC score. It can be seen that with the addition of noise, the fraud detection rate and MCC rates deteriorate, as expected. The worst performance, i.e. the largest decrease in accuracy and MCC, is from majority voting of DT+NB and NB+GBT. DS+GBT, DT+DS and DT+GBT show gradual performance degradation, but their accuracy rates are still above 90% even with 30% noise in the data set.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2806420, IEEE Access



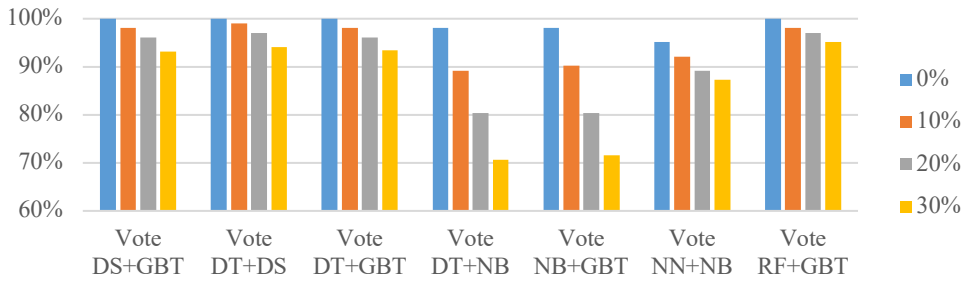


**FIGURE 1. Fraud detection rates with different percentages of noise**

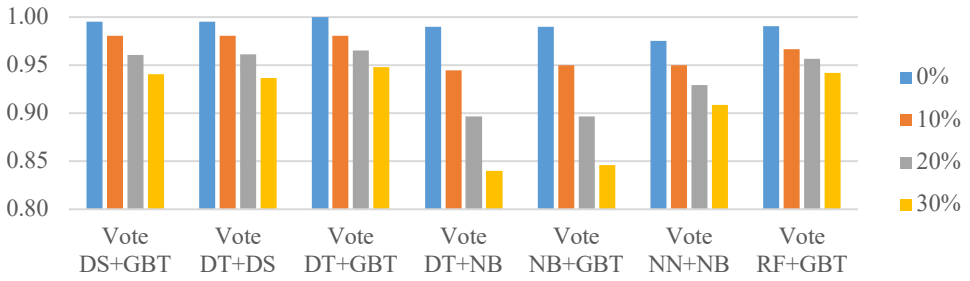


**FIGURE 2. MCC scores with different percentages of noise**

## V. CONCLUSIONS

A study on credit card fraud detection using machine learning algorithms has been presented in this paper. A number of standard models which include NB, SVM, and DL have been used in the empirical evaluation. A publicly available credit card data set has been used for evaluation using individual (standard) models and hybrid models using AdaBoost and majority voting combination methods. The MCC metric has been adopted as a performance measure, as it takes into account the true and false positive and negative predicted outcomes. The best MCC score is 0.823, achieved using majority voting. A real credit card data set from a financial institution has also been used for evaluation. The same individual and hybrid models have been employed. A perfect MCC score of 1 has been achieved using AdaBoost and majority voting methods. To further evaluate the hybrid models, noise from 10% to 30% has been added into the data samples. The majority voting method has yielded the best MCC score of 0.942 for 30% noise added to the data set. This shows that the majority voting method is stable in performance in the presence of noise.

For future work, the methods studied in this paper will be extended to online learning models. In addition, other online learning models will be investigated. The use of online learning will enable rapid detection of fraud cases, potentially in real-time. This in turn will help detect and prevent fraudulent transactions before they take place, which will reduce the number of losses incurred every day in the financial sector.

## REFERENCES

[1] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916–5923, 2013.

[2] A. O. Adewumi and A. A. Akinyelu, "A survey of machine-learning and nature-inspired based credit card fraud detection techniques," *International Journal of System Assurance Engineering and Management*, vol. 8, pp. 937–953, 2017.

[3] A. Srivastava, A. Kundu, S. Sural, A. Majumdar, "Credit card fraud detection using hidden Markov model," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 1, pp. 37–48, 2008.

[4] The Nilson Report (October 2016) [Online]. Available: https://www.nilsonreport.com/upload/content_promo/The_Nilson_Report_10-17-2016.pdf

[5] J. T. Quah, and M. Sriganesh, "Real-time credit card fraud detection using computational intelligence," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1721–1732, 2008.

[6] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C., "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[7] N. S. Halvaiee and M. K. Akbari, "A novel model for credit card fraud detection using Artificial Immune Systems," *Applied Soft Computing*, vol. 24, pp. 40–49, 2014.

[8] S. Panigrahi, A. Kundu, S. Sural, and A. K. Majumdar, "Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning," *Information Fusion*, vol. 10, no. 4, pp. 354–363, 2009.

[9] N. Mahmoudi and E. Duman, "Detecting credit card fraud by modified Fisher discriminant analysis," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2510–2516, 2015.

[10] D. Sánchez, M. A. Vila, L. Cerda, and J. M. Serrano, "Association rules applied to credit card fraud detection," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3630–3640, 2009.

[11] E. Duman and M. H. Ozcelik, "Detecting credit card fraud by genetic algorithm and scatter search," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13057–13063, 2011.

[12] P. Ravisankar, V. Ravi, G. R. Rao, and I. Bose, "Detection of financial statement fraud and feature selection using data mining

techniques," *Decision Support Systems*, vol. 50, no. 2, pp. 491–500, 2011.

[13] E. Kirkos, C. Spathis, and Y. Manolopoulos, "Data mining techniques for the detection of fraudulent financial statements," *Expert Systems with Applications*, vol. 32, no. 4, pp. 995–1003, 2007.

[14] F. H. Glancy and S. B. Yadav, "A computational model for financial reporting fraud detection," *Decision Support Systems*, vol. 50, no. 3, pp. 595–601, 2011.

[15] D. Olszewski, "Fraud detection using self-organizing map visualizing the user profiles," *Knowledge-Based Systems*, vol. 70, pp. 324–334, 2014.

[16] J. T. Quah and M. Sriganesh, "Real-time credit card fraud detection using computational intelligence," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1721–1732, 2008.

[17] E. Rahimikia, S. Mohammadi, T. Rahmani, and M. Ghazanfari, "Detecting corporate tax evasion using a hybrid intelligent system: A case study of Iran," *International Journal of Accounting Information Systems*, vol. 25, pp. 1–17, 2017.

[18] I. T. Christou, M. Bakopoulos, T. Dimitriou, E. Amolochitis, S. Tsekeridou, and C. Dimitriadis, "Detecting fraud in online games of chance and lotteries," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13158–13169, 2011.

[19] C. F. Tsai, "Combining cluster analysis with classifier ensembles to predict financial distress" *Information Fusion*, vol. 16, pp. 46–58, 2014.

[20] F. H. Chen, D. J. Chi, and J. Y. Zhu, "Application of Random Forest, Rough Set Theory, Decision Tree and Neural Network to Detect Financial Statement Fraud–Taking Corporate Governance into Consideration," In *International Conference on Intelligent Computing*, pp. 221–234, Springer, 2014.

[21] Y. Li, C. Yan, W. Liu, and M. Li, "A principle component analysis-based random forest with the potential nearest neighbor method for automobile insurance fraud identification," *Applied Soft Computing*, to be published. DOI: 10.1016/j.asoc.2017.07.027.

[22] S. Subudhi and S. Panigrahi, "Use of optimized Fuzzy C-Means clustering and supervised classifiers for automobile insurance fraud detection," *Journal of King Saud University-Computer and Information Sciences*, to be published. DOI: 10.1016/j.jksuci.2017.09.010.

[23] M. Seera, C. P. Lim, K. S. Tan, and W. S. Liew, "Classification of transcranial Doppler signals using individual and ensemble recurrent neural networks," *Neurocomputing*, vol. 249, pp. 337-344, 2017.

[24] E. Duman, A. Buyukkaya, and I. Elikucuk, "A novel and successful credit card fraud detection system Implemented in a Turkish Bank," In *IEEE 13th International Conference on Data Mining Workshops (ICDMW)*, pp. 162–171, 2013.

[25] C. Phua, K. Smith-Miles, V. Lee, and R. Gayler, "Resilient identity crime detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 533–546, 2012.

[26] M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation" *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[27] Credit Card Fraud Detection [Online]. Available: https://www.kaggle.com/dalpozz/creditcardfraud

[28] R. Saia and S. Carta, "Evaluating Credit Card Transactions in the Frequency Domain for a Proactive Fraud Detection Approach," In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications*, vol. 4, pp. 335–342, 2017.

[29] ISO 8583-1:2003 Financial transaction card originated messages [Online]. Available: https://www.iso.org/standard/31628.html