Machine Learning Model for Credit Card Fraud Detection- A Comparative Analysis

Pratyush Sharma, Souradeep Banerjee, Devyanshi Tiwari, and Jagdish Chandra Patni
School of Computer Science, University of Petroleum and Energy Studies Dehradun

Abstract: *In today's world, we are on an express train to a cashless society which has led to a tremendous escalation in the use of credit card transactions. But the flipside of this is that fraudulent activities are on the increase; therefore, implementation of a methodical fraud detection system is indispensable to cardholders as well as the card-issuing banks. In this paper, we are going to use different machine learning algorithms like random forest, logistic regression, Support Vector Machine (SVM), and Neural Networks to train a machine learning model based on the given dataset and create a comparative study on the accuracy and different measures of the models being achieved using each of these algorithms. Using the comparative analysis on the F_1 score, we will be able to predict which algorithm is best suited to serve our purpose for the same. Our study concluded that Artificial Neural Network (ANN) performed best with an F_1 score of 0.91.*

Keywords: *Machine learning, credit card fraud detection, random forest, accuracy, neural network, SVM.*

Received June 10, 2020; accepted February 17, 2021
https://doi.org/10.34028/iajit/18/6/6

1. Introduction

Credit card fraud is an umbrella term used to refer to the use of credit cards to buy services or goods to elude payment. This includes identity theft, identity assumption, or even a fraud spree. Fraud Detection is the method of monitoring the behaviour of the transactions done by the cardholder to detect any unauthorized transactions. Traditional methods of fraud detection have been used for a long period; however, these methods are very time-consuming and often inefficient. Therefore, a combination of machine learning and artificial intelligence is required for the effective detection of fraud.

The paper includes the collection of data from thousands of credit card users followed by pre-processing of data and creation of user's profile. This is followed by finding association among the data set and using different ML algorithms; inconsistency like the transaction is then observed. The basic principle of fraud detection using machine learning is based on the concept of training a model which understands the transaction records that are already known to be legit and detects any variation from these transactions.

The dataset was fetched from the Kaggle website titled 'Credit Card Fraud Detection' [5]. The dataset has total 31 columns including the class column that is to be predicted. The dataset has total 284,807 data points or rows. There are two classes to be predicted in the dataset: '0' or '1', i.e., 'genuine' or fraudulent' transaction. Principle Component Analysis (PCA) has already been applied to the dataset to protect the confidentiality of the users. PCA or Principle Component Analysis is a statistical technique to decrease the dimension of the features space by performing feature extraction [23]. The idea behind the approach is to reduce the dimensionality of the attributes in the dataset while still retaining the variation and correlation among them to the maximum possible extent. Attributes named 'V1' to 'V28' are masked using PCA. Due to this the feature analysis and selection of these 28 attributes were limited. The remaining 2 features are 'amount' and 'time'.

After feature selection 30 attributes were left including the class column. Prior to development of the model, data balancing needs to be considered. This includes dealing with the problem of a huge imbalance in the dataset. The paper introduces the use of Synthetic Minority Oversampling Technique (SMOTE) as a means for balancing the data which ensures that the finding does not imply false conclusions. As less than 0.5% of transactions are fraud among 284,807 transactions so even if a model incorrectly deems a fraud transaction to be legal the accuracy of the model still will be over 99% which is unacceptable. To solve this problem oversampling using SMOTE is being used. SMOTE or Synthetic Minority Oversampling Technique means duplication of examples in the minority class before fitting the model. SMOTE works by linear interpolation of the minority class. The synthetic records are generated by applying K nearest neighbour to examples in the minority class. This method is efficient as new examples are created which are relatively close to the already existing examples. After SMOTE algorithm was applied using 'SMOTE' function under imblearn library in python, total data points returned were 358, 208.

The cogency of the paper stands out, as the paper

provides a detailed comparison between four of the most used ML techniques-logistic regression, Support Vector Machine (SVM), random forest Classifier, Artificial Neural Network (ANN). The paper also introduces the use of SMOTE as a means for balancing the data which ensures that the finding does not imply false conclusions. F_1 Score has been used to evaluate which works efficiently in the evaluation of result in case of imbalanced classes; since in financial data analysis majority of datasets are imbalanced, F_1 score acts as the superior metric to evaluate a model versus Accuracy as used in many other previous works of literature

2. Related Work

There are many techniques available to detect fraud transactions. It is very difficult to detect the fraud, or they can be detected after the fraud happens. This happens because the fraudulent transactions are small as compared to total transactions. The authors in the paper [8], compared 7 techniques to detect such transactions. ANN got the best results for all parameters such as accuracy at 99.71%, detection rate at 99.68%, and false alarm rate at 0.12%. Although ANN takes the most time and computes power to train. SVM has the maximum false alarm rate at 5.2% and a detection rate of 85.45% not being comparable to other better techniques. Fuzzy logic has the worst detection rate at 77.8%. Decision trees are balanced towards complexity to train and results acquired with accuracy at 97.93%, detection rate at 98.52%, and false alarm rate at 2.19%. Random forest is a decision tree regression and classification technique that works well with both categorical and numerical data [6]. The authors tested random forest and SVM classifier to detect fraudulent transactions from the dataset. The pre-processing was done to avoid missing values and scale feature values. The authors concluded that imbalanced data did not work well with SVM as compared to random forest classifiers. Another advantage of using the random forest technique was the introduction of new data points did not have a major impact on the model since it used a subset of data with different decision trees. Each tree has a very low chance to impact others and hence also avoid bias and overfitting to an extent. In paper Mohankumar and Karuppasamy [15] studied random forest classification to detect fraudulent credit card transactions. The dataset used has values masked through the PCA algorithm. Scaling of feature values was done to reduce variance among features. SMOTE algorithm has been used to balance data. The balanced data contains 175000 classes. Random forest classifier is used for binary classification of data points. From the results published in the paper, the precision-recall curve has an equal value of around 0.85. Random forest classifier has become one of the most common techniques used in e-commerce to detect credit card frost due to its flexibility and scalability it provides for large datasets.

The computational power required in training the random forest model is low as compared to better state of art techniques like ANN. Although ANN is not being deployed in real-time e-commerce solutions to a large extent due to computational and time constraints. The class imbalance is the major problem in the current datasets are available that mislead the research [17]. In the paper, the authors discussed balancing data for efficient analysis, regression, and classification problems. The major techniques they studied were Random oversampling and under sampling, statistical oversampling and under sampling, SMOTE, Feature Selection, Hybrid Sampling, Cost-effective Learning, and Ensemble Learning. From reviewing the multiple research papers, we found the SMOTE technique is used commonly and another one is feature selection [9, 14, 16]. These two techniques provide the best results for balancing problems in data analysis.

More complex models can be used to predict labels for financial data. Building such models require much more time and expertise.

Razooqi *et al*. [19] proposed using fuzzy logic to adjust weights to use a genetic algorithm along with ANN which led to even better results and a very low FN rate. Training time increased drastically but the result was a lot better for ANN. Maes *et al*. [12] proposed a Bayesian network for predicting labels for financial data. The model gave good results for even small datasets and training time was reduced with parameters adjustment for the network through ANN. Shirgave *et al*. [20] the authors proposed the supervised learning technique random forest to classify the alert as fraudulent or legit and paved the way for using a semi-supervised machine learning algorithm for the classification of alerts. Lakshmi and Kavila [10] compared various methods like decision trees and random forest and found that random forest classifier proves better than decision trees and logistic regression for the accuracy for logistic regression, Decision tree, and random forest classifier are 90.0, 94.3, and 95.5 respectively. Based on the comparison among the three methods random forest classifier is the better choice over logistic regression and decision tree. And havarapu Bhanusri *et al*. [2] highlight that using machine learning algorithms we cannot determine the names of fraud and legit transactions for the given dataset and subsequent work needs to be done in that area. Sorournejad *et al*. [21] explains that the choice of algorithms should be such that minimizes False Positive and False Negative rate and maximizes True Positive and True Negative rate and ensure a good detection rate for a credit card fraud detection system. Carsten [3] mentioned that engine performance can be improved by applying Genetic Algorithm (GAs) to ANNs for credit card fraud detection.

3. Implementation

To develop a machine learning model which can predict credit card frauds by seeing the transaction, we need a good amount of data of previous transactions of the customers of a bank. Now the dataset that is available from the bank has Principal Component Analysis or PCA already applied to it to hide the confidential data of the customers. Before pushing the data to train the model, 'amount' feature was standardised using 'StandardScaler' function under sklearn library in python shown in the Figure 1, leading to values being ranged between 0 to 1. This was required since transaction amount in the dataset varied from $0.0 up to $1000, maximum values being ranged near $1-10. Hence, data points with value too high may lead to creating bias during training of the ML model. After feature selection 30 attributes were left including the class column, shown in the Table 1.

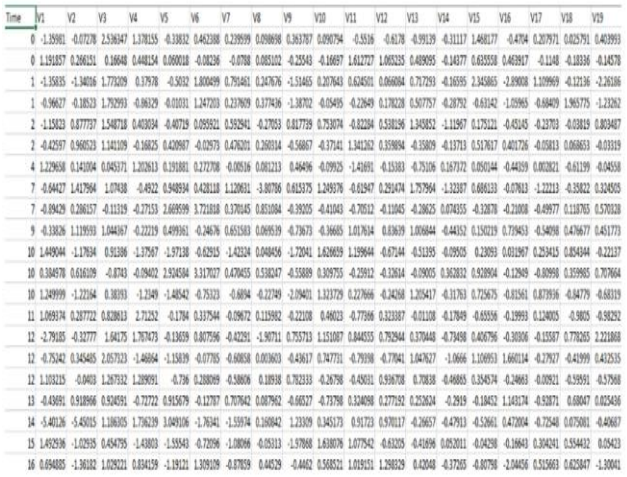

Figure 1. Converted data set.

Table 1. The correlation values of selected features.

Attribute	Correlation	Attribute	Correlation
V1	-0.424	V20	0.159
V2	0.491	V21	0.130
V3	-0.566	V22	0.046
V4	0.708	V23	-0.026
V5	-0.384	V24	-0.082
V6	-0.410	V25	0.040
V7	-0.481	V26	0.027
V8	0.052	V27	0.089
V16	-0.597	V28	0.078
V17	-0.558	Amount	0.036
V18	-0.464		
V19	0.268	Class	1.000

Features with correlation more than 0.5 or less than -0.5 have more effect on the classification. Time feature was dropped since it was starting from t=0 to $t=t_i$ which does not have any effect on the classification.

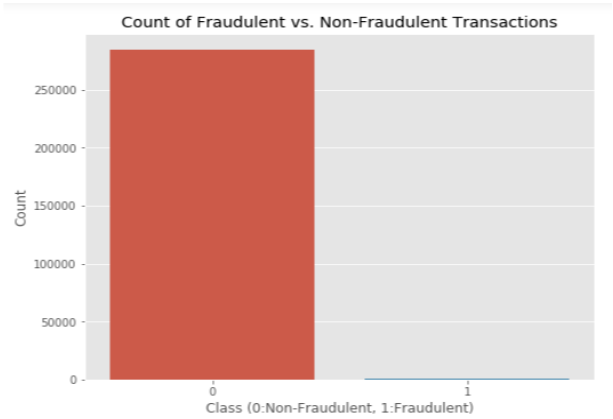

Figure 2. Histogram for fraudulent and Non-fraudulent Transactions.

The dataset post feature analysis and selection is quite overbalanced since the number of fraud transactions present in the dataset is much lower than the actual number of transactions, shown in the Figure 2. So, if we use the data set directly to train the model then we are going to get a high accuracy, but the system is going to label a transaction-safe even if it is a fraud due to bias towards class having a larger number of data points. So, to avoid such false negatives, we need to balance the data set. For balancing, we are going to use the oversampling mechanism. In oversampling what we do is try to increase the underrepresented minority class by using a specific technique. In this case, we are going to use the SMOTE or Synthetic Minority Oversampling Technique. SMOTE tries to increase the number of minority target class by using the features of the neighbours following these three steps:

1. The algorithm takes a set of points from the minority class, let it be A, and for $x \in A$, we need to find k nearest neighbors of x by calculating the Euclidean distance between x and other points in A.
2. We take the value of sampling rate N according to the imbalanced proportion for each point x in A, N random samples are being selected from the k nearest neighbours and then they form the new set A1.
3. For each point *y* in A1, we now use the following formula to find the new samples.

$$Z = x+rand(0,1)*|x-y| \quad (1)$$

Z is a coordinate of the derived data point, *x* and *y* are the original data points coordinates being used as a reference.

Once the balancing of dataset was done, the data was split into train, validation, and test data. 70/30 rule has been followed, 70% as training data, 15% was validation data and 15% was test data. Such a ratio was chosen because the dataset was primarily large and did not require any more data points for training as it may induce variance and result in a bias for classification [18].

We have utilized the dataset to train several machine learning models using the following algorithms:

1. Logistic regression.
2. Support vector machine.
3. Random forest algorithm.
4. Neural network.

Once the model is trained, we are going to test the accuracies of these models and then create a comparative visual result to show the difference in each of these algorithms.

4. Proposed Techniques

We are going to use the above four algorithms that to train the machine learning model. Brief description of each of these techniques are being given below:

4.1. Logistic Regression

Logistic Regression is a binary classification technique. It uses a logistic regression curve to estimate parameters of the logistic model from input data. It predicts discrete categories through the final output which is a continuous curve. As per the Figure 3 a cut-off is applied for both categories which are usually 0.5. The final output lies on the logistic function curve. If the output is greater than equal to 0.5, the model classifies the input to class 1 and if the final output is less than 0.5, it classifies the input into class 0.

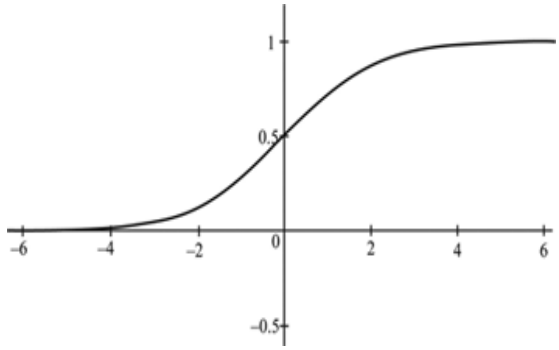

Figure 3. Logistic function.

The logistic model is the same as putting linear regression as input to a sigmoid function as given below.

Considering a simple model with single input x the $\Phi(z)$ is calculated as follows [11]:

$$\Phi(z) = \frac{1}{1+e^{-z}} \quad (2)$$

z is any real input and e is exponential constant.

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \quad (3)$$

p is interpreted as probability of the dependent variable y. β_0 and β_1 are shared parameters, x is data point [11].

Parameters' efficiency was tested through the Grid Search CV function in sklearn. model selection library.

4.2. Support Vector Machine

It is a classification and regression analysis algorithm. It uses the plotting of feature data points. Data points are first plotted in a 2D plane and then the dimensionality of the plot is increased until the data points of the 2 classes are linearly separable. As shown in the Figure 4 a linear boundary called hyper-plane is used to separate the parameter space into two half-spaces corresponding to predict the respective classes. We can also say that SVM is the result of finding the most likely logistic model. Finding such as hyperplane is an optimization problem that involves the hard margin [22] that is calculated as:

$$y_i(w*x_i - b)>=1, \text{ for all } 1<=i<=n. \quad (4)$$

Where x_i is a p dimensional real vector and w is a normal vector to the hyperplane.

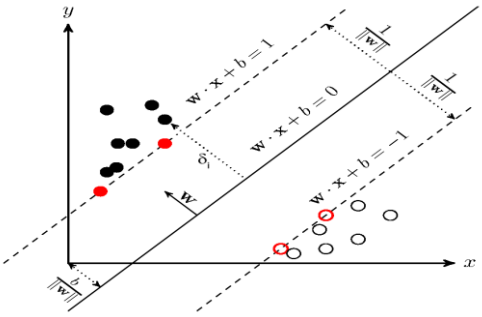

Figure 4. SVM.

The algorithm aims to maximize the distance between hard margin and soft margin that is distance between hard margin and nearest point from either class. Soft margin [22] is calculated as:

$$\min([1/n \sum_{i=1}^{n} \max(0,1 - y_i(w * x_i - b))] + \lambda\|w\|^2 \quad (5)$$

Support Vector Machine is a non-probabilistic technique although Platt scaling could be used for probabilistic classification.

To better map non-linear relations among features and labels, non-linear functions are used to separate data points of both classes on the hyperplane. One such function is Radial Basis Function.

Preferred Parameters for training of our model: C= 10, gamma= "auto", rest are being set at default values. Parameters' efficiency were tested through Grid Search CV function in sklearn.model_selection library.

4.3. Random Forest Classifier

Random Forest is a supervised learning algorithmic approach in which can be used for both classification and regression problems. As per the given Figure 5, the random forest algorithm uses a forest or collection of decision trees to obtain the classification result. The

more the number of trees the better the accuracy of the result.

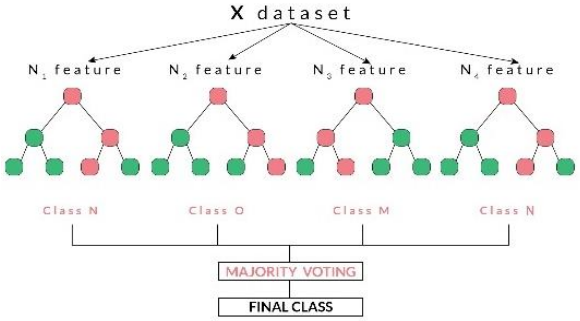

Figure 5. Random forest classifier.

Random forest is mainly based on two stages, creation of the random forest and prediction from the random forest of first stage.

To decide how to choose the split point for creating two nodes or choosing the root node we need to perform information gain. This information gain can be performed using two criteria mainly: Gini Impurity and Entropy.

Gini impurity is defined as the probability of incorrect choice of classification of the input data using the given classification [7], calculated as:

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i)) \quad (6)$$

C denotes the total number of classes and *p*(*i*)) denotes the probability of randomly picking the class *i*.

Entropy gives the measure for the homogeneity of the samples [7]. Entropy is calculated as:

$$E(S) = \sum_{i=1}^{c} - p_i \log 2\, p_i \quad (7)$$

Where *S* is feature and p_i is the probability of picking class *i*.

Preferred Parameters for the training of our model: n_estimators=300, rest are being set at default values. Parameters' efficiency was tested through the grid search CV function in sklearn.model_selection library.

4.4. Artificial Neural Network

This is a mathematical model to mimic biological neurons' capabilities in taking decisions. The biological neuron takes inputs, processes them in the axon part of its cell, and provides output. Similarly, ANN takes input, processes it through a function, and provides one single output per neuron or node.

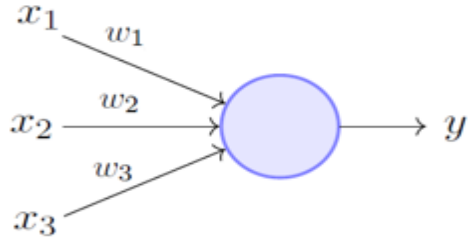

Figure 6. Perceptron model (Minsky-Papert in 1969).

As per the above Figure 6, x_1, x_2, x_3 are inputs and w_1, w_2, w_3 are weights for each input respectively and y is the final output.

The given figure shows a perceptron or a single neuron model in an ANN. We can update weights to adjust the value of y. The values x_1w_1,x_2w_2, and x_3w_3, passes through a mathematical function [1] to get the output *y*. For equalizing the values of x_iw_i, if x_i=0, a bias value b_i has been included as $x_iw_i+b_i$.

$$y = \sum_{i=1}^{n} x_i\, w_i + b_i \quad (8)$$

Wi is weight factor and bi is bias factor, *y* is the output and *xi* is the input.

A single perceptron, however, cannot be enough to map non-linear relations among features and classes or values we are trying to predict. For this multi-layer model is used shown in the Figure 7.

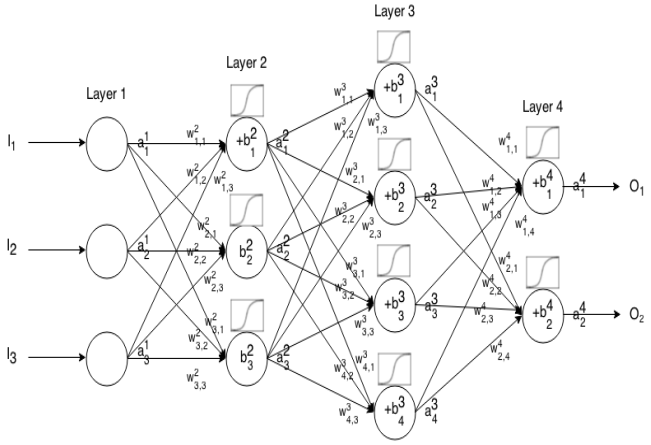

Figure 7. ANN model.

Where, l_i=inputs for respective layer, a_i=output from a node for respective layer, w_i=weight value of a node for respective layer, b_i=bias value of a node for respective layer, O_i=final output values

For each node, an activation function is used to cap the value for each node for the output from each node to be comparable for back propagation. These functions can be the Sigmoid function, Rectified Linear Unit (ReLu) or Tanh are some common ones to use.

Loss functions are used to calculate how far the final output value is compared to the actual output. The most common loss function is the quadratic cost function [1]. Squaring the difference in this function leads to punishment for a large difference in outputs.

$$C = 1/2n \sum_x ||y(x) - a^L(x)||^2 \quad (9)$$

Where *C*=loss value, *n*=number of nodes, *y*(*x*)=actual output, *a*(*x*) = predicted output. *L* denotes the layer.

The model aims to find a value of weight w_i to minimize loss function value C(w). This is done through the gradient descent technique where the slope of cost or loss function c(w) is calculated, and w is stepped downward until we get close to c(w)=0. For

avoiding overshooting the value 0 and increasing loss value further adaptive gradient descent techniques are used such as 'rmsprop' or 'adam'.

Preferred Parameters for training of our model: Learning rate=0.001, epochs=500, epoch callback trigger at epoch 31 (patience=15), model shape along with activation function used for each layer=[30(relu), 30(relu), 15(relu), 5(relu), 1(sigmoid)]. Rest are being set at default values. Loss function used is binary_crossentropy and optimizer used is adam. Loss value evaluation mode used was'min'.

5. Results

After performing training of the machine learning models using all the machine learning algorithms, we compare each of the models' performance to get a comparative analysis. We have used the Macro averages of precision, recall, and F_1 score in the following Table 2:

Table 2. Test metrics.

S.No.	Model	Accuracy	Precision	Recall	F_Score
1	Logistic Regression	0.94	0.54	0.92	0.57
2	Support Vector machine	0.95	0.76	0.80	0.78
3	Random forest	0.98	0.82	0.90	0.85
4	Neural Network	0.99	0.93	0.88	0.91

F_1 score is being calculated by using the formula below [22]:

$$2*(\text{precision}*\text{recall})/(\text{precision}+\text{recall}) \quad (10)$$

Accuracy is high for all techniques, but it is not a good parameter to evaluate our models. This is due to a high imbalance between class counts in the test dataset. Hence, precision, recall, and F_1 score have been used. In simple terms, precision is a positive predictive valve that we will get through the fraction of relevant instances with retrieved instances, whereas recall is a sensitivity fraction that we will get through the total amount of retrieved relevant instances. F_1 score is the harmonic mean between precision and recall. From the models, we have created to predict each class from the test dataset Logistic regression has done the worst. Accuracy was high but false positives at 1083 such cases for the fraudulent class were too high which led to low precision value for class 1 and subsequently F_1 score.

SVM was better than logistic regression but still not great on overall F_1 score at 0.78. Precision only at 0.53 due to higher FN cases than logistic regression, but with better recall because of lower FP cases led to an overall better result.

Random forest was very good for the dataset we used to train and test the model. F_1 score was 0.85 and FP and FN were both low.

ANN was the best model to predict fraudulent transactions. F_1 score was very high at 0.91 and accuracy at 0.99. Macro precision at 0.93 tells us that even cases that were hard to predict for TP for class 1 were marked correctly and the graphical analysis is shown in the Figure 8. And with FN level very low may lead to better customer experience. Though RF performed a little worse than ANN when the training dataset was reduced performance of both were comparable.

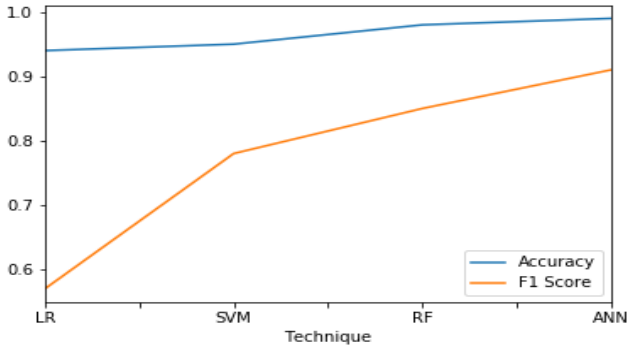

Figure 8. Graphical analysis of F1 scores.

6. Conclusions

The authors concluded that ANN was the best model (Precision-99.68%) to use for fraudulent transactions classification and in SVM (Precision-85.45%), the false alarm rate was high (5.2%) and decision tree performed average (Precision-98.52%) [8]. In our paper, the results show that random forest classifier is an upgrade over the decision tree model, and the ANN model also performed best. Paper [6] concluded that imbalanced data was performing worse for classification problems (Accuracy-86.5%). This also aligns with the findings of our paper along with random forest performing better than SVM. Paper [15] also used SMOTE for data balancing. This paper did not use ANN and deflected the technique as a highly complex and state of art solution to be used by banks. Just compared sample balancing techniques and didn't compare various ML Algorithms for classification problems [17]. This paper concluded that data balancing helps with classification problems which go by with our findings. Papers [3, 12, 13, 19], used either fuzzy logic or genetic algorithm or naïve bayes Algorithm to decide ANN parameters which drastically increased their model performance compared to ours.

As a comparison from literature reviews, and analysing our own results, we found that random forest proved to be very efficient (F_1 score – 0.85) for training the dataset whereas ANN turned out to be the best to anticipate fraudulent transactions with an F_1 score of 0.91. We also found that ANN can learn without a need to be reprogrammed and has high accuracy. SVM and logistic regression performed average (F_1 scores are 0.78 and 0.57 respectively). Hence, we recommend random forest and ANN to be the preferred techniques to be employed for prediction of credit card fraud detection systems.

A vital implication of our study is that it provides not only a faster and efficient method to detect fraud with increased accuracy, but also provides a deep dive into the various options available to the organizations. The contribution of this paper is that this research can serve as a reference point for such organizations to decide which machine learning algorithm they can follow to have the highest probability of detecting fraud transactions correctly.

The primary constraint of the paper is the absence of an amalgamation of Machine Learning Algorithms, we could have used a combination of Fuzzy Logic or Bayes, or Genetic algorithms to get parameters for ANN. The paper mentions that since the dataset is made available from a bank, it is already encrypted (PCA is already applied) to protect confidentiality. PCA often leads to information loss if the number of Principal Components is not selected carefully. It should be noted here that the PCA conversion of 28 features was done and the time feature was dropped so feature analysis was limited. Our study, being of an investigational and informative nature, provides a great scope for future research. In truth, more research will be necessary to further improve our discovery.

Credit Card fraud detection is an intricate subject that requires a considerable proportion of groundwork before applying different machine learning algorithms to it. Besides that, it is also a critical application of data science ensuring that the privacy and money of the customer are safe. This paper demonstrated how fraud can be detected efficiently using different machine learning algorithms. Globally, there are still some banks where traditional systems for detecting fraudulent transactions have been used. The main aim in this study was to address the different machine learning algorithms and how they can be utilized in different ways to detect fraud. In the future, we can improve our classifier so that can get close to the goal of 100% accuracy. Multiple algorithms can be amalgamated together, and their results can be compounded to improve the overall accuracy of the system. Since the size of the dataset directly influences the precision of the algorithm, so with due support from the banks we can improve our system. Also, a data set with non-anonymized features would allow one to see what factors are the most critical in finding fraud. This paper also applied to a distributed environment which can resolve issues relating to privacy.

References

[1] Artificial Neural Networks- Encyclopedia of Physical Science and Technology https://www.academia.edu/15726358/Artificial_Neural_Networks, Last Visited, 2021.

[2] Bhanusri A., Valli K., Jyothi P., Sai G., Rohith R. and Subash S., "Credit Card Fraud Detection Using Machine Learning Algorithms," *Journal of Research in Humanities and Social Science*, vol. 8, no. 2, pp. 04-11, 2020.

[3] Carsten P., "Credit Card Fraud Detection Using Artificial Neural Networks Tuned by Genetic Algorithms," Doctoral Thesis, Hong Kong University of Science and Technology, 2008.

[4] Classification Accuracy is Not Enough: More Performance Measures You Can Use, Machine Learning Mastery https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use, Last Visited, 2021.

[5] Credit Card Fraud Detection: Anonymized credit card transactions labeled as fraudulent or genuine, https://www.kaggle.com/mlg-ulb/creditcardfraud, Last Visited, 2021.

[6] Devi M., Janani B., Gayathri S., and Indira N., "Credit Card Fraud Detection using Random Forest Technique," *International Research Journal of Engineering and Technology*, vol. 06, no. 3, pp. 6662-6666, 2019.

[7] Entropy: How Decision Trees Make Decisions, https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8, Last Visited, 2021.

[8] Jain N., Tiwari N., Dubey S., and Jain S., "A Comparative Analysis of Various Credit Card Fraud Detection Techniques," *International Journal of Recent Technology and Engineering*, vol. 7, no. 5S2, pp. 402-407, 2019.

[9] Kalra M. and Patni J., "Playing Doom with Deep Reinforcement Learning," *International Journal of Computer Applications*, vol. 1, pp.14-20, 2019.

[10] Lakshmi S. and Kavila S., "Machine Learning for Credit Card Fraud Detection System," *International Journal of Applied Engineering Research*, vol. 13, no. 24, pp. 16819-16824, 2018.

[11] Logistic Regression, https://en.wikipedia.org/wiki/Logistic_regression, Last Visited, 2021.

[12] Maes S., Tuyls K., Vanschoenwinkel B., and Manderick B., "Credit Cards Fraud Detection Using Bayesian and Neural Networks," *in Proceedings of the 1st International Naiso Congress on Neuro Fuzzy Technologies*, Brussel, pp. 261-270, 1993.

[13] Miller G., Todd P., and Hegde S., "Designing Neural Networks using Genetic Algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms*, San Francisco, pp. 379-384, 1989.

[14] Mishra P., Patel V., Mittal P., and Patni J., "Algorithm Analysis Tool Based on Execution Time Input Instance-based Runtime Performance Benchmarking," *International Journal of Computer Applications*, pp. 27-30, 2018.

[15] Mohankumar B. and Karuppasamy K., "Credit Card Fraud Detection Using Random Forest Technique," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 8, no. 4, pp. 4128-4135, 2019.

[16] Patni J., Billus S., Billus S., and Singh R., "Feature-Based Opinion Mining and Managed Machine Learning with Sentimental Classification Models," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 2, pp. 3992-3998, 2020.

[17] Pavithra P. and Babu S., "Data Mining Techniques for Handling Imbalanced Datasets: A Review," *International Journal of Scientific Research and Engineering Development*, vol. 2, no. 3, 2018.

[18] Racz A., Bajusz D., and Heberger K., "Effect of Dataset Sizeand Train/Test Split Ratios in QSAR/QSPR Multiclass Classification," *Molecules*, vol. 26, no. 4, 2021.

[19] Razoogi T., Khurana P., Raahemifar K., and Abhari A., "Credit Card Fraud Detection Using Fuzzy Logic and Neural Networks," *in Proceedingsof the 19th Communications and Networking Symposium*, San Diego, pp. 1-5, 2016.

[20] Shirgave S., Awati C., More R., and Patil S., "A Review on Credit Card Fraud Detection Using Machine Learning," *International Journal of Scientific and Technology Research*, vol. 8, no. 10, pp. 1217-1220, 2019.

[21] Sorournejad S., Atani Z., and Monadjemi A., "A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective," https://arxiv.org/abs/1611.06439, Cornell University, Last Visited, 2021.

[22] Support-Vector Machine, https://en.wikipedia.org/wiki/Supportvector_machine, Last Visited, 2021.

[23] Uqaili I. and Ahsan S., "Machine Learning Based Prediction of Complex Bugs in Source Code," *The International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 26-37, 2020.

Pratyush Sharma has a bachelor's degree in Computer Science with specialization in Business Analytics and Optimization from UPES, Dehradun. He is currently working in the IT industry as a software engineer. He has keen interest in application/full stack development and machine learning.

Souradeep Banerjee has a bachelor's degree in Computer science with specialization in cloud computing and virtualization from UPES Dehradun. He is currently working the IT industry as a software developer engineer.

Devyanshi Tiwari has a Bachelor's Degree in Cloud Computing and Virtualization Technology. Her research interests include machine learning, DevOps. She is currently working as a Software Engineer.

Jagdish Chandra Patni working as Associate Professor at School of Computer Science, UPES Dehradun India. He did his Ph.D. in the area of High Performance computing in 2016. He did M. Tech. and B. Tech. respectively in the year 2009 and 2004.His areas of research are Database Systems, High Performance computing, Software Engineering, Machine Learning. He has published more than 50 research articles 5 books/book chapters. He is Guest Editor/Reviewer of various referred International journals. He has delivered 15 Keynote/Guest speech in India and abroad.