

Identification Of Credit Card Fraud Detection Using Decision Tree And Random Forest Algorithm

Malathi Eswaran¹, S. Deepa², S. Hamsanandhini³, Shiwangi Ojha⁴

^{1,2,3,4}Kongu Engineering College, Erode, Tamilnadu, India

Email: ¹malathieswaran@gmail.com

Abstract. Credit Card Fraud is one of the major ethical issues faced in day to day life. It's one among the foremost common frauds nowadays. The credit card fraud may happen in any of the following ways such as card stolen; card number is overseen by the other person and Fake phone calls convincing the person to disclose their card details. The main aim of the technique used here is to detect the Fraud from the legitimate transactions. Data mining techniques are helpful in detecting the frauds or the fraudulent activities. The most enhanced technique used here are Decision tree and Random Forest algorithm to detect the fraudulent behavior.

Keywords: Data mining, Decision tree, Random Forest Algorithm.

1. INTRODUCTION

Credit card fraud is the common and unethical behavior happening in our life. Credit card fraud incidences stay at around 0.1% of all card transactions. However, the amount of each fraudulent transaction ranges in billions of dollars! It's possible to detect most fraud cases beforehand through Machine Learning algorithms. It is one of the most explored domains of fraud detection. It relies on the automatic analysis of recorded transactions. The main challenges in credit card fraud detection are:

1. Huge size of data: millions of transactions are processed every day.
2. Imbalanced data set: more than 99% of transactions are legitimate.
3. Adaptive techniques: fraudsters being aware of newly built detection techniques.
4. Availability of data: banks rarely reveal customer information. so that the data scientists get very little data to access.

1.1 Getting an overview of the Data

Taken the dataset of credit card fraud detection from dataset repository, Open the R IDE and read the data. There are 284,807 rows *31 cols. The str() function is used to convert the specified value into a string. Class only contains numeric values whereas time and amount are actual values [1].

1.2 Features of the Dataset

The legitimate class would be represented as (Class == "0") and fraud is represented as (Class == "1") transactions. The data are to broken into two classes and their densities are organized in such a way that the distribution is compared. The code for the time column and the output graph are given in Fig. 1. And Fig. 2.

```
1 kccf.true <-kccf [kccf$class == 0, ]
2 kccf.false <-kccf [kccf$class == 1, ]
3 #overlay two plots on the same graph, use:library(ggplot2)
4
5 ggplot()+
6   geom_density(data = kccf.true,
7               aes(x=Time), color = "blue",
8               fill = "blue", alpha =0.12)+
9   geom_density(data = kccf.false,
10              aes(x=Time), color = "red",
11              fill = "red", alpha =0.12)
12
```

Fig. 1. Code for time object

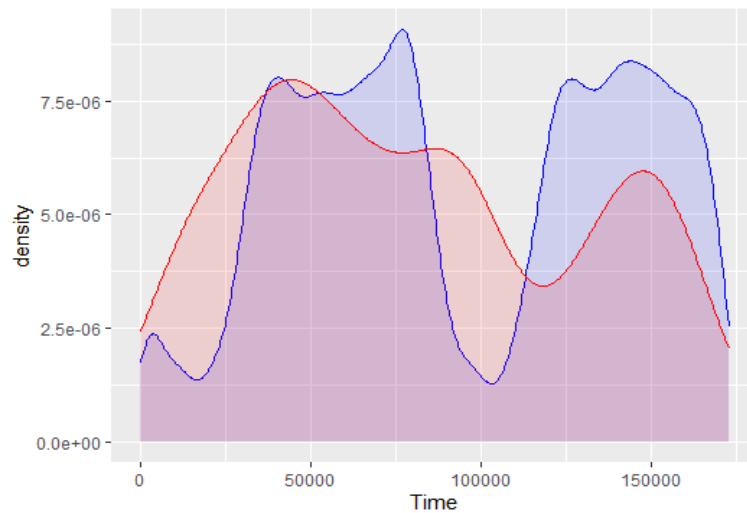


Fig. 2. Time object graph

Here blue represents legitimate transactions and red represents fraud transactions. Now, the amount is taken into consideration, where 'account' word is placed instead of time in the above code. The resultant graph is shown below.

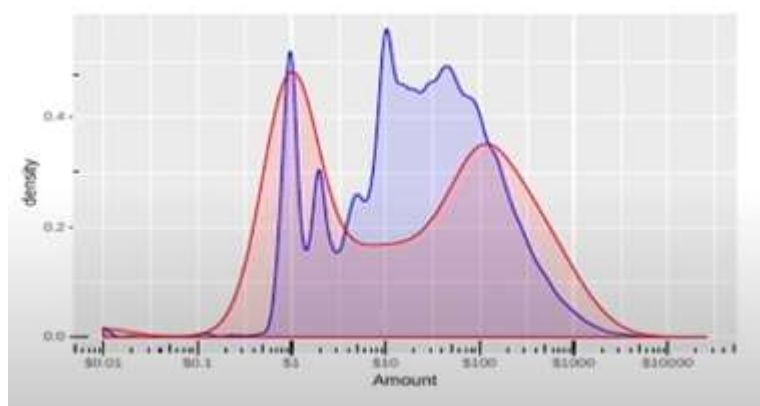


Fig. 3. Amount object Graph

Hence, the amount can be transformed logarithmically for clear visualization.

1.3 Density plot of some other features

In a similar way, the density plots of all other features (V1-V28).

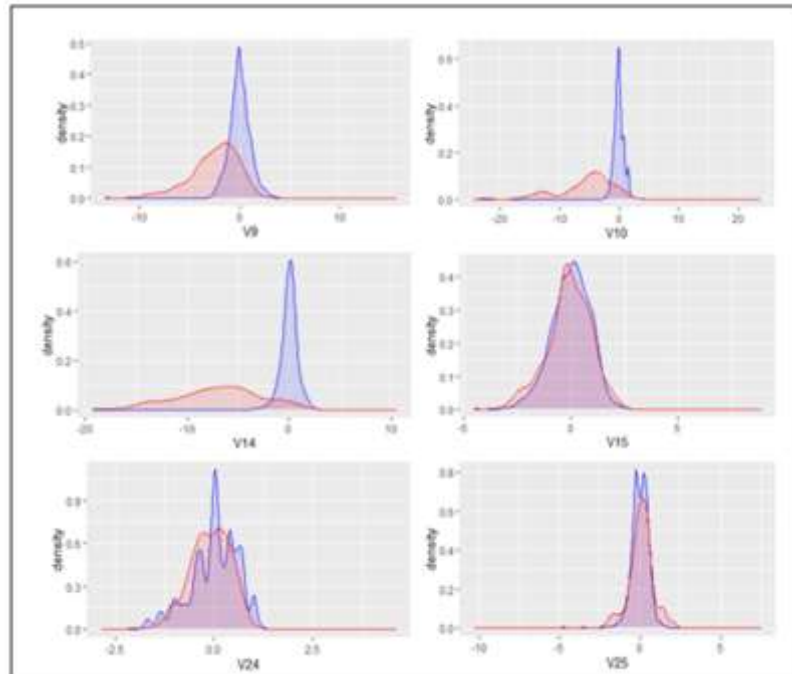


Fig. 4. Random object graph

Six features at that selected at random (V9, V10, V14, V15, V24, V25). The main difference between the above three from the below graph is that the V9 and V15 differ from each other because both the lines get overlap that is more or less similar, where in V9 the blue line that is genuine transactions is 0 that is 0 balance with density of 0.5 where the fraud transactions is from -2.9 to till -9.5. Therefore the transaction value for V9 is < -5 it shows that there is a high chance of fraud because the legitimate transaction, the value of V9 is between -3 and +3. In V10 and V14, the most the fraudulent transactions have a value of $V14 < -5$ whereas almost 99% of the transactions are legitimate are between -5 and +5 [8]. But this method cannot be used for the V15, V24, V25 because both the red and blue lines are overlapping. The differentiation of the graph with respect to V9, V10, and V14 is easier to differentiate between fraud and legitimate transactions, but it is not possible to differentiate the graph with respect to V15, V24 and V25 [2].

2. DATA CLEANING AND PREPROCESSING

Data cleaning contains the following estimation to assemble the dataset to be prepared. Detach unrelated fickle: ID, Name and Serial number these are relevant in predicting the class [3].

Check for typing error or unusual entries: "O" in place of "0", and the data which are not available.

Verifying that variable values make sense age! ≥ 100 , time interval! ≤ 0 .

The missing values are to be detached vertically and horizontally.

```
colnames(kccf)[colSums(is.na(kccf))>0]
nrow(kccf[!complete.cases(kccf),])
```

With the above code of lines, it is possible to check that missing values are not there in the data.

If there are few values missing in some places then it can be 'imputed', or can use 'mice' algorithm to remove the missing places.

Sometimes, a few variables may need some kind of transformation,

Normalization: rectilinear climbing of values in middle of 1 and 0.

$$X'_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}$$

Logarithmic transformation: log values are to be taken specifically if the dispersal of data is asymmetric on under side (e.g. Amount)

$$x'_i = \log x_i$$

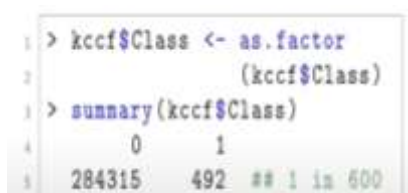
Discretization: Shattering up the span of variables to separate meantime. Sometimes can give the linguistic labels like HIGH/MEDIUM/LOW. So that it will be easy for identification.

Feature Selection: In order to have a handful of variables to keep the model simple and transparent [9]. The model uses only partitions of data that means the model is invariant to scale. So, the normalization or logarithmic transformation it does not affect the result by any degree. This model is a distinct model which means it has a type of individualization within the model and feature selection as well. So it does not require any preprocessing step. But some other model does require which can increase or improve your model efficiency [4].

3. HANDLING CLASS IMBALANCE

3.1 Balance the Data

Credit card frauds are rare events. The data has only 0.17% transactions that are fraud that means it is an imbalance classification problem.



```
1 > kccf$Class <- as.factor
2   (kccf$Class)
3 > summary(kccf$Class)
4      0      1
5 284315  492 ## 1 in 600
```

Fig. 5. Code for summary

This code gives the result that almost 1 in 600 are fraud.99.83% is 0 and .17% is 1. So, this is a heavy imbalance problem in classification which can degrade our result, unless that balancing is well done[10].

There are quite a few ways to deal with this imbalance:

Under sampling: Majority class observations are dropped to obtain a balanced dataset. One of those algorithms is called *one sided sampling*.

Oversampling: The minority class of the observations is duplicated where the balanced dataset is obtained. One of those algorithms.

Combining the two: Combining of under sampling and oversampling gives the best result [5].

3.2 Establishment of a Decision tree model

This model is uncomplicated model and yet it is constructive model for categorizing [11]. The finest thing about the model is it is clear and predictable. This domain gives a rational explanation. It branches on variable values one by one, to achieve maximum class split.

3.3 The fundamental computation

The establishment of decision tree can be done by partitioning the dataset recursively along with the certain wavering values with corresponding selected

Gini impurity:

$$I_G(p) = \sum_{i=1}^c p_i(1-p_i) = 1 - \sum_{i=1}^c p_i^2$$

Information gain or Entropy:

$$H(p) = -\sum_{i=1}^c p_i \log_2 p_i$$

These are the formula for Gini impurity or Entropy and either of these quantities need to be minimize so that the trading is efficiently done here, p_i denotes the probability of finding an observation of class i within a node and c is the number of classes [12]. Gini impurity is mostly used in classification and regression trees are part whereas entropies used an ID 3 or C4.5 will be using the Gini entropy and the library from the ecosystem to build our decision tree.

The decision tree model for credit card fraud dataset is taken .The establishment of the productive decision tree is done by the R package rpart, within a feasible time and in acceptable accuracy [13]. However, first need to split the whole data into training and test sets [14]. Training sets will build the model and Test set is to find how well the model and contemplate to reorganize the whole data to do away with inceptive prejudice.

3.4 Diagrammatic Envision of tree

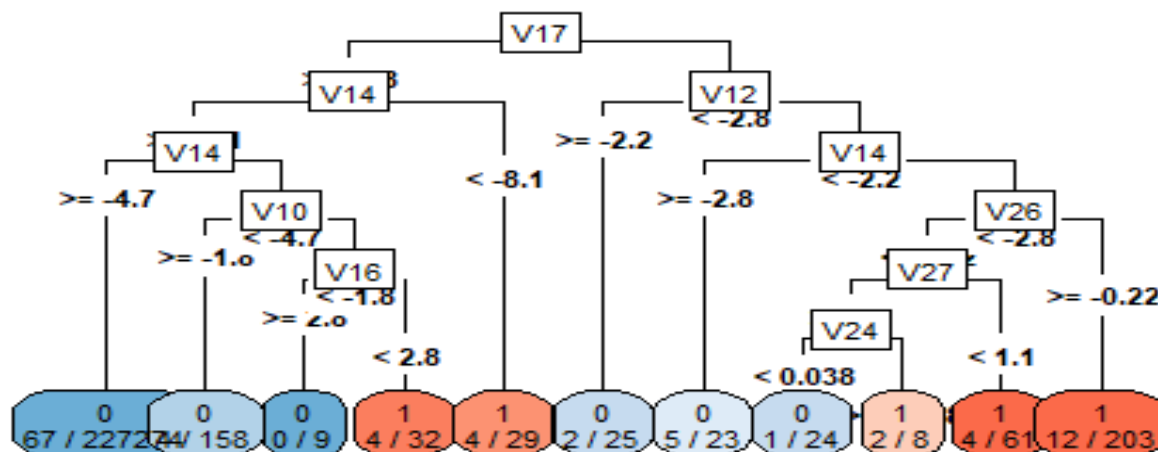


Fig. 6. Decision Tree

The left-most branch may be interpreted as:

If $V17 \geq -2.7$ and $V14 \geq -7.8$ and $V14 \geq -4.7$, then $class = 0$. So, it can also written as $V14 \geq -4.7$.

AUC matrix can able to obtain the 89% of accuracy. This is how the interpretability is obtained [6].

3.5 Incorporating variety | A bunch of trees

Reduction in unreliable predictions and increased confidence from numerous decisions is done by the bunch of trees [15]. Small quantity of trees can be built from smaller data samples. Since the time complexity of the decision tree is $O(h^2 N^3)$ where, h is the height and N is the number of observations, it speeds up the procedure.

In our case which involves

$N = 285,000$

Time = 1min

Then for 30% sample taken, the time running one tree will be just 1-2 seconds

So, the random forest method is built with 10 trees then it takes just 15 seconds which is $1/4^{\text{th}}$ the time, but the interesting thing about that is 30% sample and 10 trees that means that every data is actually represented 3 times so the use of data multiple times and different tree to classify our data and sometimes that gives a very creullistic kind of decision and more confidence in that because it comes from very different sources and different outlooks on the data [7].

Further, reduce the sample to 25% then the time for building one tree goes to below one second and can build 20 trees in just 18 seconds. Building 20 trees with 25% samples means that every observation is used almost 5 times by using 3-5 times each of the observation it's easy to reduce the time to $1/3^{\text{rd}}$ or $1/4^{\text{th}}$ and regarding our decision tree ,need not concerned with the time but more concerned with the specificity and accuracy because if a onetime build the model and can use the model multiple times to actually catch the errors or certify it or the genuine.

3.6 The Random Forest model for the kccf dataset:

The random forest method is built with R package random Forest with properly set tuning parameters. There are 3 tuning parameters which are needed to be set.

1. ntree = 10 to 100
2. samplesize = 20% to 80% of training set
3. maxnodes = 30 to 70
4. Now some values are chosen randomly number of trees is 40 with 60% specimen size of every tree and 55 is the at most number of nodes. The code is given as follows,

```
samp <- as.integer(0.60*ntr)

time.st <- Sys.time()
kccf.rF <- randomForest(Class = ..,
                        data = kccf.train,
                        importance = TRUE,
                        ntree = 40,
                        samplesize = samp,
                        maxnodes = 55)

Sys.time() - time.st
```

Fig. 7. Random forest parameters

Both the models are at most the same but anticipating the main model which is built with random forest function with its corresponding packages and its functions. Class is function too and of all other variables which are indicated by the dot and data comes from the dataset [16]. The training set established premature where three parameters 40 trees, a sample size is $0.60 \cdot ntr$ where the ntr is the number of training data and maximum node is 55.

So, when the tree is established and the tree is noticed to foretell that the class of the test data is an indistinguishable or homogeneous type of function [17]. The random forest is constructed here and registers it in the test data to do the projection well. The projection is straightforward when compared to decision tree and utilize them straightly as a new column in the testc data size. By executing this it shows as 80% of sensitivity and 100% specificity and balanced accuracy of 90.1%. As the dataset is unbalanced, the accuracy actually does not considered, so by 1% the accuracy is raised when compared to decision tree.

Random Forest typically establishes trees from specimen of the training figures which helps us in reducing time. Rather than reducing time the sensitivity and specificity is elevated. But here, both the sensitivity and specificity is reduced. Random forest package are helpful to build the trees by using the ntree, sample size and max nodes at parameters which can be tuned properly this needs a time to tune and it's a bit of trial and error process [9].

3.7 Verifying the results

Summary of the above two model:

Decision Trees	Random Forests
Pred Ref 0 1 0 54849 22 1 9 81	Pred Ref 0 1 0 56856 20 1 2 83
Accu : 0.9995 Sens : 0.9998 Spec : 0.7854 S_Ac : 0.8931	Accu : 0.9996 Sens : 1.0000 Spec : 0.8058 S_Ac : 0.9029
AUC : 0.8921	AUC : 0.9029

Fig. 8. Decision tree and Random forest with positive class as “0”

In decision tree out of 103 fraudsters 81 are acceptably categorized and 22 are uncategorized [10].

In case of Random forest, this model is helpful in categorizing the 2 fraudsters and even of number of people is reduced from 9 to 2 which is not highly necessary. So, it is superior type of categorizing where alternatively 9 legitimate transactions are uncategorized and now only 2 legitimate transactions are uncategorized, nevertheless that it is 1 in 28000. So, this enlarge the sensitivity and specificity to some extinct with positive class is taken as 0. Assume the positive class as 1, then a few more parameter are considered into our functions particularly where the calculation of those metrics that is sensitivities and specificities [11]. So if 1 is considered as the positive class then the results are given as follows,

Decision Trees			Random Forests		
Pred	Ref		Pred	Ref	
0	56849	22	0	56856	20
1	9	81	1	2	83
Accu : 0.9995 Sens : 0.7864 Spec : 0.9998 B_Ac : 0.8931			Accu : 0.9996 Sens : 0.8058 Spec : 1.0000 B_Ac : 0.9029		
AUC : 0.8931			AUC : 0.9029		

Fig. 9. Decision tree and Random forest with positive class as “1”

So, from the above output the conclusion is that first matrix remains the same, even the accuracy remains constant the only change that occurred is the specificity and sensitivity get swapped. Depending on the positive class the measures varies and the corresponding measures are to be taken according to the positive class. The measures like AUC and balanced accuracy does not brings any differences if the positive class is chosen as 1 or 0.[12].

Different measures of performance can be calculated as,

- Precision and recall
- F-score or F1-score
- Matthews Correlation Co-efficient.

Among this the familiar estimate is precision and recall. Recall and sensitivity is at most the familiar but precision is dissimilar. F-score and Matthews Correlation Co-efficient are utilized very frequently. In imbalanced class problems checking the results are done that are how well the result is obtained by the f-score or MCC. So Matthews class Co-efficient can be abbreviated to MCC. MCC can be calculated using MCC functions from ML tools package and F1-score functions from the ML metrics package [13].

3.8 Evaluation of data by other eminent scientist:

As very few data on credit card fraud are available people have used this kaggle data extensively. Here's a paper that runs 12 different Machine Learning algorithms on this dataset.

Table 1. Comparison of accuracy with the different classification algorithm

Method used	Fraud	Genuine	MCC
Naive Bayes	83.130	97.730	0.219
Decision Tree	81.098	99.591	0.775
Random Forest	42.683	99.988	0.604
Gradient Boosted Tree	81.098	99.936	0.746
Decision Stump	66.870	99.963	0.711
Random Tree	32.520	99.982	0.497
Deep Learning	81.504	99.956	0.787
Neural Network	82.317	99.966	0.812
Multi-Layer Perceptron	80.894	99.966	0.806
Linear Regression	54.065	99.985	0.683
Logistic Regression	79.065	99.962	0.786
Support Vector Machine	79.878	99.972	0.813
Decision Tree	78.640	99.984	0.841
Random Forest	80.583	99.996	0.887

So, these are the twelve algorithms that *Randhawa* actually worked out and also add the two algorithms which give us the maximum accuracy. The metrics are used are actually the specificities and sensitivities but since it actually depends positive class chosen. It is named as fraud accuracy and genuine accuracy [14]. So, the precision of distinguishing the fraud is from 32% to 83%. The top most is achieved by Naïve Bayes whereas the lowest precision is acquired by Random Forest tree. In the case of Genuine, off course most of the Genuine are correctly classified ranging from 97% to 99.988% where Random forest stimulates better categorization on the genuine transactions but it shows a very low grading on the fraud that's very unlike from what the last two algorithms is. From decision tree, 99.98% on the Genuine and 78.6% on frauds. It is little worse than a few algorithms here but the calculation of MCC that is the Matthews Correlation Co-efficient. The last two algorithms are fairly better than the other algorithms mentioned. The best among the twelve algorithms is 0.813 obtained by support vector machine and the next best is one obtained by the neural networks that are 0.812. But 81.3% by SVM and 84.1% by decision tree which is almost 3% better than best of the 12 algorithms. Random Forest method goes another 4% better 88.7%. So, the conclusion is they have used 12 algorithms and obtain 81% accuracy, where only by using two algorithms 88.7% accuracy is obtained [15].

In decision tree splitting of data into 80% of training and rest for the test sets and establishes a prototype on the 80% of the specimen and tested its precision on the rest of the 20% of the sample which is more or less 100% faultless grading on the genuine or the "0" group and about 78% of precision on the fraud that is "1" group that results in a spectacular output with 89.3% of AUC. The advantage or the finest thing regarding decision tree is that it is clear or understandable and their regulation can be acquired by crossing from the parent node of the tree to each of its child node.

The next prototype that is based on the multiple versions of the tree known as Random forest it definitely grows the precision by deliberated between AUC by 1%.

4. CONCLUSION

The conclusion obtained from both the method is that it shows the higher level of categorization when compared to other methods described in current IEEE papers. Both algorithms are quite effective with overall outcome. It is also known that the other individuals on kaggle haven't come upon with the expected output which is no less than to this specific dataset.

5. REFERENCES

- [1] Miroslav Kubat, Stan Matwin: Addressing the course of imbalanced training sets: One-Sided Selection. In: Fourteenth International Conference on Machine Learning, pp. 179-186 (1997)
- [2] Nitesh, V., Chawla: SMOTE Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence.16, 321-357 (2002).
- [3] Martin, Hisrchberg: On the complexity of learning decision trees. International symposium on Artificial Intelligence and Mathematics (1996).
- [4] Randhawa et al.: Credit Card Fraud Detection using AdaBoost and majority voting. IEEE Access.6, 14277-14284 (2016).
- [5] Linda, D., Hussien. A., John, P.: Credit Card Fraud and Detection Techniques. Banks and Bank Systems, 4(2), 57-68 (2009).
- [6] Devika, S P., Nisarga, K S., Gangana, P., Chandhini, S., Rajkumar, N.: A Research on Credit Card Fraudulent Detection System. International Journal of recent technology and engineering, 8(2), 5029-5032 (2019).
- [7] Dheepa, V., Dhanapal, R.: Analysis of credit card fraud detection methods. International journal of recent trends in Engineering. 2(3), 126-128 (2009).
- [8] Namrata, S., Swetha, P.: Document fraud detection with the help of data mining and secure substitution method with frequency analysis. International Journal of advanced computer research, 2(4), 149-156 (2012)
- [9] Gayathri, R.: Investigation of data mining techniques in fraud detection: credit card. International journal of computer applications. 82(9), 12-15 (2013).
- [10] Anita B. Desai, Ravindra Deshmukh: Data mining techniques for fraud detection. International conference of computer science and information technologies. 4(1), 1-4 (2013).
- [11] Amanze, B.C., Onukwugha, C.G.: Data mining application in credit card fraud detection system. International journal of trend in research and development. 5(4), 23-26 (2000).
- [12] Santhosh Baboo, S., Preetha, N.: Analysis of spending pattern on credit card fraud detection. IOSR journal of computer engineering, 17(2), 61-64 (2015).
- [13] Malini, N., Pushpa M.P : Analysis on credit card fraud detection techniques by data mining and big data approach. International journal of research in computer applications and robotics. 5(5), 38-45 (2017).

- [14] Richard J. Bolton and David J. Hand: Statistical fraud detection. *Statistical Science* 17(3), 235-255 (2002).
- [15] Sri Gowtham kumar .P, Sumanth Reddy.P and Mary Posaonia. A: Credit card fraud detection using machine learning. *International journal of engineering and technology*. 9(2), 4118-4123 (2019).
- [16] Xu, X.; Sun, Y.; Krishnamoorthy, S.; Chandran, K. An Empirical Analysis of Green Technology Innovation and Ecological Efficiency Based on a Greenhouse Evolutionary Ventilation Algorithm Fuzzy-Model. *Sustainability* 2020, 12, 3886.
- [17] K.Venkatachalam, A.Devipriya, J.Maniraj, M.Sivaram, A.Ambikapathy, Iraj S Amiri, "A Novel Method of motor imagery classification using eeg signal", *Journal Artificial Intelligence in Medicine Elsevier*, Volume 103, March 2020, 101787