Credit Card Fraud Detection Using Machine Learning Algorithms

Dr.A.S.MuthanandhaMurugavel[1], P.Jeevitha[2], R.Rajana[3], R.M. Danushree[4]
[1]Assistant Professor, Dept of Information Technology
[2, 3, 4]Dept of Information Technology
[1, 2, 3, 4] Dr.Mahalingam College of Engineering and Technology, Pollachi .

Abstract*- Credit card frauds are easy and friendly targets. It refers to loss of sensitive credit card information. E-commerce and many other online sites have increased the online payment modes, increasing the risk for online frauds. Many machine learning algorithms can be used for detection of fraud. This research shows several algorithms that can be used for classifying transactions as fraud or genuine one. Credit Card Fraud Detection dataset was used in the research. The main aim of the paper is to design and develop a fraud detection method for Transaction Data by analysing the past transaction details of the customers. This paper investigates and checks the performance of Decision tree, An Artificial Neural Network (ANN), XG Boost and Logistic Regression algorithms on highly skewed credit card fraud dataset. The results indicate about the accuracy for Decision tree, An Artificial Neural Network (ANN), XG Boost and Logistic Regression algorithms classifiers are 90.6, 88.3, 96.2 and 97.5 respectively.*

Keywords*-* Credit Card fraud detection, logistic regression, XGBoost, Decision tree, ANN

I. INTRODUCTION

Credit Card Fraud Transactions are unauthorized and unwanted usage of an account by someone other than the owner of that account. In Today's world high dependency on internet technology has enjoyed increased credit card transactions but credit card fraud had also accelerated as online and offline transaction. Credit card frauds are easy targets. Without any risks, a significant amount can be withdrawn without the owner's knowledge, in a short period. Fraudsters always try to make every fraudulent transaction legitimate, which makes fraud detection very challenging and difficult task to detect. Necessary prevention measures can betaken to stop this abuse and the behaviour of such fraudulent practices can be studied to minimize it and protect against similar occurrences in the future. There are many fraud detection solutions and software which prevent frauds in businesses such as credit card, retail, e-Commerce, Insurance, and Industries.

Credit Card Fraud Detection identifies the transactions that are fraudulent into two classes of legit class and fraud class transactions. They are several techniques that are designed and implemented to solve to credit card fraud detection by using many techniques such as genetic algorithm, migrating birds optimization algorithm, local outlier factor. Machine Learning Algorithms like, Isolation Forest Algorithm, Forest Artificial Neural Network , Fuzzy Logic , Genetic Algorithm , Logistic Regression ,Decision Tree , Support Vector Machines ,Bayesian Networks , Hidden Markov Model ,K-Nearest Neighbour. These algorithms are employed to analyse all the authorized transactions and report the suspicious ones. These reports are investigated by professionals who contact the cardholders to confirm if the transaction was genuine or fraudulent. The investigators provide a feedback to the automated system which is used to train and update the algorithm to eventually improve the fraud-detection performance over time. In the end of this paper, concludes about results of algorithms are made and collated.

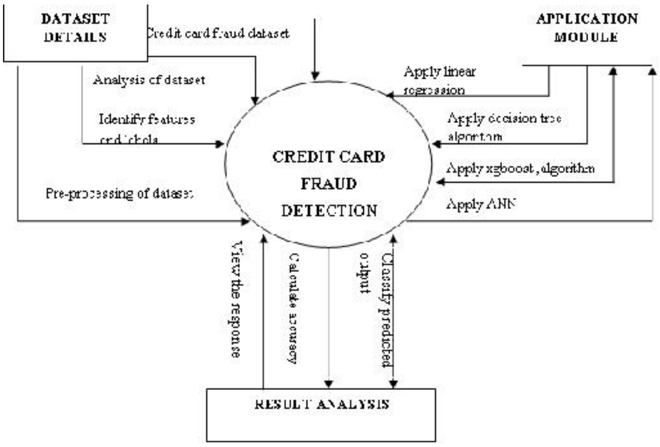


Figure 1

II. LITERATURE SURVEY

Multiple Supervised and Semi-Supervised machine learning techniques are used for fraud detection , In this paper we have compared certain machine learning algorithms for detection of fraudulent transaction and find accuracy of each algorithms.Many Supervised machine learning algorithms like

Isolation Forest Algorithm, Forest Artificial Neural Network , Fuzzy Logic , Genetic Algorithm , Logistic Regression ,Decision Tree , Support Vector Machines ,Bayesian Networks , Hidden Markov Model ,K-Nearest Neighbour are used to detect fraudulent transactions in real-time datasets. Feedback mechanism to solve the problem of concept drift. By using Naïve Bayes classifier the size of the training dataset is aggregate model when compare to other model. Two methods under random forests are used to train the large for behavioural features of normal and abnormal transactions.

They are Random-tree-based random forest and CART-based. Even though random forest obtains good results on small set data, there are still some problems in case of imbalanced data. The future work will focus on solving the above-mentioned problem.

The algorithm of the random forest itself should be improved. Performance of Logistic Regression, K-Nearest Neighbour, and Naïve Bayes are analysed on highly skewed credit card fraud data where Research is carried out on examining meta-classifiers and meta-learning approaches in handling highly imbalanced credit card fraud data. Through supervised learning methods can be used there may fail at certain cases of detecting the fraud cases.

III. DATASET DESCRIPTION

The dataset that is used in this paper is obtained from Kaggle .The datasets contains transactions made by credit cards in September 2013 by European cardholders.

This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

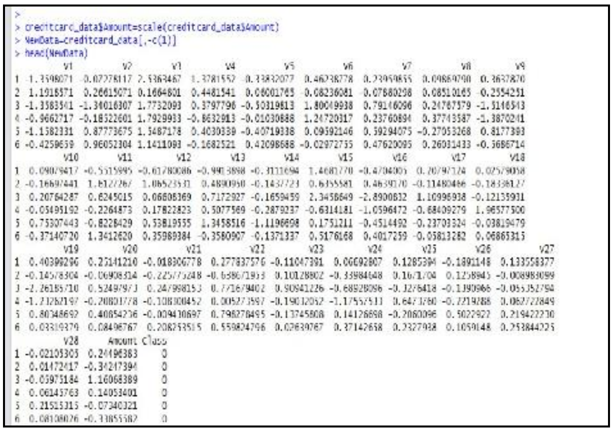
It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we can use an Artificial neural network (ANN)to provide the original features and more background information about the data. Features V1, V2, …V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

IV. METHODOLOGY

4.1 Reading dataset and preprocessing

- Using read. csv function, the csv dataset credit card is read into the dataframe variable creditcard_data.

- We have used head function and tail function to display the first 5 rows and last five rows.
- We have data composed of attributes with varying scale. So we have to re-scale the attributes to same range.
- we will scale our data using the scale() function. We will apply this to the amount component of our creditcard_data amount. Scaling is also known as feature standardization. With the help of scaling, the data is structured according to a specified range. Therefore, there are no extreme values in our dataset that might interfere with the functioning of our model.
- scale() function in R Langauge is a generic function which centers and scales the columns of a numeric matrix.
  - The center parameter takes either numeric alike vector or logical value.
  - If the numeric vector is provided, then each column of the matrix has the corresponding value from center subtracted from it.



```
>
> creditcard_data$Amount=scale(creditcard_data$Amount)
> NewData=creditcard_data[,-c(1)]
> head(NewData)
          V1          V2        V3         V4          V5          V6          V7          V8         V9
1 -1.3598071 -0.07278117 2.5363467  1.3781552 -0.33832077  0.46238778  0.23959855  0.09869790  0.3637870
2  1.1918571  0.26615071 0.1664801  0.4481541  0.06001765 -0.08236081 -0.07880298  0.08510165 -0.2554251
3 -1.3583541 -1.34016307 1.7732093  0.3797796 -0.50319813  1.80049938  0.79146096  0.24767579 -1.5146543
4 -0.9662717 -0.18522601 1.7929933 -0.8632913 -0.01030888  1.24720317  0.23760894  0.37743587 -1.3870241
5 -1.1582331  0.87773675 1.5487178  0.4030339 -0.40719338  0.09592146  0.59294075 -0.27053268  0.8177393
6 -0.4259659  0.96052304 1.1411093 -0.1682521  0.42098688 -0.02972755  0.47620095  0.26031433 -0.5686714
          V10        V11         V12        V13        V14        V15        V16         V17         V18
1  0.09079417 -0.5515995 -0.61780086 -0.9913898 -0.3111694  1.4681770 -0.4704005  0.20797124  0.02579058
2 -0.16697441  1.6127267  1.06523531  0.4890950 -0.1437723  0.6355581  0.4639170 -0.11480466 -0.18336127
3  0.20764287  0.6245015  0.06608369  0.7172927 -0.1659459  2.3458649 -2.8900832  1.10996938 -0.12135931
4 -0.05495192 -0.2264873  0.17822823  0.5077569 -0.2879237 -0.6314181 -1.0596472 -0.68409279  1.96577500
5  0.75307443 -0.8228429  0.53819555  1.3458516 -1.1196698  0.1751211 -0.4514492 -0.23703324 -0.03819479
6 -0.37140720  1.3412620  0.35989384 -0.3580907 -0.1371337  0.5176168  0.4017259 -0.05813282  0.06865315
          V19         V20          V21          V22         V23         V24        V25        V26          V27
1  0.40399296  0.25141210 -0.018306778  0.277837576 -0.11047391  0.06692807  0.1285394 -0.1891148  0.133558377
2 -0.14578304 -0.06908314 -0.225775248 -0.638671953  0.10128802 -0.33984648  0.1671704  0.1258945 -0.008983099
3 -2.26185710  0.52497973  0.247998153  0.771679402  0.90941226 -0.68928096 -0.3276418 -0.1390966 -0.055352794
4 -1.23262197 -0.20803778 -0.108300452  0.005273597 -0.19032052 -1.17557511  0.6473760 -0.2219288  0.062722849
5  0.80348692  0.40854236 -0.009430697  0.798278495 -0.13745808  0.14126698 -0.2060096  0.5022927  0.219422230
6  0.03319379  0.08496767  0.208253515  0.559824796  0.02639767  0.37142658  0.2327938  0.1059148  0.253844225
          V28      Amount Class
1 -0.02105305  0.24496383     0
2  0.01472417 -0.34247394     0
3 -0.05975184  1.16068389     0
4  0.06145763  0.14053401     0
5  0.21515315 -0.07340321     0
6  0.08108026 -0.33855582     0
```

Figure 2 Reading dataset

4.2 Splitting into training and testing sets

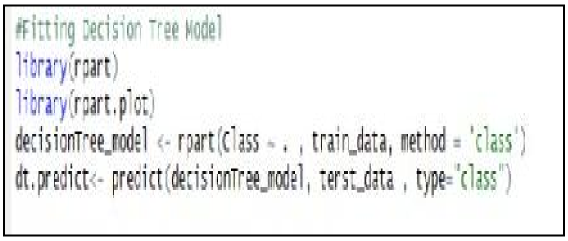
- We split the dataframe into training set and testing set using the function sample.split.
- sample.split function splits the data using given ratio in our case 80% for training set and 20% for testing set.

- Dim() is a function returns the dimension of the training set and testing set.
- The set. seed() function sets the starting number used to generate a sequence of random numbers – it ensures that you get the same result if you start with that same seed each time you run the same process.

4.3 Training the model

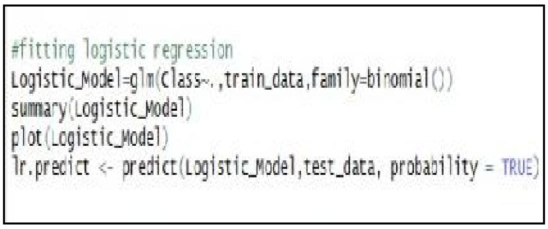
- The model is trained using four different machine Learning algorithms namely,Xgboost, decision tree, Artificial Neural networks (ANN) and logical regression algorithms.
- Glm() is used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution.

- The parameter family provides the convenient way to specify the details of models used by functions
- RPART -Recursive Partitioning And Regression Trees.
- predict () is used to returns vector of predicted responses of Rpart object.
-



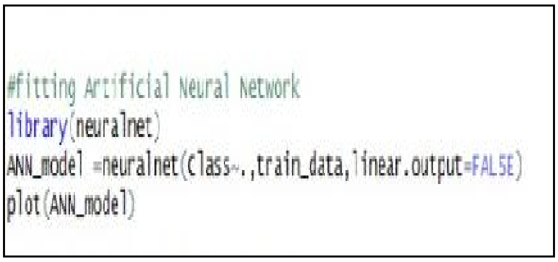
```
#Fitting Decision Tree Model
library(rpart)
library(rpart.plot)
decisionTree_model <- rpart(Class ~ . , train_data, method = 'class')
dt.predict<- predict(decisionTree_model, terst_data , type='class')
```

Figure 3 Training the dataset for decision tree algorithm



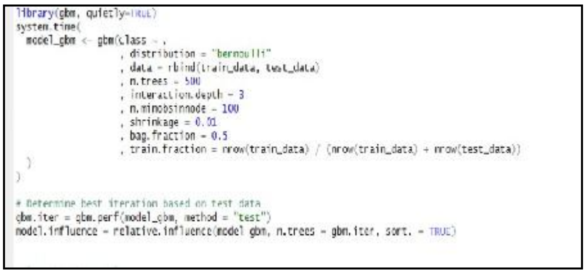
```
#fitting logistic regression
Logistic_Model=glm(Class~.,train_data,family=binomial())
summary(Logistic_Model)
plot(Logistic_Model)
lr.predict <- predict(Logistic_Model,test_data, probability = TRUE)
```

Figure 4 Training the dataset for logistic regression algorithm



```
#Fitting Artificial Neural Network
library(neuralnet)
ANN_model =neuralnet(Class~.,train_data,linear.output=FALSE)
plot(ANN_model)
```

Figure 5 Training the dataset for ANN algorithm



```
library(gbm, quietly=TRUE)
system.time(
  model_gbm <- gbm(Class ~ .
                   , distribution = "bernoulli"
                   , data = rbind(train_data, test_data)
                   , n.trees = 500
                   , interaction.depth = 3
                   , n.minobsinnode = 100
                   , shrinkage = 0.01
                   , bag.fraction = 0.5
                   , train.fraction = nrow(train_data) / (nrow(train_data) + nrow(test_data))
  )
)

# Determine best iteration based on test data
gbm.iter = gbm.perf(model_gbm, method = "test")
model.influence = relative.influence(model gbm, n.trees = gbm.iter, sort. = TRUE)
```

Figure 6 Training the dataset for XGBoost algorithm

The approach that this paper proposes, uses the latest machine learning algorithms like Decision tree, An Artificial Neural Network (ANN), XGBoost and Logistic Regression algorithms. First of all, we obtained our dataset from Kaggle, a data analysis website which provides datasets where we have 492 frauds out of 284,807 transactions.In this section of the R data science project,using read. csv function, the csv dataset credit card is read into the dataframe variable creditcard_data. We have used head function and tail function to display the first 5 rows and last five rows. Functions like table(), summary(), names(), var(), sd() are also used.We have data composed of attributes with varying scale. So we have to re-scale the attributes to same range.To accomplish that we use scale().scale() function in R Langauge is a generic function which centers and scales the columns of a numeric matrix. The center parameter takes either numeric alike vector or logical value. If the numeric vector is provided, then each column of the matrix has the corresponding value from center subtracted from it.We split the dataframe into training set and testing set using the function sample.split().

This sample.split() function splits the data using given ratio in our case 80% for training set and 20% for testing set.Dim() is a function returns the dimension of the training set and testing set.The set. seed() function sets the starting number used to generate a sequence of random numbers .

It ensures that you get the same result if you start with that same seed each time you run the same process.The model is trained using four different machine Learning algorithms namely, XGBoost, decision tree, Artificial Neural networks (ANN) and logical regression algorithms.

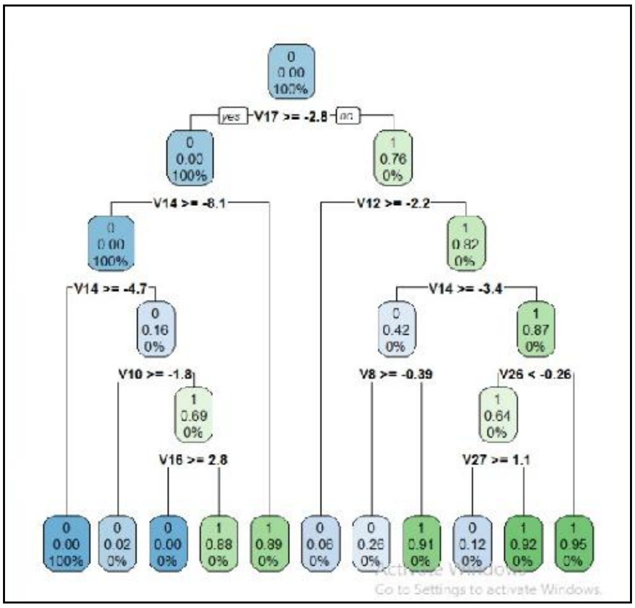


Figure 7 decision tree

Glm() is used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution.The parameter family provides the convenient way to specify the details of models used by functions. predict () is used to returns vector of predicted responses of Rpart object.This function creates Receiver Operating Characteristic (ROC) plots for one or more models.

A ROC curve plots the false alarm rate against the hit rate for a probabilistic forecast for a range of thresholds. The area under the curve is viewed as a measure of a accuracy.Higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s.This are the process takes place in our project to calculate the accuracy level.

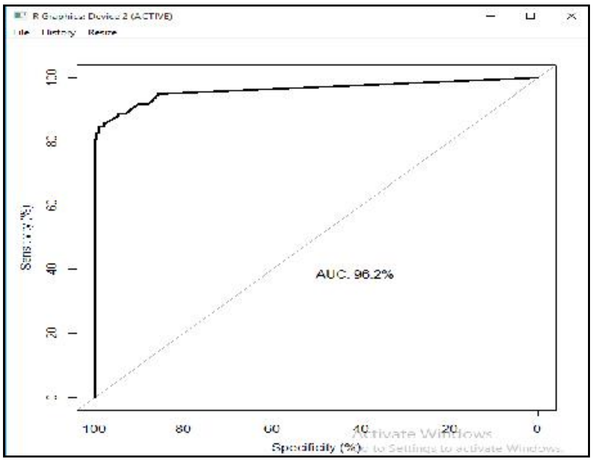


Figure 8 AUC for XGBoost algorithm

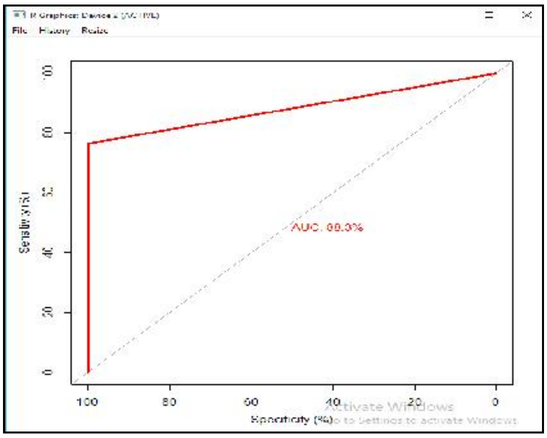


Figure 9 AUC for ANN algorithm

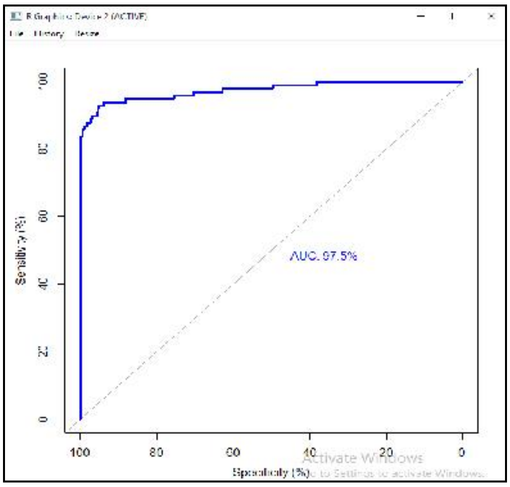


Figure 10 AUC for Logistic regression algorithm

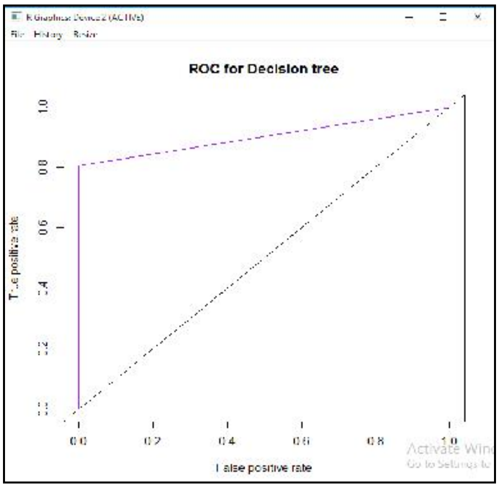


Figure 11 AUC for Decision tree algorithm

V. RESULT

Since the entire dataset consists of transaction records, its only a fraction of data that can be made available if this project were to be used on a commercial scale. Where the

ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.Being based on machine learning algorithms, the program will only increase its efficiency over time as more data is put into it.

AUC for our model is

a. An artificial neural network (ANN) -88.3
b. XGBoost algorithm -96.2
c. Logistic regression algorithm -97.5
d. Decision tree algorithm -90.27

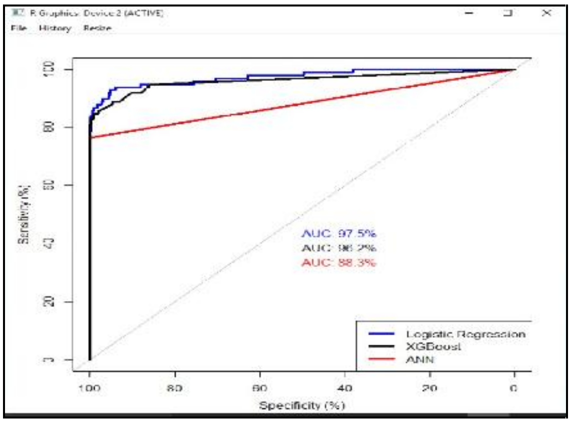


Figure 12 Comparision of algorithm

VI. CONCLUSION

Credit card fraud lead to loss of sensitive information and its considered as act of criminal dishonesty. This article has listed out the most common methods of fraud along with their detection methods. This paper has also explained in detail, how machine learning can be applied to get better results in fraud detection along with the algorithm, pseudocode, explanation its implementation and experimentation results. This paper proves that the Logistic Regression algorithm reach over 97.5% accuracy which is higher than XGBoost(96.2),Decision tree(90.27) and an Artificial neural network (ANN) (88.3).This high percentage of accuracy occurredin spite of imbalance between the number of valid and number of genuine transactions in the dataset. Being based on machine learning algorithms, the program will only increase its efficiency over time as more data is put into it.We finally observed that logistic regression gave better results.

REFERENCES

[1] Jiang, Changjun et al. "Credit Card Fraud Detection: A Novel Approach Using Aggregation Strategy and Feedback Mechanism." IEEE Internet of Things Journal 5 (2018): 3637-3647.

[2] "Credit Card Fraud Detection: A Realistic Modeling and a NovelLearning Strategy" published by IEEE transactions onneural networks and learning systems, vol. 29, no.8, august 2018

[3] Sam maes,Karl Tuyls,Bram Vanschoenwinkel. "Credit Card Fraud Detection Using Pipeling and Ensemble Learning ".researchgate.net ICITETM 2020

[4] Toluwase Ayobami Olowookere,Olumide Sunday Adenwale. "A Framework for Detecting Credit Card Fraud with Cost- Sensitive Meta- Learning EnsembleApproach". Elsevier.com/locatw/sciaf 2020

[5] Dejan Varmedja, Mirjana Karanovic, Srdjan," Credit Card Fraud Detection - Machine Learning Methods" INFOTEH-JAHORINA, 20-22 March 2019

[6] Philip K. Chan, Salvatore J. Stolfo," Credit Card Fraud Detection using Non-uniform Class and Cost Distributions" Florida Institute of Technology 2019

[7] Vaishnavi Nath Dornadulaa, Geetha Sa ," Credit Card Fraud Detection using Machine Learning Algorithms" The Authors Published by Elsevier B.V. (http://creativecommons.org/licenses/by-nc-nd/4.0/) ICRTAC 2019

[8] S. Dhankhad, B. Far, E. A. Mohammed, "Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study", 2018 IEEE International Conference on Information Reuse and Integration (IRI) pp. 122-125. IEEE.

[9] "Credit Card Fraud Detection Based on Transaction Behaviour –by John Richard D. Kho, Larry A. Vea" published by Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017

[10] J. O. Awoyemi, A. O. Adentumbi, S. A. Oluwadare, "Credit card frauddetection using Machine Learning Techniques: A ComparativeAnalysis", Computing Networking and Informatics (ICCNI), 2017International Conference on pp. 1-9. IEEE.

[11] N. Malini, Dr. M. Pushpa, "Analysis on Credit Card Fraud IdentificationTechniques based on KNN and Outlier Detection", Advances inElectrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2017 Third International Conference on pp. 255-258. IEEE.