JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING ALGORITHMS

Vaibhav Takle
Computer Engineering
Modern Education Society's College of Engineering.
Pune, India.
vaibhavdtakle@gmail.com

Vallabh Nandal
Computer Engineering
Modern Education Society's College of Engineering.
Pune, India.
vallabhnandal11@gmail.com

Chetan Ujade
Computer Engineering
Modern Education Society's College of Engineering.
Pune, India.
ujadechetan@gmail.com

Priyanka Rupanawar
Computer Engineering
Modern Education Society's College of Engineering.
Pune, India.
priyarupanawar10@gmail.com

Prof. Shraddha Khonde
Computer Engineering
Modern Education Society's College of Engineering.
Pune, India.
Shraddha.khonde@mescoepune.org

Abstract :

Financial fraud is a rising problem in the financial industry with long-term ramifications, and while numerous strategies have been developed to address this issue, To automate the evaluation of enormous volumes of sophisticated data, data mining has been successfully used to financial databases. Data mining has also played a crucial role in the identification of credit card fraud in online transactions. Credit card fraud detection is a data mining challenge. It becomes difficult for two reasons: For starters, regular and fraudulent behaviour patterns differ a lot, and second, credit card fraud data sets are heavily skewed.

On severely skewed and unbalanced credit card fraud data, this study explores and compares the performance of Logistic Regression, XGBoost, with several sampling methodologies such as under-sampling, over-sampling, and SMOTE, and Decision Tree. European cardholders provided a credit card transaction dataset with 284,786 transactions. Both raw and pre-processed data are used in these procedures. The methods' accuracy, sensitivity, precision, and recall are utilised to assess their performance. The results demonstrate that the most accurate classifiers are Logistic Regression, XGBoost, and Decision Tree.

I.INTRODUCTION

A credit card is a small, thin plastic or fibre card that contains personal information such as a portrait or signature and allows the cardholder to charge items and services to his linked account, which is debited on a regular basis.

These days, card information is read by ATMs, swiping machines, retail readers, banks, and online transactions. Each card has a unique card number, which is incredibly important; the card's security is largely determined by the physical security of the card as well as the privacy of the credit card number.

Due to the massive increase in credit card transactions, there has been a considerable increase in fraudulent instances. A range of data mining and statistical methods are used to detect fraud. Several fraud detection systems employ artificial intelligence and pattern matching. It is vital to identify fraud in a method that is both effective and secure.

Credit card theft is on the rise, and fraud-related financial losses are increasing. The Internet and online transactions are becoming more popular as new technology arises. The bulk of these transactions are made with credit cards. Credit card fraud losses in London were estimated to reach US$844.8 million in 2018. Fraud detection and prevention can assist to cut down on these losses. It is necessary to implement detection. As technology advances at a rapid rate, a

variety of frauds occur. As a result, there are several machine algorithms that are used to identify fraud, and many machine learning models are being developed to address the problem and compare model performance. Data sampling strategies will be utilised to improve the model.

II. PROBLEM STATEMENT

As a result of fraud transactions, card fraud is on the rise. Financial loss as a result of credit is on the rise. Every year, as a result of fraud, millions of dollars are lost. Hundreds of billions of dollars have been squandered. There is a scarcity of research to examine the fraud. Many machine learning approaches are widely used to detect real-world credit card fraud.

III. OBJECTIVES

The project's purpose is to identify credit card fraud using machine learning algorithms based on time and transaction amount.

IV. PROPOSED SYSTEM

This study's recommended solutions for detecting credit card system fraud are as follows. Different machine learning algorithms are examined to determine which one is better for identifying fraud transactions and may be utilised by credit card merchants, such as Logistic Regression, XGBoost, and Decision Tree. Figure is an architectural diagram for illustrating the overall system structure.

Algorithm steps:

Step 1: Read the Dataset.

Step 2: Handling missing values in dataset.

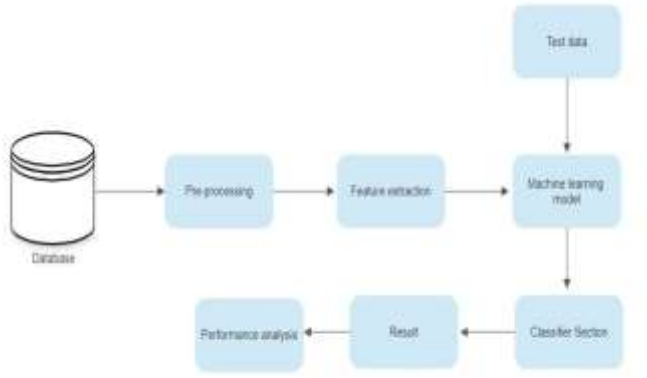
Step 3: Checking Distribution of the class

Step 4: To balance the data set, random sampling is applied.

Step 5: Separate the dataset into two parts: one for training and the other for testing.

Step 6: To determine the effectiveness of various algorithms, accuracy and performance indicators were determined.

Step 7: Then, based on the efficiency of the supplied dataset, choose the best algorithm.



System Architecture

V. TRANSACTION DATABASE

This information includes transactions from Europe cardholders in September 2013. There are 492 fraudulent transactions out of a total of 2,84,807 transactions. The data is imbalanced since there are fewer fraud cases than transactions. The data collection has undergone a PCA transformation and now only contains numeric values. Many background details are removed due to privacy and confidentiality issues, leaving just PCA transformed data. Only time and money are not PCA converted; all other given values (v1, v2, v3, v4, v5, v6, v7, v8, and so on) are numeric values that have been PCA transformed. The standard transaction has a value of 0 and the fraud feature class has a value of 1.

Fig. credit card transaction dataset

VI. DESIGN AND IMPLEMENTATION OF ALGORITHMS

We forecasted the conclusion by first understanding the issue statement and data, then doing statistical analysis and visualisation, and then determining whether the data is balanced. Because the data in this collection is uneven, it is oversampled, then scaled with standardisation and normalisation before being evaluated using a variety of machine learning approaches. Several tools are required for every data science project, including Numpy, a numeric Python library, and pandas, and for data visualisation, matplotlib and seaborn, both based on matplotlib. Jupyter notebook is used in this project to analyse the entire code, which allows the code to be shown as a block of codes, making it easier to execute each section and find faults. To construct a user interface for training and evaluating the algorithms, the python sklearn module is utilised. The test and train buttons can be used to train or test the data.

We are employing the sample approaches listed below because to the extremely unbalanced dataset.

Undersampling :- Here for balancing the class distribution, the non-fraudulent transctions count will be reduced to 396 (similar count of fraudulent transctions).

Oversampling :- Transactions that are not fraudulent will be counted in the same manner as fraudulent transactions are.

SMOTE :- Oversampling approach for synthetic minorities. It is another oversampling approach that creates fake data using the closest neighbour algorithm.

A. Machine learning algorithms

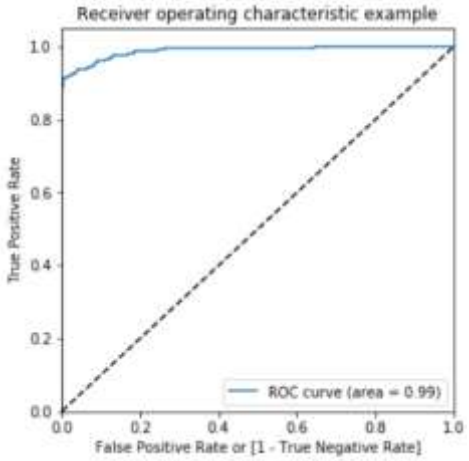
Logistic Regression

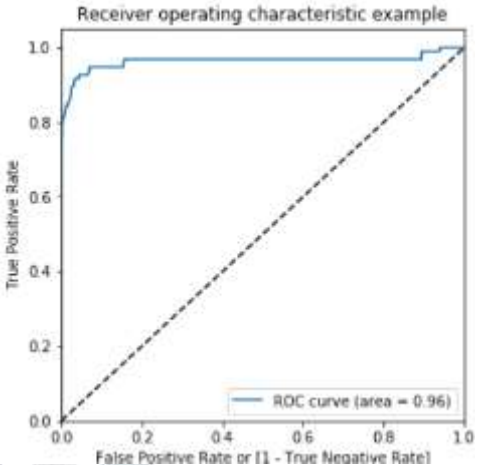
Logistic regression is a supervised classification method that predicts the probability of a binary dependent variable from the dataset's independent variable, i.e., logistic regression predicts the probability of an outcome with two values: zero or one, no or yes, false or true. The difference between logistic regression and linear regression is that logistic regression creates a straight line, whereas linear regression produces a curve. Logistic regression develops logistic curves that display values between zero and one based on the use of one or more predictors or independent variables.

A regression model with a categorical dependent variable that analyses the relationship between numerous independent factors is known as logistic regression. Binary, multiple, and binomial logistic regression models are among the numerous types of logistic regression models. Based on one or more parameters, the binary logistic regression model is used to estimate the likelihood of a binary answer.

Results of the implementations :

1.Logistic Regression with Under-sampling :

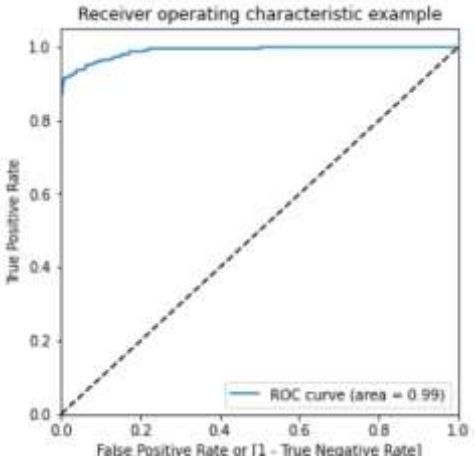


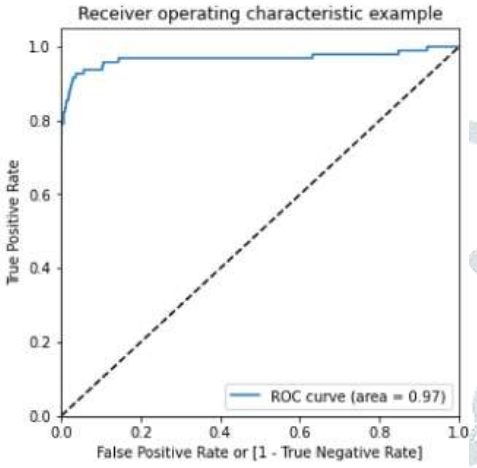


Model summary

- Train set
  - Accuracy = 0.95
  - Sensitivity = 0.92
  - Specificity = 0.98
  - ROC = 0.99
- Test set
  - Accuracy = 0.97
  - Sensitivity = 0.86
  - Specificity = 0.97
  - ROC = 0.96

2.Logistic Regression with Oversampling :

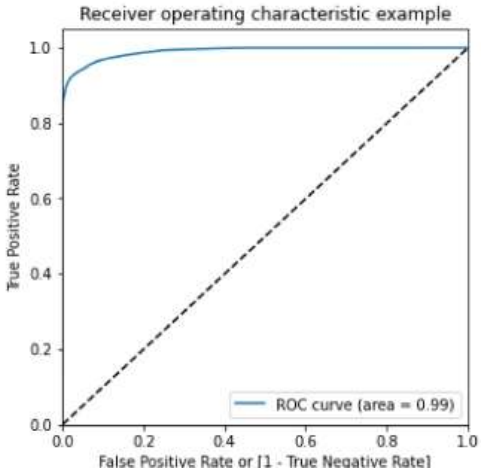


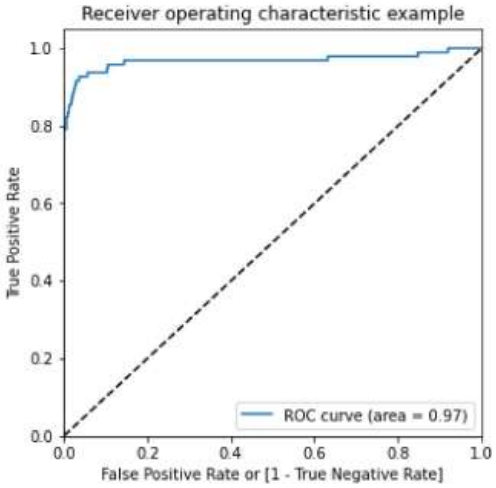


Model summary

- Train set
  - Accuracy = 0.95
  - Sensitivity = 0.92
  - Specificity = 0.97
  - ROC = 0.98
- Test set
  - Accuracy = 0.97
  - Sensitivity = 0.89
  - Specificity = 0.97
  - ROC = 0.97

2.Logistic Regression with SMOTE :





Model summary

- Train set
  - Accuracy = 0.95
  - Sensitivity = 0.92
  - Specificity = 0.98
  - ROC = 0.99
- Test set
  - Accuracy = 0.97
  - Sensitivity = 0.90
  - Specificity = 0.99
  - ROC = 0.97

XGBoost :

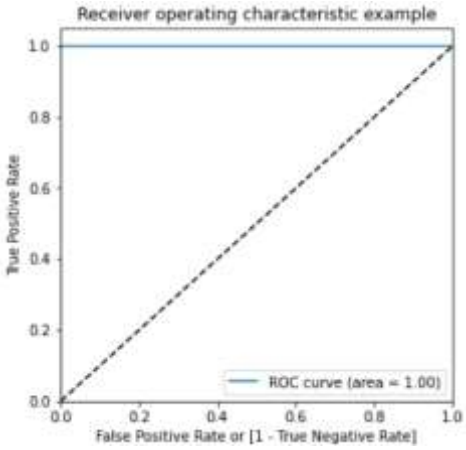
Friedman's work inspired the development of Extreme Gradient Boosting (XGBoost), a high-performance machine learning programme. Approximation of the Greedy Function:

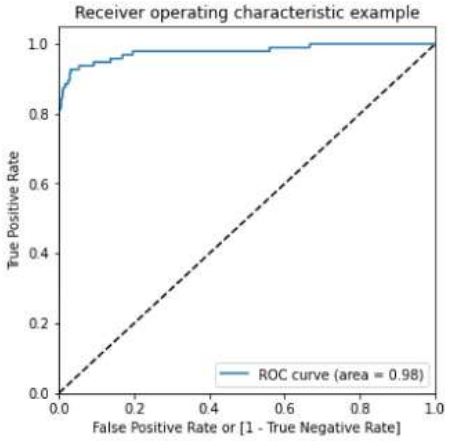
A Gradient Boosting Machine. XGBoost implements a decision tree-based Gradient Boosting method.Gradient Boosting, which is implemented in XGBoost, is also a decision tree ensemble.Those trees are poor models individually, but when they are grouped they can be really performant.

XGBoost has a strong track record of good performance on structured data issues, and it should be a part of your data scientist toolkit, especially if you want to compete on Kaggle. XGBoost is known for its versatility and quickness in addition to its performance. Whilst gradient boosting requires to build trees one by one sequentially, XGBoost implements a way to parallelize the training of each tree, Data Scientists' training will be more efficient, and their jobs will be easier.

Results of the implementations :

1.XGBoost with Under-sampling :

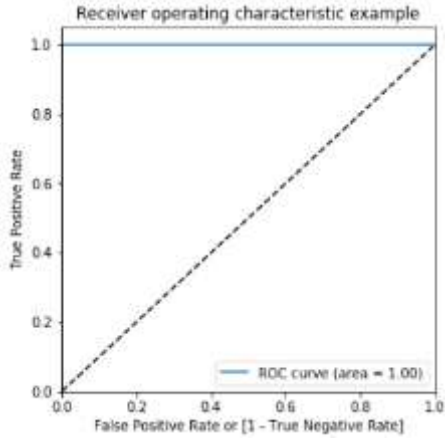


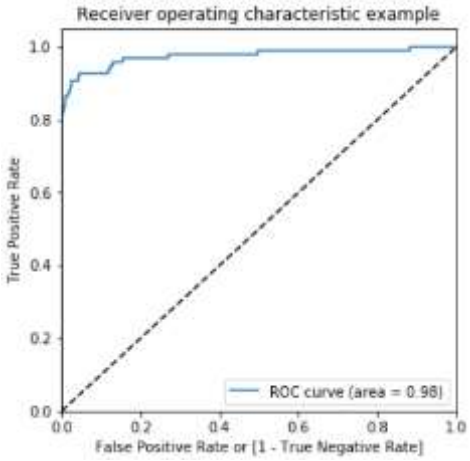


Model summary

- Train set
  - Accuracy = 1.0
  - Sensitivity = 1.0
  - Specificity = 1.0
  - ROC-AUC = 1.0
- Test set
  - Accuracy = 0.96
  - Sensitivity = 0.92
  - Specificity = 0.96
  - ROC-AUC = 0.98

2.XGBoost with Oversampling;

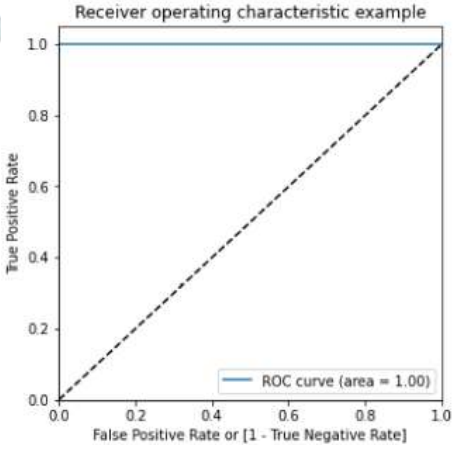


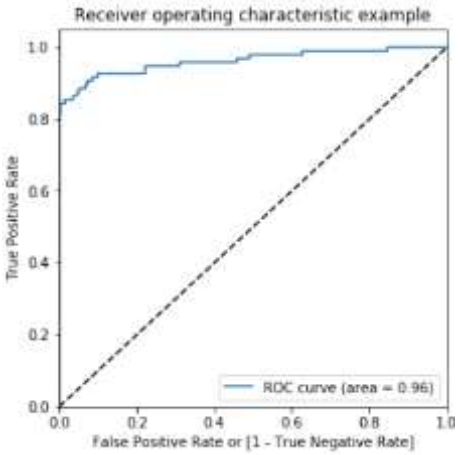


Model summary

- Train set
  - Accuracy = 1.0
  - Sensitivity = 1.0
  - Specificity = 1.0
  - ROC-AUC = 1.0
- Test set
  - Accuracy = 0.99
  - Sensitivity = 0.80
  - Specificity = 0.99
  - ROC-AUC = 0.97

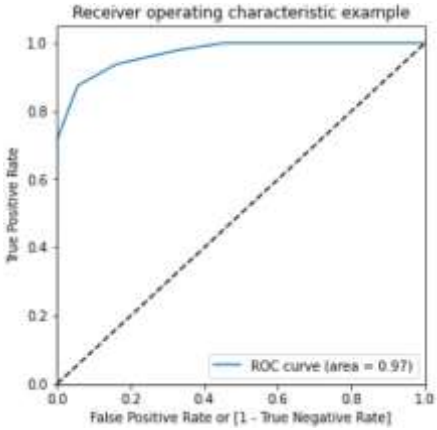
3.XGBoost with SMOTE:





Model summary

- Train set
  - Accuracy = 0.99
  - Sensitivity = 1.0
  - Specificity = 0.99
  - ROC-AUC = 1.0
- Test set
  - Accuracy = 0.99
  - Sensitivity = 0.79
  - Specificity = 0.99
  - ROC-AUC = 0.96



Model summary

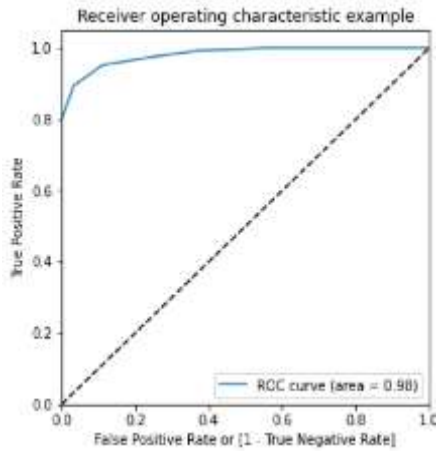
- Train set
  - Accuracy = 0.93
  - Sensitivity = 0.88
  - Specificity = 0.97
  - ROC-AUC = 0.98
- Test set
  - Accuracy = 0.96
  - Sensitivity = 0.85
  - Specificity = 0.96
  - ROC-AUC = 0.96

Decision Tree

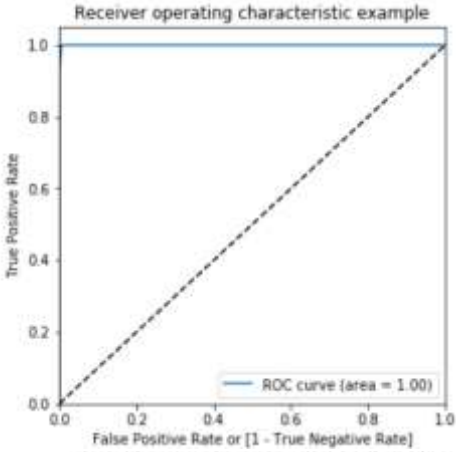
As explained on Wikipedia, a decision tree is a flowchart-like structure in which nodes represent features (e.g., weather if sunny or rainy) and leaves represent class labels that are created after all the tests have been performed. As a result, a decision tree is non-parametric. Its goal is to create groups from datasets based on conditions.

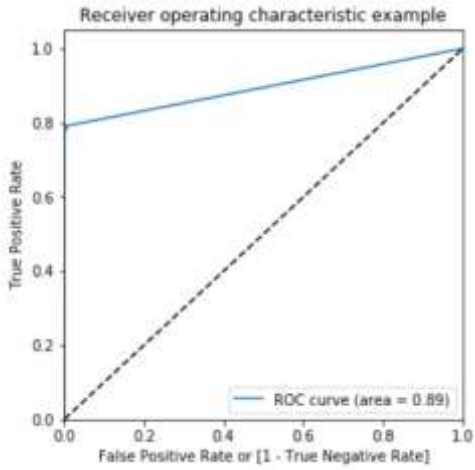
Results of the implementations :

1.Decision Tree with Under-sampling



2.Decision Tree with Oversampling:

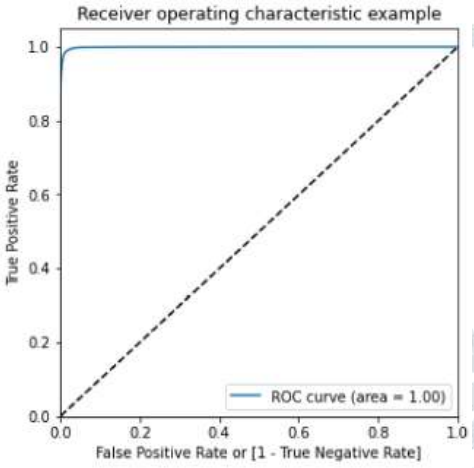


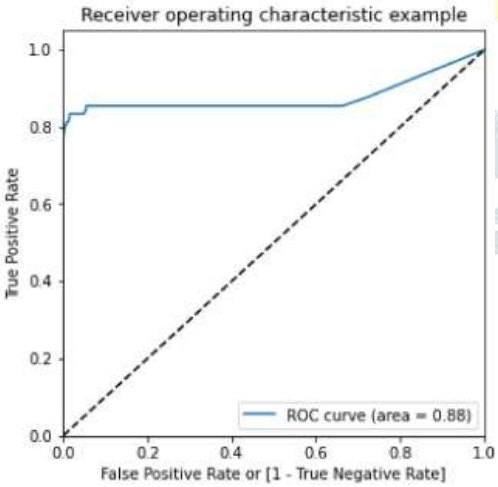


Model summary

- Train set
  - Accuracy = 0.99
  - Sensitivity = 1.0
  - Specificity = 0.99
  - ROC-AUC = 0.99
- Test set
  - Accuracy = 0.99
  - Sensitivity = 0.79
  - Specificity = 0.99
  - ROC-AUC = 0.90

2.Decision Tree with SMOTE:





Model summary

- Train set
  - Accuracy = 0.99
  - Sensitivity = 0.99
  - Specificity = 0.98
  - ROC-AUC = 0.99
- Test set
  - Accuracy = 0.98
  - Sensitivity = 0.80
  - Specificity = 0.98
  - ROC-AUC = 0.86

VII. CONCLUSION

Choosing best model: By Cost benefit analysis

We have tried several models till now with balanced data. We have noticed most of the models have performed more or less well in terms of ROC score, Precision and Recall.

But while picking the best model we should consider few things such as whether we have required infrastructure, resources or computational power to run the model or not. For the models such XGBoost we require heavy computational resources and eventually to build that infrastructure the cost of deploying the model increases. On the other hand the simpler model such as Logistic regression requires less computational resources, so the cost of building the model is less.

VIII. REFERENCES

[1] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and G. N. Surname, Random forest for credit card fraud detection, IEEE 15th International Conference on Networking, Sensing and Control (ICNSC),2018.

[2] Satvik Vats, Surya Kant Dubey, Naveen Kumar Pandey, A Tool for Effective Detection of Fraud in Credit Card System, published in International Journal of Communication Network Security ISSN: 2231 1882, Volume-2, Issue-1, 2013.

[3] Rinky D. Patel and Dheeraj Kumar Singh, Credit Card Fraud Detection & Prevention of Fraud Using Genetic Algorithm, published by International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, January 2013.

[4] Wen-Fang YU, Na Wang, Research on Credit Card Fraud Detection Model Based on Distance Sum, published by IEEE International Joint Conference on Artificial Intelligence, 2009.

[5] Salvatore J. Stolfo, Wei Fan, Wenke Lee and Andreas L. Prodromidis; "Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project"; 0-7695-0490-6/99, 1999 IEEE.

[6] S. Ghosh and D. L. Reilly, Credit card fraud detection with a neural- network, Proceedings of the 27th Annual Conference on System Science, Volume 3: Information Systems: DSS/ Knowledge Based Systems, pages 621-630, 1994. IEEE Computer Society Press.

[7]MasoumehZareapoor,Seeja.K.R, M.Afshar.Alam, Analysis of Credit Card Fraud Detection Techniques: based on Certain Design Criteria, International Journal of Computer Applications (0975 8887) Volume 52 No.3, 2012

[8] A. Shen, R. Tong, Y. Deng, "Application of classification models on credit card fraud detection", Service Systems and Service Management 2007 International Conference, pp. 1-4, 2007.

[9] G. Singh, R. Gupta, A. Rastogi, M. D. S. Chandel, A. Riyaz, "A Machine Learning Approach for Detection of Fraud based on SVM", International Journal of Scientific Engineering and Technology, vol. 1, no. 3, pp. 194-198, 2012, ISSN ISSN: 2277-1581.

[10] S. Patil, H. Somavanshi, J. Gaikwad, A. Deshmane, R. Badgujar, "Credit Card Fraud Detection Using Decision Tree Induction Algorithm", International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 4, no. 4, pp. 92-95, 2015, ISSN ISSN: 2320-088X.

[11] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and G. N. Surname, “Random forest for credit card fraud detection”, IEEE 15th International Conference on Networking, Sensing and Control (ICNSC),2018