Real-Time Credit Card Fraud Detection Using Machine Learning

Harwant Singh Arri[1], Sai Rohith[1], Gokul Sai[1], Sai Tharun[1], Batchu Akhilesh Gandhi[1*]

[1]*School of Computer Science and Engineering, Lovely Professional University*
Phagwara, Punjab, India

***Corresponding author email: eanuragshree@gmail.com**

***Abstract* – Credit card fraud occurs often and results in significant financial losses [1]. The number of online transactions has increased dramatically, and online shopping has become increasingly popular. Credit card transactions account for a significant portion of these transactions. As a result, banks and financial institutions provide services. credit card fraud detection software has a lot of utility. demand. Fraudulent transactions can take many forms. and can be classified into several types. The subject of this paper is four main fraud occasions in real-world transactions. Each a series of machine learning models are used to combat fraud. An evaluation is used to choose the optimal method. This assessment gives you a step-by-step approach to picking the right company. With regard to the type of frauds and the appropriate method with a suitable performance, we demonstrate the evaluation. Real-time credit card fraud detection is another important aspect of our project. As a result, we Consider the implementation of predictive analytics to determine if a machine learning model and an API module are appropriate. Is a specific transaction authentic or fraudulent? We also evaluate an innovative technique for dealing with the skewed data distribution. The information we used in our research came from a variety of sources according to an open source and community-maintained website.**

***Keywords:* Credit Card, Fraud Detection Software, Open Source, Realtime credit card fraud, Skewed Data Distribution**

I. INTRODUCTION

With the advancement of cutting-edge technology and global connectivity, fraud has risen dramatically. [5] There are two major strategies to avoid fraud: prevention and detection. By functioning as a layer of defence, prevention prevents fraudsters from attacking. After the prevention has failed, detection occurs. As a result, detection aids in detecting and notifying when a fraudulent transaction is initiated. Web payment gateways have recently grown more common with card not-present transactions [6] in credit card operations. According to the Nilson Report published in October 2021, online payment systems earned more than $31 trillion globally in 2020, up 7.3 percent from 2019. Credit card fraud losses worldwide increased to $21 billion in 2021, and are expected to reach $31 billion by 2025. [3] However, there has been a significant surge in fraudulent transactions, which has had a significant impact on the economy. Credit card fraud is a serious problem. There are various types of classifications. The two sorts of deceptions that are primarily identifiable in a collection of transactions Card-not-present (CNP) and Card-present (CP) frauds. These two categories can be further defined as follows: Bankruptcy fraud, theft/counterfeit fraud, and application fraud are all examples of fraud. as well as behavioural fraud Our research tries to address four issues. Natures of fraud that fall inside the CNP fraud category We suggest a method to detect them as indicated above Real-time frauds.

Machine learning is this generation's solution for replacing such approaches and working with enormous datasets that are difficult for humans to handle.

There are two types of machine learning techniques: supervised learning and unsupervised learning. Fraud detection can be done in any approach, with the dataset determining when to employ which method. Prior classification of anomalies is required for supervised learning. Several supervised algorithms have been applied in the detection of credit card fraud in recent years.

This study's data is analysed in two ways: as categorical data and as numerical data. Initially, the dataset contained categorical data. Data cleaning and other fundamental preparation procedures can be used to prepare the raw data. First, categorical data can be converted to numerical data, and then appropriate assessment procedures can be used. Second, to select the best method, categorical data is used in machine learning approaches.

The goal of this work is to find the best algorithms for each of the four fraud categories by comparing machine learning approaches and using an effective performance metric for detecting fraudulent credit card transactions.

The content of this paper is as follows. The literature review is presented in Section 2. The experimental technique, as well as the results, are presented in Section 3. Finally, the paper's conclusions and debates are offered in Section 4.

II. LITERATURE REVIEW

Many approaches have been proposed in previous studies to bring solutions to detect fraud, ranging from supervised approaches to unsupervised approaches to hybrid approaches; this necessitates a thorough understanding of the technologies involved in credit card fraud detection as well as a thorough understanding of the various types of credit card fraud. As fraud patterns expanded over time, creating new types of fraud, it became a major topic for researchers. The rest of this section goes on individual machine learning algorithms, machine learning models, and fraud detection systems that have been implemented in fraud detection. The issues that arose throughout the evaluation have been analysed in order to develop a more efficient machine learning model in the future.

Past researchers discovered many issues with fraud detection after analysing various detection models. They identified a lack of real-life data as a major difficulty in [14] and [3]. Because there is a scarcity of real-world data, of the sensitivity of data and privacy concerns [3] and [7] papers have looked into data that is unbalanced or has a skewed distribution. The reason for this is that there is a lot less of it. frauds in the transaction when compared to non-frauds Data mining techniques, according to paper [3,] are used to analyse large datasets. When working with large amounts of data, it's important to take the time to execute. Intersections of Another fundamental flaw in credit card preparation is data on transactions.

According to papers [2] and [7], the problem arises when normal transactions appear to be fraudulent in specific circumstances. On the other hand, fraudulent transactions may appear to be real. They've also run into difficulties dealing with categorical data. The majority of the attributes in credit card transaction data have categorical values. Almost many machine learning algorithms do not allow categorical values in this instance. They identified choosing detection methods and feature selection as a hurdle in detecting frauds in [3][4], because most machine learning techniques take far longer to train than to forecast. Another important factor in detecting financial fraud is feature selection. Its goal is to select out the characteristics that best represent fraud detection and its qualities. They identified fraud detection expense and lack of adaptability as problems in the fraud detection process in paper [7]. The cost of fraudulent behaviour as well as the cost of prevention should be considered while designing a system. When the algorithm is exposed to new sorts of fraud patterns and routine transactions, it loses its adaptability. Because effectiveness varies depending on the problem definition and specifications, a thorough grasp of the performance metric is required [4].

For the identification of credit card fraud, a variety of models are used. Different algorithms have been applied in those models.

Adapting the fraud detection system to new frauds can be difficult, and retraining the machine learning model owing to significant changes in fraud trends can be costly and dangerous. Tyler et al., for example, built on a framework described in [10], implemented the model, and applied it to a real-world transaction log. Logistic Regression (LR) was utilised to solve the categorization problem. Using Gaussian Mixture Models, the instances of fraudulent transactions were discretized into strategies (GMMs). To remedy the class imbalance, a synthetic minority oversampling technique was applied. Sensitivity analysis was utilised to highlight the relevance of estimations in economic value. The results show that a realistic strategy for retraining a model that takes minimum steps can perform as well as a classifier that retrains every round [11].

Another technique, known as Risk-Based Ensemble (RBE), can manage data containing concerns and produce excellent results. A highly efficient bagging model was utilised to handle unbalanced data. They employed the Naive Bayes approach to deal with the implicit noise in the transaction dataset [9]. Peter et al. assessed the efficacy of a number of deep learning methods. Recurrent Neural Networks (RNNs), Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTMs), and Artificial Neural Networks are the four topologies (ANNs). They used under sampling to overcome class imbalance and scalability issues in their project, in addition to data cleansing and other data preparation activities. The sensitivity analysis was used to determine which hyper-parameters had the greatest impact on the model's performance. They noticed that the size of the network had an impact on the model's performance. They came to the conclusion that the larger the network, the better the performance. [9]

Skewed distribution, often known as class imbalance, is a problem with credit card data. Andrea and colleagues claim that their project addresses class imbalance as well as other concerns like concept drift and verification latency. They've also shown how to employ the most relevant performance matrix in credit card fraud detection. A formal model and a robust learning technique for addressing verification latency, as well as an alert and feedback mechanism, are among the research's accomplishments. They have deemed the precision of the alerts to be the most essential measure based on experiments [13].

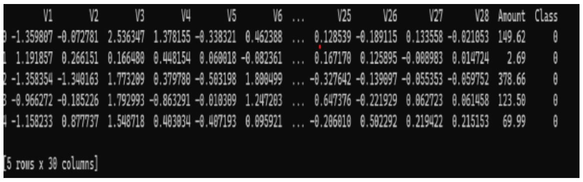
Chee et al. employed twelve standard models and hybrid approaches that included AdaBoost and majority voting to improve credit card fraud detection accuracy [14]. Both benchmark and real-world data were used to assess them. An overview of the methodologies' strengths and drawbacks was reviewed. As a performance statistic, the Matthews Correlation Coefficient (MCC) was chosen. To test the algorithms' robustness, noise was injected to the data. They also demonstrated that the majority voting mechanism was unaffected by the additional noise.

Except for accuracy, the study conducted out on extremely imbalanced data in paper [15] demonstrates that KNN performs exceptionally well in terms of sensitivity, specificity, and MCC. The paper [16] examined commonly used supervised learning techniques and gave a full evaluation of these techniques. They've also demonstrated that all algorithms alter depending on the problem.

The fraud detection method proposed in paper [17] is designed to manage class imbalance, the generation of labelled and unlabelled data, and massive dataset processing. All of the problems were overcome by the proposed system.

III. EXPERIMENTAL METHODOLOGY

We'll be working with the Kaggle Credit Card Fraud Detection dataset. V1 to V28 are the primary components produced using PCA. Because the time feature is worthless for model creation, we'll omit it. The remaining features are the 'Amount' feature, which contains the total amount of money being transacted, and the 'Class' feature, which indicates whether or not the transaction is a fraud case.



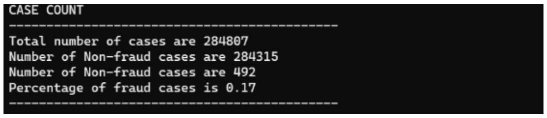
```
        V1        V2        V3        V4        V5        V6  ...       V25       V26       V27       V28  Amount  Class
0 -1.359807 -0.072781  2.536347  1.378155 -0.338321  0.462388  ...  0.128539 -0.189115  0.133558 -0.021053  149.62      0
1  1.191857  0.266151  0.166480  0.448154  0.060018 -0.082361  ...  0.167170  0.125895 -0.008983  0.014724    2.69      0
2 -1.358354 -1.340163  1.773209  0.379780 -0.503198  1.800499  ... -0.327642 -0.139097 -0.055353 -0.059752  378.66      0
3 -0.966272 -0.185226  1.792993 -0.863291 -0.010309  1.247203  ...  0.647376 -0.221929  0.062723  0.061458  123.50      0
4 -1.158233  0.877737  1.548718  0.403034 -0.407193  0.095921  ... -0.206010  0.502292  0.219422  0.215153   69.99      0

[5 rows x 30 columns]
```

Fig. 1 Kaggle Credit Card Dataset

Data Processing and EDA –

Let's have a look at how many fraud cases and non-fraud cases are there in our dataset. Along with that, let's also compute the percentage of fraud cases in the overall recorded transactions.



```
CASE COUNT
-------------------------------------------
Total number of cases are 284807
Number of Non-fraud cases are 284315
Number of Non-fraud cases are 492
Percentage of fraud cases is 0.17
-------------------------------------------
```

Fig. 2 No. of fraud cases

We can observe that there are only 492 fraud cases out of 284,807 samples, or 0.17 percent of the total samples. As a result, we may state that the data we're working with is very unbalanced and must be treated with caution while modelling and assessing it.

Data Split & Feature Selection-

We will define the independent (X) and dependent variables during this procedure (Y). We'll divide the data into a training set and a testing set based on the identified variables, which will be utilised for modelling and evaluation. Using Python's 'train test split' technique, we can simply split the data.

Modelling and testing-

Our research looks at four different types of fraud. We used the process to analyse each pattern. There are numerous numbers of the data was analysed using several methodologies. Four machine learning algorithms were prioritized in our analysis with the help of the literature

Logistic regression-

The code for Logistic regression as we kept the model in a way more simplistic manner by using the 'Logistic Regression' algorithm and as usual, fitted and stored the predicted variables in the 'lr_yhat' variable.

K-Nearest Neighbour (KNN) -

the K-Nearest Neighbour (KNN). We have built the model using the 'K Neighbour Classifier' algorithm and mentioned the 'n_neighbour' to be '5'. The value of the 'n_neighbour' is randomly selected but can be chosen optimistically through iterating a range of values, followed by fitting and storing the predicted values into the 'knn_yhat' variable.

Random forest -

The Random forest model which we built using the 'Random Forest Classifier' algorithm and we mentioned the 'max_depth' to be 4 just like how we did to build the decision tree model. Finally, fitting and storing the values into the 'rf_yhat'. Remember that the main difference between the decision tree and the random forest is that, decision tree uses the entire dataset to construct a single model whereas, the random forest uses randomly selected features to construct multiple models. That's the reason why the random forest model is used versus a decision tree.

XG BOOST -

The XG Boost model. We built the model using the 'XGB Classifier' algorithm provided by the XG boost package. We mentioned the 'max_depth' to be 4 and finally, fitted and stored the predicted values into the 'xgb_yhat'.

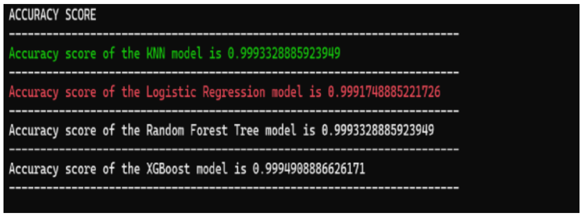
Evaluation –

Using the evaluation metrics supplied by the scikit-learn package, we will assess our constructed models during this process. Our major goal in this procedure is to select the best model for our particular situation. The accuracy score metric, the f1 score metric, and ultimately the confusion matrix will be used as assessment measures.

Accuracy Score–

Accuracy score is one of the most basic evaluation metrics which is widely used to evaluate classification models. The accuracy score is calculated simply by dividing the number of correct predictions made by the model by the total number of predictions made by the model (can be multiplied by 100 to transform the result into a percentage). It can generally be expressed as:

Accuracy score = No of correct predictions / Total no. of predictions



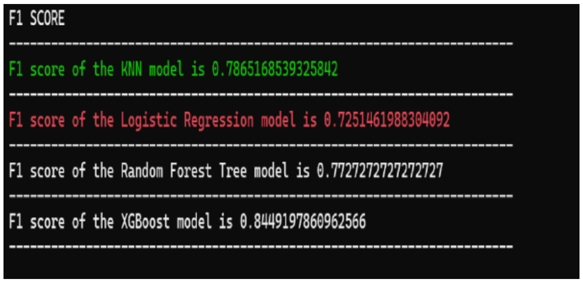
```
ACCURACY SCORE
------------------------------------------------------------------------
Accuracy score of the KNN model is 0.9993328885923949
------------------------------------------------------------------------
Accuracy score of the Logistic Regression model is 0.9991748885221726
------------------------------------------------------------------------
Accuracy score of the Random Forest Tree model is 0.9993328885923949
------------------------------------------------------------------------
Accuracy score of the XGBoost model is 0.9994908886626171
------------------------------------------------------------------------
```

Fig. 3 Accuracy Score

F1 Score –

The F1 score or F-score is one of the most popular evaluation metrics used for evaluating classification models. It can be simply defined as the harmonic mean of the model's precision and recall. It is calculated by dividing the product of the model's precision and recall by the value obtained on adding the model's precision and recall and finally multiplying the result with 2. It can be expressed as:

F1 score = 2 ((precision * recall) / (precision + recall))



```
F1 SCORE
----------------------------------------------------------------------
F1 score of the KNN model is 0.7865168539325842
----------------------------------------------------------------------
F1 score of the Logistic Regression model is 0.7251461988304092
----------------------------------------------------------------------
F1 score of the Random Forest Tree model is 0.7727272727272727
----------------------------------------------------------------------
F1 score of the XGBoost model is 0.8449197860962566
----------------------------------------------------------------------
```

Fig. 4 F1 Score

Confusion matrix -

A confusion matrix is a visualization of a classification model that shows how well the model has predicted the outcomes when compared to the original ones. Usually, the predicted outcomes are stored in a variable that is then converted into a correlation table. Using the correlation table, the confusion matrix is plotted in the form of a heatmap. Even though there are several built-in methods to visualize a confusion matrix

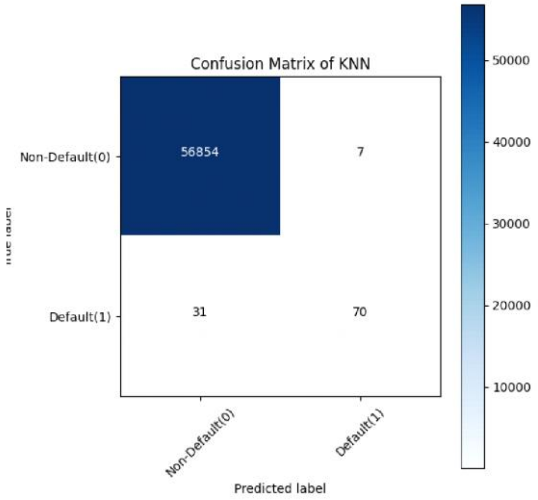


Fig. 5 Confusion Matrix of KNN

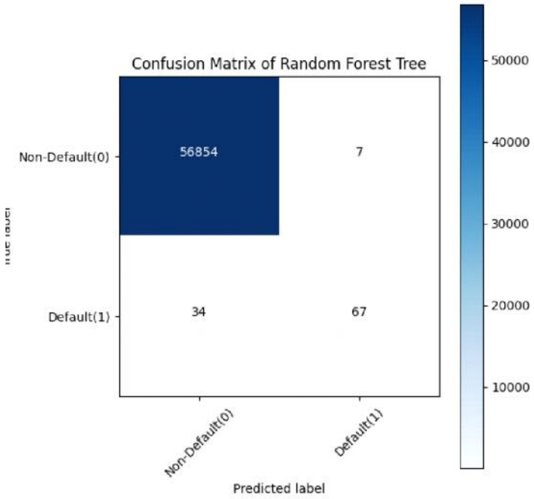


Fig. 6 Confusion Matrix of Random Forest Tree

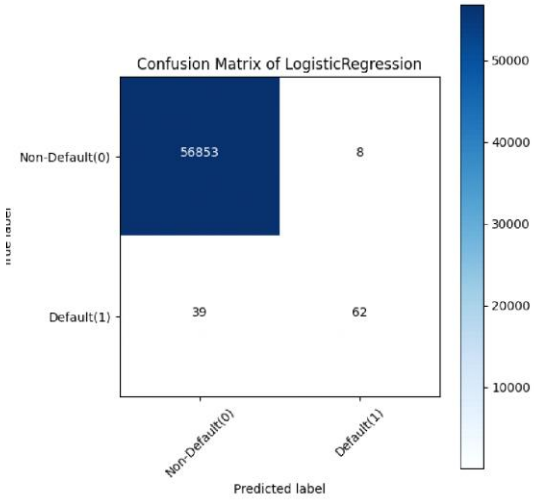


Fig. 7 Confusion matrix of Logistic Regression

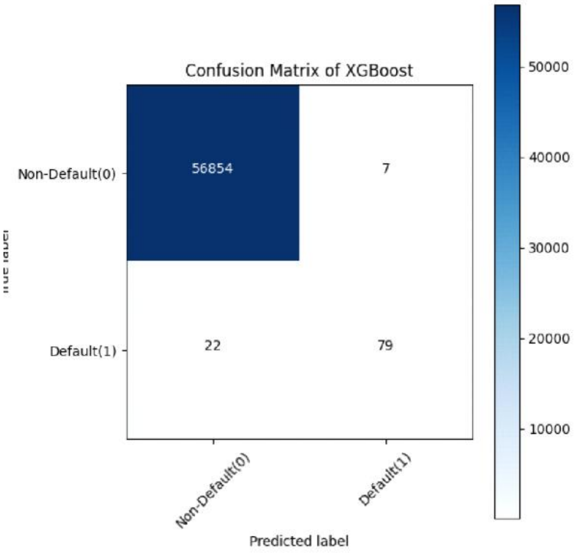


Fig. 8 Confusion Matrix of XGBoost

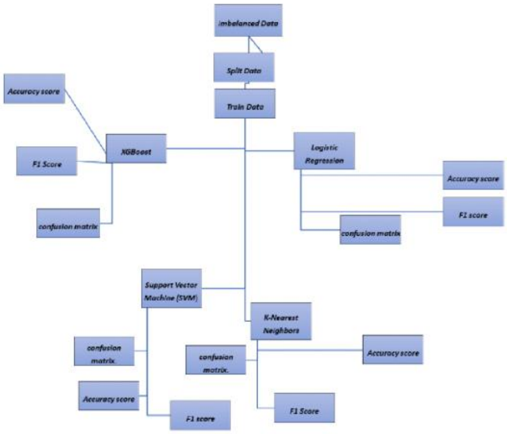


Fig. 9 Node Diagram

Fraud Detection System -

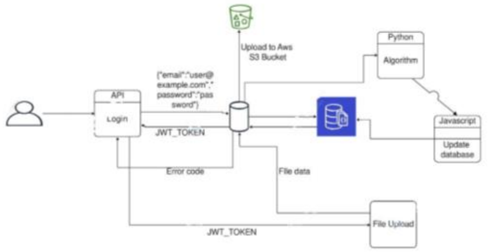


Fig. 10 Data Flow Diagram of the System

One of the group's main achievements is the ability to detect credit card theft in real time. API MODULE, FRAUD DETECTION MODELS, AWS BUCKET are the three primary components of the real-time fraud detection system. The API module is in charge of sending real-time transactions between the Fraud detection model, the GUI, and the aws bucket. The machine learning models' live transactions, expected results, and other critical data were stored in AWS bucket. The user can interact with the fraud detection system through graphical user interfaces (GUIs), which display real-time transactions, fraud alerts, and historical fraud data in a graphical representation. A message will be provided to the API module when the fraud detection model recognises a transaction as fraudulent. Then the API module will notify the end user by sending a notification and the feedback given by the end user will be stored.

IV. CONCLUSIONS

For years, researchers have been interested in credit card fraud detection, and it will continue to be an attractive topic of research in the future. This is primarily due to the fact that fraud tendencies are always changing. In this study, we propose an unique credit-card fraud detection system that uses best matching algorithms to detect four different patterns of fraudulent transactions and addresses the relevant concerns noted by previous credit-card fraud detection studies. The end user is notified over the GUI the second a fraudulent transaction occurs by addressing real-time credit-card fraud detection using predictive analytics and an API module. As soon as a suspicious transaction is spotted, this element of our system can allow the fraud investigation team to decide whether or not to proceed to the next step. As stated in the methodology, optimal algorithms that address four basic types of frauds were found through literature, experimentation, and parameter tuning. We also look at sampling approaches that effectively deal with skewed data distributions. As a result, we can conclude that applying resampling approaches has a significant impact on generating a comparatively greater performance from the classifier.

REFERENCES

1) S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and G. N. Surname, “Random Forest for credit card fraud” 15Int. Conf. Networking, Sens. Control, 2018.

2) M. Zareapoor, S.K . Seeja K.R and M. Afshar Alam, “Analysis on Credit Card Fraud Detection Techniques: Based on Certain Design criteria,” Int. J. comput. Appl., vol52, no.3, pp.35-42, 2012.

3) David Robertson,” Investments &amp; Acquisitions – September 2016 Top Card Issuers in Asia ±Pacific Card Fraud Losses Reach $21.84 Billion,” Nilson Rep., no 1096,1090.

4) J. West and M. Bhattacharya, “An Investigation on Experiment Issues in Financial Fraud Mining,” Procedia comput. Sci, vol 80, pp. 1734-1744, 2016.

5) D. S. Sisodia, N.K Reddy, and S. Bhandari,” Perfomance Evaluation of Class Balancing Techniques for Credit Card Fraud Detection”

6) G. Liu, W. Luan, Z. Li, and Y. Zhang, “A new FDS for credit card fraud detection based on behaviour certificate,” 2018.

7) Z. Zojaji, R. E. Atani and A. H. Monadjemi, “A survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented perspective,” pp.1-26,2016.

8) S. Akila and U. S. Reddy,” Risk based Bagged Ensemble (RBE) for Credit Card fraud Detection” no. Icici, pp. 670-674, 2017.

9) M. F. Zeager, A. Sridhar, N. Fogal, S. Adams, D. E. Brown, and P. A. Beling, “Adversarial learning in credit card fraud detection,”2017 Syst. Inf. Eng. Des. Symp., pp. 112±116, 2017.

10) T. Cody, S. Adams and P. A. Beling, “A utilitarian Approach to Adversarial Learning in Credit card Fraud Detection,” pp 237- 242, 2018.

11) M. Rafalo, “Real-time fraud detection in credit card transactions” Data Science Warsaw. 2017.

12) A. Dal Pozzolo, G. Boracchi, O.Calen, and C.Alippi,”credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy, “ Ieee Trans. Neural Networks Learn. Syst., pp. 1-14,2018.

13) K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, “Credit Card fraud Detection using adaBoost and majority voting” IEEE Access, vol. XX, pp. 1±1, 2018.

14) J. O. Awoyemi, A. O. Adetunmbi, and S.A.Oluwadare, “Credit Card fraud detection using machine learning techniques: A comparative analysis 2017 Int. Conf. Comput. Netw. Informatics, pp. 1-9, 2017.

15) R. Choudhary and H. K. Gianey, “Comprehensive Review On Supervised Machine Learning Algorithms,” 2017 Int. Conf. Mach. Learn. Data sci., pp. 37-43 2017.

16) G. E. Melo-Acosta, F. Duitama-Muñoz, and J. D. Arias-Londoño, “Fraud detection in big data using supervised and semi-supervised learning techniques,” Comun. Comput. (COLCOM), 2017 IEEE colomb. Conf., pp 1-6, 2017.