# Credit Card Fraud Detection on Skewed Data using Machine Learning Techniques

E.A. Amusan
Dept. of Cyber Security Science, Ladoke Akintola University of Technology, Ogbomoso, Oyo State
eaadewusi@lautech.edu.ng

O.M. Alade
Dept. of Cyber Security Science, Ladoke Akintola University of Technology, Ogbomoso, Oyo State
oalade75@lautech.edu.ng

O.D. Fenwa
Dept. of Cyber Security Science, Ladoke Akintola University of Technology, Ogbomoso, Oyo State
odfenwa@lautech.edu.ng

J.O. Emuoyibofarhe
Dept. of Information Systems, Ladoke Akintola University of Technology, Ogbomoso, Oyo State
eojustice@gmail.com

*All authors are corresponding authors.

**ABSTRACT**
The fraud associated with credit card transaction is increasing at an alarming rate and consequently resulting in huge financial loss for both the cardholders and concerned financial institutions. Most datasets for real-life problems such as this are usually imbalanced, which makes the machine learning model not robust for training purposes. Therefore, this research aimed to detect 100% of the fraudulent transactions while minimizing the incorrect classifications by first, balancing the data using under-sampling technique and then developing classification models using different machine learning algorithms such as Logistic regression, Random forest, K nearest neighbor and Decision tree classifiers. The performance of the models are evaluated based on accuracy, precision and recall and the results indicated that Random Forest recorded the highest accuracy, precision and recall of 95.19%, 97.94%, and 0.9226 respectively compared to the other three (3) algorithms.

**Keywords:** accuracy, credit card fraud, data imbalance, machine learning, precision, skewed data, under-sampling.

## 1. INTRODUCTION

The advancement in technology and availability of information has paved way for fraudsters and cybercriminals to steal credit card details and do frauds. Due to these frauds, banks, companies, product vendors and cardholders are largely affected and are susceptible to huge financial loss. Consequently, there is a need to detect fraud in credit cards transactions. It is important that financial institutions that deal with credit cards are able to recognize fraudulent credit card transactions so that customers are not wrongly charged for items which they did not purchase. The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be fraudulent. This model is then used to classify a new transaction as fraudulent or not.

In a detection problem such as this, the ratio of distribution of classes is pivotal in ascertaining precision and classification accuracy. Data imbalance is a huge challenge for conventional machine learning algorithms, which are not adequately suited for handling uneven class distributions and tend to display a bias toward the majority class accompanied by a reduced discriminatory capabilities on the minority classes (Koziarski, 2020). There are several approaches for creating a balanced dataset from imbalanced data, they are identified in literature (Jayaswal, 2020**)** as oversampling and under-sampling.

i. Oversampling: this approach resamples the minority class points to equal the total number of majority points. Repetition of the minority class points is one such type of oversampling technique. Apart from repetition, class weights can be provided to both classes. Providing the large weights to the minority class will give the same result as that of repetition.

ii. Synthetic Minority Oversampling Technique (SMOTE): this is another popular resampling approach that uses both undersampling and oversampling to balance the original dataset by randomly choosing instances from the minority class to

create new instances of the same (Mishra and Ghorpade, 2018). It differs from the conventional oversampling technique in the sense that SMOTE uses K nearest neighbour (KNN) for selecting instances of the minority class rather than just replicating same. The drawback of this approach is that it generates synthetic data which might not perfectly represent the original data.

iii. Under-sampling: this resamples the majority class points in the data to make them equal to the minority class points. In this case, a new dataset will emerge out of the original dataset using under sampling (Pozzolo, Caelen, Johnson and Bontempi, 2015). The majority class will be randomly sampled so as to equate it to the minority class. The main drawback with this approach is that a significant chunk of the data, which contains some information is not utilized**.**

In this work, the undersampling technique was used. This is because we aim to detect fraudulent transactions more correctly and undersampling does not in any way alter the fraud data in the original dataset. This paper presents how different machine learning algorithms such as Logistic regression, random forest, K nearest neighbor and Decision tree were implemented on an imbalanced dataset using resampling techniques and the rest of the paper is organized as follows. Section 2 discusses works related to this research. Section 3 describes the various machine learning algorithms. Section 4 describes the methodology which features the steps in developing the credit card fraud detection system. Section 5 highlights and discusses the experimental results of the detection system. Lastly, section 6 concludes the paper.

## 2. RELATED WORKS

Bhattacharyya, Jha, Tharakunnel & Westland (2011) evaluated the performance of logistic regression alongside support vector machines and random forests in credit card fraud detection. Their findings revealed that logistic regression maintained similar performance with different levels of under-sampling, while SVM performance tend to increase with lower ratio of fraud in the training data. Logistic regression shows significant performance, exceeding that of the SVM models with different kernels. Also, Puh and Brkic in 2019 developed a detection system for credit card fraud using SVM, Random forest and Logistic Regression. In particular, the minority class was oversampled using the SMOTE technique. In another study, Awoyemi, Adetunmbi and Oluwadare (2017) investigated the performance of naïve bayes, k-nearest neighbor and logistic regression on highly skewed credit card fraud data where a hybrid technique of under-sampling and oversampling was carried out on same. The comparative results of their findings show that k-nearest neighbour performed better than naïve bayes and logistic regression techniques. Similarly, Varmedja *et al* (2019) compared Logistic Regression (LR), Random Forest (RF), Naïve Bayes (NB) and Multilayer Perceptron (MLP) in order to determine which is most suitable for credit card fraud detection. Among the four algorithms, Random forest proved to be the best, with accuracy of 99.96%, followed by MLP with 99.93%, NB's accuracy was 99.23% and then, LR with 97.46%. However, accuracy alone as a metric is not sufficient for evaluation as the accuracies were obviously high, their precision were 96.38%, 79.21%, 16.17% and 58.82% respectively.

## 3. PERFORMANCE ANALYSIS OF THE CLASSIFICATION ALGORITHMS USED

### i. Logistic Regression Algorithm

Logistic regression algorithm is similar to linear regression algorithm, however the latter is predominantly used to predict or forecast values while the former is used for classification tasks. This algorithm is easy for both binary and multivariate classification tasks. Binomial is of 2 possible types (i.e. 0 or 1) only while multinomial is of 3 or possible types and which are not ordered (Bhanusri *et al*, 2020) in category such as very poor, poor , good, very good.

### ii. Random Forest Algorithm

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks (Devi, Janani, Gayathri and Indira, 2019)

### Working of Random Forest

The following are the basic steps involved in performing the random forest algorithm

*1.* Pick N random records from the dataset.
*2.* Build a decision tree based on these N records.
*3.* Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
*4.* For classification problem, each tree in the forest predicts the category to which the new record belongs.
Finally, the new record is assigned to the category that wins the majority vote.

### iii. K Nearest Neighbor Algorithm

The k-nearest neighbour is an instance based learning which carries out its classification based on a similarity measure, like Euclidean, Mahanttan or Minkowski distance functions (Awoyemi, Adetunmbi and Oluwadare, 2017). The first two distance measures suit continuous variables while the third is best for categorical variables. For every data point in the dataset, the Euclidean distance between an input data point and current point is calculated. These distances are sorted in increasing order and k items with lowest distances to the input data point are selected. The majority class among these items is found and the classifier returns the majority class as the classification for the input point. Equation (1) represents the Euclidean distance (Dij) between two input vectors (Xi, Xj):

$$D_{ij} = \sqrt{\sum_{k=1}^{n}(X_{ik} - X_{jk})^2} \qquad k = 1,2,3,.....n \qquad (1)$$

### iv. Decision Tree Classifier

Decision trees can be used divisibly and explicitly used to represent decisions and decision-making. The decision tree classifier provides the classification of instances where each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by the node, then moving down the tree branch corresponding to the value of the attribute. This process is then continued for the subtree rooted at the new node (Hudali, Mahalaxmi, Magadum and Belagali, 2019).

## 4. METHODOLOGY

This section describes the different stages of the experiment. The steps for building the fraud detection system to analyze the data and predict the likelihood of a transaction being fraudulent or otherwise are outlined below:

i. Data Collection

ii. data visualization

iii. Data pre-processing

iv. predictive split

v. Model implementation

vi. Performance evaluation

### A. Data collection

The dataset used in this research is the credit card fraud detection dataset from Kaggle, a data analysis website that makes datasets available for use (Kaggle.com). It is composed of 284,807 transactions made by European credit card holders among which 492 turned out to be fraudulent accounting for 0.17% of all transactions. It contains 31 features of which the first 28 are numerical input variables labeled V1, V2, V3..... V28. The other three features are 'Time', 'Amount' and 'Class'. Feature 'Time' is the transaction time in seconds 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 while feature 'Class' is the response or target variable and it takes value 1 in case of fraud and 0 otherwise.

### B. Data Visualization

The purpose of this step is to provide a sneak peek into the data distribution. It gives us insight into the data and shows if is evenly distributed or skewed, if there are outliers or not, etc. The data set is very unbalanced, highly skewed towards the non-fraud class since the target class (fraudulent transactions) is only 0.17% of all transactions as shown in Figure 1. If we use this data frame as presented to train and construct models, we will probably get a lot of false classifications due to overtraining/over fitting of the model. The resulting model will assume that the transaction is likely to be a non-fraudulent one, since almost all of the data set consists of such transactions, but we do not want our model to assume, we want our model to detect patterns that give signs of fraud.

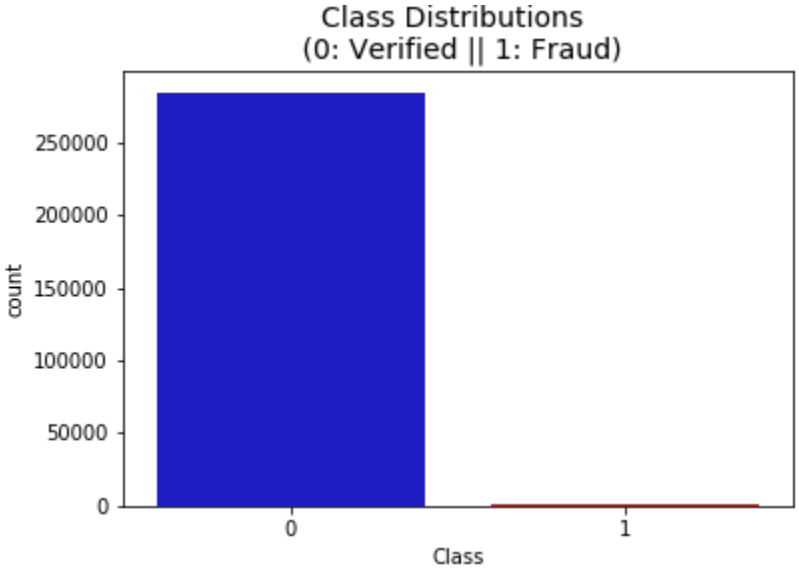


**Figure 1: Class Distributions of Fraudulent and Non-fraudulent Transactions**

Another insight into the data is that there are no null values, hence there is no need to fill up missing values.

## C. Data Preprocessing

Having observed that the distribution is highly skewed, there is therefore the need to create a balanced subset of data with the same frequency of fraudulent and non-fraudulent transactions, such that the dataset will have a ratio of near 50/50 fraudulent and normal transactions which will further help algorithms to show more precise results.
Hence, we created a sub-sample data frame from the dataset in order to have an equal amount of Fraud and Non-Fraud cases, helping our algorithms better understand patterns that determines whether a transaction is a fraudulent or not. There are 492 cases of fraud in the original dataset, so we randomly got 492 cases of non-fraud to create the new sub dataframe which was concatenated to have a balanced ratio of fraudulent and non-fraudulent transactions cases.

The balanced dataframe which presents both minority and majority class in equal proportion is as shown in Figure 2.


**Figure 2: Rebalanced (undersampled) Dataframe**

**D. Predictive Modeling Split**: At this stage, we split the data into training and testing subsets. The data is split using the train_test_split() library. The dataset was divided in ratio 70:30(train: test), where 70% of dataset is given for training of the model and 30% of dataset is used for testing the model.

**E. Model Implementation**: A predictive model is implemented based on four (4) different machine learning algorithms which are Linear regression, Random forest, K- nearest neighbour and Decision tree algorithms.

**F. Performance Evaluation**: Lastly, a confusion matrix is calculated to evaluate the performance of the algorithm based on accuracy, precision and recall.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FP} \quad (2)$$

TP- True Positive: The true positive rate represents the portion of the fraudulent transactions correctly being classified as fraudulent transactions.

TN- True Negative: The true negative rate represents the portion of the normal transactions correctly being classified as normal transactions.

FP- False Positive: The false positive rate indicates the portion of the non-fraudulent transactions wrongly being classified as fraudulent transactions.

FN- False Negative: The false negative rate indicates the portion of the non-fraudulent transactions wrongly being classified as normal transactions.

## 5. EXPERIMENTAL RESULTS

Having experimented the Logistic regression, Random forest, KNN and Decision tree classifiers on the under-sampled dataset, Table 1 shows the detection performance of the classifiers for fraudulent transaction detection evaluated by accuracy, precision and recall. The table indicates that Random Forest recorded highest precision, recall and accuracy compared to the other three algorithms. While the accuracies of the classifiers are well over 90% except that of the Decision tree classifier, the aim of this research was to detect fraudulent transactions more precisely rather than improving classifier accuracy. The classification reports and confusion matrices are represented by Figures 3-6 where class 0 means the transaction was determined to be valid and 1 means it was determined as a fraud transaction.

**Table 1: Detection Performance of Machine Learning Classifiers**

| Classifier | Accuracy | Precision | Recall |
|---|---|---|---|
| Logistic Regression | 94.87% | 0.9793 | 0.9161 |
| Random Forest | 95.19% | 0.9794 | 0.9226 |
| K Nearest Neighbour | 93.27% | 0.9653 | 0.8968 |
| Decision Tree | 89.74% | 0.9128 | 0.8774 |

## i. Result with Logistic Regression Classifier

The model used is LogisticRegression Classifier

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.92 | 0.98 | 0.95 | 157 |
| 1 | 0.98 | 0.92 | 0.95 | 155 |
| accuracy | | | 0.95 | 312 |
| macro avg | 0.95 | 0.95 | 0.95 | 312 |
| weighted avg | 0.95 | 0.95 | 0.95 | 312 |

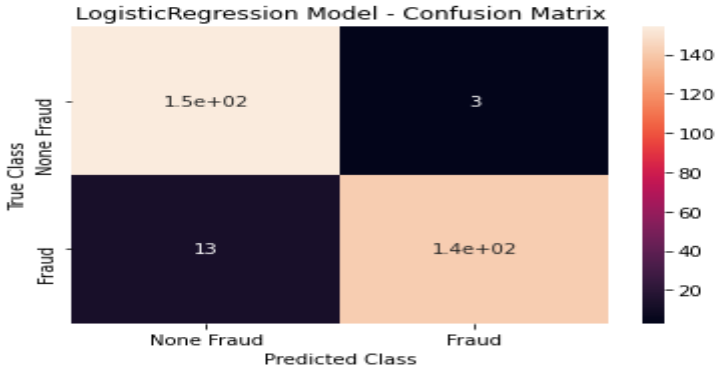


**Figure 3: Metrics for Logistic regression classifier model**

## ii. Result with Random Forest Classifier

The model used is RandomForest Classifier

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 0.98 | 0.95 | 157 |
| 1 | 0.98 | 0.92 | 0.95 | 155 |
| accuracy | | | 0.95 | 312 |
| macro avg | 0.95 | 0.95 | 0.95 | 312 |
| weighted avg | 0.95 | 0.95 | 0.95 | 312 |

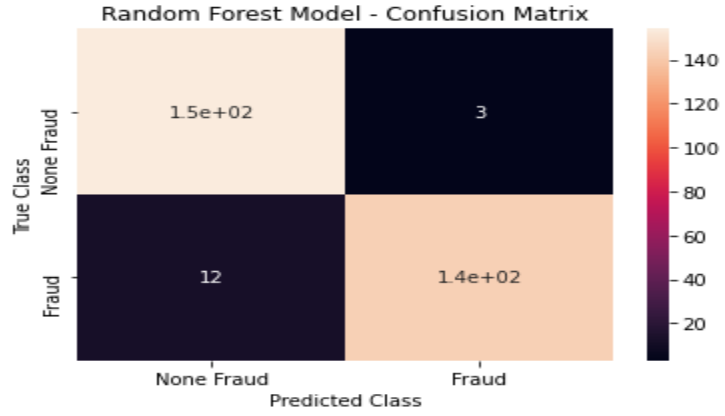


**Figure 4: Metrics for Random forest classifier model**

## iii. Result with K Nearest Neighbour Classifier

```
The model used is KNeighborsClassifier
              precision    recall  f1-score   support

           0       0.90      0.97      0.94       157
           1       0.97      0.90      0.93       155

    accuracy                           0.93       312
   macro avg       0.94      0.93      0.93       312
weighted avg       0.93      0.93      0.93       312

[[152   5]
 [ 16 139]]
```

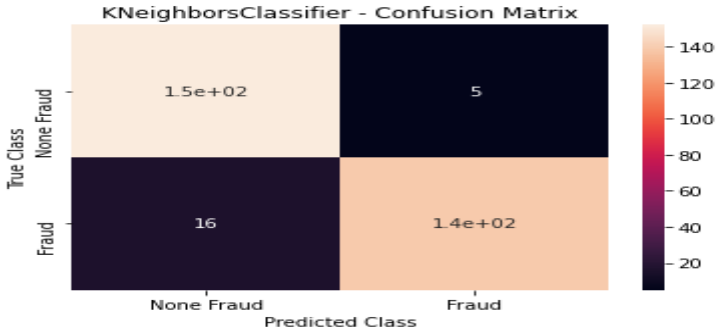


**Figure 5: Metrics for KNearest neighbour classifier model**

## iv. Result with Decision Tree Classifier

```
The model used is DecisionTreeClassifier
              precision    recall  f1-score   support

           0       0.88      0.92      0.90       157
           1       0.91      0.88      0.89       155

    accuracy                           0.90       312
   macro avg       0.90      0.90      0.90       312
weighted avg       0.90      0.90      0.90       312
```

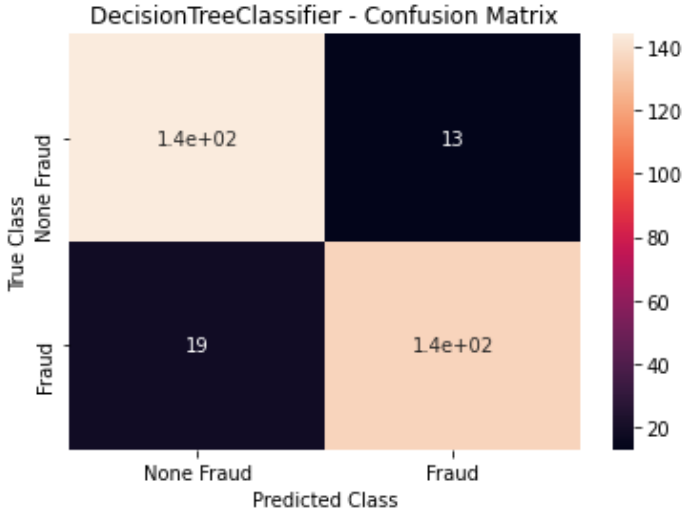


**Figure 6: Metrics for Decision tree classifier model**

## 6. CONCLUSION AND RECOMMENDATION

Credit card fraud is without a doubt a cybercrime. This paper has explained in detail, how machine learning can be applied to get better results in fraud detection. The goal of this paper was to develop a fraud detection system using certain machine learning algorithms and to evaluate the performance of same for detection of fraudulent transactions. Hence, comparison was made and it was established that Random Forest algorithm gave the best results with a precision of 97.94% and recall of 0.9226. For a detection problem of this sort, it is important to have recall with high value. The Random forest model best detects fraudulent transactions more precisely in this work.

The main issue with "Random Under-Sampling" is that we run the risk that our classification models will not perform as accurate as we would like to since there is a great deal of information loss (bringing 492 non-fraud transactions from 284,315 non-fraud transactions), hence, it is recommended that other data balancing techniques be explored.

## 7. REFERENCES

Andhavarapu Bhanusri "Credit card fraud detection using Machine learning algorithms" *Quest Journals Journal of Research in Humanities and Social Science, vol. 08(02), 2020, pp. 04-11.*

Andrea Pozzolo, Olivier Caelen, Reid Johnson and Gianluca Bontempi (2015): Calibrating Probability with Under sampling for Unbalanced Classification. In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE.

Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)* (pp. 1-9).

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, *50*(3), 602-613.

Devi Meenakshi. B, Janani. B, Gayathri. S, Mrs. Indira. N (2019). Credit Card Fraud Detection Using Random Forest, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056, p-ISSN: 2395-0072, Vol. 6 Issue: 03, pp. 6662-6666.

Hudali J.A., Mahalaxmi K.P., Magadum N.S. and Prof. Belagali S. (2019): Credit Card Fraud Detection by using ANN and Decision Tree in Journal of Advancement of Computer Technology and its Applications, HBRP publications, Volume 2 Issue 3, pp. 1-4.

Jayaswal V. (2020): Dealing with Imbalanced dataset, Techniques to handle imbalanced data, available online at https://towardsdatascience.com/dealing-with-imbalanced-dataset-642a5f6ee297

Kaggle.com. (2021). Credit Card Fraud Detection. [online] Available at: https://www.kaggle.com/mlg-ulb/creditcardfraud [Accessed 15th Jan. 2021].

Koziarski, M. (2020). CSMOUTE: Combined Synthetic Oversampling and under sampling Technique for Imbalanced Data Classification. *arXiv preprint arXiv:2004.03409*.

Mishra, A. and Ghorpade, C. (2018). *Credit Card Fraud Detection on the Skewed Data Using Various Classification and Ensemble Techniques. 2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS).*

Puh M. and Brkic L. (2019): Detecting Credit Card Fraud Using Selected Machine Learning Algorithms, MIPRO, DOI: 10.23919/MIPRO.2019.8757212, pp.1250-1255.

Varmedja D., Karanovic M., Sladojevic S., Arsenovic M. and Anderla A. (2019): Credit Card Fraud Detection - Machine Learning methods, 18th International Symposium INFOTEH-JAHORINA, IEEE. Doi: 10.1109/infoteh.2019.8717766.