Fraud Detection Model for Illegal Transactions

[1*]Musibau Adekunle Ibrahim, [2]Patrick Ozoh, [3]Oladotun Ayotunde Ojo

Department of Computer Science, Faculty of Computing and Information Technology, Osun State University, Osogbo, Nigeria

email: [1*]kunle_ibrahim2001@yahoo.com, [2]patrick.ozoh1@uniosun.edu.ng, [3]dotun4realoj@gmail.com

**Corresponding author*

Received: 03 January 2024 | Accepted: 27 February 2024 | Early access: 11 March 2024

Abstract - *Due to advancements in network technologies, digital security is becoming a top priority worldwide. This project aims to study how machine learning classifier such as random forest could be used to learn patterns in fraudulent and legitimate transactions in order to detect fraudulent transactions using Python programming language on Jupyter notebook as an Integrated Development Environment. Scikit-learn was used to develop algorithm, streamlit and heroku platforms for proper and efficient detection and classification of unauthorized transactions. This was incorporated into a web application that allows users to upload data that can be analyzed by the system to detect fraud. The Classification report and Confusion matrix have been used to evaluate each model's accuracy. Random forest as a classifier model gave an accuracy of 99.95%. At the end of this study, a web-based application has been developed to upload data and detect fraudulent in online based transactions.*

Keywords: Confusion Matrix, Financial Transaction, Credit Card, Fraud Detection and Machine Learning.

1 Introduction

Fraud is an art and crime of deceiving and scamming people in their financial transactions. Credit card fraud is a broad term used to define fraud that is committed using a payment card (David, 2021). The initial incident of credit card fraud occurs when a fraudster either steals a physical card, or illegally obtains a victim's card details. Credit card generally refers to a card that is assigned to a customer (cardholder), which usually allowing him/her to purchase goods and services within credit limit or withdraw cash in advance. It offers cardholders a time advantage, allowing them to defer repayment until a specified period by carrying it over to the next billing cycle, thus reducing immediate time constraints. The concept of fraud is present in the earliest writings of history and has since developed into an evolutionary subset of financial fraud (Berk, 2019).

The growing development of online transactions have increased rapidly over the last decade due to advancements in network technologies making it the most popular payment method for online purchases, meaning that credit cards and other online payment models are involved. Businesses, Companies, Finance companies and Institutions now provide online services such as e-commerce for easy accessibility and efficiency of online activities.
Credit card usage has enormously been increased during the last years according to Suvasini et al. (2019), 120 million cards were created in Germany and brought into use from 2004, which led to total credit card purchases of €375 billion at the same year. With respect to usage from 2005, there was an increase of 4% on the overall credit card usage (Shabad & Kavitha, 2019).

Although the use of credit cards as a payment method can be convenient for our daily transactions; people must be aware of the risks that they impose themselves while using their credit cards. More precisely, the incremental usage of credit cards gave the opportunity to fraudsters to exploit their vulnerabilities (Srivastava et al., 2019). In United States, the total losses for 2019 were as high as $3.56 billion; an increase of 10.2% comparing to the previous year. An interesting question arises as to who is responsible to pay for all those losses in case of a credit card fraud. (Srivastava et al., 2019) claim that merchants are really vulnerable in case of a credit card fraud because they are required to pay for the losses due to the so-called charge-backs. Chargebacks are requested by the consumer's bank as soon as the consumer reports a transaction as unauthorized.

The scam usually occurs when someone accesses your credit or debit card numbers from unsecured websites or via an identity theft scheme to fraudulently obtain money or property. Due to its recurrence and financial institutions, it is crucial to take preventive measures as well as identifying when a transaction is fraudulent. Necessary prevention measures can be taken to stop this abuse of fraudulent practices that can be studied to minimize it and protect against similar occurrences in the future. Due to advancement of fraudulent attacks in our society, advanced fraud detection model (FDS) is required to detect fraudulent transactions. In this paper, advanced fraud detection would therefore be developed to curb these cyber-criminal attacks.

2 Literature review

Machine learning uses algorithms to predict or classify data based on previous data therefore learning from past data characteristics to accurately classify or predict new data (Talabis, 2019). Algorithms used in machine learning to predict credit card fraud can be classified into two groups: supervised and unsupervised learning. The use of neural network is a hybrid form of machine learning that uses both supervised and unsupervised learning. The structure of this type of machine learning mimics the functions of a human brain, similarly to brain function, it uses associative memory and pattern recognition to predict outcomes of future events. According to the majority of fraud detection model, studies are based on neural networks because of its ability to learn from the past therefore allowing it to get better with time as it fed more data (Mohammed et al., 2019).

(Melo-Acosta, 2020) proposed a credit card fraud detection model that tackles scalability issues and imbalanced datasets in existing models. The main objective of the model is to reduce discrepancies such as scalability issues, low response time, and inefficiency. The model contained some datasets that were inputted for credit card fraud detection; the dataset was divided into two before analysis. This model component was replicated in the design of the model for detecting fraud to reduce scalability and increased efficiency. Mareeswari (2019) suggested an implementation of Artificial Neural Networks (ANNs) for detecting credit card fraud. Their implementation considers a sequence of transactions that have occurred at some time in the past, in order to determine whether a new transaction is legitimate or fraudulent. They believe that “looking at individual transactions” is misleading since it cannot face any periodical changes in spending behavior of a customer (Shiyang et al., 2019). They refer to their approach as “Long Short-term Memory Recurrent Neural Network (LSTM)”.

Manson (2020) suggest a different implementation of ANNs by converting the training samples into confidence values using a specific mathematical formula and then supply these values to train ANN instead of the original training samples. They call their approach as “confidence-based neural network” and they claim that it can achieve promising results in detecting credit card fraud.

Another implementation of ANNs is suggested by Maniraj et al. (2019). They use genetic algorithm; the details of which can be found in Maniraj et al. (2019). “A genetic algorithm tutorial”, Statistics and Computing could also be used to derive the optimal parameters of ANN as stated in Hand (2019). Like many other data mining techniques, ANNs make use of several parameters which need to be specified by software developers. Although the values of theses parameters can seriously affect the predicting accuracy of ANN models; a standard practice for specifying these parameters has never been established. The use of genetic algorithm which is suggested by Benson et al. (2020) can help in deciding these optimal parameters. They refer to their approach as “Genetic Algorithm Neural Network (GANN)”.

Card transactions are always unfamiliar when compared to previous transactions made by the customer. This unfamiliarity is a very difficult problem in real-world. The proposed model for this project is to design and create an application that uses machine learning algorithms that learns from previous fraudulent transactions in order to analyze online card transactions and detect fraudulent activity. A comprehensive survey conducted by Hand (2019) and his associates has revealed that techniques employed in this domain include data mining applications, automated fraud detection and adversarial detection. Unconventional techniques such as hybrid data mining or complex network classification algorithm is able to perceive illegal instances in an actual card transaction data set, based on network reconstruction algorithm that allows creating representations of deviation of one instance from a reference group, an adequate proved has been shown for the inefficient typically on medium sized online transaction.

The proposed model aims at solving some of the aforementioned problems in literature in terms of fraudulent activities that are very rampant in our society today. In the literature, it was discovered that some algorithms could not effectively detect illegal activities while some combine different methods for solving the problems of frauds, which can lead to inefficiency and low speed performance of algorithms. All these errors would be alleviated in the proposed model. The proposed model would technically improve the existing model by introducing an alert feedback interaction that would only grant authorized users access to the system and hence prevent some fraudulent activities in order to deny any illegal activities in online transactions.

3 Methodology

Card transactions are always unfamiliar when compared to previous transactions made by the customer. This unfamiliarity is a very difficult problem in real-world. The proposed system for this project is to design and create an application model that uses machine learning algorithms that learns from previous fraudulent transactions in order to analyze online card transactions and detect fraudulent activity. This allows practitioners/users to upload transaction data and the results were displayed.

Data was collected from an online anonymized dataset in Kaggle's website. The dataset contains 984 transactions and 32 features. Because of the anonymity of the dataset, most features are represented as V1-V28 which are undisclosed. Table 1 below shows basic features that have been captured when any transaction is made and would be utilized in this project.

Table 1: Raw features of credit card transactions

Attribute name	Description
Transaction id	Identification number of a transaction
Cardholder id	Unique Identification number given to the cardholder
Amount	Amount transferred or credited in a particular transaction by the customer
Time	Details like time and date, to identify when the transaction was made
Label	To specify whether the transaction is genuine or fraudulent

Scikit-learn is a machine learning tool that uses Python to develop machine learning models, this library has been employed in this research for faster processing of data since Python is a general-purpose language. Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. It allows users to build and deploy powerful data apps in minutes. Again this library has been choosing to develop the proposed system with new features to tackle some of the problems of existing models. In order to successfully perform a sufficient data preparation step for the system model, a deep understanding of the data is needed, this ensures data quality and availability of quality data being fed to the model for the model to have maximum performance. The dataset collected from Kaggle contains 269 fraudulent transactions out of 419 transactions. The difference between fraudulent and normal transactions shows a large gap, which tells us that the data is very imbalanced, this can have a negative effect on the model such that when it makes a prediction, it does so with high accuracy while unknown to the users that the algorithm is only making predictions for only one class which is the dominating class. We will need to balance it so we can build a model capable of identifying fraudulent transactions. In this case, Synthetic Minority Over-Sampling Technique (SMOT) will be used to perform the oversampling on the dataset by selecting 484 normal cases and 484 fraud transactions to make a balanced dataset. Diagrammatical representation of our model based on the above explanation is shown in Figure 1.

4 Results and Discussion

The following section explains the system development based on the modeling and designs specified in previous chapters. Code screenshots were used to highlight the functionalities of the system. It presents results for model-based machine learning techniques for predicting credit card fraud deployed using Heroku. From the data preparation, where the dataset was preprocessed and SMOT was performed on it to make a balanced dataset. A screenshot of the code used to implement the data sampling is shown in Figure 2. The new sample is created as shown in the image below. The imbalance data has 303 normal observations and 484 fraud observations, while after oversampling, the balanced dataset has 484 normal transactions and 484 fraudulent transactions. The code below is used to create the random forest model, amongst the rest (KNN, Decision tree, neural network) before creating the model, the feature selection method is used to select features fed into the model based on their importance. For neural network and KNN, they are being modelled with all the features. Figures 3 and 4 below display a screenshot of the modeled data and a graph illustrating the feature importance.

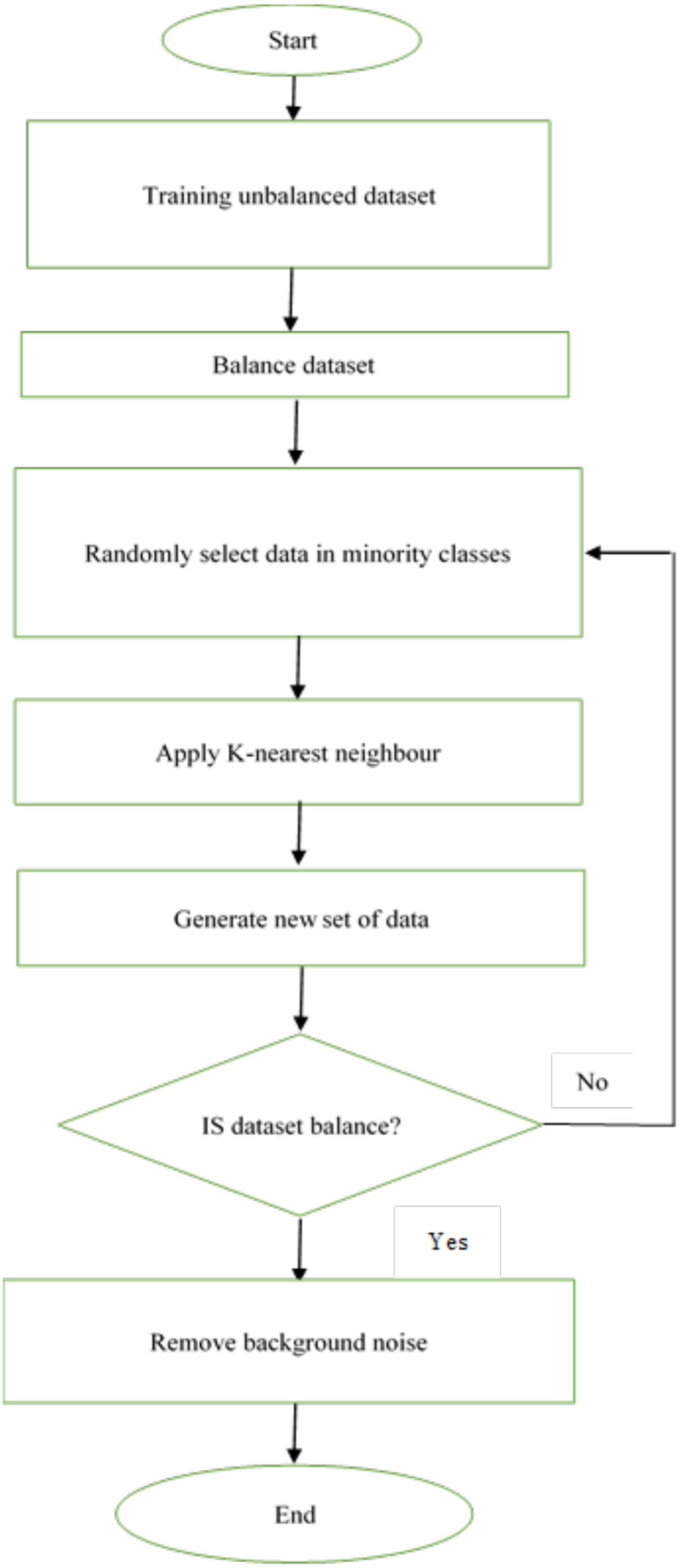
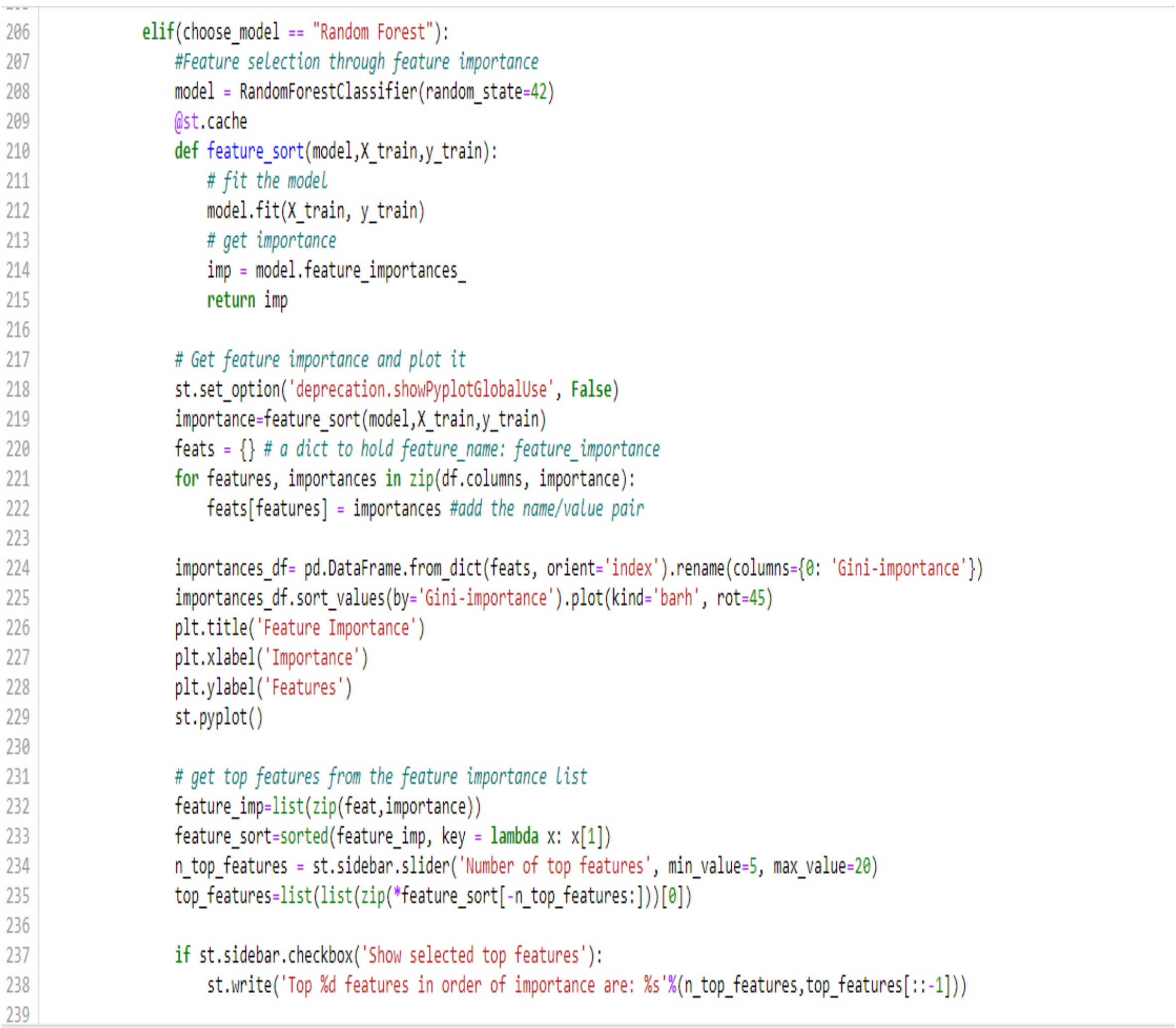
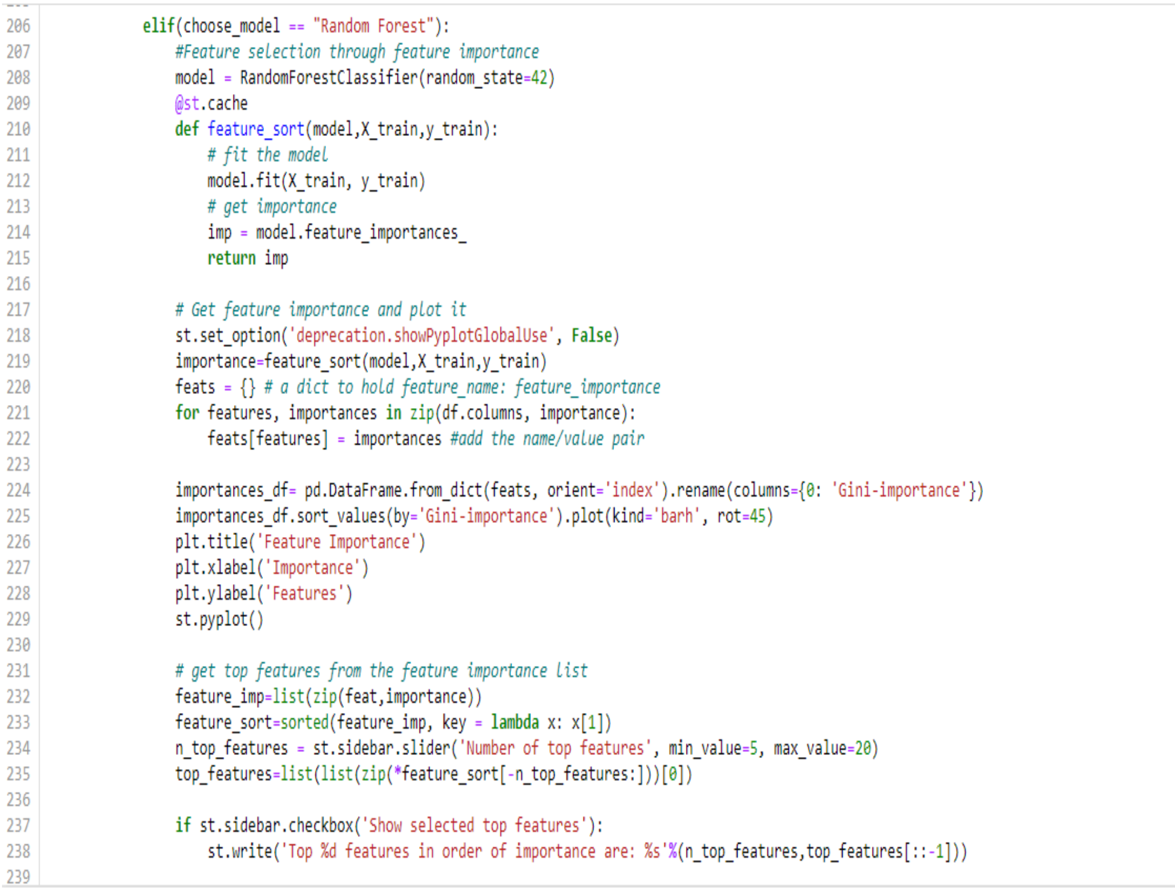


Figure 1: SMOT flowchart for prediction model



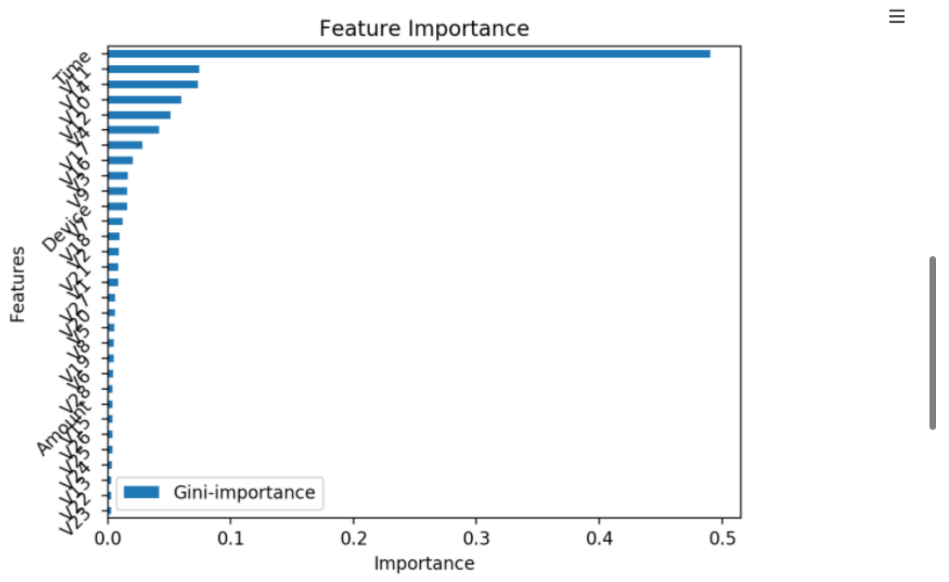
```
        elif(choose_model == "Random Forest"):
            #Feature selection through feature importance
            model = RandomForestClassifier(random_state=42)
            @st.cache
            def feature_sort(model,X_train,y_train):
                # fit the model
                model.fit(X_train, y_train)
                # get importance
                imp = model.feature_importances_
                return imp

            # Get feature importance and plot it
            st.set_option('deprecation.showPyplotGlobalUse', False)
            importance=feature_sort(model,X_train,y_train)
            feats = {} # a dict to hold feature_name: feature_importance
            for features, importances in zip(df.columns, importance):
                feats[features] = importances #add the name/value pair

            importances_df= pd.DataFrame.from_dict(feats, orient='index').rename(columns={0: 'Gini-importance'})
            importances_df.sort_values(by='Gini-importance').plot(kind='barh', rot=45)
            plt.title('Feature Importance')
            plt.xlabel('Importance')
            plt.ylabel('Features')
            st.pyplot()

            # get top features from the feature importance list
            feature_imp=list(zip(feat,importance))
            feature_sort=sorted(feature_imp, key = lambda x: x[1])
            n_top_features = st.sidebar.slider('Number of top features', min_value=5, max_value=20)
            top_features=list(list(zip(*feature_sort[-n_top_features:]))[0])

            if st.sidebar.checkbox('Show selected top features'):
                st.write('Top %d features in order of importance are: %s'%(n_top_features,top_features[::-1]))
```

Figure 2: Code screenshot on handling imbalanced data



```
        elif(choose_model == "Random Forest"):
            #Feature selection through feature importance
            model = RandomForestClassifier(random_state=42)
            @st.cache
            def feature_sort(model,X_train,y_train):
                # fit the model
                model.fit(X_train, y_train)
                # get importance
                imp = model.feature_importances_
                return imp

            # Get feature importance and plot it
            st.set_option('deprecation.showPyplotGlobalUse', False)
            importance=feature_sort(model,X_train,y_train)
            feats = {} # a dict to hold feature_name: feature_importance
            for features, importances in zip(df.columns, importance):
                feats[features] = importances #add the name/value pair

            importances_df= pd.DataFrame.from_dict(feats, orient='index').rename(columns={0: 'Gini-importance'})
            importances_df.sort_values(by='Gini-importance').plot(kind='barh', rot=45)
            plt.title('Feature Importance')
            plt.xlabel('Importance')
            plt.ylabel('Features')
            st.pyplot()

            # get top features from the feature importance list
            feature_imp=list(zip(feat,importance))
            feature_sort=sorted(feature_imp, key = lambda x: x[1])
            n_top_features = st.sidebar.slider('Number of top features', min_value=5, max_value=20)
            top_features=list(list(zip(*feature_sort[-n_top_features:]))[0])

            if st.sidebar.checkbox('Show selected top features'):
                st.write('Top %d features in order of importance are: %s'%(n_top_features,top_features[::-1]))
```

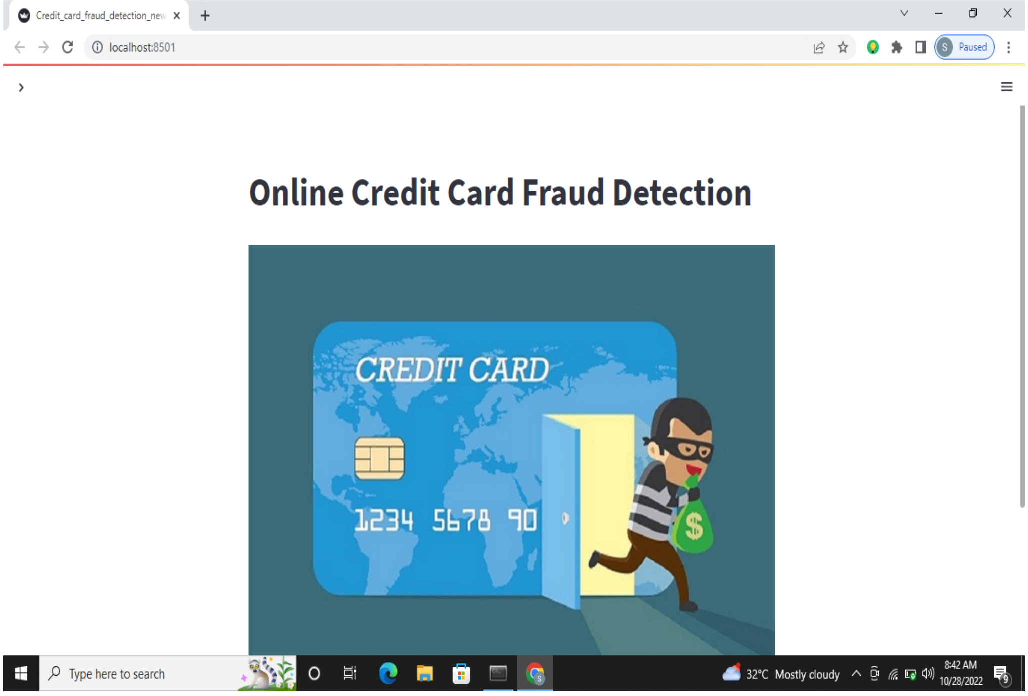
Figure 3: Code screenshot of data modelling



Top 10 features in order of importance are: ['Time', 'V11', 'V14', 'V10', 'V12', 'V4', 'V17', 'V16', 'V3', 'V9']

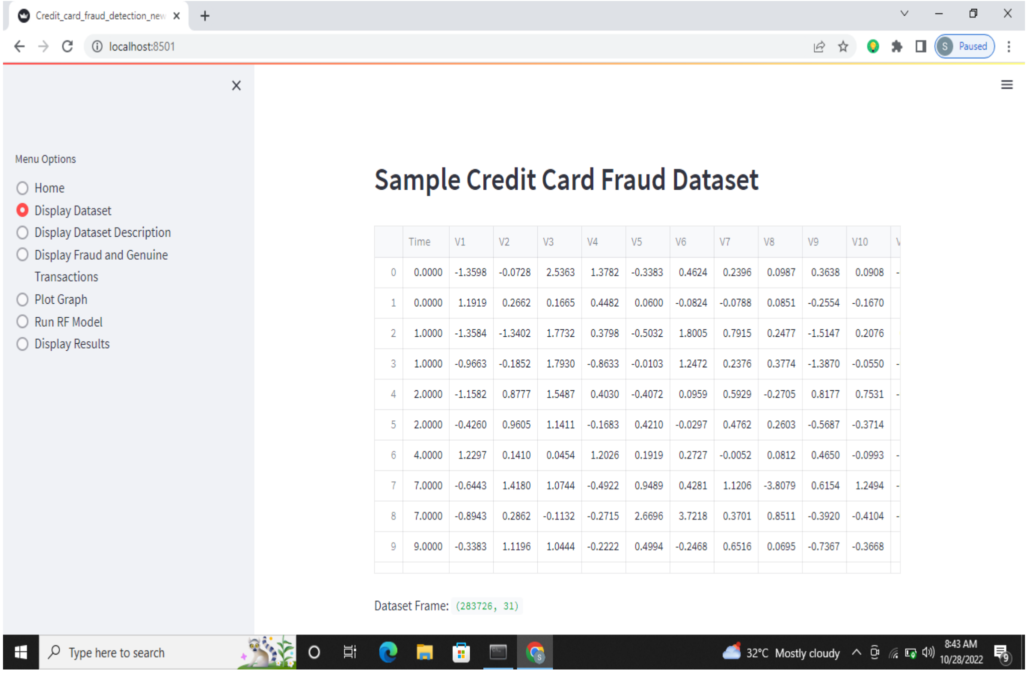
Figure 4: Feature Importance of each feature in dataset.

The system has been fully built and is ready to be used. The images below show the GUI before a dataset is uploaded and after a dataset has been uploaded. The image below in Figure 5 is the screenshot that shows the GUI welcome page of online credit card fraud detection after running it on a web application.



Online Credit Card Fraud Detection

Figure 5: GUI welcome page of online credit card fraud detection



Menu Options

- ○ Home
- ● Display Dataset
- ○ Display Dataset Description
- ○ Display Fraud and Genuine Transactions
- ○ Plot Graph
- ○ Run RF Model
- ○ Display Results

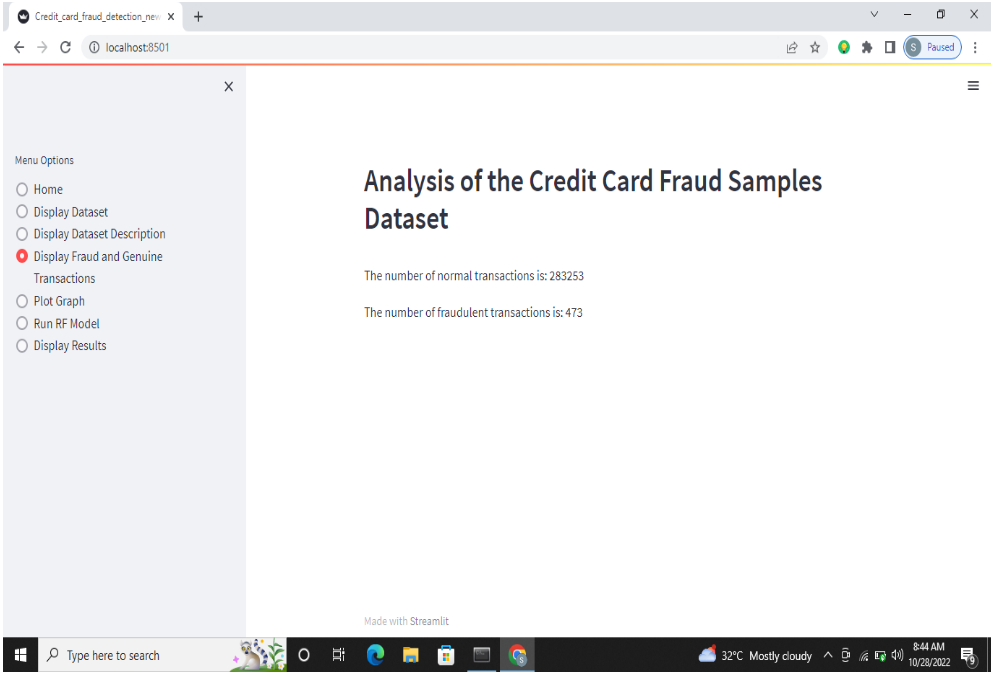
Sample Credit Card Fraud Dataset

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V
0	0.0000	-1.3598	-0.0728	2.5363	1.3782	-0.3383	0.4624	0.2396	0.0987	0.3638	0.0908	-
1	0.0000	1.1919	0.2662	0.1665	0.4482	0.0600	-0.0824	-0.0788	0.0851	-0.2554	-0.1670	
2	1.0000	-1.3584	-1.3402	1.7732	0.3798	-0.5032	1.8005	0.7915	0.2477	-1.5147	0.2076	
3	1.0000	-0.9663	-0.1852	1.7930	-0.8633	-0.0103	1.2472	0.2376	0.3774	-1.3870	-0.0550	-
4	2.0000	-1.1582	0.8777	1.5487	0.4030	-0.4072	0.0959	0.5929	-0.2705	0.8177	0.7531	-
5	2.0000	-0.4260	0.9605	1.1411	-0.1683	0.4210	-0.0297	0.4762	0.2603	-0.5687	-0.3714	
6	4.0000	1.2297	0.1410	0.0454	1.2026	0.1919	0.2727	-0.0052	0.0812	0.4650	-0.0993	-
7	7.0000	-0.6443	1.4180	1.0744	-0.4922	0.9489	0.4281	1.1206	-3.8079	0.6154	1.2494	-
8	7.0000	-0.8943	0.2862	-0.1132	-0.2715	2.6696	3.7218	0.3701	0.8511	-0.3920	-0.4104	-
9	9.0000	-0.3383	1.1196	1.0444	-0.2222	0.4994	-0.2468	0.6516	0.0695	-0.7367	-0.3668	

Dataset Frame: (283726, 31)

Figure 6: Sample credit card fraud dataset

Figure 6 above is the screenshot of the sample credit card fraud dataset with the time and volume and the dataset frame of 283,726, 31. This is a unique dataset on fraud detection, exploratory data analysis has been carried out to explore the datasets and analyze how it could be used for effective detection of illegal transactions.



Analysis of the Credit Card Fraud Samples Dataset

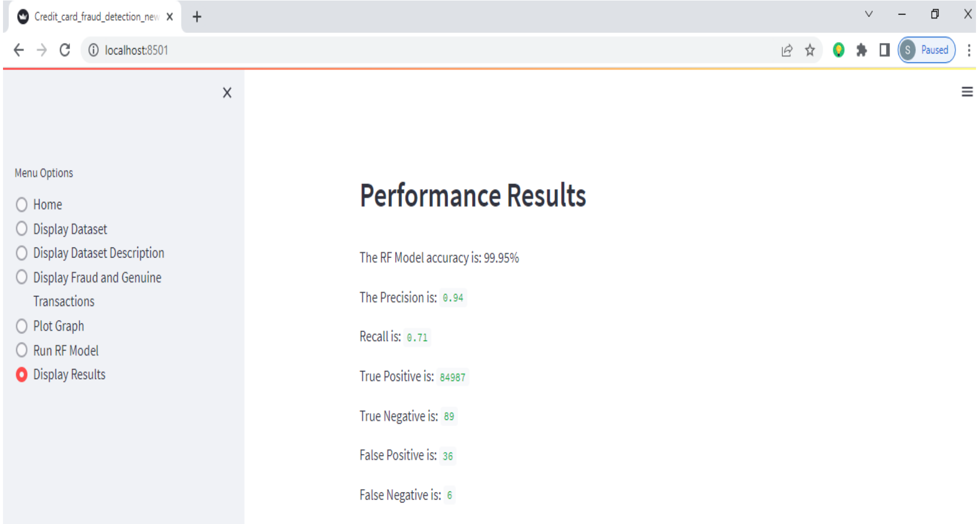
The number of normal transactions is: 283253

The number of fraudulent transactions is: 473

Figure 7: Analysis of credit card fraud samples dataset

The above image in figure 7 showcases the screenshot of the analysis of credit card fraud dataset sample. The number of normal transactions is 283,253 and the number of fraudulent transaction is 473. This indicates how powerful the developed SMOT is in terms of data classification since more than 83 percent of the transactions are normal compared to 473 fraudulent transactions that is equivalent to just 16 percent of the entire transaction. This paper would like to emphasis here that the pre-processing technique in SMOT for balancing the datasets in this research has removed about 80 percent of background noise that could have introduced errors into the experimental calculation.

Evaluation of the model has been carried out to determine the model performance to decide if it is good or bad and if it can be used effectively on other datasets and produce a good outcome. The accuracy is determined by comparing the predicted and actual data, it is the ratio of number of correct predictions to the total number of input samples. The accuracy works well when we have a balanced dataset where the number of predictions in each class is equal. Each model has its own accuracy. The accuracy of the classifier is shown in Figure 8 below.


Credit_card_fraud_detection_new
localhost:8501
Paused
Menu Options
○ Home
○ Display Dataset
○ Display Dataset Description
○ Display Fraud and Genuine Transactions
○ Plot Graph
○ Run RF Model
● Display Results

Performance Results

The RF Model accuracy is: 99.95%

The Precision is: 0.94

Recall is: 0.71

True Positive is: 84987

True Negative is: 89

False Positive is: 36

False Negative is: 6

Figure 8: Accuracy of the random forest model

A classification report is used to measure the quality of predictions from a classification algorithm. The classification report shows Precision, Recall and F1 score. This is the ability of a classifier not to label an instance positive that is actually negative. It is the ratio of the true positives to the sum of the true and false positives. The evaluation of precision-recall analysis is presented in Figure 9. Recall is the ability of a classifier to find all positive instances. It is defined as the ratio of true positives to the sum of true positives and false negatives. The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst value is 0.0, F1 scores are considered lower than accuracy measure because they embed both precision and recall into their computation. The weighted average of F1 is used to only compare classifier models, which in this case, is one. The classification report of the model is given in by Figure 10 below. In this result, it was observed that the classification accuracy produced an accuracy of 100% in some while the F1 score in some other experiments varies. For example, experimental results produced F1 scores of 1.00, o.94, 0.81 and so on for different classifier algorithms.

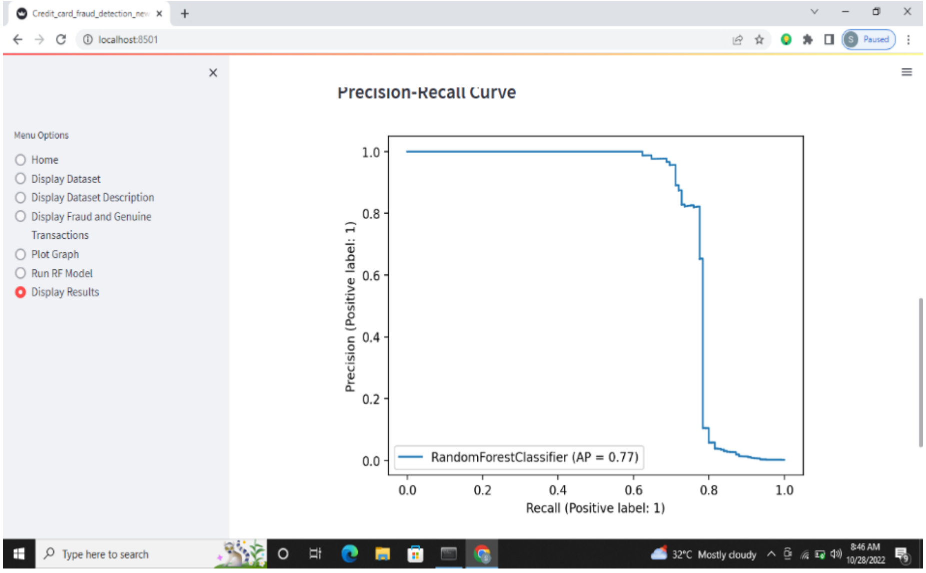
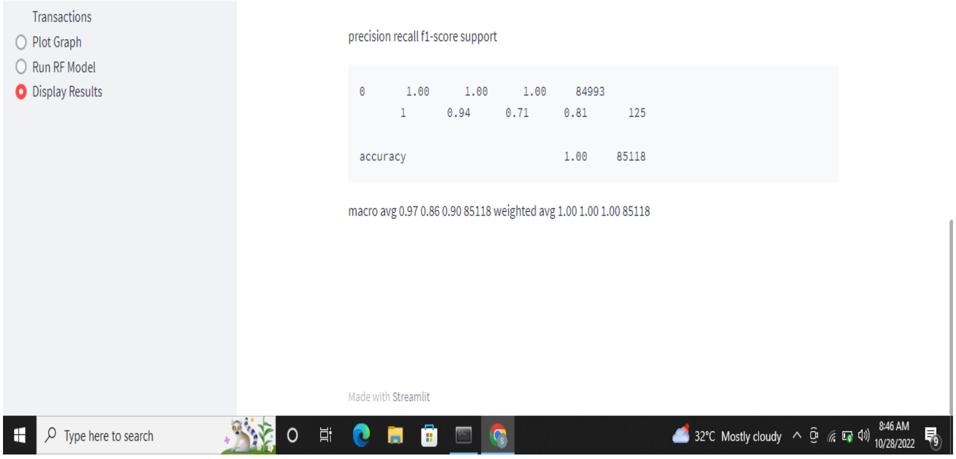


Figure 9: Precision for data analysis

By our calculation, the average classification accuracy of the experiment is 91.6%, this is a very good result compared with most of the results in the literature. In fact, SMOT becomes a state-of –the –art approach for balancing an unbalanced dataset that we have used in this research work. The recognition accuracy of 91.6% reflects the robustness, power and efficiency inherent in the developed system. Additionally, a confusion matrix that demonstrates a cross-validation performance of the random forest model has been presented to showcase how powerful our approach is in terms of data analysis. Future work could be carried to test and evaluate the performance of the developed model by applying it to solve any other unbalanced data.



precision recall f1-score support

```
0      1.00      1.00      1.00      84993
       1      0.94      0.71      0.81       125


accuracy                          1.00     85118
```

macro avg 0.97 0.86 0.90 85118 weighted avg 1.00 1.00 1.00 85118

Figure 10: Classification report

5 Conclusions

This research has developed the most common methods of fraud detection and reviewed recent findings in this field. This paper has also explained in detail, how machine learning can be applied to get better results in fraud detection along with the algorithm, code screenshots, explanation and its implementation. By applying SMOT to balance dataset, it was observed that the models performed better, Decision tree, random forest, neural network and K-nearest neighbour algorithms were used to fit and train the data. They also appear in the system to allow users to select a model of choice. The random forest gave an accuracy of 99.58, however, the efficiency decreases when trained with unbalanced transaction datasets.

References

Benson S., Edwin R.A., & Annie P. (2020). Analysis on credit card fraud detection methods. International Conference on Computer, Communication and Electrical Technology (ICCCET), IEEE, 42(2), 152-156.

Berk, R. J. (2019). What You Can and Can't Properly Do with Regression. *Journal of Quantitative Criminology, Springer*, 5(3). 756-767.

David, U. (2021). Bureau of consumer financial protection consumer credit card market report, International Conference on Computer, Communication and Electrical Technology (ICCCET), 12, 15-22.

Hand D. J. (2019), Fraud Detection in Telecommunications and Banking: Discussion of Becker, Techno metrics, 52(1), 34-38.

Maniraj, S. P., Aditya S., Shadab A. and Swarna S. (2019). Credit Card Fraud Detection using Machine Learning and Data Science, *International Journal of Engineering Research*, 8(2), 56-64.

Mareeswari,V. (2019). Prevention of Credit Card Fraud Detection based on HSVM. International Conference on Information Communication and Embedded System, 11. 33-47.

Mason, S. (2020). Looking at debit and credit card fraud. Teaching Statistics,34(3),87-9.

Melo-Acosta, G.E. (2020). Fraud Detection in Big Data Using Supervised and Semi-Supervised Learning Techniques, IEEE Colombian Conference on Communications and Computing, 2(1), 723-729.

Mohammed, E., & Behrouz F. (2019). Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study, IEEE Annals of the History of Computing, 12(3), 82-93.

Shabad, M., & Kavitha, M. (2018). Credit Card Fraud Detection Using Neural Networks at Merchant Side, *Journal of Computational and Theoretical Nanoscience*, 15(4),3373-3375.

Shiyang X., Guan Jun L., Zhenchuan Li., Lutao Z., Shuo W., & Changjun J. (2019). Random forest for credit card fraud detection. IEEE 15th International Conference on Networking, Sensing and Control, 15, 23-38.

Srivastava, A., Kundu, A., Sural, S. & Majumdar, A. (2018). Credit Card Fraud Detection Using Hidden Markov Model. IEEE Transactions on Dependable and Secure Computing, 5(1),37-48.

Suvasini P., Amlan K., Shamik S., & Majumdarb A.K. (2019). Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning. Information Fusion, 10 (4), 354-363.

Talabis, M. (2019). Information Security Analytics. Waltham: Syngress is an imprint of Elsevier, 5(1), 1-12.