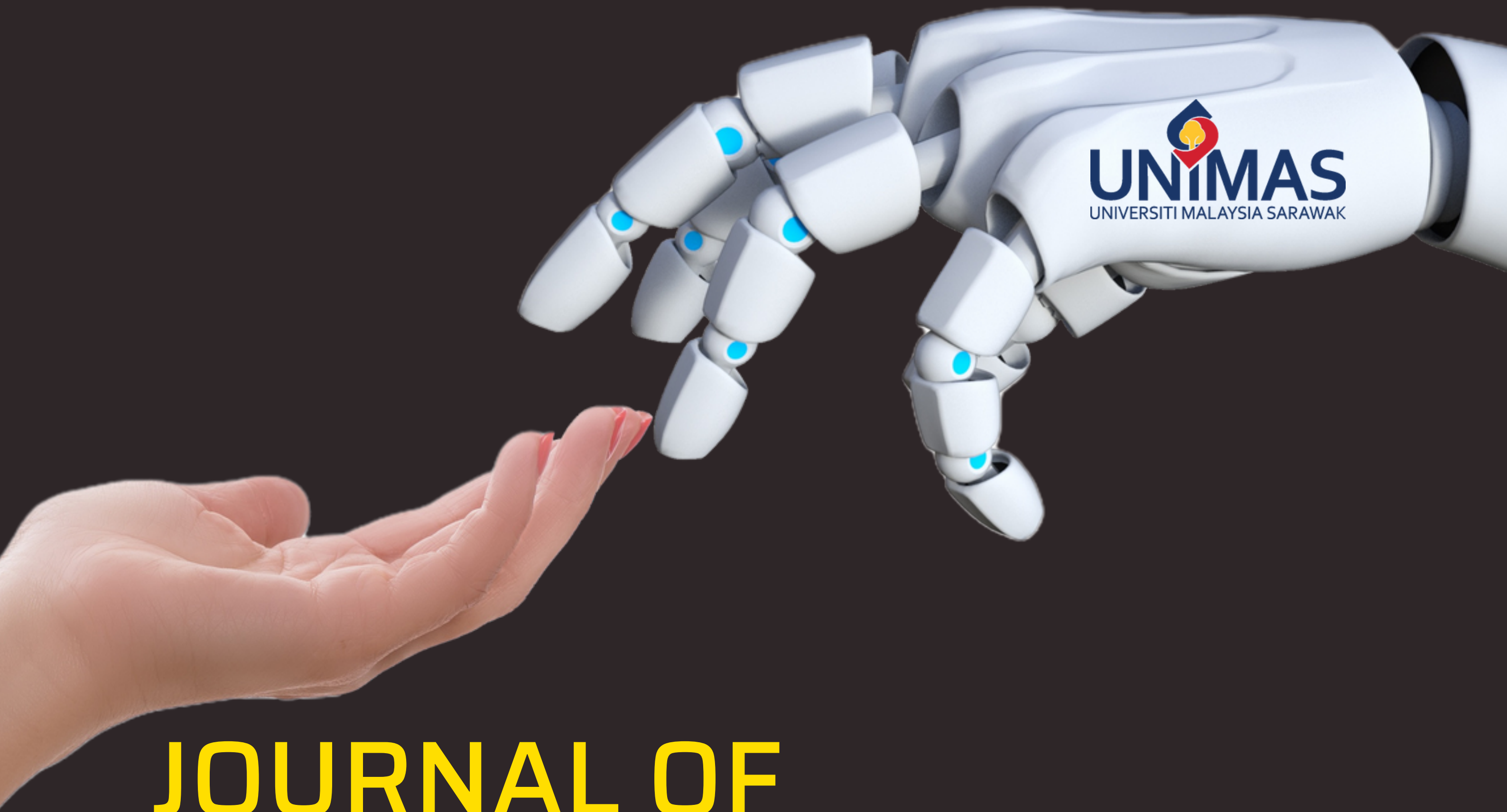


JOURNAL OF COMPUTING AND SOCIAL INFORMATICS

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITI MALAYSIA SARAWAK

eISSN 2821-3777
9772821377005

Journal of Computing and Social Informatics

The Journal of Computing and Social Informatics (JCSI) is an international peer-reviewed publication that focuses on the emerging areas of Computer Science and the overarching impact of technologies on all aspects of our life at societal level. This journal serves as a platform to promote the exchange of ideas with researchers around the world.

Articles can be submitted via *www.jcsi.unimas.my*

Assoc Prof Dr Chiew Kang Leng

Chief Editor
Journal of Computing and Social Informatics
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak
94300 Kota Samarahan
Sarawak, Malaysia

Contents

Development of SMS Spam Filtering App for Modern Mobile Devices

[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: 27 October 2023 | Accepted: 16 January 2024 | Early access: 26 January 2024

Abstract - *Short Messaging Service spam has been known to be the unwanted or unintended messages received on mobile phones. This paper has presented a review of current methods, existing problems, and future research directions on spam classification techniques of mobile SMS spams. The methodology involves collecting a large dataset of SMS messages, both legitimate and spam, to train and evaluate various machine learning algorithms. Feature extraction techniques have been employed to capture relevant information from SMS messages, such as the presence of specific keywords, the length of message, and the sender's identity. The experimental results on the proposed spam filtering system achieves a high level of accuracy with a low false-positive rate, thereby minimizing the chances of legitimate messages being classified as spam. The system effectively detects and blocks a significant portion of spam messages, providing mobile users with a reliable defense against unwanted SMS communications. The findings of this study reveal that machine learning algorithms, particularly ensemble methods like Random Forests, perform well in SMS spam filtering on mobile devices.*

Keywords: Spam Message, SMS Filtering, Modern Mobile Devices, Machine Learning, Random Forest Algorithm, Android Based App.

1 Introduction

This paper presents a detailed description of SMS Spamming Filtering Application (Android based app), which is to be used by most Android smartphone users to protect their Android devices from any harmful spams messages. The SMS Spamming Application will be a mobile based app exclusively for devices built with Android operating system. SMS is one of the popular communication services in which a message is sent electronically. The reduction in the cost of SMS services by telecom companies has led to the increased use of SMS. However, mobile users have become increasingly concerned regarding the security of their client confidentiality. This is mainly due to the fact that mobile marketing remains intrusive to the personal freedom of the subscribers, which has attracted and resulted into SMS Spam problem. A spam message is generally any unwanted message that is sent to a user's mobile phone. Spam messages include advertisements, free services, promotions, awards, etc. People are using SMS messages to communicate rather than emails because while sending SMS messages there is no need for an internet connection, and it is simple and efficient, which has led to a lot of spam messages (Gómez-Adorno, 2017)

Spam has been a large problem on the internet for as long as e-mail and personal computers have been ubiquitous. As a result, numerous methods have been proposed to reduce the ease at which spammers can retrieve messages on the internet. Previous efforts to fight spam on the internet have not totally eradicated it but rather increasing difficulty for those in the business of email spamming (Yadav et al., 2020). Various studies have been conducted by different researchers to resolve these problems, for instance, Li et al. (2020) presented a machine learning-based SMS spam filtering system that utilized features such as sender reputation, message length, and frequency of specific keywords to determine whether an SMS is a spam or not. The system achieved a high accuracy rate of

95% in detection and classification of spam messages. Similarly, Santos et al. (2021) proposed a hybrid approach combining rule-based and machine learning techniques to effectively filter SMS spam with a precision of 97% and a recall of 95%. Moreover, advancements in machine learning algorithms have demonstrated promising results in SMS spam filtering. Pham et al. (2018) explored the application of Support Vector Machines (SVM) and Naive Bayes classifiers for SMS spam detection on mobile devices. Their study achieved an accuracy of 94% with SVM and 91% with Naive Bayes. The problem at hand is the inadequate SMS spam filtering systems designed for mobile devices. The rising use of mobile devices and Short Message Service (SMS) have resulted in a surge of unsolicited and unwanted SMS spam messages. Findings in this research reveal that existing filtering techniques have not been able to effectively address these issues, allowing spam messages to infiltrate users' inboxes.
This leads to privacy invasion, wastage of network resources, and potential security risks for mobile users. The consequences include user frustration, decreased productivity, network congestion, and susceptibility to fraudulent activities. Therefore, there is a pressing need to develop robust and accurate SMS spam filtering solutions specifically tailored for mobile devices to alleviate these problems and provide users with a spam-free messaging experience.

On this note, this paper therefore aims at developing an Android smartphone users with a mobile-based security App using Python programming language. The proposed App would be developed in such a way that when the App is installed on the mobile phone, the entire system would have the capability to filter out unwanted messages through its various interfaces.

2 Literature review

In this section, related publications on SMS spam detection and classification papers would be reviewed in order to determine their strength and weaknesses. Zainal et al. (2022) developed a spam detection model using Rapid Miner and Weka tools; they applied two malware tools to perform their experiments on the UCI repository dataset. The research outputs demonstrated that both tools can produce almost a similar result on the same dataset with the same classification algorithms. El-Alfy (2019) has recently suggested a new method to identify spam messages on both email and SMS platforms. They tested many methods and features to achieve the best set of features with low level of model complexity. In their research, they applied Support Vector Machine (SVM) and Naïve Bayes techniques with eleven different features due to the nature of their datasets. It was finally discovered that the model complexity of the developed system was very high and hence could not be used for detecting big datasets. Zainal et al. (2022) introduced a model for spam messages filtering to remove background noise and unwanted materials from bulk messages. The developed model was evaluated in terms of performance in spam messages detection using text classification algorithms on mobile phones. Filtering, training, and updating features could be performed on any Android mobile phones. It was discovered that the research outputs of their experiments revealed that the developed model could be used to filter out spam messages even with small memory usage and good classification accuracy. In another research, Chan et al. (2019) proposed two approaches for classifying and eliminating SMS spam messages, their approach was focused on the weight and length of the message; series of experiments were performed on the selected database and they achieved a remarkable result in terms spam detection and classifications. Uysal et al. (2019) developed a new approach for filtering SMS spam messages. In their approach, a hybrid method comprises of chi-square and information gain algorithm for spam messages was applied. Moreover, the authors also presented an android-based SMS spam filtering method using Bayesian approach. Based on their outputs, their method is efficient and can classify both ham and spam messages even with high degree of classification accuracy. In Serrano et al. (2019), a technique for detecting spam messages using extrinsic information was investigated. All experimental tests were performed in Weka environment using 10-fold cross validation approach. The authors achieved good classification and detection accuracy with a low memory usage. Junaid (2019) proposed a system to detect and classify SMS spam messages on a mobile phone by applying different classifiers. In their results, it was concluded that supervised learning algorithms could be used to build original model. At the end, the developed model achieved a classification accuracy of over 80%. Choudhary (2017) investigated a system for detection and classification of spam messages. The authors extracted ten unique features and applied them for detecting unwanted messages. The techniques adopted in their approach were True Positive (TP) rate, False Positive (FP) rate, precision, and F-measure. In their research, the authors compared various classification algorithms, and among them, the Random Forest algorithm achieved the best results with a classification accuracy of 96.1% TP rate.

In a similar research, Suleiman (2017) proposed a technique for removing SMS spam messages using hybrid technique. They applied the hybrid method for feature selection, and extracted some spam messages features. Selected features were then compared on various algorithms in order to determine the best.

This section has so far reviewed the advantages of recent developed approaches in detecting and filtering SMS spam messages while also noting their weaknesses and limitations. According to the literature, it has been discovered that most of the SMS spam detection techniques are not accurate enough in terms of detection of unwanted messages and classification. Therefore, this current study would propose a machine learning technique to identify SMS spam messages with high performance and acceptable classification accuracy.

3 Methodology

The public dataset of SMS labelled messages were obtained from UCI Machine Learning Repository. This study finds that there are only 5,574 labelled messages in the dataset, with 4827 of the messages belong to real messages while the other 747 messages belong to spam messages as shown in Table 1. Nonetheless, this dataset consists of two named columns starting with the message labels (ham or spam) followed by strings of text messages and three unnamed columns.

Table 1: Type of features of dataset

Data Set	Legitimate	Spam	Total
SMS spam	4827	747	5574
DIT SMS spam	0	1353	142
British English SMS	450	425	875
Total	5,277 (67.6%)	2,525(32.4%)	7802

As shown in Table 1, the dataset has 67.6% of Ham message and 32.4% of Spam message. It has been discovered that this dataset contains some unwanted features and therefore requires preprocessing. The purpose of preprocessing is to convert a raw data into a form that can fit into a machine learning. The process of data preprocessing involves background noise removal, sampling and formatting. This paper uses a combination of content-based and user-based features for developing a robust system for efficient detection and classification of spam messages. Content-based features include the words, phrases, and patterns that are commonly found in spam messages, while user-based features include the sender's phone number, frequency of messages, and time of day the message was sent. The system design of SMS spam filtering typically involves several components, including data preprocessing, feature extraction, detection and classification algorithms. Feature extraction includes transformation of SMS messages into a set of features that can be used by the machine learning algorithm. This paper employed Naïve Bayes classifier for data classification to classify the dataset as spam or ham. In Figure 1, the flow chart diagram shows the steps of an SMS spam filtering, which start from the components and messages in raw data followed by preprocessing stage through various stages of algorithmic steps to detect the spam messages on mobile devices as either spam or non-spam. The implementation of an SMS spam filtering system typically involves using a programming language or tool to develop the components described in the system design. There are many programming languages and tools available for implementing SMS spam filtering systems, including Python, Java, and MATLAB. In this paper, several Python libraries, including scikit-learn and NLTK have been used to implement the data preprocessing, feature extraction, and detection algorithm. This paper employed Naive Bayes on a classification task involving spam SMS messages and the model was able to classify over 97 percent of all the SMS messages correctly as spam or non-spam. Figure 4 is the screenshot of the SMS spam filtering search icon, which grant permission for searching any messages on the app and checking the spam filtering commands within the system. The search icon help can help to detect and classify any message to determine if it belongs to spam or not

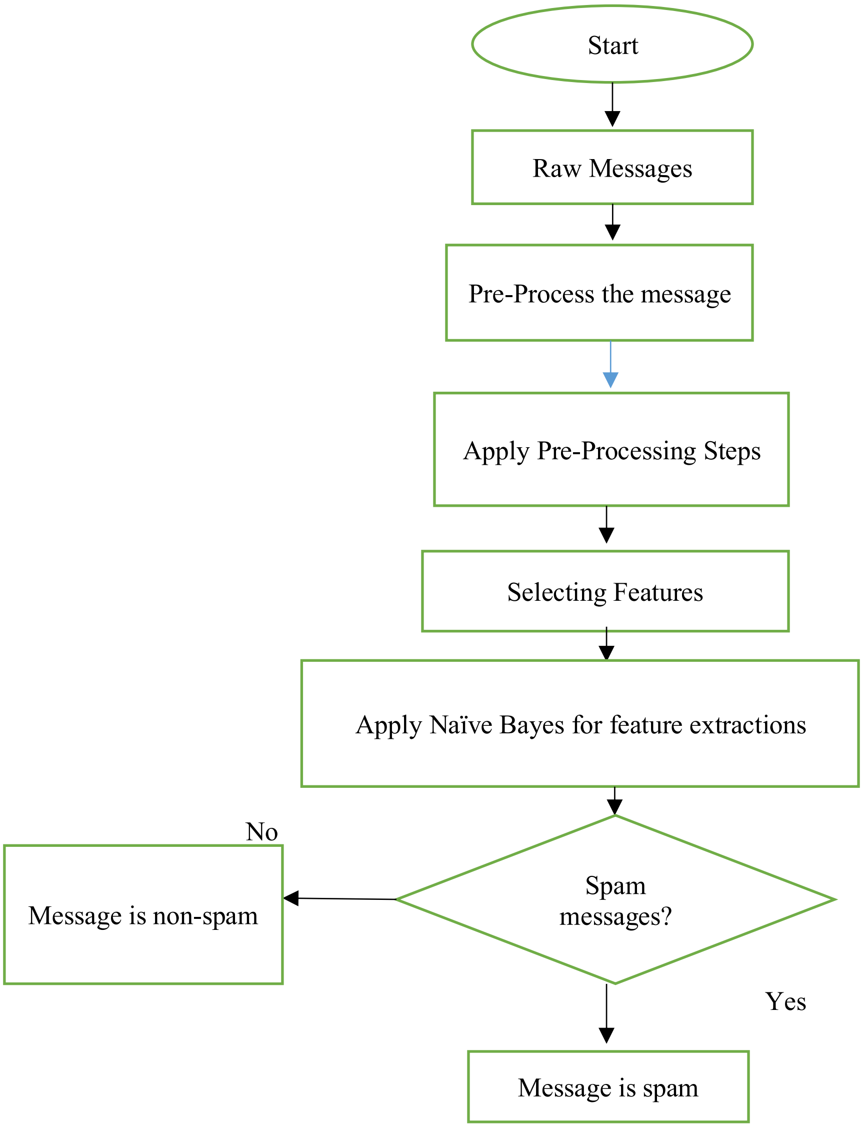


Figure 1: Flowchart diagram of the system

4 Results and Discussion

The proposed system was successfully tested to detect spam messages on mobile phones. It basically detect spam messages by the developed app that includes normal messages, spam messages and filtered spam messages. Based on the above, the application is user friendly and meets all the requirements usability and security of personal data. This application contains an additional features, which includes some security measures to protect and guide our data against cybercriminals on mobile devices. This additional feature incorporated in the system is our major contribution to knowledge in this paper since in literature, most researchers did not security capability in their systems and they are mostly on desktop not on mobile device like our system.
Figure 2 displays the SMS spam filtering app using a splash-screen that boots to the main app. This icon was displayed for over 10 seconds before launching to the main app on a mobile device or on enumerator. In Figure 3, it shows the front page for the filtering technique where all the images from the phone are stored. This was achieved by the work permission handler, which allows the developed mobile app to accept and store information. In the second tab, there is a feature for detection and classification of spam messages on mobile phone. As presented in Figure 4, the developed app shows the message details of the SMS spam filtering app. It detects all messages that enter into your mobile phones with the help of the permission handler that was previously installed in the flutter. It runs on android version 10.2.0 to grant access to the mobile phones.

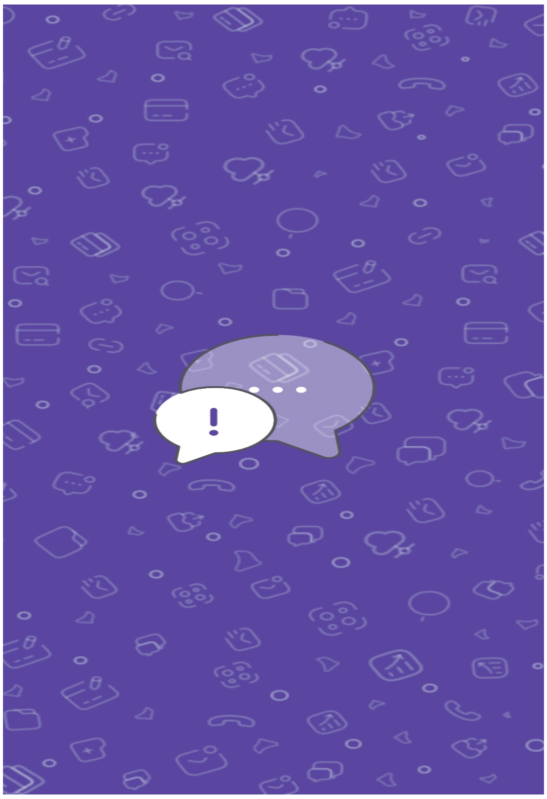
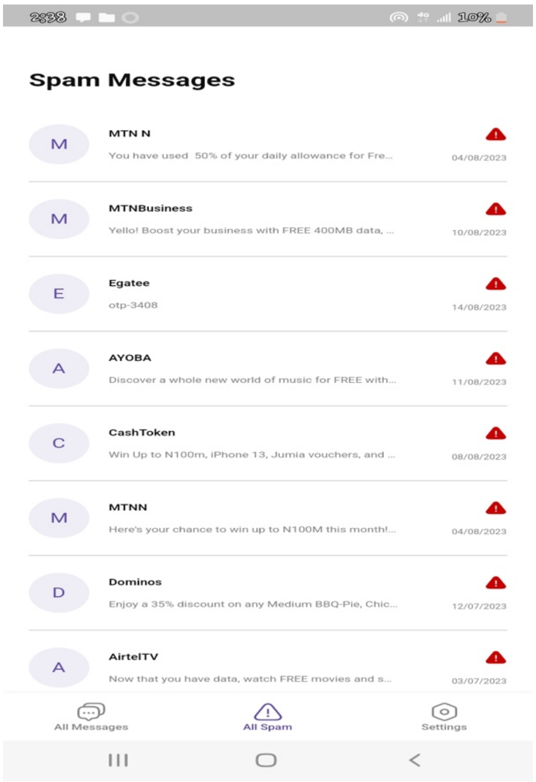


Figure 2: Front gage for spam filtering system

Overall, the developed system was able to detect and classify messages received by mobile devices as either spams or non-spams using different experimental results as presented in this paper.



Spam Messages

M MTN N
You have used 50% of your daily allowance for Fre... 04/08/2023

M MTNBusiness
Yello! Boost your business with FREE 400MB data, ... 10/08/2023

E Egatee
otp-3408 14/08/2023

A AYOBA
Discover a whole new world of music for FREE with... 11/08/2023

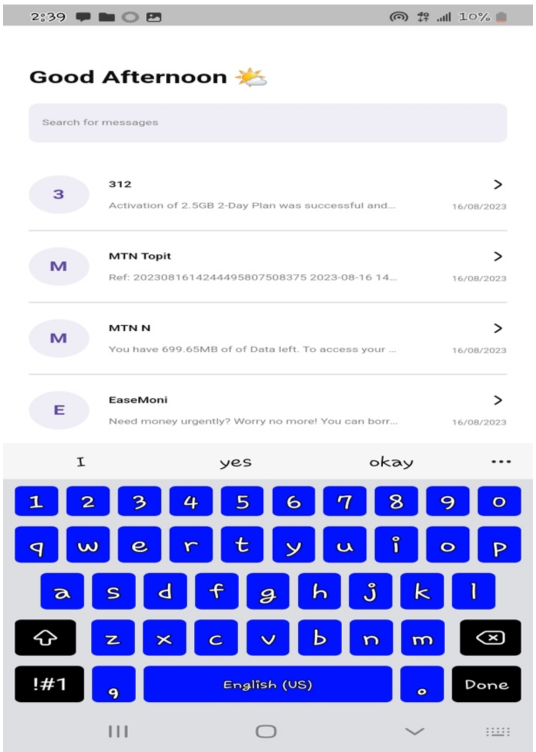
C CashToken
Win Up to N100m, iPhone 13, Jumia vouchers, and ... 08/08/2023

M MTNN
Here's your chance to win up to N100M this month!... 04/08/2023

D Dominos
Enjoy a 35% discount on any Medium BBQ-Pie, Chic... 12/07/2023

A AirtelTV
Now that you have data, watch FREE movies and s... 03/07/2023

Figure 3: Front page of SMS spam filtering system



Good Afternoon

Search for messages

3 312
Activation of 2.5GB 2-Day Plan was successful and... 16/08/2023

M MTN Topit
Ref: 20230816142444958075O8375 2023-08-16 14... 16/08/2023

M MTN N
You have 699.65MB of of Data left. To access your ... 16/08/2023

E EaseMoni
Need money urgently? Worry no more! You can borr... 16/08/2023

Figure 4: Search icon on SMS spam app

5 Conclusions

This paper studied some related research papers in the field of spam messages detection and classification with a view to developing a new approach for alleviating existing problems in this research area. About seventeen research papers have been selected and reviewed in order to understand the existing techniques in this field of study. The knowledge acquired in the literature review in this paper has been put together to propose a new method for addressing common challenges facing SMS spam filtering system. The proposed system contains some additional features that could be used to eliminate problems or limitations in spam detection and classification. This paper has contributed to knowledge in the area of security by denying unauthorized access to SMS spam filtering model and the developed app is currently running on mobile devices. This is a robust system that is potable, secured and efficient in terms of separating unwanted messages from useful ones. Generally, the developed system has been compared and evaluated with the existing techniques, the proposed system achieved higher classification and detection accuracy when compared with the state-of-art method in this research field. Future research direction in this field could be achieved by applying the proposed system for preventing hackers or unknown users from gaining access to detection systems.

References

Al-Hasan A.A., & El-Alfy E.-S.M. (2019). Dendritic cell algorithm for mobile phone spam filtering, Procedia Computer Science 52244-251.

Chan, P.P., Yang, C., Yeung, D.S., & Ng, W.W. (2019). Spam filtering for short messages in adversarial environment, Neurocomputing, Vol. 155, 167-176.

Choudhary, N., & Jain, A.K. (2019). Towards filtering of SMS spam messages using machine learning based technique", in International Conference on Advanced Informatics for Computing Research, Springer., 18-30.

El-Alfy, E.-S.M., & Al-Hasan, A.A. (2019). Spam filtering framework for multimodal mobile communication based on dendritic cell algorithm, Future Generation Computer Systems, Vol. 64, 98-107.

Gómez-Adorno, H., Pinto, D., Sidorov, G., & Villaseñor-Pineda, L. (2017). A linguistic approach and ensemble methods for SMS spam detection. Expert Systems with Applications, 68, 96-109.

Junaid, M.B., & Farooq, M. (2019). Using evolutionary learning classifiers to do mobilespam (SMS) filtering, in Proceedings of the 13th annual conference on Genetic and evolutionary computation, 1795-1802.

Li, B., Zhang, B., & Lee, W. C. (2020). SMS spam filtering based on keywords and spammers: A multi-view classification approach. Expert Systems with Applications, 39(10), 9229-9236.

Pham, D. T., Dang, T. D., & Nguyen, L. H. (2018). SMS spam filtering on mobile devices using machine learning techniques. In Proceedings of the International Conference on Advanced Computational Intelligence (ICACI) (pp. 435-440).

Santos, M. F., Cardoso, J. S., & Oliveira, H. P. (2021). Combining rule-based and machine learning classifiers for SMS spam filtering. Expert Systems with Applications, 41(4), 1933-1943.

Serrano, J.M.B., Palancar, J.H., & Cumplido, R. (2019). The evaluation of ordered features for SMS spam filtering, in Iberoamerican Congress on Pattern Recognition, Springer., 383-390.

Suleiman, D., & Al-Naymat, G. (2017). SMS spam detection using h2o framework, Procedia Computer Science, Vol. 113, 154-161.

Uysal, A.K., Gunal, S., Ergin, S., & Gunal, E.S. (2019). A novel framework for SMS spam filtering, in 2012 International Symposium on Innovations in Intelligent Systems and Applications, IEEE., 1-4.

Yadav, K., Saha, S., Kumaraguru, P., & Kumra, R. (2020). Take control of your SMSes: Designing an usable spam SMS filtering system, IEEE 13th International Conference on Mobile Data Management, MDM.

Zainal, K., Sulaiman, N. & Jali, M. (2022). An analysis of various algorithms for text spam classification and clustering using rapidminer and weka", International Journal of Computer Science and Information Security, Vol. 13, No. 3, (2022), 66-77.

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.

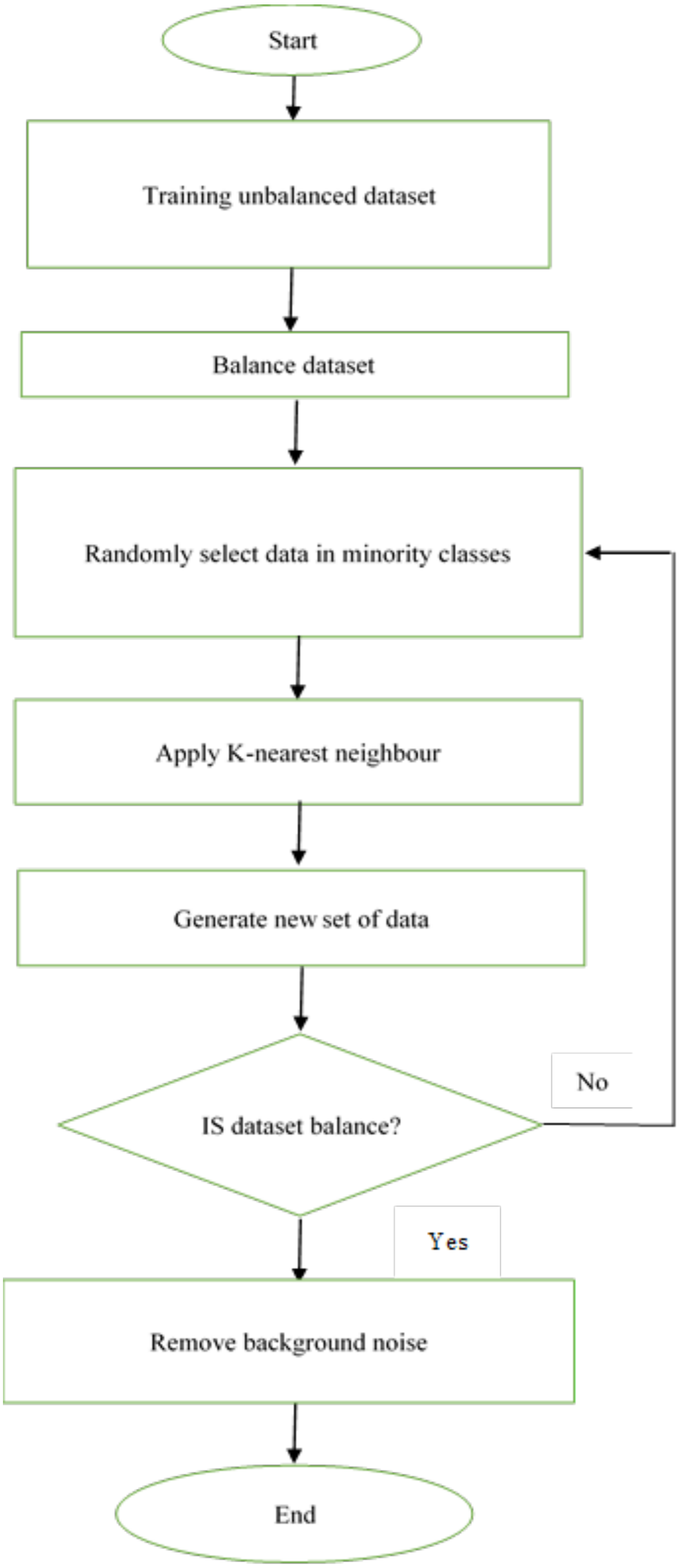
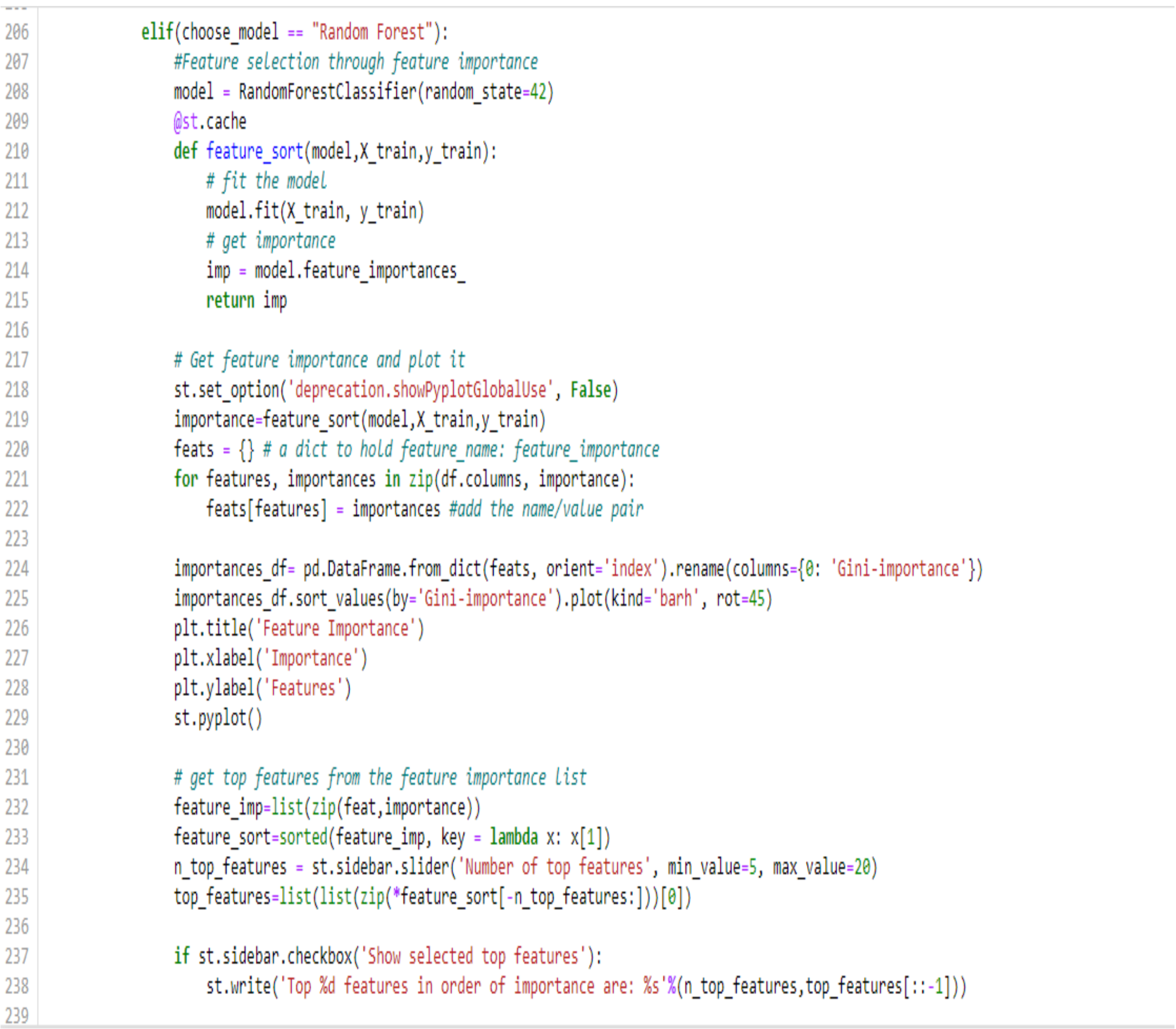
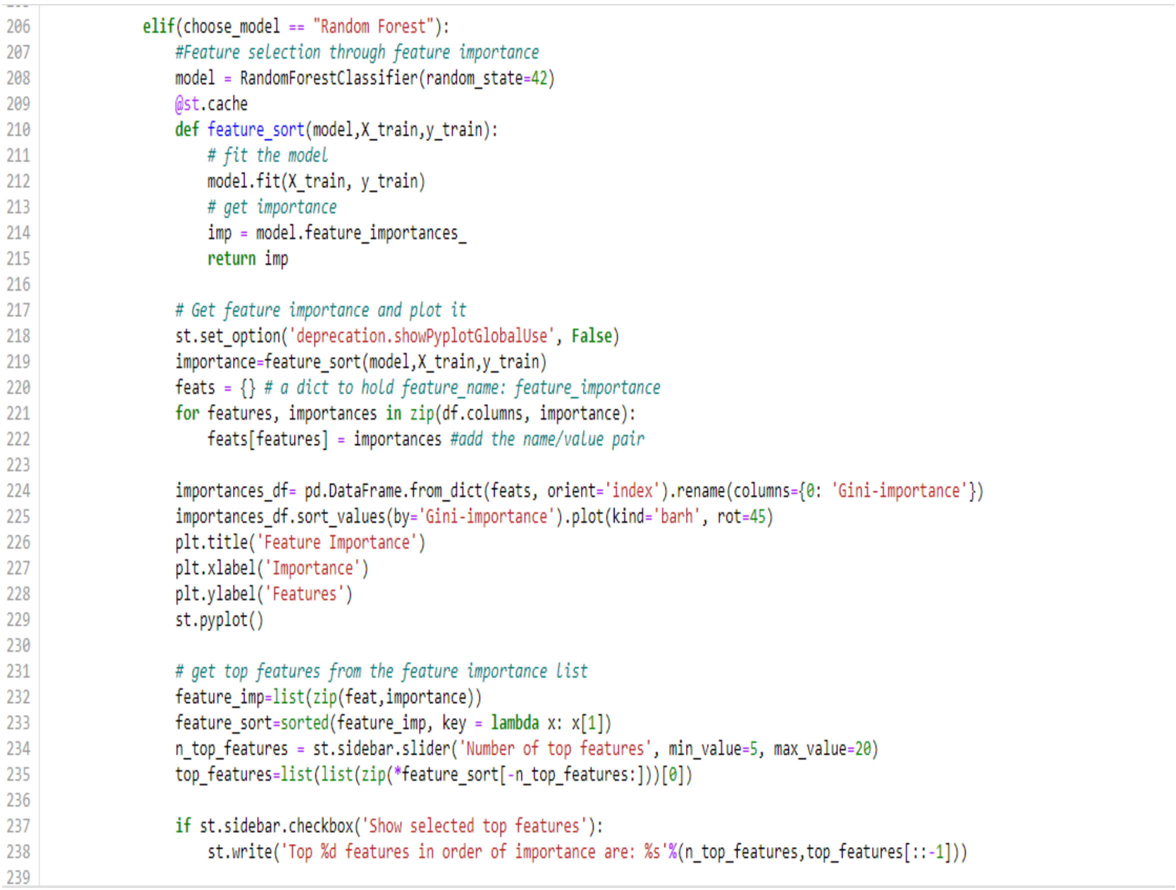


Figure 1: SMOT flowchart for prediction model



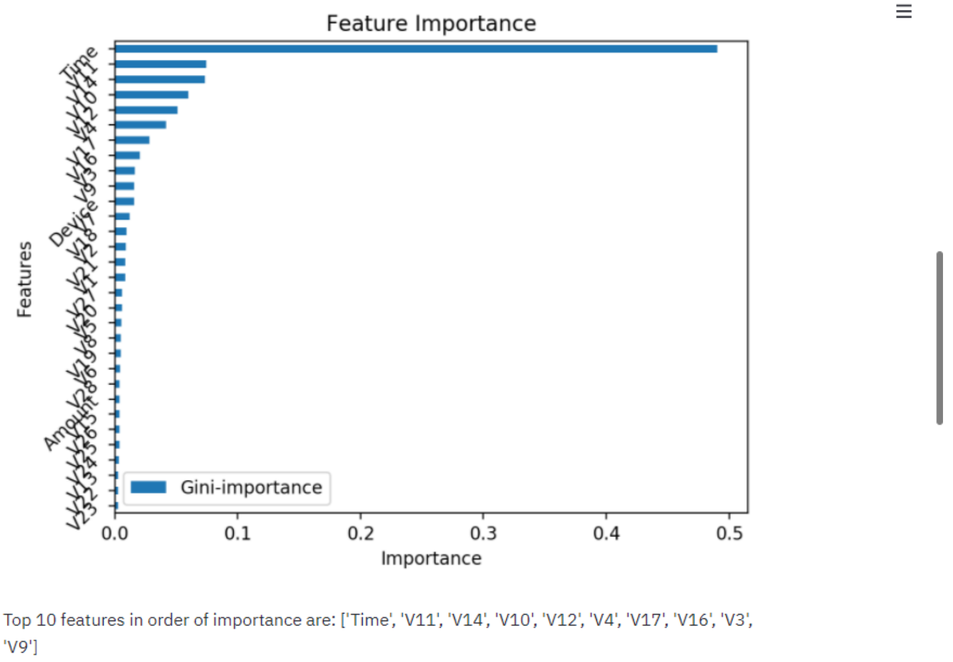
```
elif(choose_model == "Random Forest"):
    #Feature selection through feature importance
    model = RandomForestClassifier(random_state=42)
    @st.cache
    def feature_sort(model,X_train,y_train):
        # fit the model
        model.fit(X_train, y_train)
        # get importance
        imp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

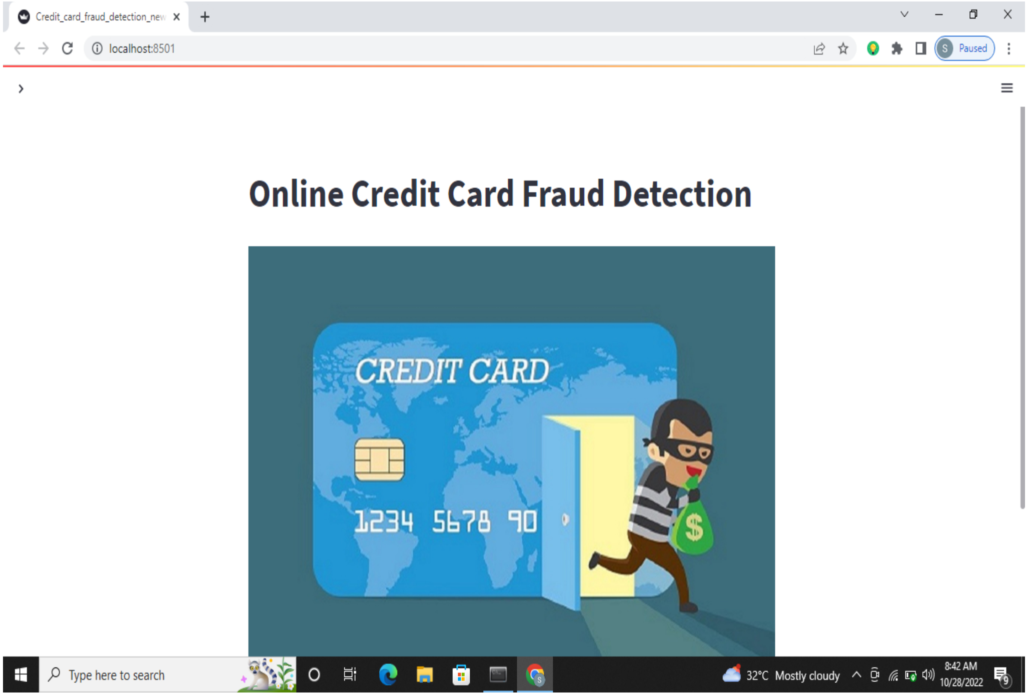
Figure 3: Code screenshot of data modelling



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

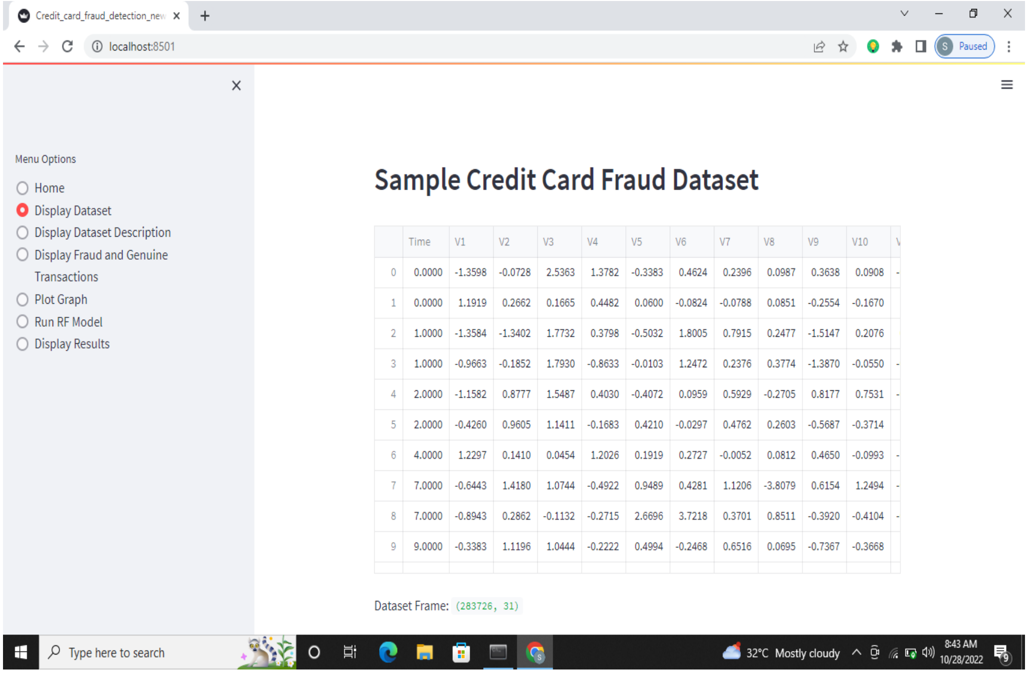
Figure 4: Feature Importance of each feature in dataset.

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

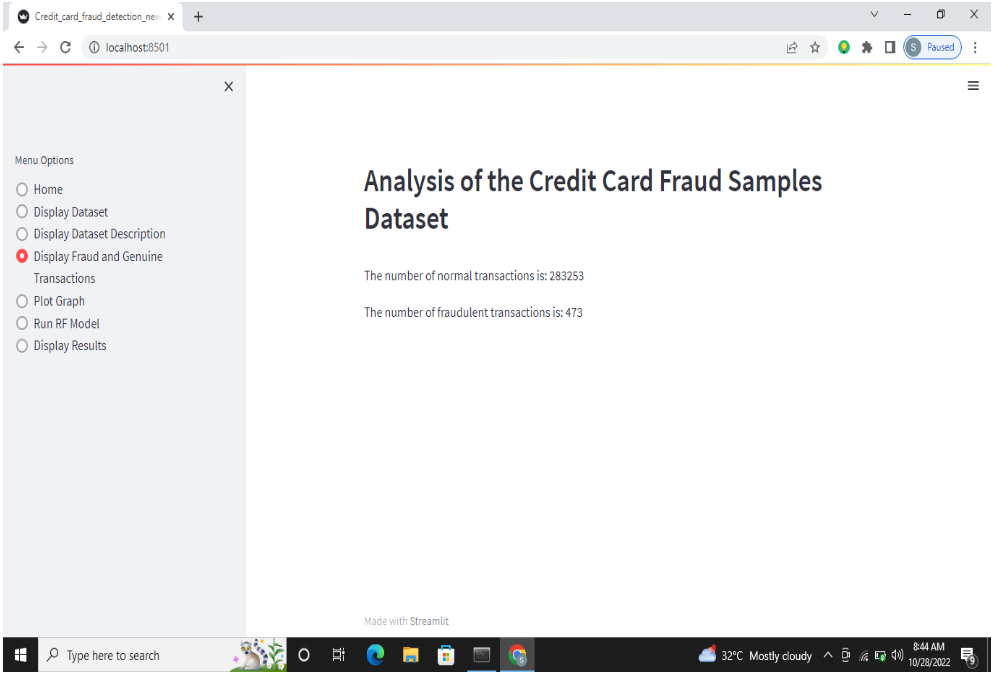
Sample Credit Card Fraud Dataset

	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

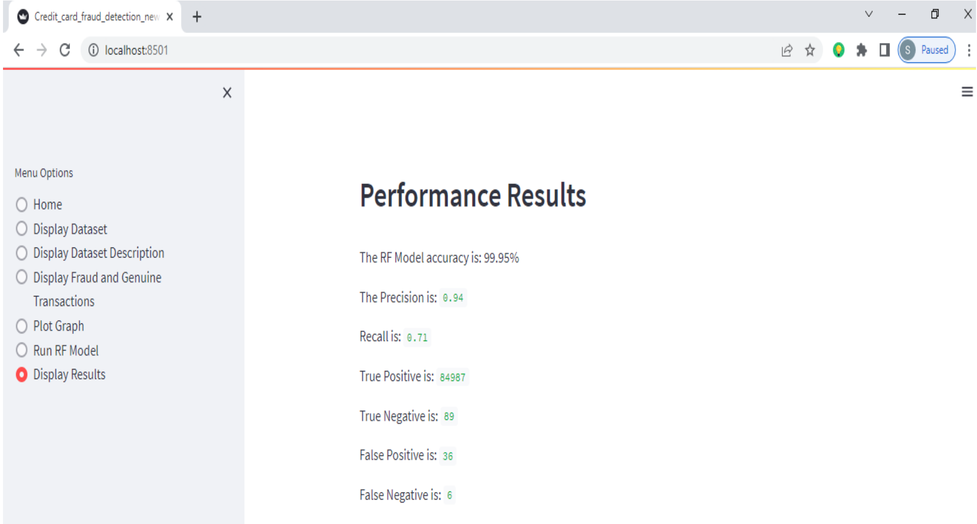
The number of normal transactions is: 283253

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.

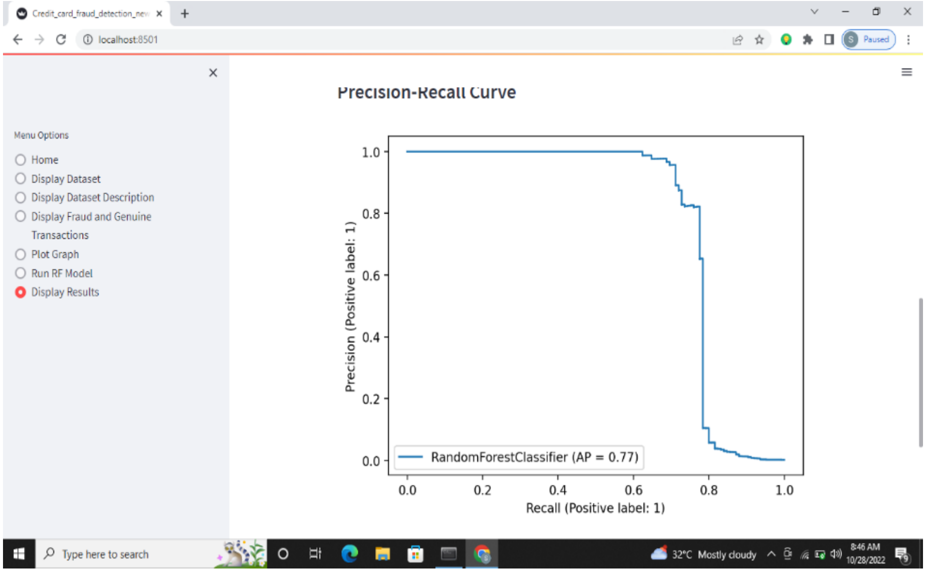
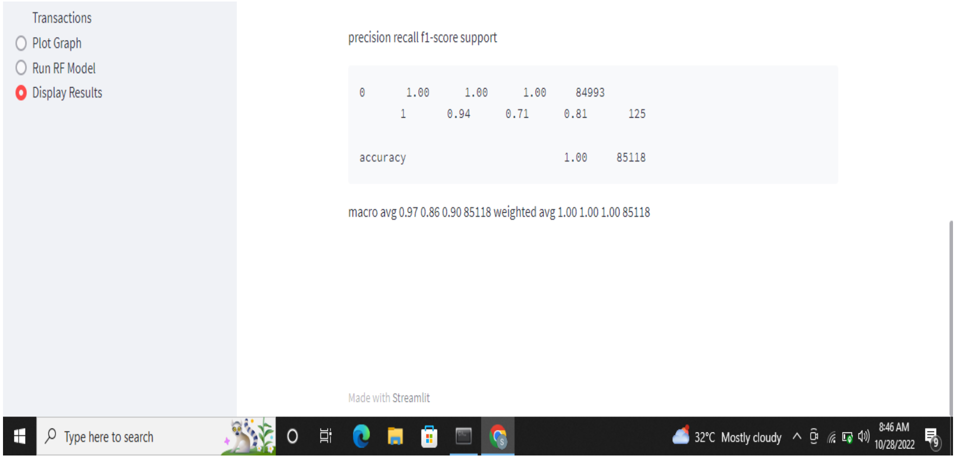


Figure 9: Precision for data analysis

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.