UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Fraudulent Detection Model Using Machine Learning Techniques for Unstructured Supplementary Service Data

Ayorinde O. Akinje & Fuad Abdulgalee
School of Computing
Universiti Teknologi Malaysia,
81310 UTM Johor Bahru, Johor, Malaysia
Email: ayorinde@graduate.utm.my

Submitted: 28/2/2021. Revised edition: 26/7/2021. Accepted: 28/7/2021. Published online: 15/11/2021
DOI: https://doi.org/10.11113/ijic.v11n2.299

***Abstract*—The increase in mobile phones accessibility and technological advancement in almost every corner of the world has shaped how banks offer financial service. Such services were extended to low-end customers without a smartphone providing Alternative Banking Channels (ABCs) service, rendering regular financial service same as those on smartphones. One of the services of this ABC's is Unstructured Supplementary Service Data (USSD), two-way communication between mobile phones and applications, which is used to render financial services all from the bank accounts linked for this USSD service. Fraudsters have taken advantage of innocent customers on this channel to carry out fraudulent activities with high impart of fraudulent there is still not an implemented fraud detection model to detect this fraud activities. This paper is an investigation into fraud detection model using machine learning techniques for Unstructured Supplementary Service Data based on short-term memory. Statistical features were derived by aggregating customers activities using a short window size to improve the model performance on selected machine learning classifiers, employing the best set of features to improve the model performance. Based on the results obtained, the proposed Fraudulent detection model demonstrated that with the appropriate machine learning techniques for USSD, best performance was achieved with Random forest having the best result of 100% across all its performance measure, KNeighbors was second in performance measure having an average of 99% across all its performance measure while Gradient boosting was third in its performance measure, its achieved accuracy is 91.94%, precession is 86%, recall is 100% and f1 score is 92.54%. Result obtained shows two of the selected machine learning random forest and decision tree are best fit for the fraud detection in this model. With the right features derived and an appropriate machine learning algorithm, the proposed model offers the best fraud detection accuracy.**

***Keywords*—Fraud, fraud detection, Unstructured supplementary service data (USSD), Machine learning**

I. INTRODUCTION

Making payment these days has never been so easy with a click, dialing codes from our mobile phones to perform financial transaction services that would have to require going into a bank to carry out or physical cash payment, Banks have taken the advantage of mobile phone banking to reach low-end customers without smartphones [1] producing Alternative Banking Channels (ABCs) service, offering various financial services, from cash withdrawal, transfers, deposits, to paying for electricity bills, phone bills, phone top-ups, traveling expenses, and television cable subscriptions which can be utilized from anywhere and anytime [2], one of the services of this ABC's is Unstructured supplementary service data (USSD) a Global Mobile Communicative System (GSM) capability that enables high speed, two-way communication between mobile phones and applications [3], which can be used for payment of utility bills, phone top-ups, and other financial services.

We've all enjoy these electronic channel services like paying for services, banks transactions and other e-services all from our mobile phones with just dialing codes ending with the hash sign (#) to make transactions, it has made life easier for us all, but it comes along with its challenges and shortcomings, the same questions come on our mind each time the services is utilized "How secure it is", fraud has been the major challenges these e-service systems had to battle with since its inception, fraudsters take advantage of innocent users on these platforms to maximize the vulnerabilities in the system for their financial

gain. Current measures to control its security flaws and threats are preventives measures such as short authentications PINs, encryption channels, this preventive measure has several disadvantages and drawbacks at the client end, the communication channel, and vulnerabilities in its security policy [4]. This preventive measure in place have failed with poor detection, high false positive, performances, still leaving the detection issue unsolved. Machine learning techniques have been employed in several sectors to mitigate fraud issues and this has shown potential improvement in fraud detection across the sectors.

This paper investigates selected classifiers performance, deriving the best feature of the dataset used and later employing different data sampling methods to solve data imbalance which affect model's poor performance, the classifiers are Support vector machine (SVM), Knieghbor, Decision Tree, Random Forest, Gradient Boosting and logistic regression. The rest of the paper are as follows section 2 presents related work reviews, Section 3 cover the USSD framework, 4 presents brief and basic information about our choice of selected Machine learning algorithms, section 5 presents the dataset, methodology and implementation of the methods, Section 6 and 7 covers performance evaluation and result discussion and lastly Conclusion.

II. Related Work

There are very few related works when fraud detection is USSD is being discussed, related work done to address fraud detection in USSD is still abstract and without implementation, the author [5] proposed a fraud detection model that will use Bayesian's algorithm for data calculations and analysis but its prediction based on prior history is a major drawback. Reasons behind few related works towards USSD this might be a result of the service is not being used globally as a financial alternative service since USSD financial transactions are limited to underdeveloped countries due to this, studies on USSD fraud detection are very limited or can be said it is still minute due to its low patronization for financial services such as payment and banking services in developed countries. Other close related of fraud detection work can be used as a reference point when discussing fraud detection, credit card fraud detects, mobile banking fraud, and other forms of fraud detection works using related machine learning algorithms.

Recent studies in credit card fraud detection have shown several techniques and approaches used in credit card fraud detection to limit fraud cases. [6], Worked on machine learning algorithms to recognize credit card fraud and proposes a framework, a sum of 12 machine learning algorithms were used for identifying credit card fraud using calculations to extend from standard neural systems to profound learning models. [7] Used Generative adversarial networks (GAN) for improving classification effectiveness in credit card fraud detection, GAN was trained to output mimicked minority class examples, which were then merged with training data into an augmented training set so that the effectiveness of a classifier can be improved. [8] worked on a combination of an automatic classifier with manual revision explored to improve data mining for FDS in credit cards. [9] propose a CNN-based fraud detection framework, to capture the intricate patterns of fraud behaviors learned from labeled data. Abundant transaction data was represented by a feature matrix, on which a convolutional neural network is applied to identify a set of latent patterns for each sample to address imbalanced data. [10] worked on improving class imbalanced in credit card FDS proposing a Scalable Real-time Fraud Finder (SCARFF) which integrates Big Data tools (Kafka, Spark, and Cassandra) with a machine learning approach which deals with imbalance, non-stationarity, and feedback latency. [11] did work on an offline transaction in credit card FDS to improve detection using Long Short-Term Memory (LSTM) networks to incorporate transaction sequences and also integrate state-of-the-art feature aggregation strategies and they report their results employing traditional retrieval metrics.

[12] used a support vector machine (SVM) along with fuzzy clustering for detecting fraudulent usage of mobile phones to address an anomaly when a call pattern does not match with any of the normal patterns. [13] proposes to develop an efficient SVM-based fraud classifier to solve the issue of highly imbalanced SB datasets, to produce SB training data, they combine the hierarchical clustering a labeling strategy and then utilize a hybrid data sampling method to classify their data. [14] who compares the performance of four different machine learning classifiers, SVM, Random Forest, logistic regression and Decision tree on skewed data, their performance was evaluated with their, precision, accuracy, specificity, sensitivity. [15] presented a Xgboost-based fraud detection model with features engineering and visualization, [16] they analysed three classifiers, logistic regression, decision tree, and random forest and proposed a model, its system design consists two-component first, to deals with data pre-processing framework responsible for processing data efficiency, while its second is an analytical model for fraud prediction.

III. USSD BANKING

Due to the benefit of USSD such as easily implementable, user-friendly menu-driven [1], suited to low-end non-smartphone devices without internet connection [17], USSD interbank instant payments platform received a significant boost in 2018, growing by 35 percent in one year due to transactions by customers. According to the report on interbank instant payment published by NIBSS Scheme, in 2017, instant transfer transactions worth $ 236,243 was performed with the USSD codes of various banks in the country, this significantly grew to $ 669.100 in 2018 a 35 percent growth within a year. This channel has grown from 25 percent usage in 2017 to 35 percent in 2018 and still growing while the use of mobile apps has grown by just one percent when compared to 2017 [18]. The service is now the best communication platform available to provide mobile financial services to low-income consumers who leave in remote areas without banks and has been embraced by all local banks.

Across countries like Uganda, Nigeria, Tanzania, Kenya, Zambia, Ghana, India, and Argentina, mobile banking is the most convenient and user-friendly alternative to traditional banks, for these countries with a high number of unbanked

people. In these countries, unbanked citizens are still very high in number, Mobile money is an easy-to-use substitute for financial transactions when compared against conventional banks. Kenya's M-Pesa (a mobile money platform) accounted for 43% of the nation's GDP in 2013, 45% in 2015, and 49% in 2016 [19]. Despite these success stories, however, the operators and customers of these services have been threatened by fraudulent activities, as fraudsters continue to exploit vulnerabilities of mobile money services for their financial gain [20]. In 2015, the volume of mobile money fraud in Uganda accounted for 53%, Tanzania for 42%, Kenya for 12%, and Ghana for 23% of all mobile money transactions. [21] respectively. Approximately 89% of fraud involving financial services in Nigeria took place through electronic channels in 2018, while only 11% were none [22].

There are security concerns in USSD, mobile banking services, and other related mobile services, but customers and its service providers still prefer them because of the advantages such as fast, easy to use, convenience in paying bills, availability, etc. Banks promote these services [23] as it helps to handle more customers with improved customer services without compromising on service quality at a reduced operational cost even where there is no physical branch.

A. USSD Mobile Banking Service

USSD mobile banking is a pull-based service that, run as real-time open session and interactive based services. Unlike SMS it doesn't operate on a store and forward mode making its response time is much faster and reliable for interactive application compares to SMS, it is supported by mobile phones available on any GSM network [24], all these features make its operations simple and user-friendly even on low-end mobile phone devices which most of the low-income customers use.

Apart from financial service USSD is used to render other forms of service such as weather information, sports updates, market survey, news, reservation applications, prepaid call-back service, location-based content services, Order confirmations, etc. Despite all these services rendered with USSD and convenience it offered customers in accessing financial services it has its security challenges, some of USSD specific security issues include:(a) Replay attack on USSD request and response message is possible (b) There is sometimes a network delay in USSD request and response messages that can be exploited in terms of its request and the integrity of its response. (c) Confidential information such as PIN, the customer number is displayed when using USSD mobile banking services. Any person viewing the transactions can exploit this vulnerability later. (d) Misuse of dirty USSD codes for operations such has Change PIN code, Factory reset, Display IMEI number resulted in the loss of critical information. (e) The system allows the use of default PIN and doesn't enforce a periodic change of PIN policy (f) The system uses only 4-digit numerical characters as PIN, which are not masked when entered on a mobile phone and can easily be guessed. Regardless of the industry, sector, fraud is significant in almost every financial service available today, this has led to lots research in fraud detection systems to detect fraud and reduce its impact on the economy, a review of fraud detection systems done over the years has highlighted types of fraud detection systems used and issues with them that have affected their results.

IV. Machine Learning Algorithms

Several supervised and semi-supervised learning techniques have been used in fraud detection models to measure, process and predictions of their datasets, machine learning algorithm such as support vector machines (SVM), random forest (RF), XGBoost.

A. Support Vector Machines (SVM)

SVM is a common algorithm in machine learning used for classification and regression. SVM modeling includes two steps, first train the dataset, to obtain a model & then to use this model to predict information from the dataset. A supervised learning algorithm analyzes data for classification and regression. SVM separates the two classes of data by a hyperplane and separates the classes in the best way, in this way it maximizes the distance between the closet points (support vectors) to the hyperplane on both classes is maximizes the margin between the support vectors on both sides of it as shown in Fig. 4.1.

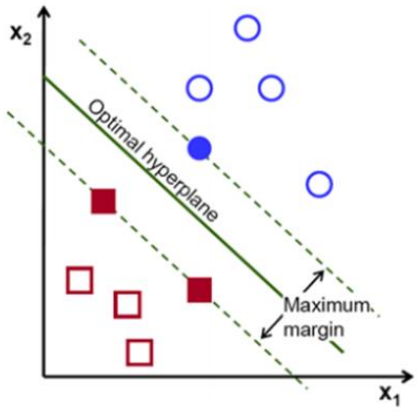


Fig. 4.1 SVM Model Graph [14]

B. Random forest (RF)

Random forest is also used for regression and classification, it corrects overfitting in its trained set an advantage over decision tree, A subset of the training set is randomly sampled so that each tree is trained and created, then a decision tree is constructed, afterwards each node divided into a feature selected from a random subset of the full feature set, this subset is defined using a subset ratio parameter creating an ensemble of random trees using the resulting created tree to determine the final classification outcome, random forest training is extremely fast for large data sets with multiple features and data instances since each tree is trained independently of the other, It provides a good estimate of generalization error and is resistant to over-fitting [25].

$$P(c|x)=(P(x|c)P(c))/(P(x)) \quad (1)$$

P(c|x) = Posterior Probability, P(x|c) = Likelihood, P(c) = Class Prior Probability, P(x) = Predictor Prior probability.

C. XGBoost

Optimized gradient boosting (Extreme Gradient Boosting) a decision tree-based algorithm that makes use of gradient boosting framework improving it through optimization and algorithmic enhancements, compared to other machine algorithms, XGBoost reduces runtime and better performance. Sub-trees from the original tree are sequentially constructed to reduce the previous tree errors in each subsequent tree. The new sub-trees thus update the previous residuals, thereby reducing costs.

V. METHODOLOGY, DATASET AND IMPLEMENTATION OF THE METHODS

A. Methodology

Among the problem situation which has been attributed to the USSD poor detection, insufficient features representation has affected its performance, as this affects the selection of the features to represent the data for optimum performance of the model, feature derivation will be introduced to solve this, as new features from the dataset will improve its detection rate, to solve its second problem an efficient fraudulent detection model is design to detect fraudulent activities using derived features.

To complete this study, its framework was divided into two phases, Phase 1 covers an overview of the dataset, its preprocessing, and feature derivation, as the features derived such as transaction type, amounts, transaction frequency will be used as input in Phase 2, consisting of four parts, short memory prediction, evaluation of short prediction threshold construction and model update as the final output of this phase would be a trained USSD fraud detection model.

B. Dataset

This section presents an exploratory analysis of the dataset for this study, the dataset is a total of 6362620 transactions, with 5 transaction types consisting of CASH_OUT with 2237500 transactions, PAYMENT with 2151495 transactions, CASH_IN with 1399284 transactions, TRANSFER with 532909 transactions, DEBIT with 41432 transactions, there are 11 variables (columns), these columns descriptions are as follow:

- step: mapping a unit of time in the real world. In this case, 1 step is 1-hour, total steps of 744 steps a month's simulation (30 days).
- type: this contains transaction types, CASH-IN, CASH-OUT, DEBIT, PAYMENT and, TRANSFER
- amount: this contains the amount of the transaction in local currency
- nameOrig: the customer who started (initiated) the transaction
- oldbalanceOrig: this contains the initial balance before the transaction
- newbalanceOrig: this contains the customer's balance after the transaction.
- nameDest: this is for the recipient ID of the transaction.
- oldbalanceDest: this contains the initial recipient balance before the transaction.
- newbalanceDest: this contains the recipient's balance after the transaction.
- isFraud: this contains identifies a fraudulent transaction (1) and non-fraudulent (0)
- isflaggedFraud: this contains flags as illegal attempts to transfer more than 200.000 in a single transaction.

During Transfer and Cash-out, the major fraud in the dataset occurs so that this is very important to our feature derivation, after checking and cleaning the null value data, examining the time of fraudulent transaction took place helps in profiling clients based on their transaction frequency/consistency, the dataset was a total of 744 steps for a period of 30 days (1 step to 1 hour) so the Valid and Fraudulent transaction rate for these 744 steps (30 days) are check.

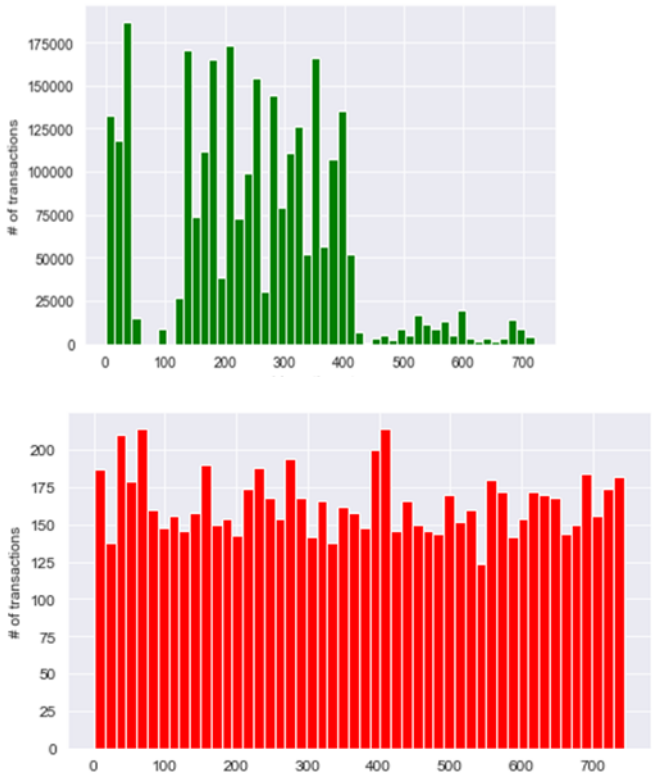


Fig. 5.1 Valid and fraudulent transaction over 774 steps

Fraudulent transaction by hour seems more constant when compared to the valid transaction that is made within the day, comparing both valid and fraudulent transaction across the 744 steps, Fig. 5.2 confirms that fraudulent transaction is constant but all within a price range as indicated with the red space in

the figure, this makes it difficult to detect fraud. So, to simplify this for the FDM to detect its fraud feature derivation is introduced, with this fraud detection will guarantee based on features that indicate that the transaction is a fraud.

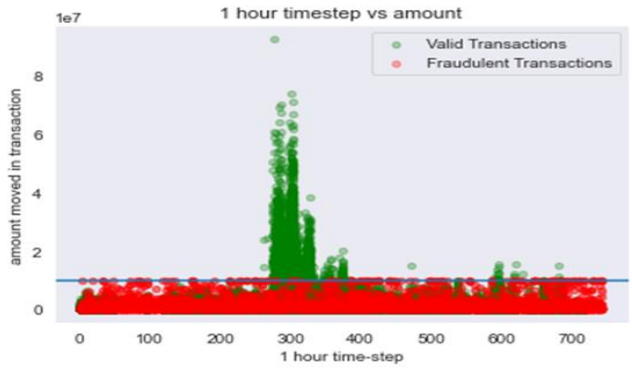


Fig. 5.2 Valid, fraudulent transaction and amount transacted within the 744 steps

C. User Profiling Based on Aggregation

To profile the user based on their mode and frequency of transaction, data was aggregated according to sum of transaction, sum of transfer, cash out, destination, type of transaction, time of transaction, how the profile perform transactions throughout the month all this in regards to each user, categorizing the user to either a high transaction profile, normal transaction profile and low transaction profile based on their transaction performance.

D. Feature Derivation

Features are derived and extracted from the dataset based on relevance of each feature in identifying the fraud transaction, and importance of each features to the model. The features were derived based on transaction frequency within a day, week, month, night, or day transaction, either cash_out or transfer transaction, and the number of each transaction to a specific destination. The steps involve in deriving the feature are illustrated in Fig. 5.3, after initializing it was declared to do for I in the customer list, append 1 if I in customer list else append 0 following the steps all through till the last feature is derived, and a new dataset is generated.

After the steps in Fig. 5.3 were completed, Table 5.1 shows the feature extracted, a new dataset is generated with all these features inclusive, the new dataset consisting of four raw features and nine derived features used in this study.

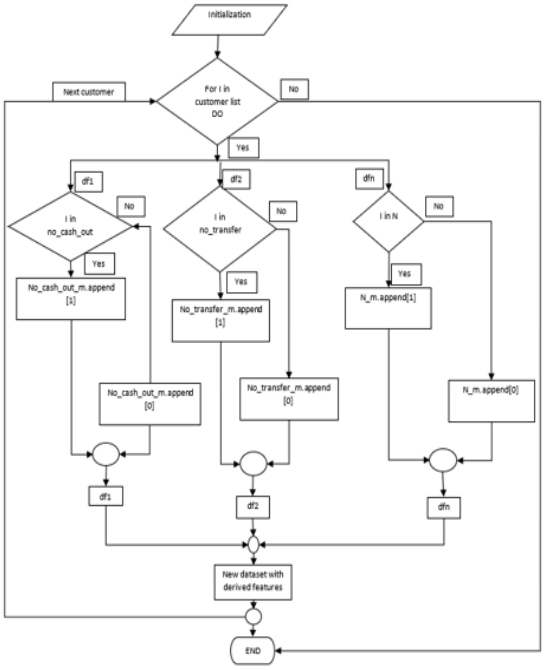


Fig. 5.3 Feature's derivation steps

TABLE 5.1. DERIVED FEATURES

Features	**Description**
Amount frequency	Amount frequency for each customer to group them within a related group, the sum of the amount and max amount for each customer is collected and then classify them based on their frequency.
No_cash_out	The number of cash-out by each customer is taken into account and counted for those who made a cash-out
No_transfer	Each customer's transfer is also counted and see how much each customer transfers and to which destination the transaction is made.
No_day_trans	Transaction made before the 12 hours (<12) is regarded as a transaction within the day, so they are categorized as a day transaction for each customer.
No_ night_trans	Transaction made by the 12 hours or after these 12 (>=12) hours is counted as a night transaction
No_destinations	Account who made a transfer to more than one destination is counted
Hour_of_day	The total number of steps was split into 24 hours per day and transaction made by each customer within each hour of the day is recorded
Day_of_the_week	Each customer transaction is counted on each day of the week they conducted a transaction.
Day_of_month	Each customer daily transaction is counted and recorded

E. Data Pre-processing

After derivation of features were completed, data pre-processing is the next step, cleaning the data removing noisy data, filling missing values and data transformation into its appropriate forms suitable for the model.

- **Standardization:** Standardization brings the data into common format this was use to remove outliers and preventing unequal contribution from derived features giving a levelled playing ground without being bias.

- **Scaling:** The data was scaled between 1 and 0 so the machine learning model won't prioritize some to features with higher value, so the model will perform faster and better since the features are on a similar scale.

F. Features Selection

Feature selection method was introduced selecting the best fit feature set for the model, to see how it can improve the model performance, the feature selected were in set then each set was retrained and the model was tested on with this selected sets with the machine learning classifiers.

G. Model Construction

The model construction is separated into several parts, the first part in this phase is the construction of the short memory prediction using machine learning techniques, its second phase is the prediction evaluation by comparing with the new transaction, and its final phase is model update by analyzing the trends of the prediction error using statistical methods.

- **Feature Selection Phase**: After the best set of features has been selected to improve the model's accuracy performance, each set of features is then evaluated to determine which set best fits the model according to their performance.

- **Construction of short memory prediction phase:** The short term memory prediction as shown in Fig. 5.2 in an offline preparation part, this memory predictor is trained with features extracted from previous phase, the memory predictor is trained based on the transaction windows of a month as this is use to predict transaction mainly for customers whose transaction trend doesn't change, while for short term memory predictor, it is trained with short term transaction windows this is based on daily transactions as this is used to predict sudden change in customers transaction, clustering approach is used to group customers into different groups based on their transaction behaviours, frequency, and recency, in which customers belonging to the same cluster/group have similar transaction patterns using range partitioning, then Using Sliding-Window approach, transactions are aggregated into their respective groups of window either short term or otherwise, the memory predictor is trained with the following machine learning techniques random forest (RF), artificial neural network (ANN), support vector machine (SVM) and XG Boost, then output from this phase are used as input in the next phase in prediction evaluation phase.

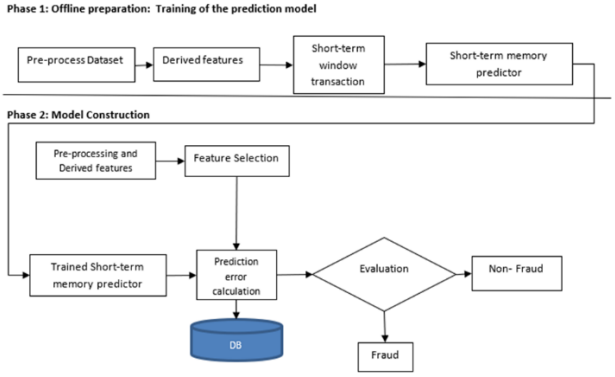


Fig. 5.2. Model Construction Phase

H. Model training and testing

In training the model, the selected classifiers were utilized Kneighbors, logistic regression, SVM, decision tree, random forest and gradient boosting, 70% of the data was used in training the model. The classifiers training result are shown in Table 5.2 respectively for each classifier used in model training.

TABLE 5.2. MODEL TRAINING RESULTS

Classifiers	Accuracy	Precision	Recall	F1 Score
KNeighbors	99.91	99.82	100	99.91
Logistic Regression	51.54	50.83	93.87	65.95
Decision Tree	100.0	100.0	100.0	100.0
Random Forest	100.0	100.0	100.0	100.0
Gradient Boosting	90.72	85.08	98.76	91.41
SVM	56.67	56.36	59.12	57.71

The remaining 30% of the data was used in the phase to test the performance of the model across each of the selected classifiers user in the training phase, result of this section is discussed section 7.

VI. PERFORMANCE MEASURES

Accuracy, false-positive rates, and true positive rates, precision, and F measure described in Table 6.1 will is used to evaluate the performance measure for this model, True Positive (TP) and True Negative (TN) means correct prediction, False Negative (FN), fraudulent observations classified as legitimate, False Positive (FP), legitimate observations wrongly classified as fraudulent ones, refer to the

correct prediction made by the model with True Positive the nature of the fraudulent scenario.

TABLE 6.1. DESCRIPTION OF PERFORMANCE MEASURES

Performance Measure	Description
Accuracy	Correctly classified as a legitimate transaction and fraudulent transaction $\frac{TP+TN}{TP+FP+TN+FN}$
True Positive Rate (TPR)	Measure the frequency of correctly predicted transaction of the model as normal $\frac{TP}{TP+FN}$
True Negative Rate (TNR)	Measure the frequency of correctly predicted fraud transaction of the model as fraud $\frac{TN}{TN+FP}$
Precision	The ratio of positive occurrences correctly predicted to the total of positive observations predicted $\frac{TP}{TP+FP}$
F measure	The weighted average of recall and precision $F1-Score=\frac{2TP}{2TP+FP+FN}$

VII. EXPERIMENT RESULT AND DISCUSSION

After the features were extracted, we tried it with selected classifiers but before prior to the dataset being used on the classifier it was divided into two parts, training and testing, in a ratio of 70% for training and 30% for the testing part, the results observed are tabulated, from the results some of the classifiers did not record all the evaluation metrics, KNeighbors, Logistic Regression, SVM only records the model accuracy which was 99.87% across the three, while the accuracy is a very good start, the other sections of our evaluation metrics cannot simply be ignored as this will result in high false positives with the model, the poor performance may be as a result of data imbalance or overfitting on these classifiers part but as we move on we'll see if these model will improve in their performance when data sampling is introduced, Decision Tree, Random Forest, Gradient Boosting were all 100% across all the evaluation metrics used as illustrated in Fig. 7.1, training period was more than 20 minutes, with all these considered we further need to improve its performance evaluation metric and find ways to reduce its training time to minimal, as this ensure a faster FMD, so we move to the next phase to address data imbalance, introducing under-sampling and over-sampling respectively, observing from the result we obtain, use of imbalanced data affected the performance of the model in some classifiers and this will increase the model false positive results which is a major attribute to poor model performance resulting in high false alarms.

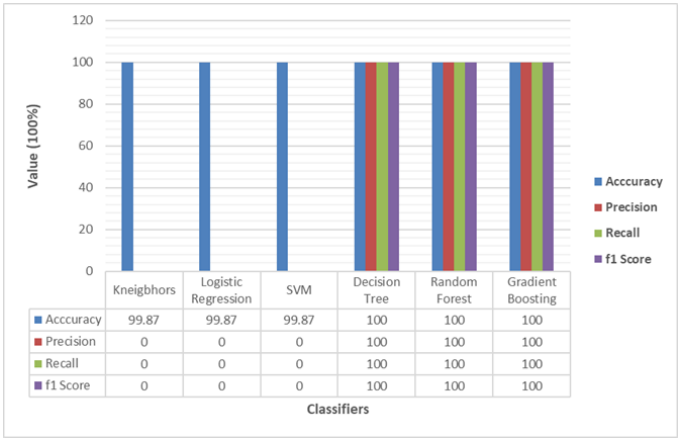


Fig. 7.1. First Results after features were derived illustrated

A. Data Under-sampling:

After obtaining the first result, and seeing some of the evaluation metrics this might be a result of the huge data imbalance in the dataset, so first we try using under-sampling data used to balance the data imbalance Data improve sampling to improve the performance then we retrain and see how the performance.From the results obtained, all evaluation metrics were recorded but the results were poorer with some classifiers such as KNeighbors, Logistic Regression, SVM with accuracy falling as low as 61.11%, Recall at 6,67%, and F1 score at 12.5% compared to its initial results at 99.87%, other Decision Tree, Random Forest, Gradient Boosting classifiers still produced good results of 100% across all the evaluation metrics as illustrated in Fig. 7.2, with these results, despite using under-sampling for our data imbalance the previous poor classifiers were still poor, this affirms that data under-sampling won't improve these classifiers performance which may be a result of overfittings but we still need the model to improve its performance with this classifiers so we try data Over-sampling to balance if this will improve the performance of the classifier.

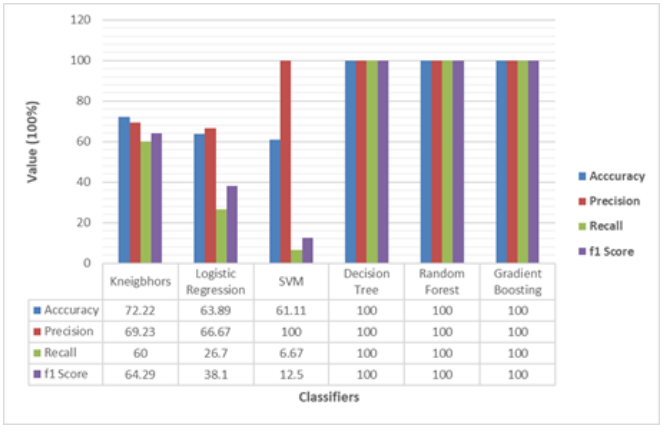
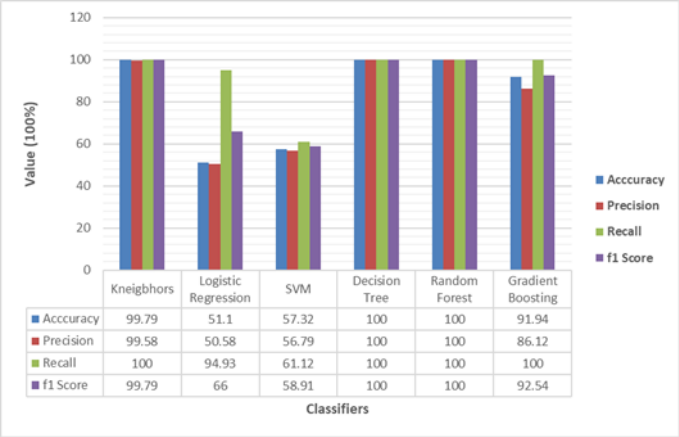


Fig. 7.2. Features derived with under-sampled data balancing illustrated

B. Data Oversampling:

With the results, we've seen after under-sampling has been for data imbalance our next resolution was to use data over-sampling for data imbalance and compare the results with the two previous results, we've obtained. Results show that some of the classifier performance improves with data oversampling with KNeigbors joining other classifiers with very good results average of 99.5% across all the evaluation metrics, though some classifier performance remains poor despite data oversampling, as we continue we'll see if the next section will improve these classifiers performance so now we left with SVM and logistic regression, Gradient Boosting performance reducing, its precision reducing to 86.12% compared to 100% previously recorded but we can still take this result as a good one, Decision tree and Radom forest performance was still at 100%, training time was 18 min 4 s, almost 2 minutes shorter than train time compared to time with just the derived feature data alone, to further improve the model performance and reduced its training time we implore feature selection.



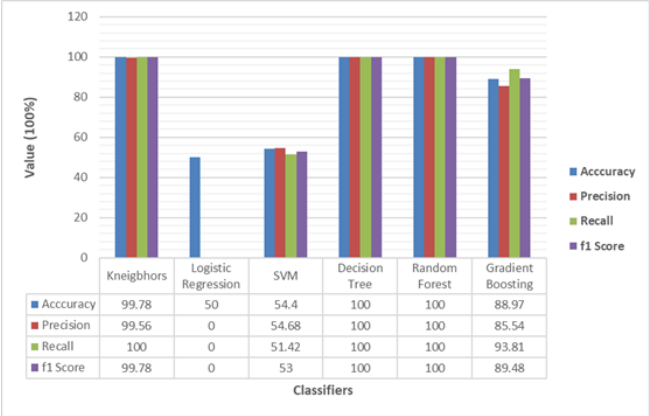
	Kneigbhors	Logistic Regression	SVM	Decision Tree	Random Forest	Gradient Boosting
Acccuracy	99.79	51.1	57.32	100	100	91.94
Precision	99.58	50.58	56.79	100	100	86.12
Recall	100	94.93	61.12	100	100	100
f1 Score	99.79	66	58.91	100	100	92.54

Fig. 7.3. Features derived with over-sampled data balancing illustrated

C. Evaluation of the Feature Selection

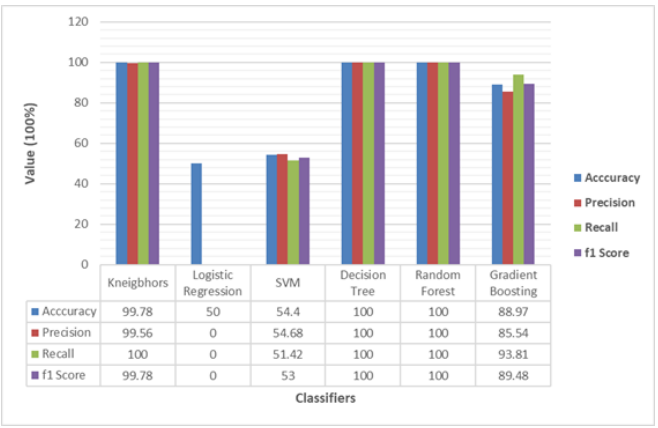
Several features have been derived from our dataset, but not all of them can influence the performance of the model as some of the features will only lengthen the training time of the model, so we used the feature selection to select the best features for this model, but before concluding which features are the best, we considered two main objectives how the selected features improve the performance of the model at the same time reduced the train time.

First, we'll try with five ('Amount', 'MaxAmount', 'No_day_trans', 'No_destinations', 'No_cash_out') features selected and see how the model performance improved with these few features. The train time with these selected features was 10 minutes, while the train time dropped halfway some classifier performance did not improve, logistic regression performance accuracy further reduce other evaluation metrics were all zero, no improvement was seen with SVM, gradient boosting performance also reduced with an average of 10% across the metrics, decision tree and random forest performance still stood at 100% as this is clearly illustrated in Fig. 7.4, Fig. 7.5 shows the result for the next selected features ('Amount', 'MaxAmount' 'Hour_of_day', 'Day_of_month', 'Day_of_the_week'), there were no significant changes in performance in all the classifiers, train time increased by 21 seconds, so this set of features were just as important as the previously selected features.



	Kneigbhors	Logistic Regression	SVM	Decision Tree	Random Forest	Gradient Boosting
Acccuracy	99.78	50	54.4	100	100	88.97
Precision	99.56	0	54.68	100	100	85.54
Recall	100	0	51.42	100	100	93.81
f1 Score	99.78	0	53	100	100	89.48

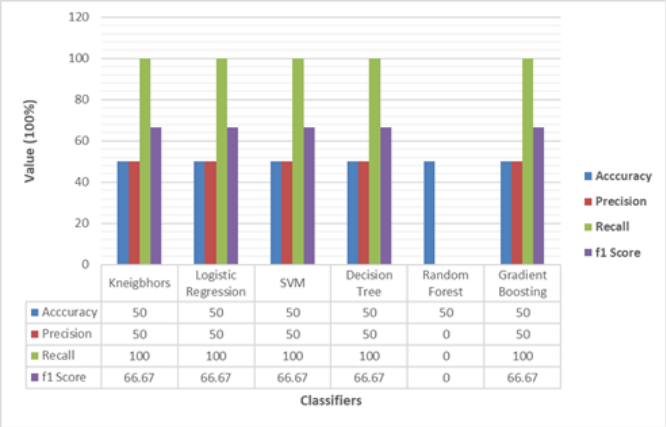
Fig. 7.4. The first set of selected features illustrated



	Kneigbhors	Logistic Regression	SVM	Decision Tree	Random Forest	Gradient Boosting
Acccuracy	99.78	50	54.4	100	100	88.97
Precision	99.56	0	54.68	100	100	85.54
Recall	100	0	51.42	100	100	93.81
f1 Score	99.78	0	53	100	100	89.48

Fig. 7.5. The second set of selected features illustrated

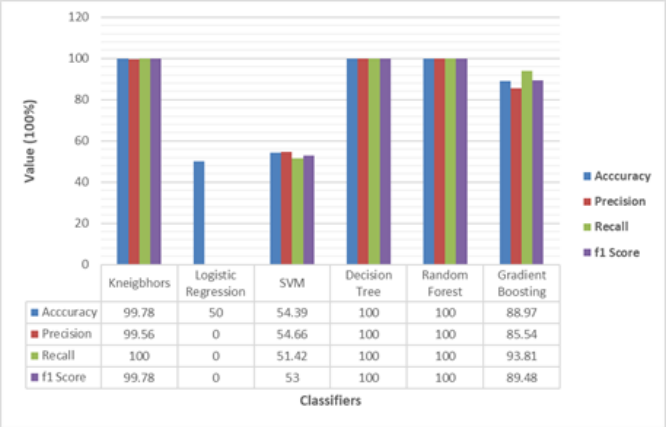
The third set of selected features ('No_transfer', 'No_trans_day', 'No_night_trans', 'Hour_of_day', 'Day_of_month', 'Day_of_the_week') results shown in Fig. 7.6, performance across all the classifier were all the same with accuracy at 50%, precision 50%, recall 100% and f1 score at 66.67% except for random forest with just 50% accuracy recorded other evaluation metrics were zero, this shows the selected features has few or no importance to the model.



	Kneighbors	Logistic Regression	SVM	Decision Tree	Random Forest	Gradient Boosting
Acccuracy	50	50	50	50	50	50
Precision	50	50	50	50	0	50
Recall	100	100	100	100	0	100
f1 Score	66.67	66.67	66.67	66.67	0	66.67

Fig. 7.6. The third set of selected features illustrated

The last set of selected features ('Amount', 'MaxAmount', 'No_transfer', 'No_trans_day', 'No_night_trans) results detailed in Fig. 7.7, the performance wish was similar to the first and second set of our selected features, indicates that the features, all have some equal influence on the model, apart from the third set which has little or no influence on the model.



	Kneighbors	Logistic Regression	SVM	Decision Tree	Random Forest	Gradient Boosting
Acccuracy	99.78	50	54.39	100	100	88.97
Precision	99.56	0	54.66	100	100	85.54
Recall	100	0	51.42	100	100	93.81
f1 Score	99.78	0	53	100	100	89.48

Fig. 7.7. The last set of selected features illustrated

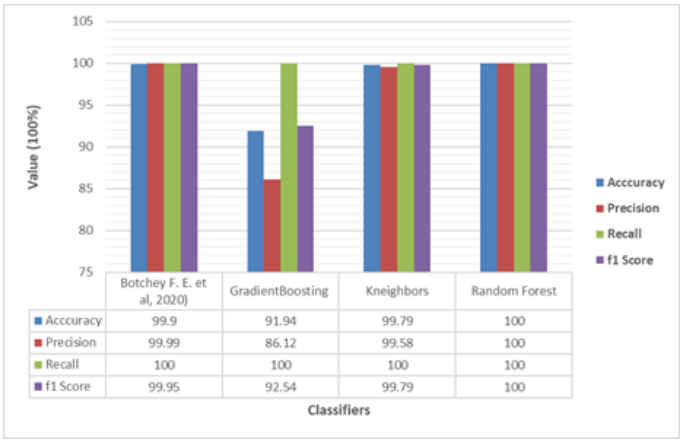
With the features selection three classifiers stood out with good performance result across their evaluation metrics, the classifiers are Kneigbhors, decision tree, random forest both two out of the classifier result were at 100% all through while kneigbhors was at an average of 99% in all the selected features as shown in Table 5.3, with this, it further confirms that all the derived features contribute to the model performance even though some might be of little contribution but this little improves the model performance.

Decision tree and random forest outperformed other classifiers when their evaluation metrics are compared throughout the results obtained, in an environment where decision tree is used, random forest will always perform better and that's what we've seen so far, since decision tree performance was very good we'll expect the random forest to perform better over decision tree as the random forest is a collection of several decision trees and then it randomly selects observations/rows, specific features/variables to build multiple decision trees from and then averages the results.

Despite data under-sampling, oversampling and feature selection SVM and logistic regression performance remains poor compare to other selected classifiers used, this may be a result of data overfitting as random fluctuation in the trained data might have impacted the performance of these classifiers negatively. Considering the main objective of this study is to investigate a suitable machine learning algorithm capable of detecting USSD fraud, we will explore why these classifiers performs poorly in future work

D. Evaluation of the Proposed Model.

To evaluate the proposed model, reference is made to the related work that used the same dataset and its work was to analyze the efficiency of Support Vector Machines, Gradient Boosted Decision Trees, and Naïve Bayes Algorithms (Botchey F. E. *et al.*, 2020) work on mobile money fraud prediction and the same evaluation metrics to measure the performance of the classifiers. Features derivation were not used in the author's work but selected features according to the author's five best features Transfer, newbalanceOrig, oldbalanceDest, amount, and oldbalanceOrg were selected for their model construction and both data oversampling and undersampling were used. The results show that gradient boosted classifier was the classifier with the best result so we benchmark the result with three of our best classifier results from Fig. 7.3.



	Botchey F. E. et al, 2020)	GradientBoosting	Kneighbors	Random Forest
Acccuracy	99.9	91.94	99.79	100
Precision	99.99	86.12	99.58	100
Recall	100	100	100	100
f1 Score	99.95	92.54	99.79	100

Fig. 7.8. Result comparison with selected referenced model

From Fig. 7.8, the referenced work best results were from Gradient Boost, it performed better than the lowest classifier result, gradient boosting result except for recall rate which was both at 100%, the second classifier (KNeighbors) performance of our results which was at an average of 99% across accuracy, precision, and f1 score rate this is very close to the referenced work performance with just 0.11% accuracy difference, 0.41% precision rate and 0.16% f1 score rate when compared these are below 1%, so this suggest that the classifier (KNeighbors) performance is at a very good rate, our best classifier (Random forest) results performed better than

referenced work in all evaluation metrics except for recall rate, same at 100%. In conclusion, our performance rate compared to the referenced work is higher at an average of 0.10% in performance.

VIII. CONCLUSION

In this study, a fraud detection model is investigated with a mobile money data dataset. The model shows the best result with three of our selected machines learning algorithms KNeighbour, decision tree, and random forest, with feature derived a better performance in terms of accuracy was improved, and the model performance further improved with selected features.

Decision tree and random forest outperformed other classifiers when their evaluation metrics are compared, in an environment when decision tree is used random forest will always perform and that's we see the reason since decision tree performance was very good we'll expect the random forest to perform better over decision tree as the random forest is a collection of several decision trees and then it randomly selects observations/rows, specific features/variables to build multiple decision trees from and then averages the results.

.

REFERENCES

[1] H. Gupta Ranjan & J. Lakshmi K. K. (2017). USSD—Architecture Analysis, Security Threats, Issues and Enhancements. *International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS).*

[2] OKO I. A. (2019). Electronic Fraud and Financial Performance of Quoted Commercial Banks in Nigeria. *International Journal of Advanced Academic Research*, 15-35,

[3] Globitel, "Globitel," 22 December 2019. [Online]. Available: http://www.globitel.com/ussd-gateway/.

[4] Nyamtiga *et al*. (2013). Security Perspectives for USSD Versus SMS in Conducting Mobile Transactions: A Case Study of Tanzania. International Journal of Technology Enhancements and Emerging Engineering Research, 1(3), 38-43.

[5] &. K. O. P. Vukeya K. E. (2014). A Model of Fraud Detection in Mobile Transaction via Unstructured Supplementary Service Data. *Southern African Telecommunication Networks and Applications Conference (SATNAC), Port Elizabeth, Eastern Cape, South Africa*.

[6] P. Ch. Devi. (2019). A Novel Hybrid Mechanism for Credit Card Fraud Detection onFinancial Data. *International Journal of Science Engineering and Advance Technology, IJSEAT*, 7(4), 77-80,

[7] U. A. D. S. F. P. P. Z. a. F. P. Fiore. (2019). Using Generative Adversarial Networks for Improving Classification Effectiveness in Credit Card Fraud Detection. *Information Sciences*, 479, 448-455,

[8] N. F. G. &. C. M. Carneiro. (2017). A Data Mining based System for Credit-card Fraud Detection in e-tail. *Decision Support Systems*, 91-101.

[9] C. D. T. Y. Z. L. Fu K. (2016). Credit Card Fraud Detection Using Convolutional Neural Networks. *International Conference on Neural Information Processing.*

[10] F. D. P. A. L. B. Y. A. C. O. M. Y. &. B. G. Carcillo. (2018). Scarff: A Scalable Framework for Streaming Credit Card Fraud Detection with Spark. *Information Fusion*, 41, 182-194.

[11] J. G. M. Z. K. C. S. P. P. E. H.-G. L. &. C. O. Jurgovsky. (2018). Sequence Classification for Credit-card Fraud Detection. *Expert Systems with Applications*, 234-25.

[12] S. a. P. S. Subudhi. (2016). Use of Fuzzy Clustering and Support Vector Machine for Detecting Fraud in Mobile Telecommunication Networks. *Int. J. Security and Networks*, 11(Nos. ½), 3-11.

[13] S. &. S. S. Ganguly. (2018). Online Detection of Shill Bidding Fraud based on Machine Learning Techniques. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems.*

[14] Navanshu Khare and Saad Yunus Sait. (2018). Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models. *International Journal of Pure and Applied Mathematics*, 118(20), 825-838.

[15] J. T. Z. W. a. F. G. Y. Zhang. (2020). Customer Transaction Fraud Detection Using Xgboost Model. *International Conference on Computer Engineering and Application (ICCEA), Guangzhou, China*.

[16] Patil Suraj Varsha Nemade and Piyush Kumar Soni. (2018). Predictive Modelling for Credit Card Fraud Detection Using Data Analytics. *Procedia Computer Science*, 132, 385-395.

[17] K. K. G. H. &. R. J. Lakshmi. (2019). UPI Based Mobile Banking Applications–Security Analysis and Enhancements. *Amity International Conference on Artificial Intelligence (AICAI).*

[18] Africa Mobile Money. (2019). Mobile Money Africa. [Online]. Available: https://mobilemoneyafrica.com/blog/nigeria-ussd-transactions-grew-by-35-hits-n261m.

[19] Deloitte, 6 June 2017 2015. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/lu/Documents/financial-services/Banking/lu-mobile-money-payment-industry-marketing-distribution.

[20] P. Z. A. N. W. D. N. a. D. K. L. Chatain. (2011). Protecting Mobile Money against Financial Crimes: Global Policy Challenges and Solutions, New York, NY.

[21] B. Busuulwa. (2016). Mobile Money Fraud, Crime Rate Increase in Uganda. [Online]. Available: www.theeastafrican.co.ke/business/Mobile-money-fraud-and-crime-rate-increase-in-Uganda-/2560-3415786-quaydf/index.html/.

[22] NIBSS. (2018). About NIBSS: Nigeria Inter-Bank Settlement System Plc (NIBSS) was Incorporated in 1993 and is Owned by All Licensed Banks Including the Central Bank of Nigeria (CBN). [Online]. Available: https://nibss-plc.com.ng/.

[23] M. K. C. A. A. A. H. Kelvin Chikomo. (2006). Security of Mobile Banking.

[24] E. Taskin, (2012). GSM MSC/VLR Unstructured Supplementary Service Data (USSD) Service.

[25] Campus, K. (2018). Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models. *International Journal of Pure and Applied Mathematics*, 118(20), 825-838.