University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Faculty of Computing, University of Ibadan, Ibadan, Nigeria

Volume 15 No. 1, September 2025

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/
uijslictr@gmail.com

University of Ibadan
Journal of Science and Logics in ICT Research

Development of Comparative Fake Transactions Alert Detection Models Using Machine Learning

[1]✉ Oguntunde T., [2]Abioye C. Oluwafisayo.

Department of Computer Science, University of Ibadan, Ibadan-Nigeria.
[1]tantos557@yahoo.com [2]christianahabioye08@gmail.com

Abstract
Fraudulent payment evidences are the current techniques criminals employ to defraud businesses and small scale enterprises. Researchers have processed transactions textual data however, there is still the challenge in the ability to differentiate between legitimate and fake transaction alerts. In Nigeria, fake transaction alerts pose significant challenges for financial institutions and individuals losing on their hard earned assets, citizens are sceptical on electronic transactions and several Point of Sale (PoS) businesses have fallen victims. Hence, this study was aimed at the development of a better fake transaction alert detection model to distinguish fake transaction alerts. Artificial Commercial Data for Fraudulence Discovery was collected from Kaggle website. The collected data was pre-processed. Data imbalances were handled. Support Vector Machine (SVM) and Random Forest (RF) algorithms were ensembled to simulate fake transactions alert detection models using MATLAB programming. They were trained and tested with 70% training and 30% testing datasets, respectively. Performance evaluation was done on RF and SVM classifiers using exactness, precision, recollection, F-measure as benchmarks. The data record employed for this study had 1,048,575 transactions alerts. At performance evaluation, RF model had exactness, precision, recollection and F-measure values, 97.6, 97.48, 97.54 and 97.51%, respectively. Its RMSE was 0.02376. Moreover, SVM model had exactness, precision, recollection and F-measure values of 96.1, 96.88, 98.38, and 97.63%, respectively, it has RMSE of 0.03911. Random Forest algorithm was more suitable for the development of the fake transactions alert detection because it had higher performance. This model could be adopted by financial related institutions.

Keywords: Fake transactions alert, fake alert detection, fraudulent transaction alert detection, machine learning, Random Forest, Support Vector Machine.

1. Introduction

A fake transaction alert, also known as a phishing scam or fraudulent transaction, refers to a type of deceptive activity where an individual or organization tricks someone into believing they are receiving a legitimate notification or alert regarding a financial transaction. A forged transactions message, otherwise called flickering in the public world, and it typically accomplished through text messages (SMS), which imitates a bank's proof of exchange to deceive unsuspecting victims (*Omogbolagun, 2021*).

Fake transaction alerts now pose a significant challenge for financial institutions and individuals, and it leads to financial losses and potential security risks. According to TheGuardian Daily News, criminals employ certain application software to perform this misconduct. Applications such as Instant Access, Elite SMS, Cash Prank, Millionaire false Financial Account as well as false credit text maker for Android (*Adepetun and Olayinka, 2023*).

This false transactions message is now one of the fast avenues fraudsters now swindle businesses, and it has become a common phenomenon in several places in Nigeria. It has since become a threat to businesses and startup investments. Several PoS businesses have become targets of the fraudulent act in recent times (TheGuardian, 2023). These have made detection and prevention ways of curbing this menace of utmost importance to everyone, most importantly to those in Information Technology fields.

It has however been discovered that the application of machine learning techniques offer capable methods to automatically classify alert transactions as either fraudulent or valid

Oguntunde T., and Abioye C. Oluwafisayo (2025). Development of Comparative Fake Transactions Alert Detection Models Using Machine Learning. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 15 No. 1, pp. 121 – 129.

with high accuracy. This can be achieved by training classification models on features engineered from historical alert data with fraudulent labels. The models can learn to recognize predictive patterns and characteristics of fake activities. This study developed a model that helps to detect fake alert transactions from genuine ones.

2.1 Related Works

Van Vlasselae *et al.,* (2015) proposed APATE, a new technique to identify fake charge card trades carried out in virtual shops. It combined cloud features by taking advantage of the system of vendors and credit card owners to develop a finite suspicion mark for each network object, and fundamental features developed from the attributes of incoming deals and the customer expenditure records employing the principles of RFM (Recency - Frequency - Monetary). The findings indicate that there was a considerable interdependence between intrinsic and network-based properties. The optimal models, with AUC (Area under the Curve)-scores more than 0.98, are produced by combining these two categories of features.

Soltani and Akbari (2014) used Artificial Immune Systems (AIS) to solve charge card misconduct discovery. They also created a novel model known as the AIS-based Fraud Detection Model (AFDM). For misconduct discovery, an immune system-inspired algorithm (AIRS) was applied then, refined. When compared to the base algorithm, the accuracy was improved by 25%, cost was decreased by 85%, and the system response-time was shortened by 40%.

Dhankhad *et al.,* (2018) implemented ten supervised machine learning algorithms to isolate false charge card deals using a actual data records. Additionally, these algorithms were used along-side with aggregation learning techniques to develop a powerful classifier. The very crucial elements, which could result in increased precision in identifying fraudulent credit card transactions were determined. A real-world credit card transaction was used in evaluating the effectiveness of many supervised machine learning models on detecting identity fraud or non-fraud. They also compared the systems' ecatness, True Positive Rate (TPR), False Positive Rate (FPR), Geometric-mean, recall, precision, specificity, and F1-score. With a 0.5360 FPR, SVM was found to have the greatest rating, while the stacking classifier had the lowest, at 0.0335.

Chumuang *et al.,* (2020) worked on creating a charge card misconduct monitoring system by analyzing credit card fraud risk behavior using an Internet app notification that uses a systematic technique. The developed system’s goal was to reduce the likelihood of charge card misconduct or stop the harm, which can result from it. The system notified users of credit card fraud that is deemed hazardous or suspected. The capability evaluation of the system was carried out using yardsticks such as duration, validity under the given conditions, and correctness and completeness of the information. The system's accuracy, efficiency and timeliness measurements came back at 86.7%, 80.0% and 86.7%, respectively.

Roseline *et al.,* (2022) discovered machine learning techniques applied to card transactions database so as to discover patterns. To ascertain the legitimacy about card transactions, they applied statistical models using a range of random techniques and machine learning models, including Naive Bayes, SVM, ANN, and LSTM, were employed to appropriate the issue of class imbalance in the data records. Performance evaluation of Naive Bayes, Support Vector Machine (SVM) and Artificial Neural Network were done against LSTM-RNN (ANN). Tests show that Roseline *et al* suggested model has a high degree of accuracy and yields strong outcomes.

De Sá *et al.,* (2018) employed an automated Hyper-Heuristic Evolutionary Algorithm (HHEA) to develop Fraud Bayesian Network Classifier (Fraud-BNC) for charge card fraud detection. HHEA searched for the optimal combination of fraudulent components in a given dataset and organized the knowledge about BNC techniques into taxonomy. Two methods for handling cost-sensitive classification were tested in conjunction with Fraud-BNC, which was spontaneously constructed using a data record set from PagSeguro (online payment service in Brazil). The Fraud-BNC’s grouping capacity was compared with seven different algorithms. Fraud-BNC proved to be the most effective algorithm, improving the current company's economic performance by 72.64 percent.

Osegi & Jumbo (2021) compared a proposed novel Internet-based learning method in abnormality detection called Hierarchical Temporal Memory, which is founded on the Cortical Learning Algorithms (HTM-CLA), trained with an Artificial Neural Network using the Simulated Annealing technique (SA-ANN). Additionally, a deep recurrent neural technique built on Long Short-Term Memory ANN (LSTM-ANN) was compared with HTM-CLA. The ability of the systems to accurately categorize charge card fraud (CCF) using an average categorization performance ratio yard-stick is the basis for investigating their performances. Simulations conducted on the Australian and German CCF standard data record sets showed that the proposed HTM-CLA outperformed the SA-ANN in a competitive manner. In the target data records, the HTM-CLA proved a clear 2:1 advantage over the LSTM-ANN.

Albashrawi (2016) carried out a analysis enquiry studies to isolate financial misconduct with information discovery technique between 2004 and 2015, and informed academic researchers and business professionals about the latest developments. Between 2004 and 2015, among the most widely used techniques, logistic regression (LR) appears to be the most effective method for identifying financial fraud, with a 13 percent usage frequency. Both Artificial Neural Networks (ANN) and Decision Trees (DT), with an 11 percent usage frequency, are next in line. Naïve Bayes (NB) and Support Vector Machine (SVM) were represented with 6% and 9% frequency, respectively. Furthermore, the methods were applied singly or in conjunction with a collaborative method for the development of robust detection model.

Malini *et al.* (2017) explained how to spot charge card theft. This research related varieties of methods for isolating dubious charge card transactions, including sequence alignment, fuzzy logic, genetic programming, machine learning, and fuzzy logic. To enhance the optimum result for the scam detection challenge, the strategies were combined with the K-Nearest Neighbor (KNN) procedure and outlier discovery methods. It has been demonstrated that these methods minimize the incidence of wrong alarms and increase the rate of fraud detection.

Lepoivre *et al.*, (2016) used two unsupervised algorithms to create an anti-fraud project. Classification is used by them to create a bundle. The clustering approach will then be used to group each package. The model has been used on manually entered data that included numerous bank accounts. To classify the transactions, an unsupervised classification approach called Principal Component Analysis (PCA) and SIMPLE K-MEANS has been used.

3.0 Methodology

3.1 Data Collection

The data used for this research work is a Synthetic Financial Datasets generated by the PaySim mobile money simulator for Fraud Detection downloaded from Kaggle website. PaySim simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. The original logs were provided by a multinational company, who is the provider of the mobile financial service which is currently running in more than 14 countries all around the world.

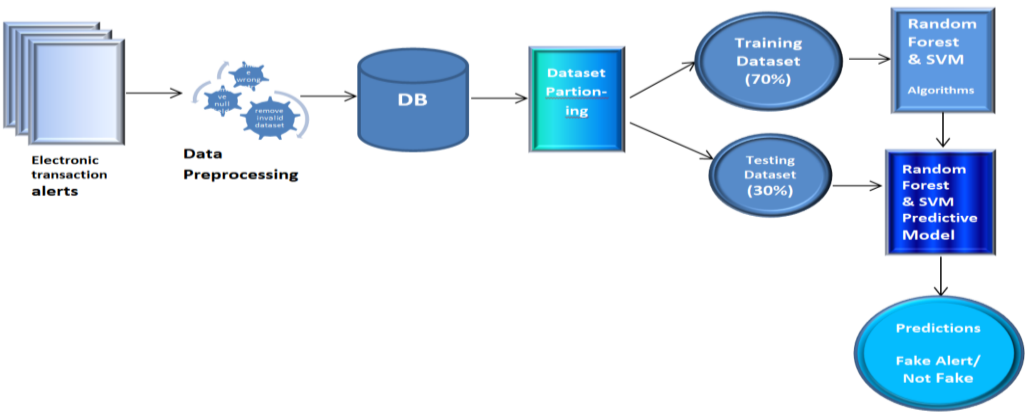

Figure 1: Fake Transactions Alert Detection Model

3.2 Data Preprocessing

Data preprocessing begins with a thorough examination of the dataset to identify and address any inconsistencies, errors, or missing values in the data.

3.2.1 Data Cleaning

The raw transaction data was cleaned by removing duplicates, outliers, and irrelevant information. Data entry errors, formatting inconsistencies, and invalid values were addressed to improve data quality and reliability. MATLAB's functions such as summary, missing, isoutlier, were used to identify missing values, outliers, duplicates, and inconsistencies. Furthermore, functions like isnan, ismissing, and fillmissing were used to detect and handle missing values by imputation or removal. Duplicates were identified using functions like unique and duplicated, and removed or merged duplicates using indexing or logical indexing operations.

3.2.2 Data Transformation

Categorical variables were changed into Arabic numeral depictions using methods like label coding. Additionally, numerals features such as transaction amount and time stamps were standardized or scaled to ensure uniformity and comparability across different scales.Data imbalances were also addressed through techniques such as under sampling of majority classes to achieve a better balanced degree of existence of dubious and non-dubious dealings.

3.2.3 Feature Engineering

Meaningful insights were derived from existing features to improve the classification ability of the classifier. Feature engineering methods includes creating transaction frequency metrics, aggregating transaction amounts over time periods, and creating metrics for fake and real transactions.

3.2.4 Deployment of Fake Transaction Alert Detection Model

A MATLAB program was written to model Support Vector Machine and Random Forest algorithms being trained using fake transaction alert dataset containing 1,048,575 transaction alert datasets. Support Vector Machines provided a model that showed the connection between very complicated input data records and yield by modifying the weights and biases. The dataset (where **isFraud** was the target while the features were Amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, and newbalanceDest) was partitioned into 70:30 for Training and testing, respectively, for both SVM and Random Forest (RF) as in codelisting 1. k-fold cross-validation techniques was used to assess the model's performance.

Hyper parameter tuning fine-tuned parameters (learning rate, regularization strength, and tree depth) using grid search to realise the maximum predictive ability. The model was trained iteratively with periodic evaluations and the predictive power of the model was estimated employing independent datasets.

```
%% Select training and testing dataset (70% and 30% respectively)
X_train = X(1:(length(fake_real)*0.7),:);
Y_train  = class(1:(length(fake_real)*0.7),:);

X_test = X((length(fake_real)*0.7)+1:end,:);
Y_test  = class((length(fake_real)*0.7)+1:end,:);
```

Codelisting 1: Data Partitioning into 70:30, Training and Testing, respectively

```
%% Create a cross-validated classifier.
cvmdl = crossval(SVMMODEL);

kloss = kfoldLoss(cvmdl);
%% Accuracy of the Classifier
Acc = (1 - rloss)*100;
disp(['Accuracy of the Classifier = ' num2str(Acc) '%'])
```

Codelisting 2: K-fold Cross-validation

Yardsticks of comparison among the classifiers were statistical (Root Mean Square Error (RMSE) and Mean Absolute Error (MAE)) and capcity metrics: exactness, correctness, recollection, and F-measure.

Random Forest is a collaborative learning method, which creates many decision trees during training, which yields the most prevalent of the classes (classifications) or the mean forecast (regression) of each of the trees to recognize distributions and relationships between normal transaction features and fraudulent behavior. The algorithm automatically picks a random subdivision of predictors for every of the tree, minimizing the impact had by irrelevant or noisy features on the final predictions.

The Random Forest classifier was trained using MATLAB's inbuilt function, **TreeBagger**, specifying the number of trees in the forest (**numTrees**) and the classification method.

Random Forest hyperparameters were tuned to optimize performance (trees numbers (numTrees), highest tree deepness, minimum leaf size, and the count of predictors) to consider for every of the fragment. Cross-validation was done for optimal set of hyperparameters that maximize performance on validation data. The model was trained iteratively, with periodic evaluations on the evaluation set to monitor progress and prevent overlearning.

Predictions were made on the test set by the trained Random Forest classifier. Capacity yardsticks of measurement: exactness, correctness, recollection, and F-measure were computed as regards the predicted as well as the actual labels.

```
%% Select training and testing dataset (70% and 30% respectively)
X_train = X(1:(length(fake_real)*0.7),:);
Y_train  = class(1:(length(fake_real)*0.7),:);

X_test = X((length(fake_real)*0.7)+1:end,:);
Y_test  = class((length(fake_real)*0.7)+1:end,:);
```

Codelisting 3: Random Subdivision of Predictors to Minimize Noise Impact

```
% Train the Random Forest classifier
numTrees = 100; % Number of trees in the forest
rf_classifier = TreeBagger(numTrees, X_train, Y_train, 'Method', 'classification');
```

Codelisting 4: MATLAB's TreeBagger, Specifying the Number of Trees in the Forest

```
%% Make predictions on the test set
y_pred = predict(rf_classifier, X_test);
```

Codelisting 5: Prediction on the Test set

```
%% Convert predicted labels to numeric values
y_predd = cell2mat(y_pred);
Y_testd = cell2mat(Y_test);
%% Evaluate the performance of the Random Forest classifier
accuracy = sum(y_predd == Y_testd) / numel(Y_test);
precision = sum(y_predd & Y_testd) / sum(y_predd);
recall = sum(y_predd & Y_testd) / sum(Y_testd);
f1 = 2 * (precision * recall) / (precision + recall);

disp(['Accuracy: ', num2str(accuracy)]);
disp(['Precision: ', num2str(precision)]);
disp(['Recall: ', num2str(recall)]);
disp(['F1 Score: ', num2str(f1)]);
```

Codelisting 6: Capacity yardsticks of measurement

The performance of the deployed models, SVM and the Random Forest, were evaluated using several standard coefficients vis-à-vis Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Error matrix (determined with True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN)), generated from MATLAB was used to determine the number of real transaction (positives) and fake transaction alerts (negatives). Correctness, recollection, F-measure, Intersection Over Union (IOU), and Overall Accuracy (OA) were determined. Then the predicted result from the developed models obtained through SVM and Random Forest were compared with the downloaded dataset to know the accuracy of the guess of every Machine Learning models.

4.1 Result and Discussion

Figure 4.1 presents a cross-section of the transaction alerts in the dataset that are categorized as fake or real transaction alert. The transaction in the dataset was first categorized into fake and real transaction by subtracting the old balance from the new balance. The value of the difference in the account balances (old and new) was compared with amount received as alert for each transaction. If the difference between the new account balance and old account balance is not equal to the amount received as alert for each transaction, the computer program will categorize the transaction as fake else such transaction is categorized as real.

In this study, Fake Transactions Alert Detection models were developed with two machine learning algorithms, Support Vector Machine (SVM) and Random Forest (RF) algorithms. The dataset used to train the Models contain a record of 1,048,575 transaction alerts. The whole dataset was partitioned into 70% (734,002) for training while 30% (314573) of the dataset for testing. MATLAB code was written to implement the two models, which were trained with the training dataset.

The Random Forest and Support Vector Machine models were evaluated using the testing dataset. Only 3.9% of the dataset were misclassified by the SVM model while 2.37% of the dataset were misclassified by the RF model. The Confusion Matrix for the RF Fake Transactions Alert Detection model correctly classified 240549 (97.1%) fake transaction alerts as fake, while 7262 (2.9%) were incorrectly classified as fake. Furthermore, 66548 (99.7%) real transaction alerts were correctly classified as real, while 213 (0.3%) were incorrectly classified as real indicating a significant improvement in the model's accuracy (Figure 2). In Figure 2 and 3, the numbers of truly classified classes were written at the diagonal of the Matrix while the numbers of falsely classified classes are written off the diagonal of the matrix.

In Figure 3, the confusion matrix for the SVM Fake Alert Detection model correctly classified 240,083 (96.9%) fake transaction alerts as fake, while 7728 (3.1%) were incorrectly classified as fake. Moreover, 62817 (94.1%) real transaction alerts were correctly classified as real, while 3944 (5.9%) were incorrectly classified as real indicating a significant improvement in the model's accuracy.

The RF and SVM values for TP, TN, FP and FN were 240549, 66548, 7262, and 213; and 240083, 62817, 7728, 3944, respectively.

Table 1: A cross-section snapshot of the PaySim Dataset

Amount	NewBalance	OldBalance	oldbalance Dest	newbalance Dest	IsFraud	Isflagged	Diff	Class
9839.64	170136	160296.36	0	0	0	0	9839.64	'real'
1864.28	21249	19384.72	0	0	0	0	1864.28	'real'
181	181	0	0	0	1	0	181	'real'
181	181	0	21182	0	1	0	181	'real'
11668.14	41554	29885.86	0	0	0	0	11668.14	'real'

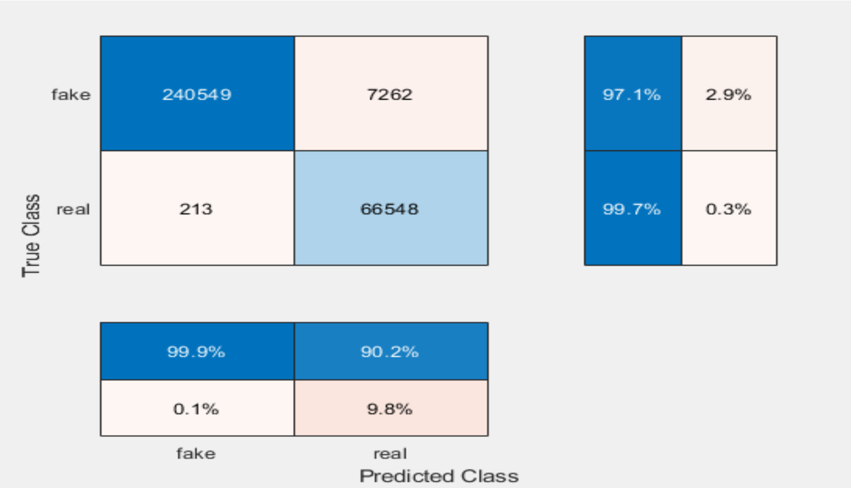

Figure 2: Confusion Matrix for RF Fake Transactions Alert Detection Model

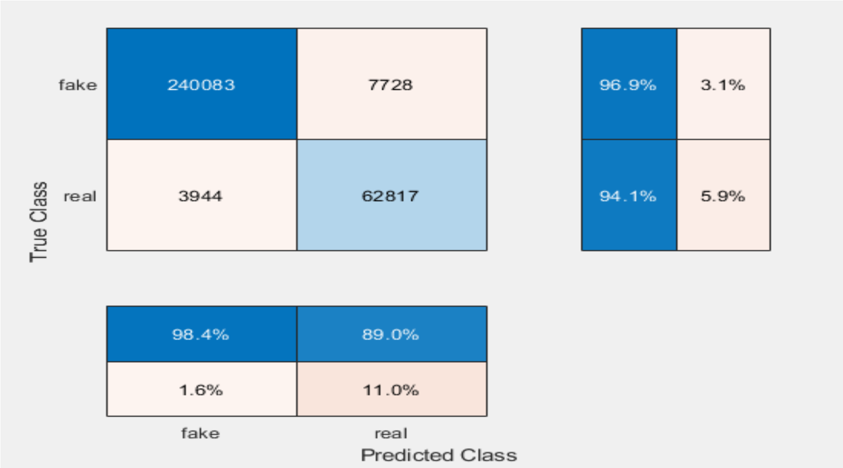

Figure 3: Confusion Matrix for SVM Fake Transaction Alerts Detection Model

Table 2: RF & SVM Values for True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN)

Model	TP	TN	FP	FN
RF	240549	66548	7262	213
SVM	240083	62817	7728	3944

The high value of TP and TN and low values of FP and FN gotten from this model is an indication that the fake transactions alert detection model by both algorithm has the ability to accurately classify transaction alerts although Random Forest algorithm has the higher TP and TN values and lower FP and FN values, which distinguished it as a better algorithm for developing fake transactions alert detection model.

The performances of Fake Transactions Alert Detection RF and SVM models were also evaluated using measures of RMSE, precision, accuracy, recall and F1 Score. Precision measures the proportion of true positive predictions among all positive predictions made by the model. The F1 score is the harmonic mean of precision and recall and provides a balance between the two metrics. A high F1 score indicates that both precision and recall are high.

In Table 3, RF had an RMSE of 0.02376 and an accuracy of 97.6%. Its precision, recall and F1 Score values were 0.97478 (a very low false positive rate), 0.97545 (indicates a very low false negative rate) and 0.97511(indicating a high balance between precision and recall), respectively. SVM model had an RMSE of 0.03911 and an accuracy of 96.1%, while its precision, recall and F1 Score values were 0.968815, 0.983838, and 0.976269, respectively. Having a high precision, recall, and F1 score suggests that the model has made very accurate positive predictions while also capturing a large proportion of the actual positive instances, indicating excellent performance.

A lower RMSE (RF=0.02376, SVM=0.03911) indicates better model performance, as it means the model's predictions are closer to the actual values. Comparing the performance of the two-models, Random Forest algorithm is more suitable for the development of the Fake Transactions Alert Detection model because it has a higher accuracy than Support Vector Machine algorithm as indicated in Figure 4.

5.0 Conclusion

Comparing the performance of the two algorithms for the development of fake alert detection models, Random Forest algorithm was more suitable for the development of the fake transactions alert detection because it had higher performance, overall, than Support Vector Machine algorithm. This model could be adopted for use cooperatively by service providers, financial institutions and their customers thereby minimising fraud by fake transactions alert.

6.0 References

Adepetun and Olayinka (2023), How fake credit alerts from fraudulent customers plunge PoS operators, others into bad debts, TheGuardian,- a Nigerian Daily Newspaper, **https://guardian.ng/news/concerns-as-more-nigerians-fall-prey-to-fake-bank-alerts/**

Albashrawi, M. (2016). Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015. Journal of Data Science, 14(3), 553-569. https://doi.org/10.6339/JDS.201607_14(3).0010

Chu muang, N., Hiranchan, S., Ketcham, M., Yimyam, W., Pramkeaw, P., &Tangwannawit, S. (2020). Developed Credit Card Fraud Detection Alert Systems via Notification of LINE Application. 2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP). doi:10.1109/isai-nlp51646.2020.9376829

De Sá, A. G. C., Pereira, A. C. M., & Pappa, G. L. (2018). A customized classification algorithm for credit card fraud detection. Engineering Applications of Artificial Intelligence, 72, 21–29. doi:10.1016/j.engappai.2018.03.0

Table 3: Performance Metrics for RF & SVM Fake Transactions Alert Detection Models

Metrics	Random Forest Algorithm	Support Vector Machine Algorithm
RMSE	0.02376	0.03911
Accuracy	97.624	96.089
Precision	0.970695	0.968815
Recall	0.999115	0.993838
F1 Score	0.9847	0.976269

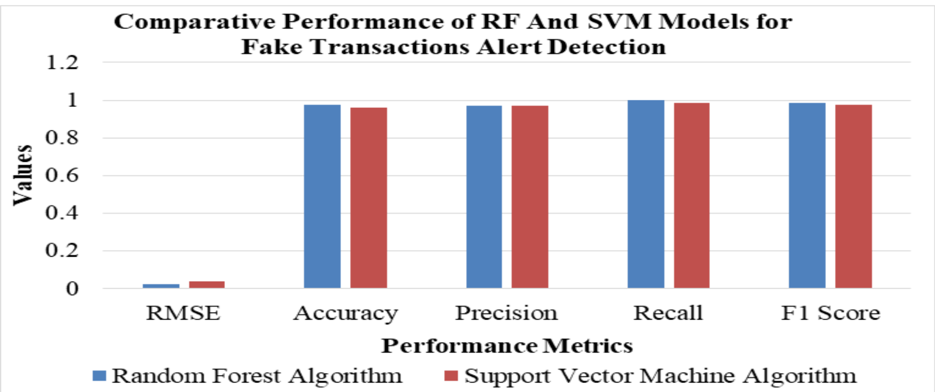

Figure 4: Comparative Performance of RF and SVM Models for Fake Transactions Alert Detection

Dhankhad, S., Mohammed, E., & Far, B. (2018). Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study. 2018 IEEE International Conference on Information Reuse and Integration (IRI). doi:10.1109/iri.2018.00025

Lepoivre, M. R., Avanzini, C. O., Bignon, G., Legendre, L., &Piwele, A. K. (2016). Credit card fraud detection with unsupervised algorithms. Journal of advances in information technology, 7(1).

Omogbolagun (2021), How fake credit alerts from fraudulent customers plunge PoS operators, others into bad debts, PUNCH,- a Nigerian Daily Newspaper, https://punchng.com/how-fake-credit-alerts-from-fraudulent-customers-plunge-pos-operators-others-into-bad-debts/

Osegi, E. N., & Jumbo, E. F. (2021). Comparative analysis of credit card fraud detection in Simulated Annealing trained Artificial Neural Network and Hierarchical Temporal Memory. Machine Learning with Applications, 6, 100080. doi:10.1016/j.mlwa.2021.100080

Roseline, J. F., Naidu, G. B. S. R., Pandi, V. S., alias Rajasree, S. A., & Mageswari, N. (2022). Autonomous credit card fraud detection using machine learning approach☆. Computers and Electrical Engineering, 102, 108132.

Soltani H., N., & Akbari, M. K. (2014). A novel model for credit card fraud detection using Artificial Immune Systems. Applied Soft Computing, 24, 40–49. doi:10.1016/j.asoc.2014.06.042

TheGuardian (2023). Concerns as more Nigerians fall prey to fake bank alerts https://guardian.ng/news/concerns-as-more-nigerians-fall-prey-to-fake-bank-alerts/

Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., &Baesens, B. (2015). APATE : A novel approach for automated credit card transaction fraud detection using network-based extensions. Decision Support Systems, 75, 38–48. doi:10.1016/j.dss.2015.04.013