Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 8, pp 68-76, August-2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.308007
Available Online at https://www.jaiet.com

Intelligent Real-Time Fraud Detection in Financial Institutions

[1]*Nwadike, U. S., [2]Emmah, V.T., [3]Matthias, D.

[1,2,3]Department of Computer Science, Rivers State University, Port-Harcourt, Nigeria
Authors E-mail: [1]nwadike.udochukwu@ust.edu.ng, [2]victor.emmah@ust.edu.ng, [3]Daniel.matthias@ust.edu.ng
*Corresponding Author: Nwadike, U. S., E-mail: nwadike.udochukwu@ust.edu.ng

Abstract: Financial institutions process high transaction volumes through digital payment channels, but conventional rule-based fraud detection systems often fail to adapt to changing fraud patterns and may produce excessive false alerts. This study developed and evaluated an intelligent real-time stacked ensemble prototype for credit card fraud detection in financial institutions. The system combined supervised fraud probability estimates from Random Forest and XGBoost with anomaly score evidence from Isolation Forest, and used Logistic Regression as a meta-classifier to generate the final transaction-level decision. The prototype was implemented in Python using Pandas, NumPy, scikit-learn, imbalanced-learn, XGBoost, Seaborn, Matplotlib, and Joblib. The Credit Card Fraud Detection dataset was used for experimentation, containing 284,807 transactions, 30 input features, and 492 fraudulent cases. The data were split using stratified 80:20 sampling; Time and Amount were standardised using training set statistics only, and SMOTE was applied only to the training partition to prevent test-set leakage. The held-out test set retained the original imbalanced distribution of 56,864 legitimate transactions and 98 fraudulent transactions. Random Forest achieved the highest precision and F1-score, with precision of 0.8454, recall of 0.8367, F1-score of 0.8410, false positive rate of 0.00026, and AUC-ROC of 0.9731. XGBoost achieved the highest base-model recall of 0.8673 and AUC-ROC of 0.9750, but produced 136 false positives. Isolation Forest achieved AUC-ROC of 0.8532 but failed to detect fraudulent transactions as a standalone binary classifier under the implemented configuration. The stacked ensemble achieved accuracy of 0.9993, precision of 0.7778, recall of 0.8571, F1-score of 0.8155, false positive rate of 0.00042, and AUC-ROC of 0.9758. The results show that the stacked ensemble produced a usable prototype-level fraud decision layer by combining supervised and anomaly based evidence, although threshold calibration, explainability, and validation on local institutional data remain necessary before operational deployment.

Keywords: Credit card fraud detection, financial institutions, stacked ensemble, Random Forest, XGBoost, Isolation Forest, SMOTE, real-time fraud detection.

I. INTRODUCTION

Digital transaction channels are now routine in financial services. Banks, merchants, fintech operators, and payment processors support card payments, electronic transfers, internet banking, mobile banking, and automated payment workflows at large scale. These channels increase transaction speed, but they also expose financial institutions to fraud patterns that can change faster than static monitoring controls. In credit card environments, fraudulent activity may involve stolen credentials, card-not-present abuse, account compromise, automated testing of card details, and unauthorised purchases. Traditional fraud detection systems often depend on rule-based screening, account checks, blacklists, fixed transaction thresholds, and manual review. These methods can detect known patterns, but they have limited adaptability when fraudsters change their behaviour. Xiang *et al.* (2024) describe a card-issuer fraud detection process in which account checking, predictive scoring, and manual audit are combined, showing the continuing role of layered controls in credit card fraud monitoring. Such systems may still generate false positives or depend on delayed review when suspicious transactions require human validation. In high-volume financial environments, delayed manual review creates an additional constraint because suspicious transactions must be screened quickly enough to support timely intervention.

Machine learning can learn fraud patterns from transaction data rather than relying only on manually defined rules. Recent studies have examined supervised classifiers, anomaly detection methods, deep learning, graph-based fraud modelling, federated learning, and ensemble learning for financial fraud detection (Ilham *et al.*, 2025; Mienye & Sun, 2023; Ileberi & Sun, 2024). These approaches have reported improved fraud detection results, but they also introduce methodological risks. Severe class imbalance can inflate accuracy, test-set leakage can overstate model performance, and single-model systems may produce unsuitable trade-offs between fraud recall and false positive burden.

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 8, pp 68-76, August-2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.308007
Available Online at https://www.jaiet.com

This study addresses these concerns by developing an intelligent real-time stacked ensemble prototype for transaction-level credit card fraud detection. The system combines Random Forest, XGBoost, and Isolation Forest within a modular Python pipeline. Random Forest and XGBoost provide supervised fraud probabilities, Isolation Forest provides anomaly score evidence, and Logistic Regression combines these base-model outputs into a final stacked ensemble decision. The evaluation uses a held-out test set that preserves the original class imbalance, while SMOTE is restricted to the training partition.

II. RELATED WORKS

Financial fraud detection research has increasingly moved from static rule-based screening toward machine learning, deep learning, anomaly detection, graph analytics, and hybrid architectures.

Prince (2025) designed a multi-stage machine learning pipeline for detecting fraudulent activity in online financial transactions, constructed on a technology stack comprising Apache Kafka, KSQL, and Apache Spark. The Isolation Forest algorithm was employed for unsupervised anomaly detection within the pipeline, and the resulting system achieved an F1-score of 91 per cent alongside a recall rate of 97 per cent. These results confirm that stream processing infrastructure, when paired with capable anomaly detection algorithms, can sustain both the throughput demands and the detection accuracy required for practical fraud detection. The study's principal limitation is the significant computational infrastructure required to operate the pipeline at production scale, which may constrain adoption in resource-limited environments.

Adepetun *et al.* (2022) conducted an empirical study of machine learning adoption across Nigerian financial institutions, examining its impact on operational efficiency and fraud prevention outcomes. Their findings indicated that ML adoption significantly reduced fraudulent payouts and improved the precision of credit risk assessments in the local banking context, providing an early empirical validation of AI-driven fraud prevention within Nigeria. The study is noteworthy for its contextual focus, though the authors acknowledged a limited emphasis on the technical architectures underlying the systems evaluated, which constrains the direct applicability of their findings to system design decisions.

Onyeama (2024) compared the effectiveness of unsupervised anomaly detection techniques for credit card fraud detection in the Nigerian financial sector, contrasting deep Autoencoder networks and Principal Component Analysis implemented using TensorFlow. Deep Autoencoders significantly outperformed PCA in reconstruction-based fraud detection, achieving higher sensitivity to anomalous transaction patterns across the evaluated dataset. The study validates the use of unsupervised deep learning approaches for fraud detection in Nigerian financial data, though the author acknowledged that such models require careful calibration of reconstruction error thresholds to balance detection sensitivity against false alarm rates.

Jemai, Zarrad, and Daud (2024) benchmarked ensemble learning models on the widely used Kaggle credit card fraud dataset, comparing the performance of XGBoost, Random Forest, and CatBoost classifiers across standard evaluation metrics. XGBoost emerged as the highest-performing model, achieving a receiver operating characteristic area under the curve of 0.99. The study provides a useful empirical reference for the selection of ensemble methods in credit card fraud classification, and the benchmark dataset used is one that has informed numerous subsequent research efforts in this field. The training overhead associated with large ensemble models was identified as a practical limitation for deployment in latency-sensitive environments where model retraining must be frequent.

Salami *et al.* (2025) evaluated AI-powered behavioural biometrics as a means of continuous authentication in digital banking environments, using Long Short-Term Memory networks to analyse keystroke dynamics and mouse trajectory data. The model achieved an accuracy of 97.9 per cent in detecting session hijacking, where a legitimate user's authenticated session is taken over by an unauthorised party. The study contributes an important perspective on the role of behavioural signals in comprehensive fraud detection, though real-time processing of high-velocity biometric telemetry was noted as a practical challenge for deployment at scale within financial institutions.

Zakaria *et al.*, (2025) designed a hybrid detection framework that integrated Graph Neural Networks with the Isolation Forest algorithm for real-time transaction fraud detection. The framework achieved a detection latency of under 100 milliseconds, meeting the operational threshold required for practical real-time intervention in payment systems before fraudulent transfers can be completed. The engineering complexity of integrating GNN-based relational modelling with streaming anomaly detection was noted as a challenge for both initial implementation and ongoing system maintenance in production environments.

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 8, pp 68-76, August-2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.308007
Available Online at https://www.jaiet.com

Lai *et al.* (2025) proposed a two-stage self-supervised learning approach for detecting financial statement fraud under conditions of limited and imbalanced labelled data, a setting that is characteristic of real institutional fraud datasets. The first stage employed an auxiliary reconstruction task to build robust data representations without requiring labelled fraud examples, whilst the second stage fine-tuned these representations using a small set of labelled instances. The approach achieved a recall of 0.881, comparing favourably with fully supervised methods that require large quantities of labelled fraud examples for training. The framework's performance was found to depend on the alignment between the auxiliary reconstruction task and the downstream fraud detection objective.

Ahmed, Tisha, and Sweet (2025) proposed a low-latency financial risk assessment framework designed for deployment at the network edge, dynamically routing transactions between deep learning and adaptive regression models based on the computational demands of each transaction context. The framework achieved a 55 per cent reduction in inference latency compared to equivalent cloud-based alternatives, making it particularly relevant for applications where network round-trip time would otherwise compromise the responsiveness of real-time detection. The distributed synchronisation architecture required to maintain consistency between edge and central system components was identified as a source of engineering complexity in operational deployment.

Olaniyi *et al.* (2026) proposed an explainable GNN-based framework for anti-money laundering, using a heterogeneous graph architecture combined with GNNExplainer to provide interpretable fraud evidence at the network level. The study identified relational features as accounting for 51 per cent of the predictive power of the model, confirming that the structural relationships between accounts carry more information than individual transaction features alone for the detection of coordinated financial crime. The real-time generation of graph-level explanations for large-scale transaction networks was identified as a computational challenge that must be addressed before the approach can be deployed effectively in live institutional systems.

Hassan, Ahmad, and Mustapha (2024) proposed an enhanced feature engineering technique specifically designed for credit card fraud detection in Nigeria, combining genetic algorithm-based feature selection with wrapper-based evaluation methods to identify the most informative subset of transaction attributes. The hybrid approach achieved a significant reduction in false positive rates compared to baseline models using standard feature sets, confirming that domain-informed feature construction can substantially improve detection performance in region-specific financial contexts. The approach's dependence on manual domain expertise for the initial construction of candidate features was identified as a limitation that may affect its reproducibility and scalability across different institutional datasets.

III. ANALYSIS OF THE PROPOSED SYSTEM

The proposed intelligent real-time fraud detection system addresses transaction-level credit card fraud detection under conditions of severe class imbalance, static-rule vulnerability, and delayed human review. The central requirement is to classify individual transactions as legitimate or fraudulent using available transaction features while reducing dependence on manually defined rules and improving the traceability of model-based decisions. The system is designed as a prototype for financial institutions and operates on historical transaction data, with the final decision generated from complementary supervised and anomaly-detection evidence.

The proposed system integrates four core capabilities: supervised classification using Random Forest and XGBoost, anomaly detection using Isolation Forest, stacked decision modelling through a Logistic Regression meta-classifier, and class-imbalance handling through SMOTE. Random Forest provides a stable supervised classification baseline; XGBoost contributes high-sensitivity fraud probability estimates; Isolation Forest supplies anomaly-score evidence; and the Logistic Regression meta-classifier combines these outputs into a final transaction-level decision. SMOTE is applied only to the training partition so that the held-out test set retains the original class distribution and provides a more realistic evaluation of fraud-detection performance.

In an operational financial institution, the equivalent real-time inputs may include transaction amount, transaction time, account or card identifiers, merchant category, payment channel, device or terminal identifier, customer and merchant location, authentication outcome, recent transaction frequency, and previous fraud-confirmation history. In the public dataset used in this study, direct identifiers are anonymised into PCA-transformed variables V1 to V28, while Amount and Time remain explicit features. The system therefore analyses the combined feature pattern rather than relying on one manually selected threshold. The main actors are the automated detection engine, the fraud analyst who reviews alerts, and the system

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 8, pp 68-76, August-2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.308007
Available Online at https://www.jaiet.com

administrator who monitors and configures the detection pipeline.

3.1 Proposed System Architecture

The proposed system follows a modular fraud detection workflow, as shown in Figure 1. A transaction dataset enters the data ingestion component, where the schema is validated. The preprocessing component then performs stratified splitting, standardisation of Time and Amount, and SMOTE resampling on the training partition. The detection engine trains Random Forest, XGBoost, and Isolation Forest. The stacked classification component converts the base-model outputs into three meta-features: Random Forest fraud probability, XGBoost fraud probability, and Isolation Forest anomaly score. A Logistic Regression meta-classifier uses these values to produce the final transaction-level fraud probability and binary decision.

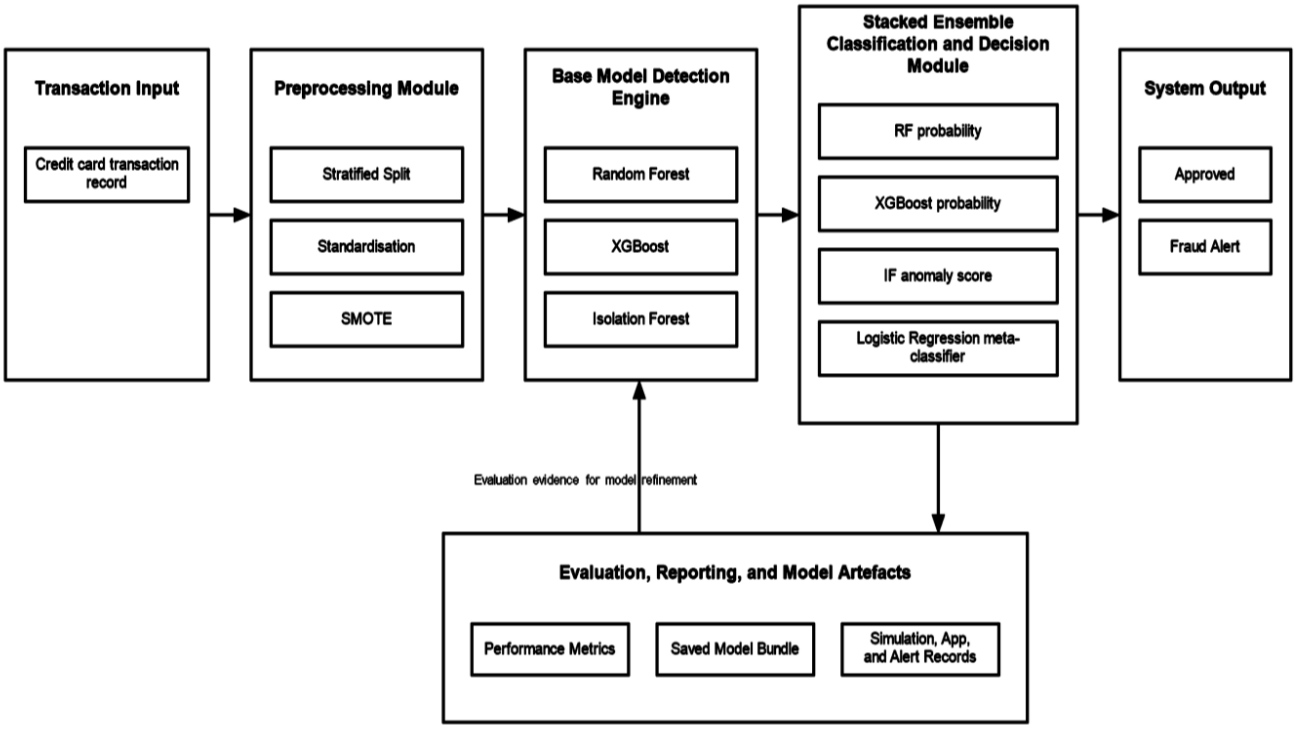

Figure 1: Proposed fraud detection system architecture

The architecture also includes reporting, visualisation, model persistence, transaction simulation, alert review, and a local application interface. These components were included so that the trained model bundle and fitted scaler could be reused to test individual transaction records and produce approval or fraud-alert decisions through the implemented prototype path.

3.2 Dataset and Preprocessing

The experimental dataset was the Credit Card Fraud Detection dataset, which contains anonymised credit card transactions made by European cardholders over a two-day period (Machine Learning Group, 2013). The dataset contains 284,807 transaction records, 30 input features, and one binary target label. The input features are Time, Amount, and 28 PCA-transformed variables labelled V1 to V28. The target variable is Class, where 0 represents a legitimate transaction and 1 represents a fraudulent transaction.

The dataset contained 284,315 legitimate transactions and 492 fraudulent transactions, giving a fraud prevalence of 0.1727%. This severe class imbalance shaped the evaluation strategy. Accuracy was reported, but precision, recall, F1-score, false positive rate, AUC-ROC, and confusion matrix values were treated as more informative evidence for fraud detection performance.

$$x_{\text{scaled}} = \frac{x - \mu_{\text{train}}}{\sigma_{\text{train}}} \quad (1)$$

The dataset was split into training and test partitions using stratified 80:20 sampling with a fixed random seed of 42. Stratification preserved fraud prevalence in both partitions. The training partition contained 227,451 legitimate transactions and

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 8, pp 68-76, August-2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.308007
Available Online at https://www.jaiet.com

394 fraudulent transactions before resampling. The held-out test partition contained 56,864 legitimate transactions and 98 fraudulent transactions. Only the Time and Amount columns were standardised because V1 to V28 were already PCA-transformed in the released dataset.

SMOTE was applied only to the training partition after the train-test split. This increased the fraudulent training records from 394 to 227,451 and balanced the training set at 227,451 legitimate and 227,451 fraudulent records. The test set remained unchanged so that evaluation measured performance under the original imbalanced distribution. Within the resampled training data, an internal 80:20 base/meta split was used to train the stacked ensemble without exposing the held-out test partition to meta-classifier fitting. Table 1 summarises the dataset and preprocessing conditions used in the experiment.

Table 1: Dataset and preprocessing summary

Item	Value
Total transactions	284,807
Feature columns	30
Target column	Class
Legitimate transactions	284,315
Fraudulent transactions	492
Fraud prevalence	0.1727%
Train/test split	80% / 20%, stratified
Training set before SMOTE	227,451 legitimate; 394 fraudulent
Training set after SMOTE	227,451 legitimate; 227,451 fraudulent
Test set	56,864 legitimate; 98 fraudulent

3.3 Model Design

Random Forest was used as a supervised ensemble classifier. It trains multiple decision trees and assigns the final class by aggregating tree-level predictions:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \dots, h_T(x)\} \quad (2)$$

Where $h_t(x)$ is the output of the t-th decision tree for transaction, x and T is the number of trees. In this study, Random Forest used 100 estimators, unrestricted tree depth, and random seed 42.

XGBoost was used as a supervised gradient-boosted classifier. It was selected because boosted tree ensembles have reported high performance on structured tabular fraud datasets and can provide high fraud recall. The implemented configuration used 100 estimators, learning rate of 0.1, maximum tree depth of 6, subsample value of 0.8, binary logistic objective, and random seed 42.

Isolation Forest was used as the anomaly detection component n. It identifies unusual observations by recursively partitioning the feature space. Its anomaly score can be expressed as:

$$s(x,n) = 2^{-\frac{E[h(x)]}{c(n)}} \quad (3)$$

Where $E[h(x)]$ is the average path length required to isolate transaction x, n is the sample size, and $c(n)$ is the expected path length in an unsuccessful binary search tree. The implemented Isolation Forest used 100 estimators, contamination of 0.002, and random seed 42.

The stacked ensemble used Logistic Regression as the meta-classifier. The meta-feature vector for each transaction was:

$$z(x) = [p_{\text{RF}}(x), p_{\text{XGB}}(x), a_{\text{IF}}(x)] \quad (4)$$

where $p_{\text{RF}}(x)$is the Random Forest fraud probability, $p_{\text{XGB}}(x)$ is the XGBoost fraud probability, and $a_{\text{IF}}(x)$ is the Isolation Forest anomaly score. The final stacked fraud probability was estimated as:

$$P(y=1|z) = \frac{1}{1+e^{-(\beta_0+\beta_1 z_1+\beta_2 z_2+\beta_3 z_3)}} \quad (5)$$

where β_0 is the intercept, β_1 to $\beta_3$are learned coefficients, and z_1 to z_3 are the three meta-features.

The implementation used Python 3.10 with Pandas, NumPy, scikit-learn, imbalanced-learn, XGBoost, Seaborn, Matplotlib, and Joblib. The experiment was executed as a single-host software implementation, and the output artefacts included CSV reports, Markdown tables, PNG figures, saved model files, simulation records, alert review records, and application interface outputs.

IV. RESULTS AND DISCUSSION

The models were evaluated on the same unmodified held-out test set. Table 2 presents the comparative performance results, while Figure 2 presents the same metrics graphically. Because the dataset was highly imbalanced, precision, recall, F1-score, false positive rate, AUC-ROC, and confusion matrix counts were used to interpret the results. The values are implemented test-set results from the Python pipeline, not simulated financial losses or live banking measurements.

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 8, pp 68-76, August-2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.308007
Available Online at https://www.jaiet.com

Table 2: Comparative model performance on the held-out test set

Model	Accuracy	Precision	Recall	F1-score	False positive rate	AUC-ROC
Random Forest	0.9995	0.8454	0.8367	0.8410	0.00026	0.9731
XGBoost	0.9974	0.3846	0.8673	0.5329	0.00239	0.9750
Isolation Forest	0.9982	0.0000	0.0000	0.0000	0.00004	0.8532
Stacked Ensemble	0.9993	0.7778	0.8571	0.8155	0.00042	0.9758

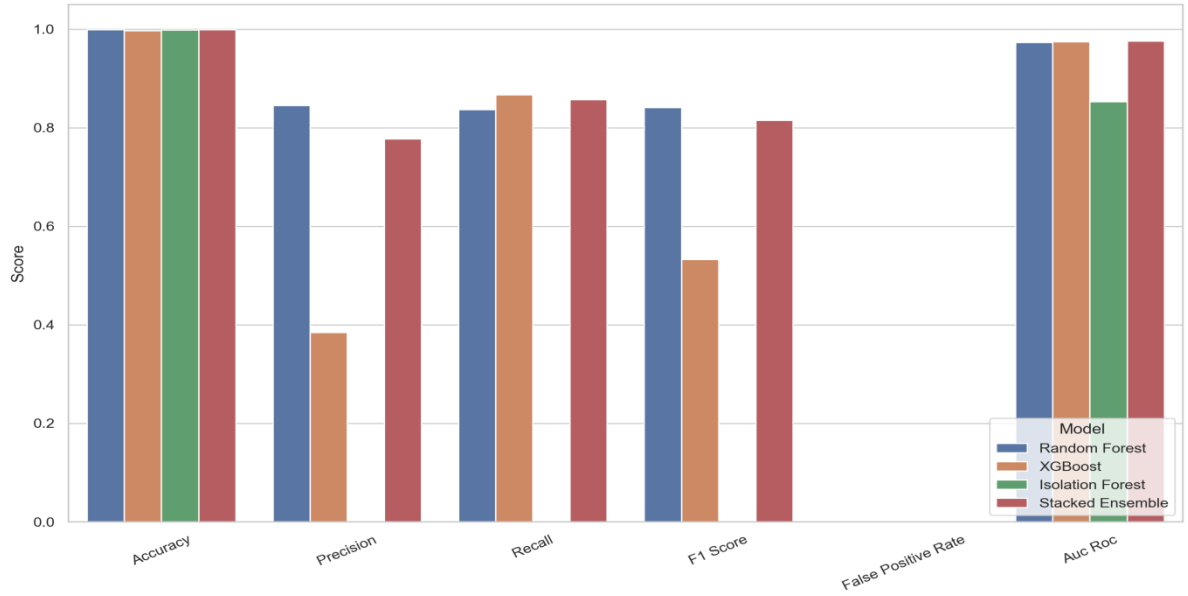

Figure 2: Comparative model performance across evaluation metrics

Random Forest achieved the highest precision and F1-score in the experiment. It correctly detected 82 of the 98 fraudulent transactions in the test set and produced 15 false positives. This result indicates a favourable balance between detecting fraud and limiting unnecessary alerts. The precision result matters operationally because false alerts increase analyst workload and can inconvenience legitimate customers.

XGBoost achieved the highest base-model recall by detecting 85 of the 98 fraudulent transactions. That sensitivity came with 136 false positives, giving a lower precision of 0.3846. This shows the trade-off between fraud sensitivity and alert quality. XGBoost was retained as a base learner because its fraud probability supplied high-recall evidence, but its standalone decision threshold produced a heavier false alert burden than Random Forest and the stacked ensemble.

Isolation Forest achieved AUC-ROC of 0.8532, showing that its anomaly scores contained some discriminatory information. Its binary classification output still failed to identify any fraudulent transaction in the test set. It classified 56,862 legitimate transactions correctly but missed all 98 fraudulent transactions. This result indicates that anomaly detection performance depends on contamination setting, score calibration, and decision threshold. Under the implemented configuration, Isolation Forest was not suitable as a standalone detector.

The stacked ensemble achieved accuracy of 0.9993, precision of 0.7778, recall of 0.8571, F1-score of 0.8155, false positive rate of 0.00042, and AUC-ROC of 0.9758. It detected 84 of the 98 fraudulent transactions and produced 24 false positives. This means the stacked ensemble improved recall over Random Forest while avoiding the much larger false positive count produced by XGBoost. It also produced the highest AUC-ROC among the evaluated models, indicating better threshold-independent ranking in this experiment.

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 8, pp 68-76, August-2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.308007
Available Online at https://www.jaiet.com

Table 3: Confusion matrix results by model

Model	True negative	False positive	False negative	True positive
Random Forest	56,849	15	16	82
XGBoost	56,728	136	13	85
Isolation Forest	56,862	2	98	0
Stacked Ensemble	56,840	24	14	84

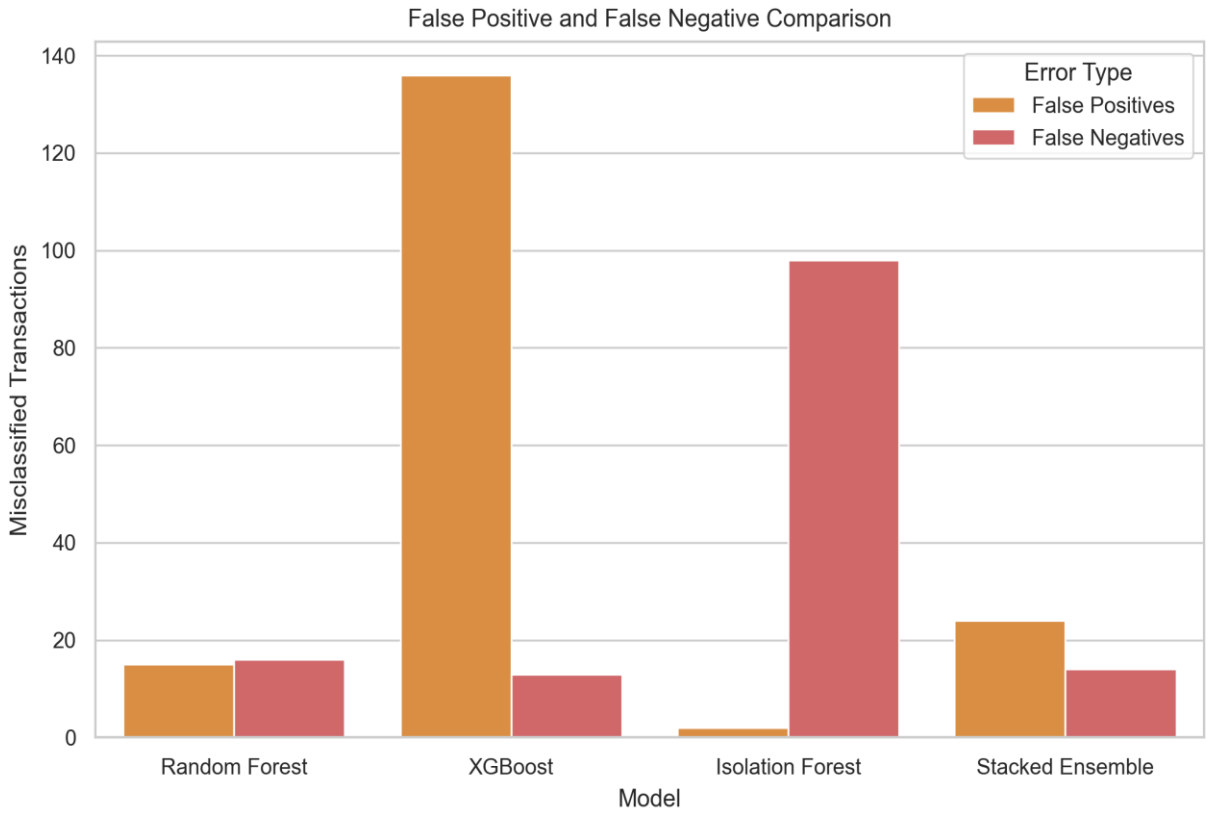

Figure 3: False positive and false negative comparison

The confusion matrix values clarify why accuracy alone is inadequate. Isolation Forest had accuracy of 0.9982 because almost every test transaction was legitimate, but it detected no fraud. XGBoost had higher fraud recall but generated a higher number of false positives. As shown in Figure 3, Random Forest produced a smaller false positive burden than XGBoost, while the stacked ensemble occupied a more balanced error position. Random Forest produced the best F1-score, while the stacked ensemble produced the best integrated system output because it combined the three forms of model evidence into a single decision layer.

The prototype interface provided additional implementation evidence. It loaded the saved model bundle and scaler, transformed selected transaction records, generated base-model predictions and scores, and returned the stacked ensemble decision. The application output included the actual class where available, Random Forest score, XGBoost score, Isolation Forest score, stacked ensemble score, base-model fraud vote count, and final decision. The stacked ensemble generated 108 fraud alerts from the held-out test partition through this trained decision path.

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 8, pp 68-76, August-2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.308007
Available Online at https://www.jaiet.com

Table 4: Sample prototype transaction decision output

Time	Amount	Actual class	Random Forest score	XGBoost score	Isolation score	Stacked score	Base-model votes	Decision
0	149.62	0	0.0000	0.0094	-0.3129	0.0001	0	Approved
0	2.69	0	0.0000	0.0045	-0.3133	0.0001	0	Approved
1	378.66	0	0.0000	0.0204	-0.2558	0.0001	0	Approved
406	0.00	1	0.6000	0.9985	-0.2957	0.9684	2	Fraud Alert
4,462	239.93	1	0.9700	0.9809	-0.2650	0.9995	2	Fraud Alert
6,986	59.00	1	0.8100	0.9980	-0.2391	0.9974	2	Fraud Alert
7,519	1.00	1	0.9900	0.9981	-0.2047	0.9997	2	Fraud Alert

The results are consistent with the literature on ensemble and imbalance-aware fraud detection. Tree-based supervised models produced the best standalone results on the structured tabular data, while the stacked ensemble combined different error profiles. The result does not show that the stacked ensemble is superior on every metric; Random Forest retained the best precision and F1-score, while XGBoost retained the highest base-model recall. The contribution of the stacked ensemble is that it provides one final decision model that improves recall over Random Forest and reduces false positives from 136 under XGBoost to 24. Compared with the existing account-checking and manual-audit process discussed by Xiang et al. (Xiang et al., 2024), the implemented prototype shifts the final transaction decision toward learned model evidence, while still requiring analyst validation and production controls before institutional use.

The study also has limitations. The dataset was public and anonymised rather than collected from a Nigerian financial institution. The PCA-transformed variables prevent direct interpretation of the original transaction attributes. The prototype was evaluated in a controlled software environment rather than a live banking transaction stream. Isolation Forest was not calibrated beyond the implemented contamination setting. The system did not include SHAP, LIME, probability calibration, cost-sensitive threshold optimisation, analyst workflow security, or production monitoring. These limitations mean that the findings should be interpreted as prototype-level evidence, not as proof of operational deployment readiness.

V. CONCLUSION

This paper presented an intelligent real-time stacked ensemble prototype for credit card fraud detection in financial institutions. The system combined Random Forest, XGBoost, and Isolation Forest outputs through a Logistic Regression meta-classifier. The implementation followed a modular Python pipeline covering data ingestion, validation, preprocessing, SMOTE-based imbalance handling, model training, stacked classification, evaluation, reporting, visualisation, model persistence, and transaction-level prototype testing.

The evaluation used the Credit Card Fraud Detection dataset containing 284,807 transactions and 492 fraud cases. SMOTE was applied only to the training partition, while the held-out test set preserved the original imbalanced distribution. Random Forest achieved the highest precision and F1-score, XGBoost achieved the highest base-model recall, and Isolation Forest failed as a standalone binary fraud detector under the implemented configuration.

The stacked ensemble achieved accuracy of 0.9993, precision of 0.7778, recall of 0.8571, F1-score of 0.8155, false positive rate of 0.00042, and AUC-ROC of 0.9758. It detected 84 of 98 fraudulent test transactions and produced 24 false positives. These results show that combining supervised fraud probabilities and anomaly score evidence can support a prototype-level final decision layer for credit card fraud detection. The article makes three contributions. First, it presents a reproducible fraud detection pipeline that links data ingestion, validation, preprocessing, imbalance handling, base-model training, stacked classification, reporting, model persistence, and prototype transaction testing. Second, it compares Random Forest, XGBoost, Isolation Forest, and the stacked ensemble under one controlled experimental procedure. Third, it interprets the final system output using precision, recall, F1-score, false positive rate, AUC-ROC, and confusion matrix evidence rather than relying on accuracy alone.

Future work should validate the system on Nigerian banking transaction data, tune classification thresholds, calibrate predicted probabilities, compare alternative meta-classifiers, improve anomaly score calibration, add explainability methods such as SHAP or LIME, and evaluate the prototype under

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 8, pp 68-76, August-2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.308007
Available Online at https://www.jaiet.com

streaming transaction conditions with audit logging, access control, monitoring, and periodic retraining.

REFERENCES

[1] Xiang, S., Zhu, M., Cheng, D., Li, E., Zhao, R., Ouyang, Y., ... & Zheng, Y. (2023, June). Semi-supervised credit card fraud detection via attribute-driven graph representation. *In Proceedings of the AAAI conference on artificial intelligence* (Vol. 37, No. 12, pp. 14557-14565).

[2] Ilham, M., Winarno, A., Lutfi, M., & Indrasetianingsih, A. (2025). Handling imbalanced fraudulent transaction data using SMOTE-Tomek and random forest: A classification approach. *Best: Journal of Applied Electrical, Science and Technology*, 7(1), 35-38.

[3] Mienye, I. D., & Sun, Y. (2023). A deep learning ensemble with data resampling for credit card fraud detection. *IEEE Access*, 11, 30628-30638.

[4] Ileberi, E., & Sun, Y. (2024). A hybrid deep learning ensemble model for credit card fraud detection. *IEEE Access*, 12, 175829-175838.

[5] Prince, M. (2025). Developing Machine Learning Models for Real-Time Fraud Detection in Online Transactions. *American Journal of IR 4.0 and Beyond (AJIRB)*, 45-51.

[6] Adepetun, A., Odutola, O., & Dopemu, E. M. (2022). Exploring the impact of machine learning on financial decision-making in the Nigerian banking sector. *International Journal of Scientific and Management Research*, 5(12), 212-221.

[7] Onyeama, J. (2024). Credit card fraud detection in the Nigerian financial sector: A comparison of unsupervised TensorFlow-based anomaly detection techniques, autoencoders and PCA algorithm. *arXiv preprint arXiv*:2407.08758.

[8] Jemai, J., Zarrad, A., & Daud, A. (2024). Identifying fraudulent credit card transactions using ensemble learning. *IEEE Access*, 12, 54893-54900.

[9] Salami, I. A., Popoola, A. D., Gbadebo, M. O., Kolo, F. H. O., & Adesokan-Imran, T. O. (2025). AI-powered behavioural biometrics for fraud detection in digital banking: A next-generation approach to financial cybersecurity. *Asian Journal of Research in Computer Science*, 18(4), 473-494.

[10] Zakaria, R. M., Rahman, M. M., Rahman, H., Rafi, M. A., Minto, A., Hossain, M. S., & Saimon, S. I. (2025). Detecting financial fraud in real-time transactions using graph neural networks and anomaly detection techniques. *Journal of Economics, Finance and Accounting Studies*, 7(6), 01-13.

[11] Lai, J., Xie, A., Feng, H., Wang, Y., & Fang, R. (2025, October). Self-supervised learning for financial statement fraud detection with limited and imbalanced data. *In Proceedings of the 4th International Conference on Artificial Intelligence and Intelligent Information Processing* (pp. 919-924).

[12] Ahmed, M. P., Tisha, S. A., & Sweet, M. R. (2025). Real-Time Hybrid Optimization Models for Edge-Based Financial Risk Assessment: Integrating Deep Learning with Adaptive Regression for Low-Latency Decision Making. *Journal of Business and Management Studies*, 7(7), 38-52.

[13] Olaniyi, O. M., Aroh, I. S., Onyii, H., Metibemu, O. C., & Akinola, O. I. (2026). Graph neural networks for multi-layered financial crime network detection: An explainable AI framework for anti-money laundering. *Journal of Engineering Research and Reports*, 28(2), 18-36.

[14] Hassan, H., Ahmad, M. A., & Mustapha, R. (2024). An enhanced feature engineering technique for credit card fraud detection. *FUDMA Journal of Sciences*, 8(4), 8-16.

Citation of this Article:

Nwadike, U. S., Emmah, V.T., & Matthias, D.. (2026). Intelligent Real-Time Fraud Detection in Financial Institutions. *Journal of Artificial Intelligence and Emerging Technologies (JAIET).* 3(8), 68-76. Article DOI: https://doi.org/10.47001/JAIET/2026.308007

***** End of the Article *****