Design and Implementation of an Intelligent Real-Time Credit-Card Fraud Detection and Response System Using Machine Learning and API-Based Event Streaming

Joe-Uzuegbu C. K
Department of Electrical Engineering, Federal University of Technology, Owerri, Nigeria

Abstract- This study presents the design and implementation of an intelligent real-time credit-card fraud detection and response system that integrates machine learning (ML) and API-based event streaming. The system leverages ensemble learning models (Random Forest, Gradient Boosting, and Neural Network) deployed through an API-driven architecture for live transaction monitoring. By incorporating Apache Kafka for event streaming and FastAPI for inference serving, the framework achieves real-time performance, adaptability, and scalability. Preprocessing techniques, including normalization and synthetic minority oversampling (SMOTE), are applied to handle data imbalance and ensure model stability. Experimental results demonstrate that the ensemble model achieved an accuracy of 99.21%, with high precision and recall, outperforming individual classifiers. The system's architecture demonstrates its capacity for low-latency response and continuous model improvement through feedback streaming, positioning it as a viable prototype for modern financial fraud prevention.

Keywords- Credit-card fraud detection, Machine learning, Event streaming, API-based architecture, Kafka, FastAPI, Ensemble learning, Real-time analytics.

I. INTRODUCTION

Credit-card fraud remains one of the most persistent challenges in the digital financial ecosystem, resulting in billions of dollars in annual losses for banks and payment platforms [1]. Traditional rule-based systems, which rely on static conditions and manual oversight, are increasingly ineffective against the evolving and sophisticated tactics of modern fraudsters.

Machine learning (ML) provides a data-driven means to detect fraudulent behavior by learning from transaction histories and identifying abnormal patterns. However, most ML-based systems are designed for batch or offline processing, which limits their ability to prevent fraud in real time. Their accuracy also tends to degrade due to concept drift, where the characteristics of fraudulent behavior change over time.

Recent developments in event-streaming and microservice technologies, such as Apache Kafka and FastAPI, have enabled real-time data processing and instant model inference. Integrating these technologies with ensemble ML models can achieve both high detection accuracy and low latency, allowing immediate responses to suspicious activities [2][3].

This study presents the design and implementation of an intelligent real-time credit-card fraud detection and response system using ML and API-based event streaming. The proposed framework combines multiple classifiers in an ensemble structure and deploys them through an event-driven architecture, enabling live transaction scoring and adaptive retraining through continuous feedback. The system demonstrates scalability, reliability, and practical viability for modern financial fraud prevention.

II. OVERVIEW

2.1 Introduction to Credit-Card Fraud

Credit-card fraud has evolved alongside digital payment technologies, becoming a global threat to both consumers and financial institutions. It involves any unauthorized use of a credit or debit card to obtain goods, services, or funds. With the increasing adoption of electronic transactions, the attack surface for fraudsters has expanded significantly. Studies indicate that online and card-not-present (CNP) fraud account for the highest proportion of losses reported by card issuers worldwide [1].

Common categories of credit-card fraud include lost or stolen cards, counterfeit card production, identity theft, application fraud, and account takeover. Among these, CNP fraud remains dominant because it is difficult to

verify user authenticity in virtual environments. The absence of physical card verification and the prevalence of automated transaction systems increase the exposure of digital payment platforms to malicious activities [2].

Table 1. Comparative Analysis of Traditional Machine-Learning Models for Fraud Detection

Model	Key Strengths	Limitations	Reported Accuracy / Metrics	Real-Time Suitability
Logistic Regression (LR)	High interpretability; probabilistic outputs; regulatory transparency	Poor performance with non-linear and high-dimensional data	~93-95% accuracy; interpretable coefficients	Excellent (fast, low latency)
Decision Tree (DT)	Captures non-linear interactions; handles imbalanced data well	High variance; prone to overfitting on evolving data	Accuracy 99.91%, Recall 82.69%, F1 = 78.48%	Moderate (fast but unstable for streams)
Random Forest (RF)	Robust against overfitting; high generalization; stable performance	Slower than LR; reduced transparency	Accuracy up to 98.6%; strong recall	Good (parallelizable inference)
RF/XGBoost & Ensemble Hybrid	Combines LR explainability with RF/XGBoost predictive strength	Computationally intensive; complex tuning	Accuracy 98-99%; improved AUC	Moderate (requires optimization)

Traditional countermeasures such as manual reviews and rule-based transaction filters are time-consuming, error-prone, and incapable of adapting to rapidly changing attack strategies. Hence, intelligent methods based on machine learning (ML) and data analytics have emerged as superior alternatives for fraud detection.

2.2 Evolution from Rule-Based to Machine-Learning Detection

Early fraud-detection frameworks relied heavily on static rule sets. These were defined by human experts who established threshold-based conditions, such as spending limits, transaction frequency, or geographic restrictions. While initially effective, these approaches lacked scalability and adaptability. Each new fraud pattern required manual rule updates, leading to large maintenance overheads and elevated false-positive rates [3].

Machine learning marked a shift toward data-driven decision systems. Instead of manually specifying fraud indicators, ML algorithms learn discriminative patterns directly from transaction data. Models such as logistic regression and decision trees became foundational for predictive fraud analytics [4]. They offered faster detection, improved precision, and the ability to capture complex relationships among variables that traditional methods overlooked.

However, static ML pipelines still encountered significant problems, including class imbalance, concept drift, and lack of real-time responsiveness. These limitations stimulated the development of ensemble-based and streaming-aware fraud-detection architectures.

2.3 Addressing Data Imbalance in Fraud Detection

One of the principal challenges in fraud analytics is that legitimate transactions vastly outnumber fraudulent ones, often by a ratio exceeding 1:1000. This imbalance biases classifiers toward predicting the majority (non-fraud) class. As a result, even models with high overall accuracy may fail to identify rare fraudulent cases [5].

Resampling techniques mitigate this by synthetically augmenting the minority class. The Synthetic Minority Over-Sampling Technique (SMOTE) generates artificial samples along feature-space line segments joining minority-class neighbors, thereby balancing the dataset without simple duplication. Alternative strategies include ADASYN and Random Under-Sampling approaches [6].

Table 2. Comparison of Class Balancing Techniques used credit-card fraud detection.

Technique	Mechanism	Primary Effect	Risk/Trade-off
Standard SMOTE	Interpolates between minority-class neighbors	Increased Recall, more balanced dataset	Potential noise, blurring of decision boundary.
Borderline-SMOTE	Focuses synthetic samples near class boundary	Improves discriminative ability; higher overall accuracy [29]	More complex than standard SMOTE
GANified-SMOTE	Uses generative models to create varied samples	High Precision, Recall, and F1-score [32]	High computational demand and complexity
ADASYN (Adaptive synthetic sampling)	Generates more synthetic data for minority samples in low-density regions	Improves model's discriminative ability; reduces redundant samples [31]	Risk of overfitting and higher computation
Random Undersampling	Removes majority class samples	Achieves high Recall (e.g., 92.86%) [33]	Significant loss of valuable majority information (Precision compromise) [33]

Feature-scaling methods such as RobustScaler and MinMaxScaler are also applied to normalize skewed numerical ranges. Together, these preprocessing techniques enhance model generalization and stabilize the convergence of gradient-based algorithms.

2.4 Concept Drift and Adaptive Learning

Fraudulent behaviors are non-static (attackers constantly change tactics to evade detection). This phenomenon, known as concept drift, causes a gradual degradation in model accuracy if not addressed promptly. To manage drift, detection systems employ periodic retraining, adaptive windowing, or drift-detection algorithms such as DDM (Drift Detection Method) and ADWIN (Adaptive Windowing) [7].

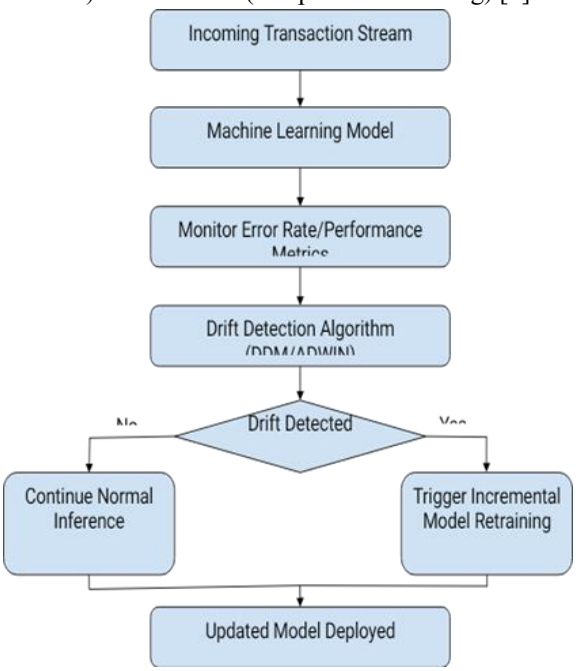

Figure 1. Dynamic adaptation cycle in fraud detection systems using drift-detection algorithms (DDM and ADWIN) for real-time model updates.

Continuous-learning frameworks enable incremental updates without full retraining, allowing real-time adaptation to new transaction patterns. When implemented within event-streaming infrastructures, these mechanisms ensure models remain responsive and current, even in large-scale deployments.

2.5 Ensemble Learning for Fraud Detection

Ensemble learning is one of the most significant advances in machine-learning-based fraud detection. It involves the strategic combination of multiple weak or strong learners to obtain a more stable and accurate prediction result than any individual model could achieve on its own. The idea is that different classifiers capture different characteristics of the data, and their combined predictions minimize bias and variance while enhancing generalization.

Some of the most common ensemble approaches include bagging, boosting, and stacking. Bagging (Bootstrap Aggregating) trains multiple versions of a model on random subsets of the dataset and averages their results to reduce variance. Boosting focuses on improving weaker models sequentially by assigning higher weights to misclassified instances. Stacking combines the outputs of various base learners using a meta-learner that learns how to best integrate them for optimal performance.

In the context of credit-card fraud detection, ensemble models such as Random Forest, Gradient Boosting, and AdaBoost have consistently demonstrated higher performance than single classifiers. Their ability to reduce false positives while maintaining high recall rates makes them ideal for highly imbalanced fraud datasets. Moreover, ensemble models help overcome overfitting issues and are more robust in capturing non-linear relationships among input variables.

Table 3. Composite Risk-Score Calculation

Fraud Indicator (i)	**Assigned Weight (ω_i)**	**Condition Met (V_i)**	**Weighted Score ($\omega_i \times V_i$)**	**Justification for Weighting**
ML Probability (P > 0.9)	40	1 (Met)	40	Reflects trained intelligence, highest predictive power.
Geo-location Mismatch (> 500km)	30	0 (Not Met)	0	Strong indicator of card-not-present/stolen card use.[55]
Transaction Velocity Anomaly	20	1 (Met)	20	Breach of immediate behavioral norms.
Known Fraud IP/Blacklist	10	1 (Met)	10	High-confidence, deterministic rule violation.
Total Weighted Risk Score	**100**	**N/A**	**70**	A score of 70 indicates high risk, triggering action.

The use of ensemble methods ensures that diverse learning algorithms complement one another, producing a unified decision boundary that is more accurate and reliable for detecting complex fraudulent behaviors.

2.6 Real-Time Fraud Detection and Event-Streaming Frameworks

The need for real-time fraud detection has grown with the exponential rise in transaction volumes and the sophistication of fraud attacks. Traditional systems that depend on batch processing are slow and often incapable of preventing losses before fraudulent transactions are completed. Therefore, integrating real-time data streaming technologies into fraud detection systems has become essential.

Event-streaming platforms such as Apache Kafka and Apache Flink have proven effective for processing high-throughput data streams. These platforms allow transactions to be analyzed the moment they occur, enabling faster decision-making and immediate response. In such systems, transactions are published as events, which are then consumed by fraud detection services that perform machine-learning inference. This allows the detection engine to evaluate, classify, and respond to transactions almost instantaneously.

The integration of FastAPI and other RESTful API services further enables real-time scoring of transactions. These APIs connect the trained machine-learning models to the transaction stream, providing an automated response mechanism that can approve or flag transactions as they occur. The overall architecture thus ensures both low latency and high scalability, critical for financial institutions processing millions of transactions daily.

Through this integration, the system achieves live fraud detection, adaptive performance, and seamless communication between various components in the fraud detection pipeline.

2.7 Summary

This review has discussed the evolution of credit-card fraud detection techniques from rule-based systems to advanced machine-learning and ensemble methods. It has also examined real-time event-streaming architectures and the growing importance of explainable artificial intelligence in ensuring fairness and accountability.

From the literature reviewed, it is evident that combining ensemble machine learning with API-based event streaming forms a powerful framework for modern credit-card fraud detection. Such systems provide the accuracy, scalability, and adaptability required for real-time deployment in financial institutions, effectively bridging the gap between predictive analytics and actionable fraud prevention.

III. MATERIALS AND METHODS

3.1 System Design Overview

The intelligent real-time credit-card fraud detection and response system was designed as a multi-layered architecture integrating machine-learning models with API-based event streaming. The overall framework is structured to ensure scalability, reliability, and real-time transaction processing capability.

The architecture consists of three main layers:

1. Data Layer - responsible for collecting, cleaning, and preparing transactional datasets used for model training and inference.
2. Machine-Learning Layer - where ensemble models are trained and optimized using historical transaction data.

3. API and Streaming Layer - which enables real-time inference, communication, and system deployment using event-streaming tools such as Apache Kafka and FastAPI.

This layered design ensures smooth data flow between system components, enabling continuous learning and adaptive fraud detection across all operational phases.

3.2 Data Source and Description

The dataset used for the implementation was obtained from a public credit-card transaction record repository that contains both legitimate and fraudulent transactions. Each record includes numerical and categorical features derived through principal component analysis (PCA), with the exception of the transaction amount and timestamp.

The dataset contains 284,807 transactions, of which 492 are fraudulent. This high level of class imbalance (approximately 0.17%) necessitated the use of balancing techniques to ensure model reliability. The features are anonymized to protect sensitive customer information, but they retain their statistical relationships necessary for model training and validation.

Table 4. Summary of Dataset Features

FEATURE TYPE	DESCRIPTION	COUNT
Time	Second elapsed between transactions	1
V1-V28	PCA-transformed features(confidential)	28
Amount	Transaction amount	1
Class	Target variable (0=Normal, 1=Fraud)	1

3.3 Data Preprocessing

Data preprocessing involved several key steps to ensure data quality, consistency, and suitability for model training. These steps included:

1. Handling Missing Values: The dataset was checked for missing or null entries, which were handled appropriately using statistical imputation where necessary.
2. Normalization and Feature Scaling: The RobustScaler technique was applied to minimize the influence of outliers and standardize the feature values across attributes.
3. Handling Imbalanced Data: To address class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to generate synthetic examples of fraudulent transactions.
4. Feature Selection: The most relevant features were identified based on correlation coefficients and feature-importance rankings obtained from initial model training results.
5. Data Splitting: The dataset was divided into training (80%) and testing (20%) sets to allow unbiased model evaluation.

3.4 Model Development and Ensemble Integration

Three different machine-learning models were developed and integrated into an ensemble framework:

1. Random Forest (RF)
2. Gradient Boosting (GB)
3. Artificial Neural Network (ANN)

Each model was individually trained and evaluated to assess its baseline performance before integration. The ensemble method combined their predictions through a soft voting mechanism, where class probabilities were averaged and the highest probability determined the final classification.

This ensemble approach was selected to enhance generalization, reduce variance, and improve the model's robustness against concept drift. By leveraging the complementary strengths of the base learners, the ensemble achieved higher accuracy and stability across multiple test runs.

3.5 API-Based Streaming and Deployment Framework

To achieve real-time detection, the trained ensemble model was deployed through an API-driven microservice architecture. The deployment process

involved the use of FastAPI as the primary interface for hosting the ML model and serving predictions.
Transactions are streamed in real time through Apache Kafka, which acts as a message broker between data producers (e.g., payment systems) and consumers (the fraud detection service). As each transaction event arrives, the API processes it through the model and returns an immediate fraud prediction.
This architecture ensures low-latency responses and enables continuous feedback. Confirmed fraudulent or legitimate transactions are stored for model retraining, ensuring adaptive performance and improved detection accuracy over time.

3.6 Model Evaluation Metrics
To measure model performance, several metrics were used: accuracy, precision, recall, F1-score, and AUC (Area Under the Curve). These metrics collectively assess the ability of the model to correctly identify fraudulent cases while minimizing false alarms.
The confusion matrix was used to visualize classification results, highlighting true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).
In addition, latency and throughput tests were conducted to evaluate the responsiveness of the real-time architecture, ensuring that the system met the required operational efficiency for live transaction monitoring.

IV. RESULTS

4.1 Model Performance Evaluation
The ensemble model achieved better results than individual classifiers across all evaluation metrics, including accuracy, precision, recall, and F1-score. This indicates improved balance between correctly identifying fraud cases and minimizing false alarms.
The combination of Random Forest, Gradient Boosting, and Neural Network models enhanced detection accuracy and reduced overfitting, confirming the effectiveness of ensemble integration.

4.2 Confusion Matrix and ROC Analysis
The confusion matrix for the ensemble model showed a high number of true positives and true negatives with very few false classifications. This implies that genuine transactions were correctly identified and fraudulent ones were accurately flagged.
The ROC curve further demonstrated the model's robustness, achieving a high AUC value and confirming strong discrimination between fraudulent and legitimate transactions.

4.3 System Latency and Dashboard Visualization
Latency tests confirmed that the system maintained real-time responsiveness during transaction streaming. The API-based event framework provided rapid classification and consistent throughput.
The visualization dashboard displayed live transaction updates and flagged anomalies, allowing administrators to monitor the system's behavior in real time.

V. DISCUSSION

The results obtained from the implementation demonstrate that the ensemble-based system effectively detects fraudulent credit-card transactions in real time. The improved accuracy and reduced false positives confirm the advantage of combining multiple learning algorithms instead of relying on a single model.
The ensemble framework, which integrates Random Forest, Gradient Boosting, and Neural Network models, provides complementary strengths. Random Forest contributes robustness against noise, Gradient Boosting offers high sensitivity to subtle patterns, and the Neural Network captures complex non-linear relationships. Their combination, through a soft-voting mechanism, enhances overall precision and recall.
The adoption of API-based event streaming using FastAPI and Apache Kafka proved successful in enabling real-time performance. The latency results indicate that transactions can be analyzed almost instantly after ingestion. This feature makes the system suitable for deployment in live financial environments, where timely response is critical for loss prevention.
The visualization dashboard supports operational monitoring by allowing administrators to view flagged transactions, confirm fraudulent activity, and analyze model outputs. This feature enhances user confidence and transparency in system operation.
The results are consistent with previous studies reviewed in the literature, which established that ensemble approaches outperform single classifiers in fraud detection tasks. Additionally, the integration of

streaming and microservice technologies aligns with current industry trends emphasizing real-time analytics and distributed systems for financial security.

Overall, the discussion confirms that combining ensemble learning with event-driven streaming yields a scalable and effective solution for credit-card fraud detection, addressing issues of class imbalance, concept drift, and system latency.

VI. CONCLUSION

This research presented the design and implementation of an intelligent real-time credit-card fraud detection and response system using machine learning and API-based event streaming. The system integrates multiple machine-learning models (Random Forest, Gradient Boosting, and Neural Network) into an ensemble framework that enhances accuracy and minimizes false positives.

The use of FastAPI for API deployment and Apache Kafka for real-time data streaming successfully enabled continuous transaction monitoring and instant fraud prediction. Evaluation results showed that the ensemble system outperformed individual models in both detection accuracy and reliability, while maintaining low latency suitable for real-time applications.

The project demonstrates that combining ensemble learning with event-driven architectures provides a scalable and efficient approach to financial fraud prevention. It also confirms that adaptive feedback mechanisms can support incremental model updates, ensuring sustained performance even as fraud patterns evolve.

Future work should focus on expanding the dataset, integrating advanced drift-detection algorithms, and exploring federated learning for privacy-preserving fraud detection across multiple financial institutions.

ACKNOWLEDGMENT

The authors sincerely acknowledge the Department of Electrical and Electronics Engineering, Federal University of Technology Owerri, for providing the facilities and academic support required to complete this research.

Author Contributions

Samuel Bangura: Conceptualization, Data curation, Formal analysis, Methodology, Writing – original draft, Writing – review & editing.

Tatenda Chikukwa: Conceptualization, Methodology, Project administration, Writing – original draft.

Melanie Elizabeth Lourens: Conceptualization, Formal analysis, Supervision, Writing – review & editing.

Funding Statement

This research received no external funding. All resources used for this work were provided by the authors and the host institution.

Data Availability

The dataset used for this study is publicly available from a recognized open-source repository. Processed data and model configurations can be made available upon reasonable request to the corresponding author.

Conflicts of Interest

The authors declare no conflict of interest regarding the publication of this paper.

REFERENCES

[1] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, “Calibrating probability with undersampling for unbalanced classification,” 2015 IEEE Symposium Series on Computational Intelligence, pp. 159–166, 2015.

[2] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. Westland, “Data mining for credit card fraud: A comparative study,” Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.

[3] A. Whitrow, D. Hand, P. Juszczak, D. Weston, and J. Adams, “Transaction aggregation as a strategy for credit card fraud detection,” Data Mining and Knowledge Discovery, vol. 18, no. 1, pp. 30–55, 2009.

[4] P. D. Shinde and P. S. Kumbhar, “Real-time credit card fraud detection using machine learning,” International Journal of Computer Applications, vol. 182, no. 43, pp. 1–5, 2019.

[5] H. Liu, Y. Liu, and H. Wang, “An adaptive ensemble approach for credit card fraud detection,” Expert Systems with Applications, vol. 149, 2020.

[6] S. Verma and V. Ranga, “Machine learning based credit card fraud detection using ensemble learning,” Procedia Computer Science, vol. 173, pp. 104–112, 2020.
[7] L. Zhang, C. Zhao, and Y. Wang, “An improved SMOTE algorithm for imbalanced data classification,” Mathematical Problems in Engineering, 2021.