International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

A Real-Time Bank Transaction Fraud Detection Framework Using Apache Kafka and Machine Learning

Sudheer Kamana [*1]
M.Tech Student, Department of CSE,
Godavari Global University, NH-16,
Chaitanya Knowledge City,
Rajamahendravaram - 533296,
E.G.Dist, Andhra Pradesh.
sudheerk@ggu.edu.in

D.Phani Kumar [*2]
Asst Prof,Department of CSE,
Godavari Global University, NH-16,
Chaitanya Knowledge City,
Rajamahendravaram - 533296,
E.G.Dist, Andhra Pradesh.
phanikumar@ggu.edu.in

Abstract— **The enormous rise in digital banking, online financial services has increased the size, speed, and depth of financial transactions significantly, thus intensifying the vulnerability to predatory actions. Conventional fraud detection patterns based on rule-based logic and batch processing can often not generate timely and accurate fraud detection in high-volume transactional settings. To overcome these shortcomings, this paper will introduce a real-time bank transaction fraud detection system that combines Apache Kafka event streaming distributed over various banking flagellation channels to a set of machine learning (ML) fraud detection algorithms applied in real time by Kafka consumers. Transaction data is preprocessed, processed into features and normalized and then executed through supervised ML models that have been trained on past transaction datasets. The framework uses algorithms like Random Forest and are effective at capturing the complex trends in transactions and differentiating legitimate and fraudulent transactions with low latency. Transaction fraud in quickly, generating real-time notifications and allowing immediate preventative measures; The system is based on a modular and flexible architecture that ensures fault tolerance, high throughput, and component independence, which is adapted to real-world banking applications. The system has been able to identify fraud in near real time with experimental implementation and execution results indicating that it is able to perform reliably. In sum, the suggested solution increases fraud prevention, minimizes financial loss, and improves security and customer confidence in modern online banking platforms.**

Keywords— **Conventional Fraud Detection, Apache Kafka, Machine Learning (ML), Random Forest, Supervised Models, Digital Banking.**

I. INTRODUCTION

The unprecedented growth of digital banking, online payment services, and cashless financial services has resulted in a staggering number of banking operations, and the speed at which those operations occur. Although these innovations provide greater customer convenience and operational efficiency, they also present high-order and developing fraud risks to financial institutions. Live transaction fraud is a crucial issue to banks because its timely detection can cause them to lose a lot of money, regulatory fines, and customer confidence. Therefore, the need to create intelligent, scalable, and real time fraud detection systems has become part of contemporary banking settings. Nearly all traditional fraud detection systems rely on fixed rule-based systems and offline batch processing. Whereas these techniques have the ability to detect familiar fraud patterns, they fail to respond quickly to new behavioral fraud and cannot adapt smoothly to new transaction high-throughput environments. Additionally, batch-based analytics do not provide the means to process real-time transaction streams and, as such, cannot be used to detect fraud in real time. Such restrictions point to the necessity of real-time data-processing systems and smart learning algorithms that could process transactional behaviour in real-time.

New innovations in streaming big data engines and machine learning methods could provide solutions to these problems. Apache Kafka, a distributed event-streaming system, provides reliable, low-latency ingestion and processing of feeds of massive transaction volumes, whereas machine learning models can train on such patterns and difference between legitimate and legitimate funds movements. These technologies are easily integrated to provide scalable, fault-tolerant, and adaptive fraud detection. This article presents the idea of a real-time bank transaction fraud detection model which uses both Apache Kafka and trained machine learning models. The system takes live transaction data and does real-time preprocessing and feature extraction using minimal latency and then classifies transactions without fraud or deception. The proposed framework can improve banking security by facilitating scalable implementation and allowing the detection of fraud as soon as possible. The rest of this paper outlines related work, system architecture and methodology, experimental results, and conclusions.

Scope

The project scope is aimed at developing and deploying a real-time fraud detection system in the digital banking setting with Apache Kafka and Machine Learning in place. The system targets processing high volume banking transactions created via a wide variety of digital platforms including online banking platforms, ATMs, mobile apps, and payment gateways. The framework addresses real-time data ingestion, preprocessing, feature engineering, model training, and fraud classification with low latency.The proposed framework is restricted by supervised machine learning-oriented fraud detection based on the structured transaction data. It lays a focus on scalable streaming architecture, mistakes resilience and grant-wanted deployment applicable in the context of real world bank infrastructures. Performance metrics included in the project are accuracy, precision, recall, and F1-score. Yet, it uses no sophisticated graph-based fraud analytics or deep sequential modeling and

no full-scale cloud deployment yet in its implementation. Its design is scalable and can be expanded to the enterprise-scale (financial) institutions that require real-time fraud detection and intelligent transaction risk evaluation.

Objectives

The main goal of this project is to design a scalable and low-latency real-time bank transaction fraud detection system through the combination of Apache Kafka with machine learning models trained in supervised mode. The system will support a distributed streaming architecture that can achieve real-time ingestion and processing of banking operations on digital channels. It contains the preprocessing operations (normalization of data, categorical encoding, and feature engineering) to establish transaction data in a way that is accessible to machine learning analysis. Supervised models such as Decision Tree and Random Forest classifier are taught and tested to distinguish between genuine and fraudulent transactions. The framework is aimed at reducing the latency in fraud detection through real-time inference of streaming data and generating immediate warnings to facilitate prompt preventative measures within banking systems. The architecture is also modular and scalable, providing fault tolerance and high throughput, as well as flexibility to emerging patterns of fraud. With these combined goals, the project will focus on facilitating the security of digital banking, minimizing financial resources wastage, and increasing customer confidence in the current financial services offered over the internet.

II. LITERATURE SURVEY

The section offers an overview of the major contributions to the research field in the area of banking and financial transaction fraud detection, with a specific focus on machine learning, deep learning, and real-time data streaming architectures. The studies reviewed are on supervised and unsupervised fraud detection models, how to deal with highly imbalanced datasets, behavior-based anomaly detection, and how to build scalable architectures to process high-velocity streams of transactions. Recent developments bring an increasing number of distributed streaming platforms such as Apache Kafka and Spark in combination with intelligent analytics to low-latency, real-time fraud detection. These pieces collectively offer both the theoretical and practical bases of designing scalable, intelligent and deployable fraud detection systems in contemporary digital banking contexts and in turn serve as inspirational to the proposed Kafka-based real time ML framework.

The study by Wu and Lee (2025) [1] brought forward a deep learning-based credit card fraud detection model on advanced neural structures and claimed higher accuracy as compared to the traditional machine learning approaches. Despite its success, the study concentrated on the offline data and failed to tackle the challenges of real-time streaming deployment. Jabeen et al. (2025) [2] presented a hybridized deep learning architecture integrating multiple classifiers to improve the existence of fraud on highly imbalanced transaction data; the framework was not presented with real-time transaction pipelines.In [3], Baishlan et al. described a systematic review of methods to detect credit card fraud using machine learning, pointing to the superiority of ensemble and deep learning techniques in comparison with classical classifiers. Although it was largely analyzed, there was a need to study scalable implementations in real-time, which was not investigated. A comparative study of real-time-data streaming platforms to detect fraud is done by Daksa and Suryadevara (2025) [4], which showed the benefits of Kafka-based software, but the research included no intelligent ML-driven decision engine.Motie et al. (2024) [5] used graph neural networks (GNNs) to model intricate connections among transactions and users, and it was found to be accurate in terms of detecting fraud. Nevertheless, graph models are not applicable in time-sensitive banking settings due to their own computational complexity. In the article by Cherif et al. [6] [6], an encoder-decoder graph neural network was developed to detect credit card fraud, and this model better recognized coordinated fraud attacks but was resource-intensive due to large computational requirements.Yu et al. (2024) [7] investigated transformer-based detectors on fraud and demonstrated better results in teaching longterm transactional relationships. However, the research was restricted to offline experimentation only and it did not represent real-time streaming limitations. The paper by Mienye et al. (2024) [8] presented a hybrid deep learning algorithm, enhanced with generative methods, to tighten the imbalance between data, better recalls, yet nothing was said about its RL implementation in streaming systems.

Zhao et al. (2023) [9] conducted a thorough review on various deep learning methods for credit card fraud detection, focusing on LSTM and attention-based models for sequential transaction analysis. While effective, the authors noted challenges with interpretability and real-time deployment. Malik and Singh (2022) [10] developed the hybrid machine learning design that integrates feature engineering and ensemble classifiers, reaching a higher accuracy but the system is batch based.Scalability The scalability of a streaming framework built on Spark was presented by Carcillo et al. (2018) [11], who created SCARFF, a credit card fraud detector based on Spark, proving the viability of real-time fraud analytics. The framework, on the other hand, was complex in infrastructure and costly in its computation. Transaction aggregation strategies to detect fraud were proposed earlier by Whitrow et al. (2009) [12], which later were ineffective, because they were based on fixed aggregation rules (Whitrow et al., 2009).Afroz et al. (2020) [13] introduced a behavior-based real-time fraud detection solution based on streaming analytics and indicated reduced detection latency. The system was not modularly scalable, even though it got promising results. Gupta and Singh (2021) [14] proposed a Kafka-based fraud analytics architecture using machine learning models and emphasized that Kafka is appropriate to use with financial streaming systems, but there were no reported evaluation metrics.On the whole, the studies find that machine learning and deep learning models are far superior compared to traditional rule-based systems in fraud detection. Recent studies point clearly at an orientation toward real-time and streaming elements of a fraud detection architecture although a number of practical issues have not been addressed.

Problem Gaps Identified

A critical review of the available literature highlights that there are a number of gaps in the research on

real-time bank transaction fraud detection. The foundational studies are typically oriented on offline and batch-based analysis and are thus restrictive in application in the real-life banking system where real-time response is needed. Numerous methods are based on single-source transaction data and cannot incorporate continuous streaming data through distributed banking channels. Although deep learning and ensemble models are proven to be accurate, real-time deployment, scalability, and latency constraints are overlooked. Moreover, the problems of class diazage, interpretation of models, and modularity of systems are not appropriately tackled. Little work has offered end-to-end architectures implementing scalable, fault-tolerant, and low-latency fraud detection using Apache Kafka and machine learning models. Such limitations exemplify the necessity of a dynamic, short response and deployable, fraud detection system, which is addressed through the proposed system.

III. PROPOSED METHODOLOGY

The framework proposed is a Real-Time Bank Transaction Fraud Detection System that merges an Apache Kafka-based streaming infrastructure with the smart data saying of currency based on Machine Learning (ML) models. The architecture is structured such that it handles transaction streams with high propagation speed, derives significant behavioral characteristics and categorizes transactions in real time as legal or illegal. Structured transaction data is enhanced into optimally represented features after preprocessing. Various machine learning methods including Decision Tree (DT), Random Forest (RF), and Neural Networks (NN) are used to acquire patterns of transaction behavior. The trained models are tested with the help of conventional performance measures, and the top-performing model is implemented into a Flask-based web app to monitor fraud in real-time and create alerts.

1) Data Preprocessing and Integration

Let the transaction dataset be defined as:

$$D = (xi, yi) \mid i = 1,2, \dots, N$$

Where e.g. xi,yi is a transaction feature vector, such as:

1. Transaction amount
2. Timestamp
3. Location
4. Payment mode
5. Account history attributes
6. Merchant category

$yi \in \{0,1\}$ represents the fraud label

0 → Legitimate

1 → Fraudulent

2) Feature Normalization

In order to have the same scaling and shorter convergence of ML models, numeric characteristics are standard-scaled as shown:

$$xi_{norm} = xi - \mu\sigma$$

Where:

μ = mean of the feature

σ = standard deviation

Categorical Encoding: Categorical variables, e.g., payment mode, type of transaction or whereabouts are coded in:

1. Label Encoding
2. One-Hot Encoding
3. This transforms the non-numeric attributes to appropriate representations.
4. Behavioral Feature Engineering

Further derived features are calculated to reflect the fraud trends:

a. Deviation in transaction frequency.
b. Amount anomaly score
c. Location variance
d. Delay in processing transactions.

Final feature vector presentation:

$$xi = [f1, f2, f3, \dots, fk]$$

Where f_k represents engineered fraud-sensitive features.

3) Streaming Data INcorporation with Kafka Data Broker

Apache Kafka is the system backbone or distributed streaming. Kafka Producers share real time transaction events to target topics. Kafka Consumers maintain constant subscribements to subjects and get streams of transactions. Transaction messages are preprocessed and sent to the ML inference engine.

Streaming of transaction flow can be mathematically defined as:

$$T = t1, t2, t3, \dots, tm$$

Where each t_j represents a real-time transaction event.

This ensures low-latency ingestion and high-throughput processing.

4) Model Training and Prediction

The classification function is defined as:

$$\hat{y}_i = f(x_i)$$

Where:

- $f(x_i)$ is the trained ML classifier
- $\hat{y}_i$ is predicted fraud label

Decision Tree (DT)

Decision Tree splits feature space recursively:

$$f_{DT}(x_i) = \text{Decision Rule}(x_i)$$

Random Forest (RF)

Random Forest aggregates predictions from multiple trees:

$$\hat{y}_i = \text{Majority}(T_1(x_i), T_2(x_i), \dots, T_K(x_i))$$

Where:

- $T_k(x_i)$ represents prediction of the k^{th} tree
- K = number of trees

RF reduces overfitting and improves generalization.

Neural Network (NN)

The Neural Network learns nonlinear feature relationships:

$$\hat{y}_i = \sigma(W_2 \cdot \phi(W_1 x_i + b_1) + b_2)$$

Where:

- W_1, W_2 = weight matrices
- b_1, b_2 = bias terms
- ϕ = activation function (ReLU)
- σ = sigmoid function

Neural Networks capture complex fraud patterns including sequential and behavioral anomalies.

4) Real-Time Fraud Detection

Once trained, the selected model is integrated with Kafka Consumer module.

For each incoming transaction:

1. Consume transaction stream
2. Preprocess and normalize features

3. Apply trained classifier
4. Generate fraud probability
5. Trigger alert if:
$$P(\text{Fraud} \mid x_i) > \theta$$

Where $\theta$is fraud threshold.

Fraudulent transactions are forwarded to alert topic and administrative dashboard.

5) Evaluation Metrics for the Proposed Model

The models are evaluated using:

Accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision :

$$Precision = \frac{TP}{TP + FP}$$

Recall:

$$Recall = \frac{TP}{TP + FN}$$

F1-Score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

Due to class imbalance in fraud detection, **Precision, Recall, and F1-score** are considered more significant than Accuracy.

The best-performing model is selected and deployed in the Flask-based real-time fraud detection web application.

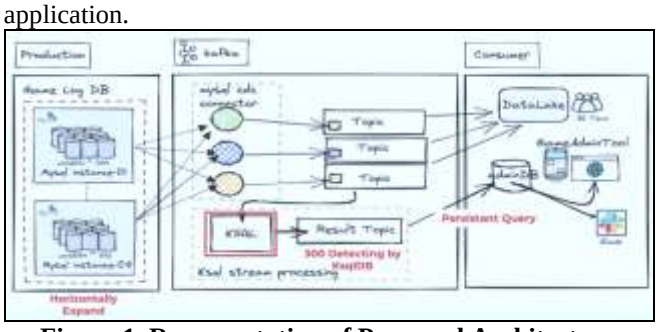

Figure 1. Representation of Proposed Architecture

Figure 1 represents the general layout of the proposed Real-Time Bank Transaction Fraud Detection System. The framework starts with an ongoing generation of transaction data by banking platforms like ATM, mobile applications, and payment gateways. Kafka Producers capture these transaction events and publish them into streaming topics. High-throughput, fault-tolerant transaction data obtains are provided by the streaming layer and then real-time processes the incoming data into transaction data, which undergoes preprocessing and feature engineering modules (normalization, encoding, and reporting anomaling features). The trained features are classified into training and validation datasets (80 percent training and 20 percent testing split) to develop an offline model.

Decision Tree and other machine learning models such as random forest and neural networks are trained using historical labeled transaction data. Once the performance metrics have been evaluated, the most successful model is deployed to be used in real-time inference. The deployed model is combined with Kafka Consumers to classify incoming transactions in real-time. A flask-based web interface is used to flag suspicious transactions and select alerts. Transactions and predictions are saved and audited and retrained in the future.

This architecture ensures:

1. Low-latency fraud detection
2. High scalability
3. Fault tolerance
4. Adaptive learning capability

Bank Transaction Fraud Detection Dataset

To evaluate the proposed framework, a structured bank transaction dataset was used. The dataset contains both numerical and categorical transaction attributes such as:

- Transaction amount
- Account ID
- Payment mode
- Merchant category
- Timestamp
- Location
- Fraud label

Each observation represents a transaction labeled as fraudulent or legitimate.

Dataset Characteristics

- Structured numerical features
- Encoded categorical attributes
- Behavioral engineered features
- Binary classification label

The dataset was preprocessed through:

- Missing value handling
- Label encoding
- Feature normalization
- Train-test split (80%–20%)
- Class imbalance handling

The processed dataset serves as the input to train and validate Decision Tree, Random Forest, and Neural Network models. The integration of real-time Kafka streaming with trained ML classifiers enables continuous fraud monitoring and instant detection within the deployed Flask web application.

IV. Empirical Results

To assess the effectiveness of the proposed Real-Time Bank Transaction Fraud Detection Framework, several machine learning models were conducted with Apache Kafka and Machine Learning. Structured data on bank transactions were analyzed experimentally using data elements like amount of transaction, place and time of transaction, mode of payment, type of merchant, and past behavioral variables. Preprocessing activities such as normalization, categorical encoding, feature engineering, and balancing of classes were performed before training to overcome the inequality between fraudulent and genuine transactions. Because the number of fraud cases in the dataset is often considerably fewer, appropriate preprocessing guaranteed a better generalization of the model.

Models of classification that were assessed include:

1. Decision Tree (DT)
2. Random Forest (RF)
3. Neural Network (NN)

International Journal of Engineering Research and Science & Technology

Performance was measured using standard evaluation metrics:

1. Accuracy
2. Precision
3. Recall
4. F1-Score

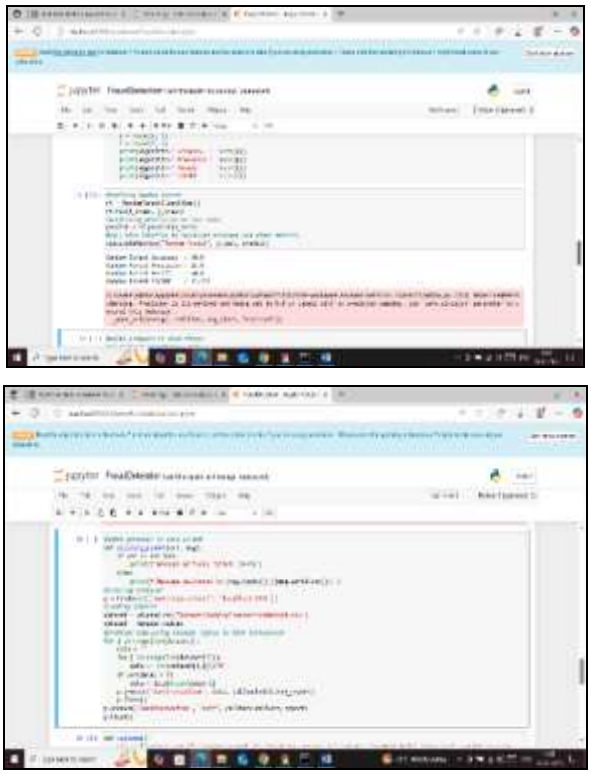

Figure 2: Represent use of RF algorithm and Kafka Functions

Figure 3. Represents the Fraud Detection Interface

Figure 3 shows the prediction interface of the proposed system on fraud detection.

V. CONCLUSION AND FUTURE WORK

This paper presented a predictive analytics infrastructure based on AI to identify and predict outbreaks of infectious diseases early on, using epidemiological data, climate data, population movement data, and social media data. Structured and unstructured data were effectively fused with the help of a whole-scale preprocessing pipeline, which entailed normalization and NLP-driven text processing. The experimental analysis with Decision Tree, Random Forest, and Neural Network models showed that the Neural Network model could outperform the first two verified by its capacity to approximate nonlinear and spatio-temporal outbreak patterns. Implementation of the trained model using Flask-based web application confirmed the usefulness of the system in practice by offering real-time prediction of the outbreak, confidence scores, and graphical feedback as a source of decision support.

Future directions include improving temporal modeling with better deep learning models (e.g. LSTM and Transformer) and incorporating real-time streams of data and making model decisions more understandable through explainable AI methods. By extending the framework to facilitate multi-disease and multi-region prediction, severity estimation and integration with IoT and geospatial surveillance systems will build on the critical role in precision public health and epidemic preparedness.

REFERENCES

[1] Y. Wu and J. Lee, "A Deep Learning Method of Credit Card Fraud Detection," *Mathematics*, vol. 13, no. 5, p. 819, 2025.

[2] M. Jabeen, M. Ali, and J. Kim, "Enhanced Credit Card Fraud Detection Using Deep Hybrid Learning Techniques," *Mathematics*, vol. 13, no. 12, p. 1950, 2025.

[3] N. Baisholan et al., "A Systematic Review of Machine Learning in Credit Card Fraud Detection," *Computers*, vol. 14, no. 10, p. 437, 2025.

[4] R. Daksa and L. Suryadevara, "A Comparative Study on Real-Time Data Streaming for Fraud Detection," *Procedia Computer Science*, vol. 213, pp. 329–338, 2025.

[5] S. Motie et al., "Financial Fraud Detection Using Graph Neural Networks," *Expert Systems with Applications*, vol. 234, 2024.

[6] A. Cherif et al., "Encoder–Decoder Graph Neural Network for Credit Card Fraud Detection," *Expert Systems with Applications*, vol. 222, 2024.

[7] C. Yu et al., "Credit Card Fraud Detection Using Advanced Transformer Models," *arXiv preprint*, 2024.

[8] I. D. Mienye et al., "A Hybrid Deep Learning Approach with Generative Models for Fraud Detection," *Technologies*, vol. 12, no. 10, 2024.

[9] L. Zhao et al., "Deep Learning for Credit Card Fraud Detection: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, 2023.

[10] E. F. Malik and A. K. Singh, "Credit Card Fraud Detection Using a Hybrid Machine Learning Framework," *Mathematics*, vol. 10, no. 9, 2022.

[11] F. Carcillo et al., "SCARFF: A Scalable Framework for Streaming Credit Card Fraud Detection," *Information Fusion*, vol. 41, 2018.

[12] J. Whitrow et al., "Transaction Aggregation as a Strategy for Credit Card Fraud Detection," *Data Mining and Knowledge Discovery*, vol. 18, no. 1, 2009.

[13] S. Afroz et al., "Behavior-Based Real-Time Credit Card Fraud Detection Using Streaming Analytics," *IEEE Access*, vol. 8, 2020.

[14] A. Gupta and D. Singh, "Real-Time Financial Fraud Analytics Using Kafka and Streaming ML," *IEEE Big Data*, 2021.