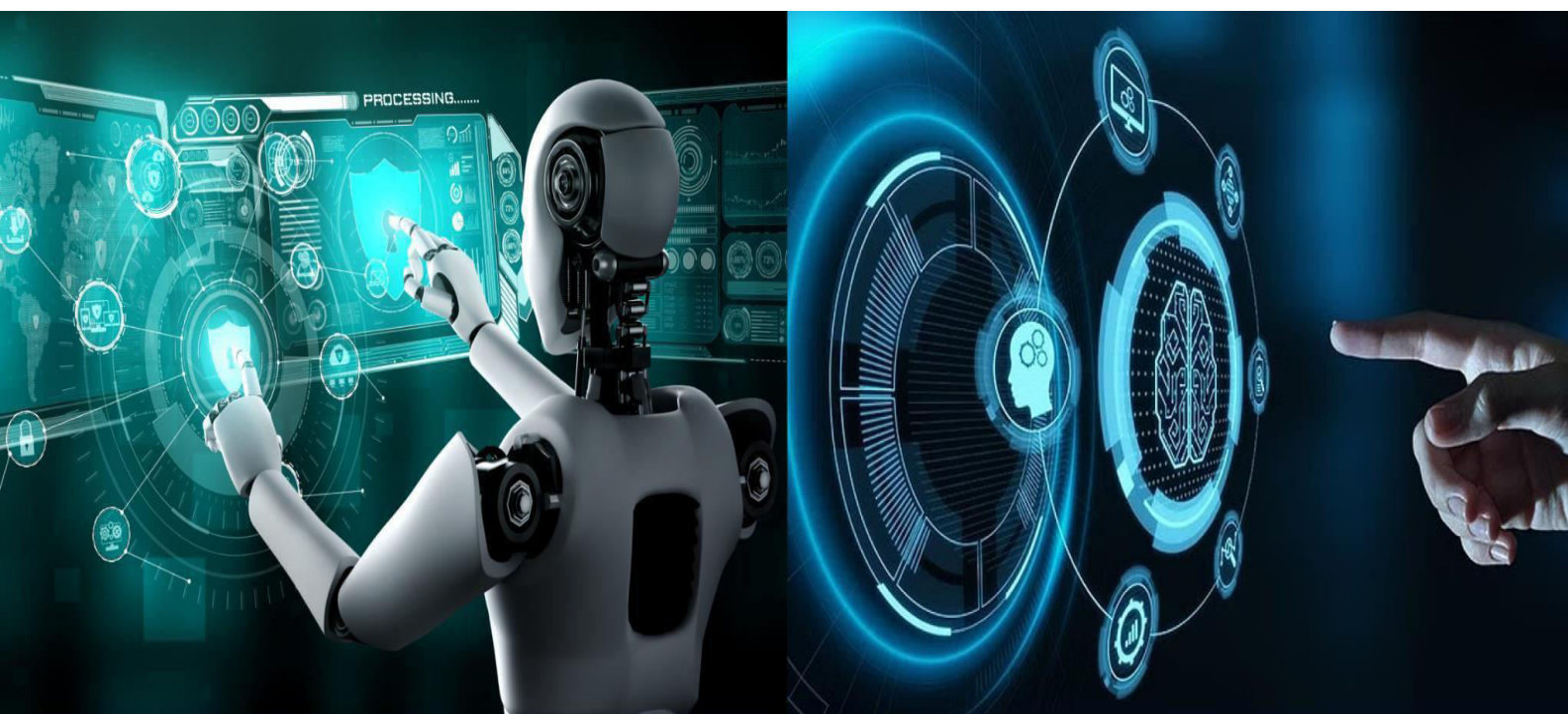


International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Fraud Detection on Bank Payment

Usha K¹, Mohammed Moinuddin², Rajabhakshi M³, Amith M⁴, Darshan P⁵

Mentor, Department of Computer Science and Engineering, Jain Institute of Technology, Davanagere,
Karnataka, India¹

UG, Department of Computer Science and Engineering, Jain Institute of Technology, Davanagere,
Karnataka, India^{2,3,4,5}

ABSTRACT: With the swift expansion of digital banking and online payment platforms, financial fraud has emerged as a significant concern for both banks and customers. Conventional methods for detecting fraud often struggle to pinpoint suspicious transactions in real-time. This project introduces a Real-Time Fraud Detection System for Bank Payments that utilizes machine learning, Apache Kafka, and Streamlit to efficiently and accurately identify fraudulent transactions.

The system in question employs a trained machine learning model to evaluate transaction characteristics such as the transaction amount, account balances, transaction timing, and details of the sender and receiver. Apache Kafka is utilized for real-time data streaming, facilitating the ongoing collection and processing of transaction information. The system categorizes transactions as legitimate or fraudulent based on defined probability thresholds, ensuring quick and dependable fraud detection.

KEYWORDS: Real-Time Fraud Detection, Bank Payment Fraud, Machine Learning, Apache Kafka, Streamlit, Data Streaming, Transaction Analysis, Predictive Modelling.

I. INTRODUCTION

In the future, enhancing this fraud detection system could involve linking it with actual banking and payment gateway APIs to analyze live transaction data. Implementing advanced machine learning and deep learning models can refine prediction accuracy and more effectively manage complex fraud patterns. The system could also be upgraded with automatic model retraining and concept drift detection, allowing it to adapt to changing fraud behaviours over time. Adding features like automated transaction blocking, real-time notifications, and cloud-based deployment can further enhance the system's scalability, security, and practical usability in real banking scenarios.

Beyond the existing functionality, the fraud detection system can be improved by integrating user behaviour analysis and historical transaction profiling to more accurately identify unusual patterns. Utilizing ensemble and deep learning models, like LSTM networks, can bolster the system's capability to recognize sequential and time-sensitive fraud activities. Including real-time visualization dashboards that offer detailed analytics, fraud trends, and performance metrics can aid bank administrators in overseeing system behaviour. Additionally, implementing enhanced authentication measures, data encryption, and audit logging will strengthen system security and ensure compliance. The system may also be deployed on cloud platforms with containerization to enable high availability, fault tolerance, and accommodate large-scale transaction processing, making it ideal for enterprise-level banking applications.

II. RELATED WORK

Numerous research studies have tackled the issue of detecting financial fraud through the use of machine learning techniques. Previous fraud detection systems were based on rules, which often proved ineffective against changing fraud patterns and resulted in high rates of false positives. To address these shortcomings, researchers have introduced supervised machine learning models, such as logistic regression, random forest, and gradient boosting, to learn intricate transaction patterns and enhance detection accuracy.

With the rising volume of transactions and the necessity for timely responses, recent research has concentrated on architectures for real-time fraud detection. Apache Kafka has gained popularity as a streaming platform to facilitate low-



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

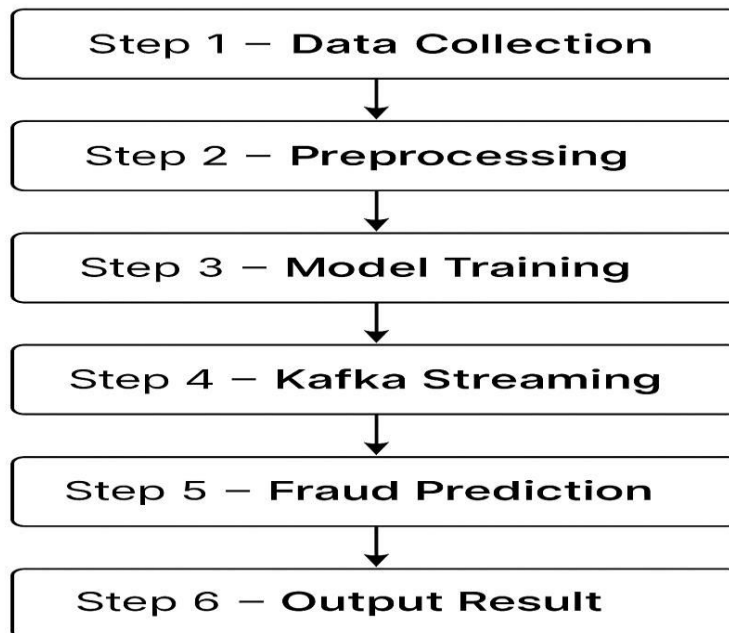
latency ingestion and processing of transaction data. Solutions like SCARFF have shown that Kafka-based streaming pipelines, coupled with machine learning models, can effectively identify fraudulent transactions almost in real-time. Additional studies have investigated the integration of Kafka with stream processing engines like Apache Spark and Apache Flink to bolster scalability and throughput.

Another significant area of research addresses challenges such as data imbalance and concept drift, which frequently occur in fraud detection systems. Given that fraudulent transactions are uncommon, several studies recommend sampling techniques and adaptive learning methods to sustain model performance over time. Recent research has also highlighted the importance of probability-based fraud scoring rather than solely binary classification to enhance risk evaluation and decision-making.

While existing studies predominantly concentrate on backend processing and model performance, fewer efforts have been made to create lightweight, user-friendly interfaces for real-time fraud monitoring. In contrast, the proposed system merges Kafka-based real-time streaming with a trained machine learning model and a Streamlit-based web application that includes login functionality and live visualization. This methodology not only guarantees effective fraud detection but also enhances usability and transparency, making the system applicable for both practical and academic uses.

III. SUGGESTED APPROACH

Simple Methodology



A. Design Considerations:

- Transaction Data Generation.
- Data Streaming using Kafka Producer.
- Real Time Message Handling by Kafka.
- Transaction Consumption using Kafka Producer.
- Data Preprocessing.
- Fraud Prediction using Machine Learning.
- Transaction Classification.
- Result Visualization using Streamlit.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

B. Description of the Proposed Algorithm:

The proposed algorithm is designed to identify fraudulent behaviour bank payment transactions in real time by combining machine learning with a Kafka-based streaming architecture. The algorithm begins by receiving transaction details entered by the user or generated through simulated inputs. These details are forwarded to a Kafka Producer, which publishes the transaction data to a designated Kafka topic for real-time processing.

A Kafka Consumer continuously listens to the topic and retrieves incoming transactions as soon as they are available. Once a transaction is received, the algorithm performs data preprocessing, which includes selecting the required features, handling missing values, and applying feature scaling using a pre-trained scaler to maintain consistency with the training data. The pre-processed data is then passed to a trained machine learning model.

The machine learning model computes a fraud probability score for each transaction. Based on a predefined threshold value, the algorithm classifies the transaction as either legitimate or fraudulent. The classification result, along with the associated probability score, is sent to the Streamlit-based user interface.

The algorithm displays the prediction result in real time, providing clear visual indicators for legitimate and fraudulent transactions. The entire process runs continuously, enabling real-time monitoring and timely fraud detection for bank payment transactions.

IV. PSEUDO CODE

Step 1: Load the trained machine learning fraud detection model and feature scaler.

Step 2: Initialize the Streamlit application and authenticate the user through the login page.

Step 3: Accept transaction details from the user or simulated input.

Step 4: Send the transaction data to the Kafka Producer.

Step 5: Publish the transaction data to the Kafka topic and consume it using the Kafka Consumer.

Step 6: Preprocess the consumed transaction data and apply feature scaling.

Step 7: Predict the fraud probability using the trained machine learning model.

Step 8: Classify the transaction as legitimate or fraudulent based on a predefined threshold.

Step 8: End.

V. SIMULATION RESULTS

The results demonstrate that the system successfully detects fraudulent and legitimate transactions in real time. Transactions with higher fraud probability values were correctly identified as fraudulent, while transactions with lower probability values were classified as legitimate. The Kafka-based streaming architecture ensured low-latency data transfer, allowing continuous transaction monitoring without interruption. The Streamlit interface clearly displayed transaction details, fraud probability, and classification results, making the system easy to understand and use. Overall, the simulation results confirm that the proposed system is effective, responsive, and capable of handling real-time fraud detection scenarios. During the simulation, the system maintained stable performance while processing multiple transactions continuously through the Kafka pipeline. The Kafka Producer and Consumer operated efficiently without data loss, demonstrating reliable message delivery and real-time processing capability. The fraud detection model produced consistent probability scores, allowing clear separation between legitimate and fraudulent transactions. Visual indicators in the Streamlit application, such as success and alert messages, helped users quickly interpret the results. These observations confirm that the integration of machine learning with Kafka streaming and a Streamlit-based interface provides an effective and practical solution for real-time bank payment fraud detection. The findings from the simulation likewise suggest that the system responds quickly to incoming transactions, with minimal delay between data entry and fraud prediction. The use of Apache Kafka enables asynchronous and scalable message handling, ensuring smooth real-time processing even when multiple transactions are submitted consecutively. The machine learning model effectively assigns probability scores that reflect the risk level of each transaction, improving transparency in fraud classification. These results demonstrate that the proposed system is suitable for real-time monitoring scenarios and can serve as a reliable prototype for further development in real-world banking applications.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

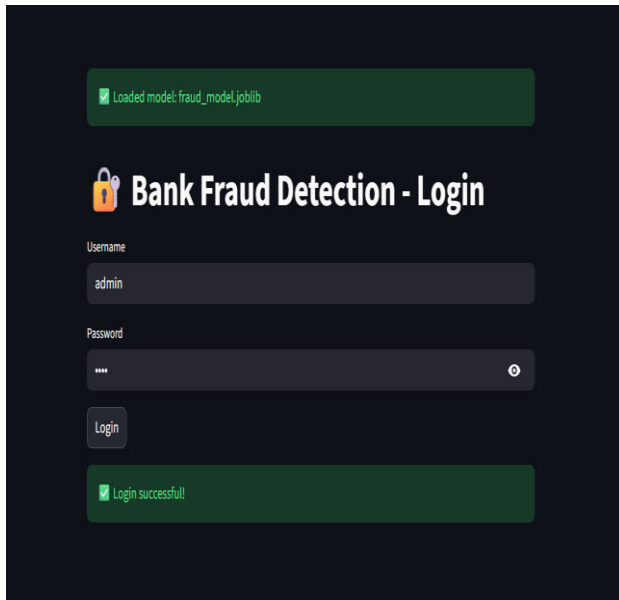


Fig 1:Login Page

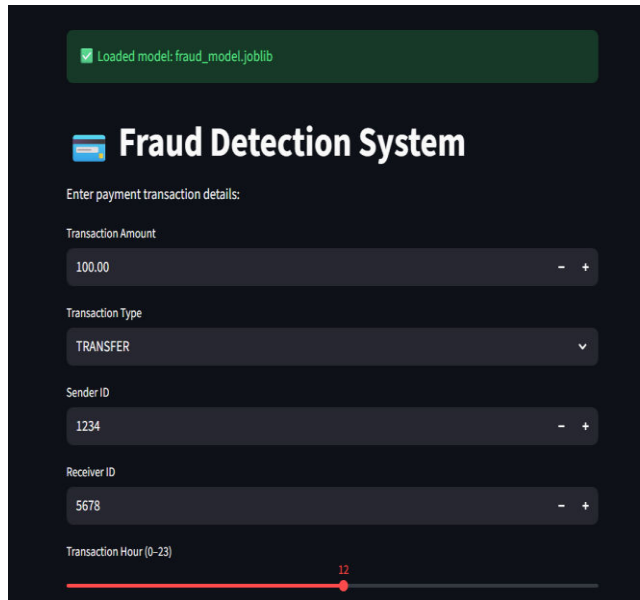


Fig 2:Data Entry

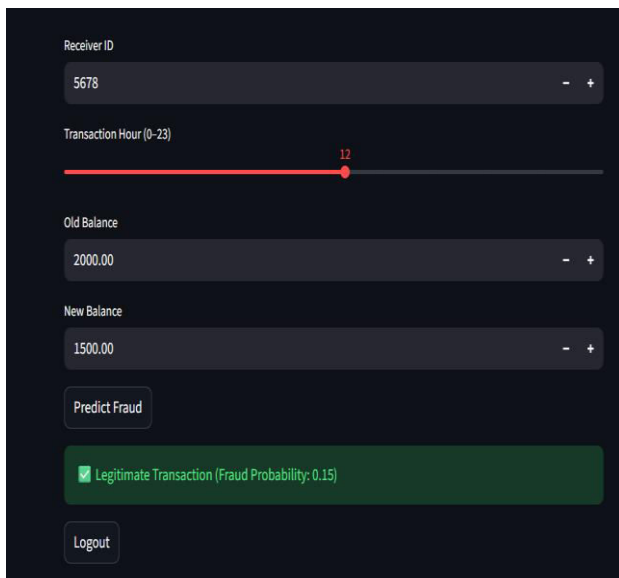


Fig 3:Legitimate Transaction

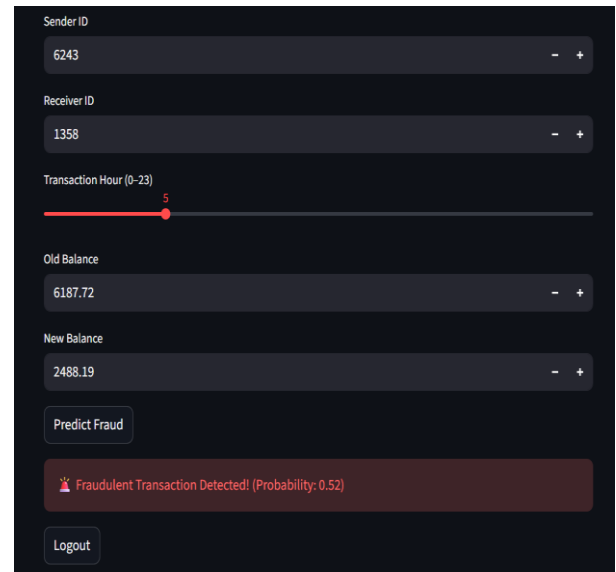


Fig 4:Fradulent Transaction

VI. CONCLUSION AND FUTURE WORK

This project successfully creates a System for Detecting Bank Payment Fraud in Real-Time using Machine Learning, Apache Kafka, and Streamlit. The system processes transaction data instantly, analyzes it with a trained machine learning model, and accurately categorizes transactions as either legitimate or fraudulent based on probability scores. The streaming architecture based on Kafka guarantees low latency and dependable data transmission, while the Streamlit application offers a secure and intuitive interface with live visualization of results. Simulation outcomes show that the system can efficiently identify fraudulent transactions and respond promptly, making it appropriate for real-time monitoring applications. Overall, the project achieves its goals and presents a viable prototype for contemporary fraud detection systems.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

In the future, the system can be enhanced by incorporating real banking or payment gateway APIs to handle live transaction data. More advanced machine learning and deep learning models can be introduced to increase detection accuracy and address complex fraud patterns. The system could also be expanded with automated responses to fraud, such as transaction blocking and alert notifications. Furthermore, cloud-based deployment and scalable architectures could be utilized to accommodate high-volume transaction processing, rendering the system suitable for actual banking environments.

REFERENCES

- [1] Dal Pozzolo, A., Bontempi, G., Snoeck, M., & Snoeck, M. (2018). Adversarial drift detection in fraud detection. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12), 6319–6331. <https://doi.org/10.1109/TNNLS.2017.2783390>
- [2] Dal Pozzolo, A., Caelen, O., Bontempi, G., Snoeck, M., & Snoeck, M. (2015). Adaptive machine learning for credit card fraud detection. *IEEE Intelligent Systems*, 30(4), 38–45. <https://doi.org/10.1109/MIS.2015.45>
- [3] Bahnsen, A. C., Aouada, D., & Ottersten, B. (2015). Cost-sensitive decision trees for fraud detection. *Expert Systems with Applications*, 42(13), 5637–5649. <https://doi.org/10.1016/j.eswa.2015.02.020>
- [4] Kaggle. (2023). Credit card fraud detection dataset. Retrieved from <https://www.kaggle.com>
- [5] Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. *Proceedings of the NetDB*, 1–7.
- [6] Zaharia, M., Das, T., Li, H., Hunter, T., Shenker, S., & Stoica, I. (2013). Discretized streams: Fault-tolerant streaming computation at scale. *Proceedings of the 24th ACM Symposium on Operating Systems Principles*, 423–438. <https://doi.org/10.1145/2517349.2522737>
- [7] Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, 18(1), 30–55. <https://doi.org/10.1007/s10618-008-0116-z>
- [8] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), 1–58. <https://doi.org/10.1145/1541880.1541882>
- [9] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?” Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. <https://doi.org/10.1145/2939672.2939778>
- [10] Streamlit Inc. (2023). Streamlit: The fastest way to build data apps. Retrieved from <https://streamlit.io>



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  ijircce@gmail.com



www.ijircce.com

Scan to save the contact details