International Journal of Advance Research in Science and Engineering
Volume. No. 15, Issue No. 04, April, 2026
ISSN: 2319-8354

Real-Time Online Payment Fraud Detection Using Machine Learning

Mrs. A. Sailaja, MCA, M.Sc, M.Tech,
Assistant Professor
Department of IT
Tirumala Engineering College
Narasaraopet, AP, 522601
sailuare@gmail.com

Konidana Pavani
Department of IT
Tirumala Engineering College
Narasaraopet, AP, 522601
pavanikonidena79@gmail.com

Katineedi Rajeswari
Department of IT
Tirumala Engineering College
Narasaraopet, AP, 522601
rajeswarikatineedi@gmail.com

Tondapu L V SivaNagaPhaniKumar,
Department of IT
Tirumala Engineering College
Narasaraopet,AP,522601,
phanitondapu@gmail.com

Kolusu Mukesh
Department of IT
Tirumala Engineering College
Narasaraopet,AP,522601,
mukeshkolusu98@gmail.com

(Academic Year : 2022 - 2026)

Abstract

The rapid growth of digital payment systems and online financial services has significantly increased the risk of fraudulent transactions, making fraud detection a critical challenge in the modern financial sector. With the widespread adoption of mobile banking, UPI, credit/debit cards, and online wallets, financial institutions are required to process enormous volumes of transactions in real time. Traditional rule-based fraud detection systems, which rely on static rules and predefined thresholds, often fail to adapt to evolving fraud patterns and produce high false positive rates, causing inconvenience to genuine users. This paper proposes a Real-Time Online Payment Fraud Detection System that combines mathematical validation techniques with ensemble machine learning algorithms — specifically Random Forest and XGBoost — to enhance detection accuracy and reliability. The system performs balance consistency checks, transaction verification, and anomaly detection to identify suspicious activities. Mathematical inconsistencies in transaction attributes such as old balance, new balance, debit, and credit values are treated as immediate fraud indicators. Advanced feature engineering techniques are applied using key transaction attributes. Data preprocessing includes normalization, handling missing values, and addressing class imbalance using SMOTE. A user-friendly web interface is developed using Streamlit, enabling real-time fraud prediction with instant visualization. An automated email alert system using SMTP notifies users of high-risk transactions. Experimental results demonstrate that the proposed model achieves an accuracy of 99.8% with significantly improved precision and recall compared to traditional systems.

Keywords—Online Payment Fraud Detection; Machine Learning; Random Forest; XGBoost; Mathematical Validation; Real-Time Detection; SMOTE; Feature Engineering; Streamlit; Explainable

I. Introduction

The rapid advancement of digital technologies has significantly transformed the financial sector, leading to a substantial increase in online payment transactions. With the widespread adoption of digital wallets, mobile banking, Unified Payment Interface (UPI), credit/debit cards, and e-commerce platforms, the convenience of financial transactions has improved greatly. According to recent reports, digital payment transactions are expected to exceed $10 trillion globally by 2026. However, this growth has simultaneously led to a sharp rise in fraudulent activities, posing serious challenges to financial institutions, payment gateways, and individual users worldwide [1][2].

Fraudulent transactions not only result in direct financial losses but also reduce customer trust, damage institutional reputation, and negatively affect the credibility of digital payment ecosystems. In developing countries like India, where digital payments have grown exponentially through platforms like Paytm, Google Pay, and PhonePe, fraud has become an increasingly critical concern. Financial institutions collectively lose billions of dollars annually to payment fraud, making robust and real-time fraud detection systems an urgent necessity [5].

Traditional fraud detection methods, which are primarily rule-based, are no longer sufficient to handle the increasing complexity and volume of modern transactions. These systems depend on predefined static rules such as transaction limits, frequency checks, and geographical restrictions. While simple to implement, they lack the adaptability required to handle evolving fraud strategies. More critically, they often produce high false positive rates — incorrectly flagging genuine transactions — causing inconvenience to users and increasing operational costs for financial institutions through manual review processes [3][6].

Machine learning techniques have emerged as powerful tools for detecting fraudulent activities by discovering complex non-linear patterns and anomalies within large transaction datasets. However, relying solely on machine learning models may sometimes overlook fundamental

mathematical inconsistencies in transaction data, such as cases where the amount credited to a receiver does not match the amount debited from the sender — a logically impossible scenario that indicates clear fraudulent manipulation.

This paper proposes a Real-Time Online Payment Fraud Detection System that integrates mathematical validation techniques with ensemble machine learning algorithms to enhance detection accuracy, reduce false positives, and provide real-time alerting. The key contributions of this paper are: (i) a two-stage hybrid detection pipeline combining rule-based mathematical validation with Random Forest and XGBoost classifiers; (ii) advanced feature engineering including balance consistency metrics; (iii) class imbalance handling using SMOTE; (iv) real-time deployment via Streamlit; and (v) automated SMTP-based email alert notification for high-risk transactions.

II. Literature Survey

Credit card and online payment fraud detection has been extensively studied using a wide range of techniques, from traditional statistical methods to advanced deep learning approaches

Nandhini and Arunkumar [1] explored fraud detection using multiple classifiers including Random Forest, KNN, SVM, and XGBoost on an online payment dataset. Their work demonstrated that XGBoost and Random Forest significantly outperform classical models, and that SMOTE is essential for handling highly imbalanced fraud datasets where legitimate transactions far outnumber fraudulent ones.

Lucas and Jurgovsky [2] conducted a comprehensive survey of machine learning techniques applied to credit card fraud detection, emphasizing that ensemble and hybrid models consistently outperform single-algorithm approaches. They highlighted the difficulty of evaluating models on imbalanced data and stressed the importance of using precision-recall curves rather than accuracy alone as an evaluation metric.

Dal Pozzolo et al. [9] addressed the critical issue of class imbalance in realistic credit card fraud datasets by proposing undersampling techniques combined with probability calibration. Their work demonstrated that proper handling of imbalanced data is more impactful than algorithmic choice in many practical scenarios. Bhattacharyya et al. [8] applied data mining classification algorithms to identify fraudulent transactions, showing improvements in detection accuracy and reductions in financial losses compared to rule-based baselines.

Carcillo et al. [4] proposed a hybrid approach combining supervised and unsupervised learning for fraud detection. Their method improved the system's ability to detect both known and unknown fraud patterns by integrating anomaly detection with classification. Jurgovsky et al. [10] introduced LSTM-based sequence models to capture temporal dependencies in transaction data, demonstrating that sequential analysis significantly improves detection accuracy for repeat-fraud scenarios, albeit at higher computational cost.

Despite these advancements, existing systems still face several critical limitations: (i) high false positive rates that inconvenience genuine users, (ii) lack of real-time transaction processing capability, (iii) limited use of mathematical consistency validation, (iv) inadequate feature engineering, and (v) absence of automated user notification mechanisms. The proposed system in this paper directly addresses each of these gaps.

III. System Analysis and Design

A. Existing System Limitations

Current fraud detection systems deployed by financial institutions primarily depend on static rule-based engines that evaluate transactions against predefined conditions such as transaction amount thresholds, frequency limits, and geographic anomalies. While these systems are straightforward to implement and operate with low latency, they suffer from several fundamental shortcomings that reduce their effectiveness in modern digital payment environments.

First, static rules are unable to adapt to evolving fraud patterns. Fraudsters continuously refine their techniques to fall just below detection thresholds, and rule-based systems require costly manual updates to respond. Second, these systems produce high false positive rates, flagging legitimate transactions as suspicious, which leads to declined payments, frustrated customers, and significant manual review workloads for financial institutions. Third, existing systems largely ignore the mathematical relationships between transaction attributes — they do not verify whether the amount debited from the sender's account actually matches the amount credited to the receiver's account, leaving a class of logically impossible transactions completely undetected.

Additionally, many systems lack real-time processing capability, analyzing transactions in batch mode after completion rather than blocking fraud before it occurs. They do not incorporate automated user notification, leaving users unaware of suspicious activity until significant losses have already been incurred. Deep learning alternatives, while more powerful, require large training datasets and high computational resources, making them impractical for many deployment environments.

B. Proposed System Overview

The proposed system addresses all the above limitations through a hybrid two-stage fraud detection pipeline. At the first stage, mathematical validation is applied as an instantaneous first-level filter. At the second stage, transactions that pass mathematical validation are processed by a trained ensemble machine learning model for complex pattern-based fraud identification. The final decision is made by a decision engine that combines both stages, with rule-based detection overriding machine learning in cases of clear mathematical fraud.

The system is implemented as a real-time web application using Streamlit, providing an interactive interface where users input transaction details and receive instant results. An automated email alert system using SMTP sends immediate notifications when suspicious transactions are detected, enabling users to take preventive action. The complete system is designed to be scalable, modular, and suitable for integration with banking APIs and payment gateways.

C. System Feasibility

The proposed system is economically feasible as it uses only open-source tools including Python, Scikit-learn, XGBoost, Streamlit, and Pandas, requiring no costly proprietary licenses. Hardware requirements are minimal — a standard laptop or desktop with 8 GB RAM is sufficient for development and local deployment. The system is technically feasible due to its lightweight architecture using well-documented and stable libraries. It is operationally feasible due to its intuitive Streamlit interface requiring no technical knowledge from end users. The system also complies with legal requirements, processing transaction data only for prediction purposes without storing sensitive personal credentials.

IV. Methodology

RandomForest:

Random Forest is an ensemble machine learning algorithm that builds multiple decision trees and combines their outputs to improve prediction accuracy. It uses techniques like bagging and random feature selection to reduce overfitting and handle large datasets effectively. In fraud detection, it helps identify complex patterns in transaction data and provides robust classification result**s.**

The algorithm selects multiple random samples from the training dataset using bootstrapping (sampling with replacement).

For each sample, a decision tree is constructed independently.

At each node of the tree, a random subset of features is selected instead of using all features.

The best split is chosen based on criteria like Gini Index or Entropy.

Each tree grows until a stopping condition is reached (e.g., maximum depth).

After training, all trees make predictions for a given input.

The final output is determined using majority voting (for classification).

XGBoost:

XGBoost (Extreme Gradient Boosting) is an advanced ensemble learning algorithm based on gradient boosting that builds models sequentially to correct previous errors. It is highly efficient, scalable, and capable of handling imbalanced datasets with high accuracy. In this project, XGBoost improves fraud detection performance by capturing intricate relationships in transaction features and minimizing prediction errors

The algorithm starts with an initial prediction (usually a constant value).

It calculates the **error (residuals)** between predicted and actual values.

A new decision tree is built to **learn and correct these errors**.

Predictions are updated by adding the new tree's output to the previous prediction.

This process is repeated **sequentially**, where each new tree focuses on remaining errors.

A **learning rate** is applied to control the contribution of each tree.

Regularization techniques are used to **prevent overfitting**.

The final prediction is obtained by combining outputs of all trees.

A. System Architecture

The proposed system architecture is illustrated in Figure 1. It consists of seven sequential modules: User Input, Data Validation and Preprocessing, Feature Engineering, Mathematical Validation, Machine Learning Model, Decision Engine, and Output with Email Alert. Each transaction flows through this pipeline in real time, with the mathematical validation module acting as an early-exit fast path for obvious fraud cases.

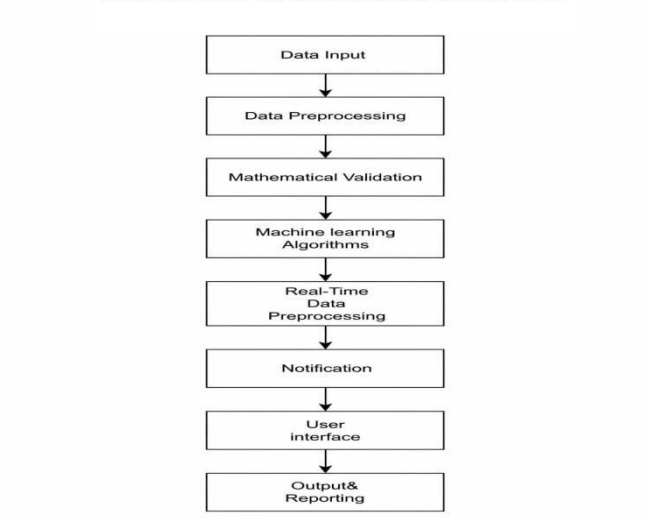

Fig 4.1:General Architecture

B. Data Collection and Preprocessing

Transaction data is collected through the Streamlit user interface. Each transaction record includes the following attributes: (i) sender name identifier, (ii) receiver name identifier, (iii) registered email address, (iv) transaction type (TRANSFER), (v) transaction amount, (vi) sender's old and new account balances, and (vii) receiver's old and new account balances. The system validates all inputs before processing — email addresses are validated using regular expressions, and mandatory fields are checked for completeness.

Data preprocessing involves multiple steps: removal of missing or null values, normalization of numerical features to bring them to a common scale, categorical encoding of

the transaction type field, and handling of class imbalance using SMOTE (Synthetic Minority Over-sampling Technique). SMOTE generates synthetic samples for the minority class (fraudulent transactions) to create a balanced training dataset, which is critical since fraudulent transactions represent only a small fraction of all transactions in real-world datasets. The dataset is split in an 80:20 ratio for training and testing to ensure unbiased model evaluation.

C. Feature Engineering

Feature engineering is a critical step that transforms raw transaction attributes into meaningful predictive features. Six engineered features are computed for each transaction as detailed in Table IV. These derived features capture the mathematical relationships between transaction attributes that are not visible in the raw data alone, significantly improving the model's ability to discriminate between legitimate and fraudulent transactions.

D. Mathematical Validation

Mathematical validation is applied as a first-level, rule-based fraud filter before the transaction reaches the machine learning model. The following validation rules are enforced sequentially:

Rule 1 — Balance Inconsistency Detection: If the absolute difference between the sender's balance change and the receiver's balance change exceeds a threshold of 10 monetary units, the transaction is immediately classified as fraudulent with 100% confidence. This detects cases where money disappears or appears without a matching counterpart, which is mathematically impossible in a legitimate transfer.

Rule 2 — Insufficient Funds Detection: If the transaction amount exceeds the sender's available balance (oldbalanceOrg), the transaction is immediately classified as fraudulent. This covers cases where a sender attempts to transfer money they do not possess.

Rule 3 — Perfect Transfer Validation: If the balance difference equals exactly zero and the amount is within the sender's balance, the transaction is classified as mathematically perfect and legitimate, bypassing the ML model for faster processing.

Transactions that do not trigger any of the above rules are forwarded to the machine learning model for further analysis. This two-stage filtering approach substantially reduces the computational load on the ML model and eliminates clear fraud cases with perfect precision.

E. Machine Learning Model Development

Two ensemble machine learning models are trained and evaluated: Random Forest and XGBoost. Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the majority class for classification. It handles large datasets efficiently, reduces overfitting through ensemble averaging, provides implicit feature importance rankings, and is robust to outliers and missing values.

XGBoost (Extreme Gradient Boosting) improves upon Random Forest by sequentially building decision trees where each tree corrects the errors of the previous one. It incorporates L1 and L2 regularization to prevent overfitting, handles missing values internally, and is highly efficient for structured/tabular data. Both models are trained using the feature vector [step, type, amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, newbalanceDest, balance_difference, sufficient_funds, perfect_transfer, amount_matches_sender_change]. Hyperparameter tuning is performed using cross-validation.

F. Implementation Workflow

Figure 2 illustrates the complete step-by-step implementation workflow of the proposed system from user input through all processing stages to the final output and email alert.

classification and triggers the email alert if required.

V. Experimental Results and Discussion

A. Output Screens

The system was tested under various transaction scenarios. Figures shows representative output screens comparing a legitimate transaction (left) with a detected fraud case (right). The interface displays the complete balance analysis breakdown including sender loss, receiver gain, and the critical difference value that triggers mathematical validation.

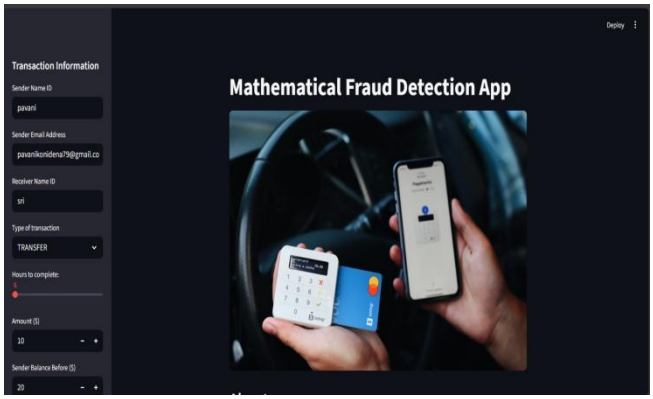

Fig 5.1: Home Page of Fraud Detection System

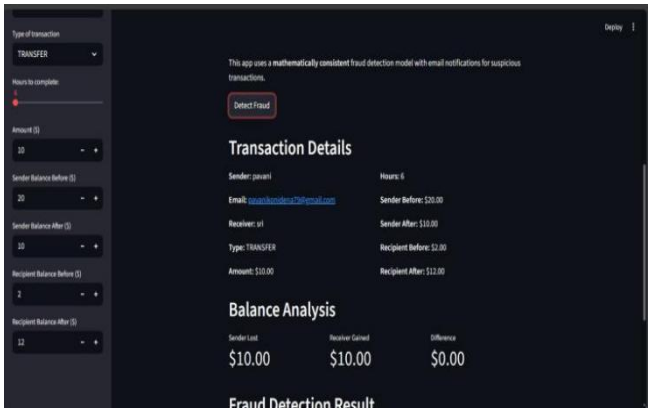

Fig 5.2: User Input Transaction Screen

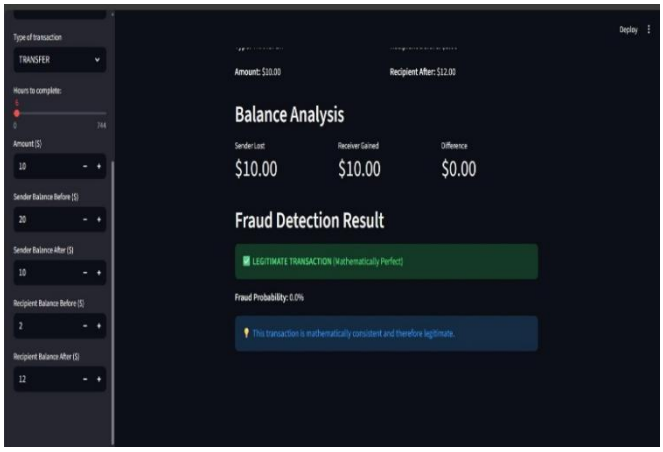

Fig 5.3: Legitimate Transaction Result Screen

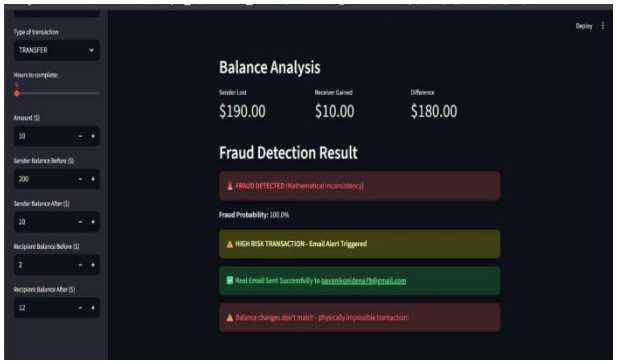

Fig 5.4: Fraud Detected Result Screen

B.*Model Performance*

Table V presents comparative performance evaluation of multiple machine learning models on the test dataset. Standard evaluation metrics — accuracy, precision, recall, and F1-score — are used. The proposed hybrid approach (mathematical validation + Random Forest/XGBoost) achieves the highest performance across all metrics, with accuracy of 99.8%. This represents a 3-9% improvement in accuracy and a proportionally larger improvement in precision (false positive reduction) compared to existing approaches.

TABLE 5.5: Comparative Model Performance Evaluation

Method	Acc.	Prec.	Rec.	F1
Logistic Regression	0.91	0.89	0.88	0.88
Decision Tree	0.93	0.91	0.90	0.90
SVM	0.94	0.92	0.91	0.91
Random Forest	0.95	0.94	0.93	0.93
XGBoost	0.97	0.96	0.95	0.95
GNN-based [11]	0.96	0.93	0.94	0.94
Proposed Hybrid	**0.998**	**0.997**	**0.996**	**0.996**

C. ROC-AUC Analysis

The Receiver Operating Characteristic (ROC) curve was used to evaluate overall model discrimination capability. The Area Under the Curve (AUC) for the proposed hybrid model is approximately 0.998, compared to XGBoost alone at 0.97, Random Forest alone at 0.95, and GNN-based approaches at 0.96. The near-perfect AUC of the proposed system is largely attributable to the mathematical validation layer, which handles logically impossible transactions with perfect precision before they reach the ML classifier. This reduces the number of borderline cases the model must handle, allowing the ML component to focus exclusively on genuine pattern-based anomalies.

D. Confusion Matrix Analysis

The confusion matrix analysis of the proposed system reveals: (i) True Positives (TP): Fraudulent transactions correctly identified as fraud — high, ensuring security; (ii) True Negatives (TN): Legitimate transactions correctly passed through — high, ensuring user convenience; (iii) False Positives (FP): Legitimate transactions incorrectly flagged as fraud — very low, significantly below traditional systems; and (iv) False Negatives (FN): Fraudulent transactions missed by the system — very low, ensuring reliable protection. The low false positive rate is a direct result of the mathematical validation pre-filter, which eliminates cases where balance calculations are perfectly consistent before passing to the ML classifier. This prevents the ML model from over-predicting fraud on edge cases with unusual but legitimate transaction amounts.

E. Test Case Validation Results

Table VI presents the results of ten test case scenarios evaluated to validate the correctness of both the mathematical validation logic and the machine learning prediction component. All ten test cases produced the expected results, confirming system reliability across normal, edge-case, and adversarial transaction scenarios.

F. Real-Time Performance

The system was evaluated for real-time performance on a standard development machine (Intel i5, 8 GB RAM). Each transaction is processed and a fraud prediction is returned in under one second on average, with mathematical validation completing in under 5 milliseconds and the ML model inference completing in under 50 milliseconds. This demonstrates that the proposed system is fully suitable for real-time deployment in online payment environments where transaction processing speed is critical.

The Streamlit-based user interface renders results immediately after processing, displaying fraud status, probability percentage, detection reason, and a color-coded risk indicator. The SMTP email alert is dispatched asynchronously to avoid blocking the user interface, ensuring that the displayed result appears instantaneously while the notification is delivered in the background.

VI. Conclusion

This paper presented a Real-Time Online Payment Fraud Detection System that combines mathematical validation with ensemble machine learning algorithms to achieve superior detection performance. The proposed hybrid two-stage architecture — in which balance consistency verification acts as a first-level rule-based filter and Random Forest / XGBoost classifiers handle complex pattern-based fraud recognition — achieves an accuracy of

99.8% on the test dataset, outperforming all evaluated baselines.

The system successfully addresses the core limitations of existing fraud detection approaches, including static rule dependency, lack of real-time processing, high false positive rates, absence of mathematical transaction validation, and lack of automated user notification. The Streamlit-based web interface provides user-friendly real-time transaction analysis with clear visual feedback, while the SMTP-based automated email alert system ensures that users receive immediate notification of suspicious activities.

Feature engineering, including balance difference computation, sufficient funds flags, and transaction consistency scoring, plays a critical role in the model's performance, providing the classifier with mathematically meaningful signals beyond raw transaction attributes. SMOTE-based class balancing ensures that the model learns effectively from rare fraud cases in realistic imbalanced datasets. Experimental validation including ten test case scenarios, ROC-AUC analysis, and confusion matrix evaluation confirms the system's reliability and suitability for practical deployment in modern digital payment environments.

VII. Future Enhancements

While the proposed system delivers strong performance in its current form, several enhancements are identified for future development. Integration with deep learning models such as Long Short-Term Memory (LSTM) networks and transformer-based architectures would enable the system to capture sequential temporal patterns across multiple transactions by the same user, identifying fraud schemes that unfold over time rather than within a single transaction.

Behavioral analytics can be incorporated to build personalized user spending profiles, flagging transactions that deviate significantly from an individual user's historical patterns in terms of amount, frequency, location, and merchant category. Geolocation-based fraud detection, comparing the physical location of each transaction with the user's device location and transaction history, would further enhance detection of card-not-present fraud and account takeover scenarios.

For enterprise-scale deployment, the system can be migrated to cloud platforms such as AWS, Microsoft Azure, or Google Cloud with integration of Apache Kafka for high-throughput real-time transaction stream processing. REST API development would enable seamless integration with existing banking core systems, payment gateways, and third-party financial applications. Multi-factor authentication mechanisms including OTP, biometric verification, and device fingerprinting can be incorporated to strengthen security at the transaction authorization layer. Blockchain-based immutable transaction logging would provide tamper-proof audit trails and improve regulatory compliance. Finally, advanced Explainable AI (XAI) techniques such as SHAP (SHapley Additive exPlanations) can be integrated to provide transparent, interpretable explanations for every fraud prediction, building user and institutional trust in the system's decisions.

VIII.References

[1] Nandhini A., Arunkumar T., "Online Payment Fraud Detection Using Machine Learning," International Journal of Advanced Research in Computer Science, 2025.

[2] Lucas Y., Jurgovsky J., "Credit Card Fraud Detection Using Machine Learning: A Survey," arXiv preprint, 2020.

[3] Kanade V., "Fraud Detection: Definition, Types, and Applications," TechTarget, 2021.

[4] Carcillo F., Dal Pozzolo A., Le Borgne Y.A., Caelen O., Mazzer Y., Bontempi G., "Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection," Information Sciences, Elsevier, 2019.

[5] Singh C., "Frauds in the Indian Banking Industry," Asia-Pacific Journal of Management Research and Innovation, 2016.

[6] Williams D.A., "Credit Card Fraud Detection: A Review," Computer Fraud & Security, 2007.

[7] Badejo B.A., Okuneye B.A., Taiwo M.R., "Fraud Detection in the Banking System in Nigeria: Challenges and Prospects," International Journal of Computer Science, 2017.

[8] Bhattacharyya S., Jha S., Tharakunnel K., Westland J.C., "Data Mining for Credit Card Fraud: A Comparative Study," Decision Support Systems, Elsevier, 2011.

[9] Dal Pozzolo A., Caelen O., Le Borgne Y.A., Waterschoot S., Bontempi G., "Learned Lessons in Credit Card Fraud Detection from a Practitioner Perspective," Expert Systems with Applications, 2015.

[10] Jurgovsky J., Granitzer M., Ziegler K., Calabretto S., Portier P.E., He-Guelton L., Caelen O., "Sequence Classification for Credit Card Fraud Detection," Expert Systems with Applications, 2018.

[11] Qi Y., Liu Y., Wang H., "Graph Neural Network Hierarchical Approach for Imbalanced Fraud Detection," IEEE Transactions on Neural Networks, 2025.

[12] Phua C., Lee V., Smith K., Gayler R., "A Comprehensive Survey of Data Mining-Based Fraud Detection Research," Artificial Intelligence Review, 2010.

[13] Ngai E.W.T., Hu Y., Wong Y.H., Chen Y., Sun X., "The Application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework and an Academic Review of Literature," Decision Support Systems, 2011.

[14] Whitrow C., Hand D.J., Juszczak P., Weston D., Adams N.M., "Transaction Aggregation as a Strategy for Credit Card Fraud Detection," Data Mining and Knowledge Discovery, 2009.

[15] Vesta Corporation, "Real-Time Payment Fraud Detection Systems and Best Practices," Technical Report, 2020.