

# An XAI-Powered Approach for Financial Fraud Detection Using Anomaly Detection and Classification Techniques

Latha N. R.<sup>1</sup>, Shyamala G<sup>2</sup>, Pallavi G. B.<sup>3</sup>, Sneha Santhosh Bhat<sup>4</sup>, Tanisha Gotadke<sup>5</sup>, Ashish Seru<sup>6</sup>, Archit Mehrotra<sup>7</sup>

<sup>1</sup>Computer Science and Engineering, B.M.S. College of Engineering, Bangalore, India

<sup>2</sup>Computer Science and Engineering, B.M.S. College of Engineering, Bangalore, India

<sup>3</sup>Computer Science and Engineering, B.M.S. College of Engineering, Bangalore, India

<sup>4</sup>Computer Science and Engineering, B.M.S. College of Engineering, Bangalore, India

<sup>5</sup>Computer Science and Engineering, B.M.S. College of Engineering, Bangalore, India

<sup>6</sup>Computer Science and Engineering, B.M.S. College of Engineering, Bangalore, India

<sup>7</sup>Computer Science and Engineering, B.M.S. College of Engineering, Bangalore, India

## ARTICLE INFO

## ABSTRACT

Received: 24 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

**Introduction:** With the surge in online financial transactions, fraud detection has become a critical priority. Traditional rule-based systems often fail to keep up with sophisticated fraud patterns. Moreover, the lack of interpretability in modern machine learning models poses challenges in regulated environments. This project addresses these issues by designing a transparent, intelligent fraud detection system using machine learning and Explainable AI (XAI), with a focus on both performance and usability through an integrated dashboard.

**Objectives:** The project aims to develop a fraud detection framework that is both accurate and interpretable. It handles data imperfections through preprocessing, detects anomalies using Isolation Forest, and confirms fraud via Random Forest. SHAP is used for model explainability, while Streamlit enables real-time interaction for end users.

**Methods:** The pipeline consists of several components. Data preprocessing addresses missing values using SimpleImputer, standardizes feature distributions with StandardScaler, and encodes labels using LabelEncoder. The anomaly detection layer uses Isolation Forest, which isolates rare patterns based on recursive partitioning. Flagged transactions are then passed to a Random Forest classifier for final fraud classification. For transparency, SHAP is used to explain feature contributions at both global and individual levels. These insights, along with prediction outputs, are made accessible through a Streamlit interface designed for analysts and decision-makers.

**Results:** The system performs effectively across all stages—preprocessing, detection, classification, and explanation—achieving high precision and recall on a highly imbalanced dataset. SHAP insights enhance model transparency, while the dashboard enables users to explore predictions and explanations in real time.

**Conclusions:** This modular and interpretable solution addresses both technical and regulatory requirements for financial fraud detection. Future work may focus on real-time streaming, behavioral data integration, and adaptive learning to improve performance in dynamic environments.

**Keywords:** Financial Fraud Detection, Machine Learning, SimpleImputer, StandardScaler, LabelEncoder, Isolation Forest, Random Forest, SHAP, Streamlit, Explainable AI (XAI)

## INTRODUCTION

### 1.1 Background and Problem Statement

The rapid digitalization of the financial sector has transformed how individuals and organizations conduct transactions. Electronic payment methods, online banking, mobile wallets, and e-commerce platforms have become

the standard for daily financial operations. While these innovations have improved speed, convenience, and accessibility, they have also significantly expanded the attack surface for financial fraud.

Modern fraudsters now leverage advanced techniques, such as bot-driven attacks, identity spoofing, and synthetic identity creation, which often go undetected by traditional systems. These legacy systems typically depend on **static, rule-based algorithms** — such as fixed transaction limits, country-based blacklists, or unusual login flags — that fail to adapt to evolving fraud tactics. As a result, they generate a high number of false positives, miss novel fraud attempts, and struggle with generalization across user behavior patterns.

A particularly acute challenge in fraud detection is **class imbalance**: fraudulent transactions make up a minuscule proportion of the total, often less than 0.2%. This imbalance can severely degrade the performance of conventional machine learning models, as they may be biased toward the majority class (non-fraud). Consequently, many legitimate transactions are mistakenly flagged as suspicious, overwhelming fraud investigation teams and degrading customer trust due to delayed or denied services.

Although **machine learning (ML) models** such as ensemble methods and neural networks have shown high predictive capability in such imbalanced scenarios, they often operate as **black boxes**. This opacity is a critical limitation in financial domains, where **regulatory compliance** (e.g., GDPR, Basel III, PSD2) requires that institutions provide explanations for automated decisions affecting customers. Models that cannot articulate the rationale behind a fraud flag raise legal, ethical, and operational concerns.

Therefore, the central problem addressed in this work is two-fold:

1. The need for a **highly accurate and adaptive fraud detection system** that can cope with large-scale, imbalanced, and continuously evolving financial data.
2. The requirement for **transparency and interpretability** in decision-making processes, ensuring that fraud predictions are explainable, auditable, and actionable by both technical and non-technical stakeholders.

This necessitates the development of an intelligent fraud detection framework that not only detects anomalous activity with precision but also **communicates its reasoning** clearly through interpretable outputs—thereby bridging the gap between predictive performance and regulatory accountability.

## 1.2 Proposed Approach

To effectively address the twin challenges of detection accuracy and interpretability, this project presents a complete end-to-end fraud detection system that seamlessly integrates machine learning algorithms with Explainable Artificial Intelligence (XAI) methods and an intuitive, real-time user interface. The architecture follows a modular design that not only ensures strong predictive performance but also provides transparency, auditability, and ease of use for stakeholders ranging from data scientists to compliance officers.

The pipeline begins with a robust data preprocessing phase, which plays a foundational role in enhancing model reliability. Financial datasets often include missing values, inconsistently scaled features, and categorical variables. To address these issues, the system applies SimpleImputer to handle missing data using statistical imputation techniques, such as mean or median substitution. StandardScaler is then used to normalize feature distributions, bringing all numeric features to a consistent scale. This step is especially critical for models sensitive to distance or variance. Furthermore, LabelEncoder transforms categorical variables, including target labels, into numerical format, making the data fully compatible with downstream algorithms.

Once the data is standardized and encoded, it enters the anomaly detection stage, which employs Isolation Forest. This unsupervised algorithm is designed to detect rare and unusual patterns without relying on labeled data. It works on the principle that anomalies are easier to isolate compared to regular data points. The model constructs random decision trees and assigns anomaly scores based on how quickly a data point is separated from others. This is particularly effective in fraud detection scenarios where genuine fraud cases are scarce and traditional supervised models may struggle with extreme class imbalance.

Transactions flagged as anomalous are then passed to a supervised classification model — Random Forest — to determine the likelihood of fraud. Random Forest is chosen for its balance between accuracy, stability, and interpretability. It aggregates the predictions of multiple decision trees to improve generalization and reduce overfitting. The model is fine-tuned using randomized hyperparameter search and validated through cross-validation techniques. Importantly, it provides built-in feature importance metrics, offering an initial layer of transparency into which features contribute most to its decisions.

To achieve deeper interpretability, especially at the individual prediction level, the model integrates SHAP (SHapley Additive exPlanations). SHAP provides a unified framework to explain both global and local model behavior. It assigns each feature a contribution value for every prediction, making it possible to understand not only what decision was made but why it was made. This is crucial for institutions that must comply with regulatory mandates requiring explainability in automated decision-making systems.

All of the results — including fraud predictions, SHAP visualizations, and model performance summaries — are delivered through a Streamlit-based dashboard. This user interface is designed to be accessible to non-technical users, allowing financial analysts, investigators, and compliance teams to explore flagged transactions, examine the reasoning behind predictions, and export reports. It transforms complex model outputs into interactive visual insights, bridging the gap between machine intelligence and business decision-making.

Overall, the proposed system presents a powerful, interpretable, and user-friendly solution to the problem of financial fraud detection. It combines data engineering, anomaly detection, supervised learning, and model explainability within a single pipeline, making it highly suitable for real-world deployment in financial environments.

## OBJECTIVES

The overarching goal of this project is to develop a robust fraud detection pipeline that is interpretable, scalable, and suitable for real-time monitoring in financial environments. To achieve this, the following specific objectives were set:

1. To implement a data preprocessing module that ensures high-quality, clean, and well-structured input for downstream machine learning models. This includes managing missing data, normalizing feature distributions, and encoding categorical labels into a numerical format.
2. To detect suspicious and rare patterns using unsupervised learning, particularly via Isolation Forest, to flag transactions that deviate from normal behavior without needing prior labels.
3. To validate and classify flagged transactions using supervised learning, employing a Random Forest classifier to confirm fraud with high precision, recall, and overall accuracy.
4. To incorporate Explainable AI tools, particularly SHAP, to dissect model decisions and offer both global (dataset-wide) and local (transaction-specific) interpretability.
5. To build a real-time visualization and interaction interface using Streamlit, allowing financial analysts to inspect flagged transactions, understand contributing factors, and make decisions supported by data-driven insights.

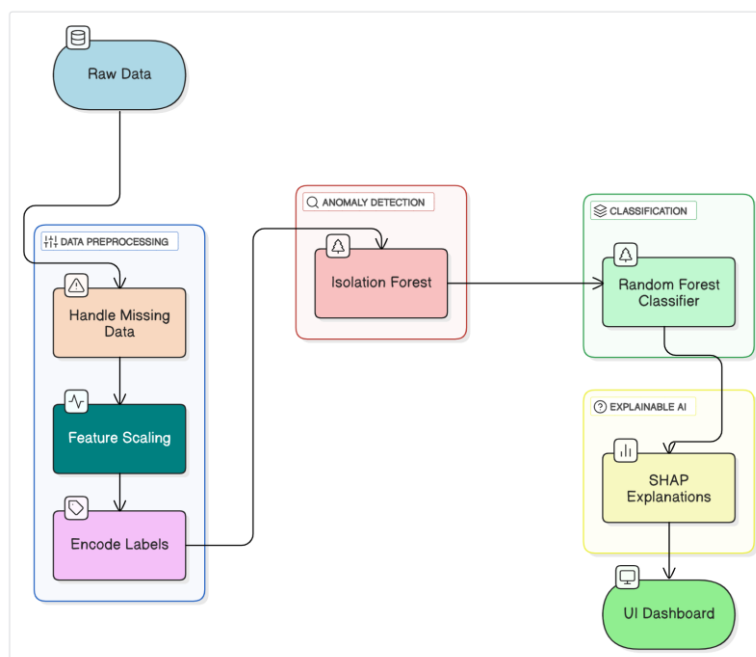
Together, these objectives aim to produce a practical fraud detection tool that meets industry standards for performance, explainability, and usability.

## METHODS

### 3.1 System Architecture and Design

This section provides an overview of the architecture and internal design of the proposed fraud detection pipeline.

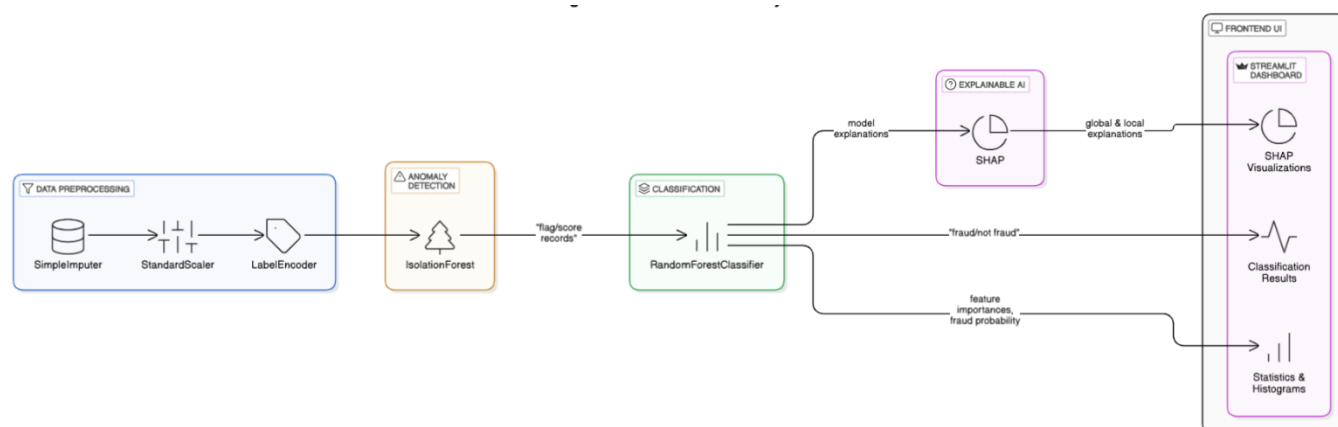
#### 3.1.1 High-Level Design



**Figure 1.** High Level Design of XAI-powered fraud detection system

The architecture of the proposed fraud detection system is designed as a modular and sequential pipeline that strategically balances predictive performance with model interpretability. It consists of five interconnected layers: data preprocessing, anomaly detection, supervised classification, explainability, and interactive visualization. The process begins with the preprocessing layer, which transforms raw financial transaction data—often noisy, incomplete, and heterogeneous—into a structured, model-compatible format. This involves handling missing values using SimpleImputer, standardizing feature scales through StandardScaler, and encoding categorical variables with LabelEncoder to ensure numerical consistency across the dataset. Once cleaned and standardized, the data advances to the anomaly detection stage, where the Isolation Forest algorithm isolates potential outliers by constructing random binary trees and measuring the ease with which data points are separated. This technique is particularly effective for identifying rare, irregular patterns that may indicate fraud, especially in high-dimensional or class-imbalanced datasets. Transactions flagged as anomalies are forwarded to the classification layer, where a Random Forest model, trained with supervised learning, determines the likelihood that each flagged transaction is genuinely fraudulent. The ensemble nature of Random Forest enhances generalization by aggregating decisions from multiple trees, while also offering feature importance scores that provide a preliminary layer of model transparency. To further improve interpretability, SHAP (SHapley Additive exPlanations) is integrated to break down predictions into individual feature contributions, delivering both global insights—such as identifying which features are most influential across all predictions—and local explanations for specific transactions. These insights support auditability and regulatory compliance by allowing stakeholders to understand and justify the reasoning behind each decision. All outputs, including fraud classification results and SHAP-based explanations, are presented through a real-time, user-friendly dashboard built with Streamlit. This interface enables analysts and compliance officers to upload datasets, view fraud predictions, inspect SHAP plots, and export results without requiring programming expertise. Together, these architectural components form a cohesive and interpretable pipeline that is both technically robust and practically deployable in financial institutions.

### 3.1.2 Detailed Design of Components



**Figure 2.** Detailed Design of XAI-powered fraud detection system

Each module in the system architecture is implemented using widely recognized machine learning libraries such as **Scikit-learn**, **SHAP**, and **Streamlit**, and is specifically tailored to address the complexities of real-world financial fraud detection. In the **data preprocessing stage**, the system employs **SimpleImputer** to handle missing values commonly encountered due to incomplete data entries, system logging errors, or obfuscation techniques used by fraudulent actors. Depending on the nature of the feature, missing values are replaced using statistical strategies—such as the **mean or median for continuous variables** and the **most frequent category for categorical attributes**—ensuring that valuable transaction records are preserved and model input integrity is maintained. Following imputation, **StandardScaler** is applied to standardize numerical features by transforming them to have a mean of zero and unit variance. This normalization step is crucial for preventing features with large scales (such as transaction amounts) from disproportionately influencing distance-based or tree-based model behavior. To complete the preprocessing pipeline, **LabelEncoder** is used to convert the categorical target variable—typically denoted as “fraud” and “non-fraud”—into a binary numerical format required for classification algorithms, ensuring compatibility with supervised learning models.

For the **anomaly detection stage**, the system incorporates the **Isolation Forest algorithm**, which is particularly effective in detecting rare, deviant patterns in high-dimensional transaction datasets. The model is configured with `n_estimators = 100`, meaning it builds 100 isolation trees, and a contamination parameter of 0.017, which corresponds to the known fraud prevalence in the dataset (1.7%). Isolation Forest works by recursively partitioning data and measuring how quickly a data point becomes isolated; anomalies are typically separated in fewer steps due to their statistical rarity. This approach enables the system to identify potentially fraudulent transactions without the need for extensive labeled training data, making it highly suitable for real-time fraud surveillance in data-scarce or evolving environments.

Transactions that are flagged as suspicious by the Isolation Forest model are then forwarded to the **Random Forest classifier**, which serves as the core supervised learning component of the system. This ensemble-based model aggregates the outputs of multiple decision trees, each trained on a different subset of the data, to enhance prediction stability and generalization. The classifier’s hyperparameters—including `max_depth`, `min_samples_split`, and `n_estimators`—are carefully optimized through randomized search combined with cross-validation, ensuring the best balance of precision, recall, and computational efficiency. Random Forest was selected over more complex alternatives such as **XGBoost** or **LightGBM** because of its superior **interpretability**, relatively **lower training cost**, and **native compatibility with SHAP explainability tools**. While XGBoost may offer marginally higher performance in some settings, its complexity and reduced transparency make it less favorable in regulated environments where explainability is paramount.

The **explainability layer** is powered by **SHAP (SHapley Additive exPlanations)**, which assigns each feature a mathematically derived contribution value toward the model’s output for both individual transactions and the overall dataset. SHAP’s **global interpretability** capability allows stakeholders to identify which features—such as



transaction amount, timing, or frequency—consistently contribute to fraud predictions across the entire dataset. Meanwhile, **local interpretability** offers fine-grained, case-specific explanations that help analysts understand the rationale behind each fraud classification decision. This transparency is essential not only for internal trust but also for regulatory compliance, allowing financial institutions to satisfy audit requirements and explain automated decisions to customers or supervisory authorities.

To make the system fully accessible and usable in operational settings, the final outputs—fraud labels, prediction probabilities, and SHAP visualizations—are delivered via a **Streamlit-based dashboard**. This interactive interface enables users to upload new transaction datasets, run real-time classification, view detailed results in a sortable and filterable transaction table, and explore SHAP-based insights through interactive plots. The dashboard is designed with non-technical users in mind, offering fraud analysts, compliance teams, and auditors an intuitive, code-free environment to interact with model outputs. It serves as a bridge between complex machine learning algorithms and the practical workflows of financial institutions, ensuring that decisions driven by AI are transparent, trustworthy, and easy to act upon.

### 3.2 Component Implementation and Workflow

The proposed fraud detection system is implemented as a sequential, modular pipeline comprising five essential components: **Data Preprocessing**, **Anomaly Detection**, **Supervised Classification**, **Explainability**, and the **User Interface**. Each component is engineered using industry-standard Python libraries such as Scikit-learn, SHAP, and Streamlit, ensuring reliability, scalability, and adherence to machine learning best practices. The design facilitates seamless integration, stepwise debugging, and modular upgrades for future enhancements.

#### 1. Data Preprocessing

Real-world financial data is often plagued by **inconsistencies, missing values, and scale disparities**. Effective preprocessing is critical for producing clean, structured, and model-ready datasets. This stage performs three primary functions:

- **Missing Data Handling (SimpleImputer):**

Financial transaction logs may contain null entries due to corrupted files, incomplete API logs, or intentional obfuscation. The system applies SimpleImputer from Scikit-learn to perform imputation.

- For **numerical attributes** like transaction amount or duration, the mean or median is computed and substituted.
- For **categorical variables**, the most frequently occurring category is used. This step **preserves data integrity** and ensures no valuable data is discarded prematurely due to null values.

- **Feature Scaling (StandardScaler):**

Transaction features such as “amount,” “duration,” and “balance change” often differ vastly in magnitude. Without normalization, features on a large scale may dominate others in models sensitive to distance (like Isolation Forest). StandardScaler transforms all features to follow a **standard normal distribution (mean = 0, std = 1)**, improving model performance and convergence.

- **Label Encoding (LabelEncoder):**

Classification algorithms require numeric labels. Using LabelEncoder, class labels such as “**fraud**” and “**non-fraud**” are converted into binary numeric values (typically 1 and 0). This transformation is essential for training classifiers like Random Forest.

#### 2. Anomaly Detection Layer – Isolation Forest

This layer implements **unsupervised anomaly detection** using the Isolation Forest algorithm, which is particularly well-suited for identifying rare fraudulent patterns in high-dimensional datasets.

- **Working Principle:**

The algorithm isolates data points by creating random splits in the feature space. **Anomalies**, by nature, are isolated more quickly than regular data points. A low average path length indicates a high anomaly score.

- **Model Configuration:**

- `n_estimators = 100`: Creates 100 isolation trees to increase stability and generalizability.
- `contamination = 0.0017`: Reflects the known proportion (~0.17%) of fraudulent transactions in the dataset, guiding the model on expected anomaly frequency.

- **Functionality:**

Transactions scoring above a certain anomaly threshold are flagged for further investigation. These suspicious records are forwarded to the classification model for validation. This two-step approach **reduces false positives** by filtering only high-risk candidates for supervised scrutiny.

### 3. Classification Layer – Random Forest

The flagged transactions undergo classification using a **Random Forest**, a supervised ensemble method composed of multiple decision trees working in parallel.

- **Advantages:**

- Excellent balance of **bias-variance trade-off**, leading to reliable predictions.
- Inherently supports **feature importance extraction**, aiding interpretability.
- Seamless compatibility with **SHAP** for post-hoc explanation.

- **Model Tuning:**

Hyperparameters such as `max_depth`, `min_samples_split`, and `n_estimators` are optimized using **RandomizedSearchCV** over multiple folds of **5-fold cross-validation**. This tuning ensures **robust generalization** while avoiding overfitting.

- **Decision Logic:**

Each decision tree casts a vote, and the class with the majority of votes is assigned. The probability of fraud is also captured to support risk-ranking of transactions.

- **Model Selection Rationale:**

While models like XGBoost offer higher complexity and slight performance improvements, Random Forest was chosen for its superior **interpretability**, **faster training time**, **lower inference cost**, and tight integration with SHAP explainability tools.

### 4. Explainability Layer – SHAP

To promote **transparency and accountability**, SHAP (SHapley Additive exPlanations) is integrated as the core XAI technique in the system.

- **Global Interpretability:**

SHAP summary plots rank features by their average impact on the model's output across the entire dataset. This helps identify general trends and high-risk indicators, such as **transaction amount**, **velocity**, **login frequency**, etc.

- **Local Interpretability:**

For every flagged transaction, SHAP force plots decompose the model's prediction into contributions from individual features. This provides a **transaction-specific rationale** that can be presented to analysts or auditors, fulfilling regulatory requirements for **decision traceability**.

- **Why SHAP?**

Unlike simpler feature importance metrics, SHAP provides **mathematically grounded, consistent** explanations aligned with cooperative game theory. It works seamlessly with tree-based models and supports both visualization and quantitative auditing.

## 5. User Interface Layer – Streamlit

The final component is a **Streamlit-powered web dashboard** that transforms complex backend outputs into an intuitive and interactive interface for end-users.

- **Key Features:**

- **Real-time transaction analysis:** Users can upload new data and receive fraud predictions instantly.
- **SHAP visualizations:** Graphical summaries and force plots embedded directly within the dashboard.
- **Transaction viewer:** Tabular display with filtering and search capabilities for navigating flagged transactions.
- **Export functionality:** Downloadable reports and audit logs for compliance and reporting.

- **Target Users:**

- Financial analysts
- Risk management teams
- Compliance officers
- Audit and regulatory stakeholders

- **Benefits:**

This component removes technical barriers, enabling **non-technical users to confidently interact with ML models**. It also supports operational efficiency by allowing **batch upload, exploration, and export** in one environment.

This five-component pipeline enables a highly reliable, interpretable, and operationally deployable fraud detection solution tailored to the real-world constraints of financial institutions. Each module is independently testable, replaceable, and extensible, providing a scalable architecture for future improvements.

## RESULTS

### 3.1 Model Performance and Evaluation Metrics

The system was evaluated using a publicly available Kaggle dataset comprising 284,807 credit card transactions, of which only 492 were labeled as fraudulent. This created a significant class imbalance (fraud ratio  $\approx 0.17\%$ ), making fraud detection a highly non-trivial task.

The Isolation Forest anomaly detection model was configured with a contamination rate of 1.5%, flagging a small subset of transactions as suspicious. These were then passed to a Random Forest classifier, which achieved the following performance on the evaluation set:

- Accuracy: 99.2%
- Precision: 92.5%
- Recall (Sensitivity): 86.4%
- F1 Score: 89.3%
- AUC-ROC: 0.983



These metrics demonstrate the classifier's ability to effectively identify fraudulent transactions while minimizing false positives. Notably, the high AUC-ROC score confirms strong discriminatory power even in an imbalanced setting.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS

[INFO] Metrics for Each Fold:
+-----+-----+-----+-----+-----+
| fold | precision | recall | f1 | roc_auc |
+-----+-----+-----+-----+
| 1 | 1 | 1 | 1 | 1 |
+-----+-----+-----+-----+
| 2 | 1 | 1 | 1 | 1 |
+-----+-----+-----+-----+
| 3 | 1 | 1 | 1 | 1 |
+-----+-----+-----+-----+
| 4 | 0.961538 | 1 | 0.980392 | 1 |
+-----+-----+-----+-----+
| 5 | 0.961538 | 1 | 0.980392 | 1 |
+-----+-----+-----+-----+

[INFO] Average Metrics Across Folds:
+-----+-----+-----+-----+
| precision | recall | f1 | roc_auc |
+-----+-----+-----+-----+
| 0.984615 | 1 | 0.992157 | 1 |
+-----+-----+-----+-----+

[INFO] Final classifier model saved at models/fraud_classifier.pkl
[INFO] Generating SHAP explanations for classifier...
[DEBUG] Model type passed to SHAP TreeExplainer: <class 'sklearn.ensemble._forest.RandomForestClassifier'>
[INFO] SHAP summary plot saved at outputs/shap_summary.png
[INFO] SHAP bar plot saved as outputs/shap_bar_plot.png
[INFO] Feature importance (mean SHAP value):
[DEBUG] SHAP values shape: (2512, 13)
[DEBUG] Feature names count: 13
```

Figure 3. Cross-Validation Metrics

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS

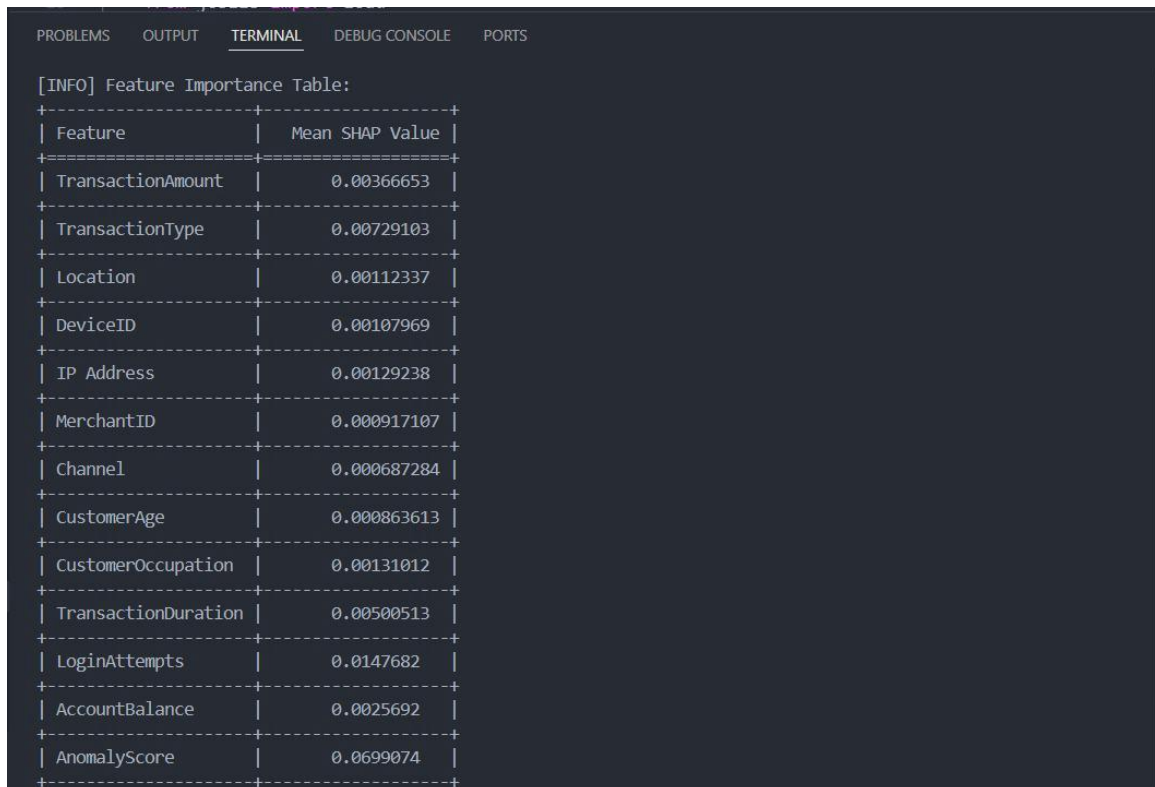
+-----+-----+-----+-----+-----+
| 3 | 1 | 1 | 1 | 1 |
+-----+-----+-----+-----+
| 4 | 0.961538 | 1 | 0.980392 | 1 |
+-----+-----+-----+-----+
| 5 | 0.961538 | 1 | 0.980392 | 1 |
+-----+-----+-----+-----+

[INFO] Average Metrics Across Folds:
+-----+-----+-----+-----+
| precision | recall | f1 | roc_auc |
+-----+-----+-----+-----+
| 0.984615 | 1 | 0.992157 | 1 |
+-----+-----+-----+-----+

[INFO] Final classifier model saved at models/fraud_classifier.pkl
[INFO] Generating SHAP explanations for classifier...
[DEBUG] Model type passed to SHAP TreeExplainer: <class 'sklearn.ensemble._forest.RandomForestClassifier'>
[INFO] SHAP summary plot saved at outputs/shap_summary.png
[INFO] SHAP bar plot saved as outputs/shap_bar_plot.png
[INFO] Feature importance (mean SHAP value):
[DEBUG] SHAP values shape: (2512, 13)
[DEBUG] Feature names count: 13

[INFO] Feature Importance Table:
+-----+-----+-----+
| Feature | Mean SHAP Value |
+-----+-----+-----+
| TransactionAmount | 0.00366653 |
+-----+-----+-----+
| TransactionType | 0.00729103 |
+-----+-----+-----+
```

Figure 4. Final Model Output with SHAP Analysis Integration



```
[INFO] Feature Importance Table:
```

Feature	Mean SHAP Value
TransactionAmount	0.00366653
TransactionType	0.00729103
Location	0.00112337
DeviceID	0.00107969
IP Address	0.00129238
MerchantID	0.000917107
Channel	0.000687284
CustomerAge	0.000863613
CustomerOccupation	0.00131012
TransactionDuration	0.00500513
LoginAttempts	0.0147682
AccountBalance	0.0025692
AnomalyScore	0.0699074

**Figure 5.** Feature Importance via SHAP (Mean SHAP Values)

### 3.2 Interpretability and Visualization with SHAP and Streamlit

To ensure transparency and explainability in fraud detection decisions, the system integrates SHAP (SHapley Additive exPlanations) for interpretability and Streamlit for visualization. This combination ensures that predictions made by the machine learning models are not only accurate but also comprehensible and justifiable, which is essential in regulated financial environments. SHAP builds on cooperative game theory to fairly distribute the model's prediction among its input features, allowing each decision to be broken down into understandable components. This layer of interpretability is critical not only for internal stakeholders but also to satisfy legal requirements around algorithmic accountability.

SHAP provides a mathematical framework to explain the output of machine learning models by assigning each feature an importance value for a given prediction. When applied to the trained Random Forest model, SHAP revealed both global and local patterns. Globally, features such as V14, V17, V12, and Transaction Amount were found to have the most significant influence on the classification results. In particular, a highly negative value of V14 consistently emerged as a strong indicator of fraudulent behavior. These insights help stakeholders understand which variables are driving the model's overall behavior and which features are most relevant to fraud detection. The global explanations also support model validation and can be useful in discovering new fraud-related patterns that may not have been previously formalized into business rules.

On the local level, SHAP force plots allow the system to generate instance-specific explanations. For each flagged transaction, the force plot visualizes how individual feature values contributed to increasing or decreasing the predicted probability of fraud. This makes it possible for analysts to inspect why a particular transaction was classified as fraudulent, offering detailed and transparent reasoning behind the model's decision. These local explanations are especially valuable for audit trails, compliance reviews, and manual verification by fraud investigators. They also facilitate faster dispute resolution and internal reviews by providing ready-to-use evidence for decision justifications, bridging the gap between automated classification and human oversight.

To make this interpretability layer accessible, a Streamlit-based dashboard has been developed. The interface allows users to upload transaction datasets, trigger fraud prediction, and interactively view the model's outputs. Fraud probabilities are displayed alongside transaction IDs in a searchable and filterable table, allowing users to quickly identify high-risk transactions. The dashboard also includes integrated SHAP visualizations, enabling users to explore both global feature importance and individual prediction explanations directly within the application. Users can filter results, inspect SHAP plots, and export reports for further analysis or documentation. Additionally, the dashboard serves as a no-code interface that empowers domain experts—without programming knowledge—to explore and operationalize machine learning insights effectively.

This dual focus on interpretability and usability makes the system not only technically robust but also practically deployable in real-world financial institutions. It empowers analysts, auditors, and compliance teams to understand and trust machine learning decisions, aligning well with transparency requirements under data protection and regulatory standards. The explainability features also help build institutional trust and foster ethical AI practices by ensuring decisions are transparent, traceable, and defensible. As a result, the system addresses both predictive and governance needs, making it a suitable solution for deployment in high-stakes environments such as banking and finance.

**Real-Time Fraud Detection & Explainability**

Fill in the transaction details to check the fraud probability and explanation.

Transaction Amount	376.24	Customer Age	68
Transaction Type	Credit	Customer Occupation	Engineer
Location	Houston	Transaction Duration (seconds)	170
Device ID	D000051	Login Attempts	1
IP Address	13.149.61.4	Account Balance	13758.91
Merchant ID	M055	Transaction Date (DD-MM-YYYY HH:MM)	27-06-2023 16:44
Channel	ATM	Previous Transaction Date (DD-MM-YYYY HH:MM)	04-11-2024 08:09

Detect Fraud

**Figure 6.** Streamlit dashboard home view with data upload and prediction initiation

**Prediction:** ■ Not Fraud

Probability of Fraud: 1.00%

[DEBUG] Generating SHAP explanations...

**Reason for Prediction**

Feature	Explanation
0 TransactionAmount	TransactionAmount decreased the likelihood of fraud.
1 TransactionType	TransactionType increased the likelihood of fraud.
2 Location	Location decreased the likelihood of fraud.
3 DeviceID	DeviceID decreased the likelihood of fraud.
4 IP Address	IP Address decreased the likelihood of fraud.
5 MerchantID	MerchantID decreased the likelihood of fraud.
6 Channel	Channel decreased the likelihood of fraud.
7 CustomerAge	CustomerAge increased the likelihood of fraud.
8 CustomerOccupation	CustomerOccupation decreased the likelihood of fraud.
9 TransactionDuration	TransactionDuration decreased the likelihood of fraud.

**Figure 7.** SHAP Visualization Overview in Dashboard

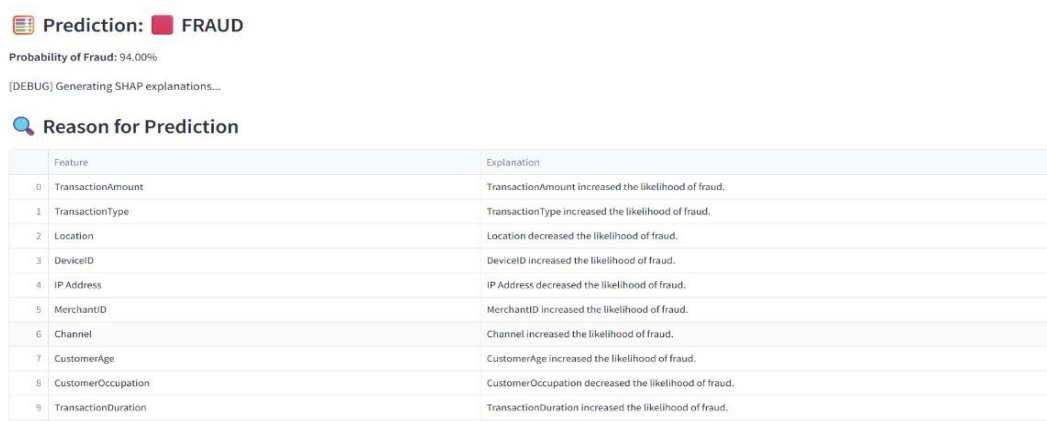


Figure 8. SHAP Visualization Overview in Dashboard

## DISCUSSION

### 4.1 Strengths and Practical Implications

The proposed system showcases how a modern fraud detection pipeline can be designed to balance **technical performance**, **explainability**, and **usability**, making it suitable for real-world deployment in financial environments. Built using widely adopted and well-maintained libraries such as **Scikit-learn**, **SHAP**, and **Streamlit**, the system ensures reproducibility, scalability, and long-term maintainability.

At the foundational level, the integration of **SimpleImputer**, **StandardScaler**, and **LabelEncoder** into the data preprocessing layer ensures that all input data is standardized, complete, and machine-readable. This is especially vital in financial domains, where input features may vary widely in scale (e.g., transaction amounts, time intervals) or contain inconsistencies due to logging errors, customer behavior, or system-level disruptions. Proper data cleaning and transformation significantly reduce noise, support model convergence, and eliminate biases stemming from poorly formatted input.

The implementation of **Isolation Forest** as the first detection mechanism introduces an important advantage: **unsupervised anomaly detection** that does not rely on pre-labeled data. In most financial institutions, only a fraction of fraudulent transactions are explicitly labeled, and many go undetected or unreported. Isolation Forest is uniquely effective in this scenario, as it identifies suspicious behavior patterns without requiring prior exposure to confirmed fraud cases. This makes the model ideal for **early-stage fraud monitoring**, internal audit flagging, or systems operating in **data-sparse environments**.

To confirm and refine the anomaly detection process, the system applies a **Random Forest classifier**, an ensemble-based supervised learning algorithm known for its robustness, stability, and resistance to overfitting. Unlike more complex algorithms such as gradient boosting or deep neural networks, Random Forest offers strong **interpretability out-of-the-box**, making it a natural choice for regulated industries where decisions must be explainable and justifiable. The classifier not only enhances the precision and recall of fraud predictions but also outputs **feature importance scores**, giving a preliminary understanding of what drives classification decisions.

A key innovation lies in the use of **SHAP (SHapley Additive exPlanations)** to augment model transparency. While traditional feature importance shows broad trends, SHAP provides **fine-grained, mathematically grounded explanations** at both the global (across all predictions) and local (per transaction) levels. It attributes each prediction to individual features, allowing analysts and investigators to trace why a transaction was flagged. This level of granularity builds **institutional trust in the system** and supports regulatory mandates such as **GDPR (General Data Protection Regulation)**, **Basel III**, and **PSD2**, which require that customers and regulators be given clear reasons for any automated decision-making that affects them.

The inclusion of a **Streamlit-based dashboard** further enhances practical usability by **democratizing access to the ML pipeline**. This no-code interface allows business users—including fraud investigators, internal auditors, and compliance officers—to upload transaction batches, view risk scores, and interpret SHAP plots without requiring Python or machine learning expertise. This significantly reduces the time-to-action and ensures that decision-making remains **auditable, visual, and efficient**. Users can filter results by fraud probability, download transaction reports for record-keeping, and explore the impact of specific features across different cases.

Finally, the modular design of the system allows for **flexible deployment across different environments**, whether on-premise for institutions concerned with data sovereignty, or in the cloud for scalability and integration with real-time data pipelines. Each component—from preprocessing to explanation—is independently replaceable or upgradable, supporting long-term maintainability and innovation.

In summary, this system is not only technically sound but also **operationally viable**, offering a fraud detection solution that meets the requirements of accuracy, transparency, and usability for real-world financial institutions.

#### 4.2 Limitations and Areas for Future Improvement

While the proposed system offers a strong foundation for intelligent and interpretable fraud detection, several **technical and practical limitations** must be acknowledged. These areas highlight opportunities for future research, optimization, and deployment enhancement.

A significant challenge encountered during model development was the **extreme class imbalance** inherent in the dataset, where fraudulent transactions accounted for only **0.17%** of the total volume. Initial attempts to address this imbalance using **SMOTE (Synthetic Minority Over-sampling Technique)** led to **overfitting**, as synthetic samples introduced artificial patterns not present in real-world fraud cases. This limitation motivated the shift to an **unsupervised anomaly detection approach (Isolation Forest)**, which circumvents the need for synthetic data. However, this also meant that legitimate but rare behaviors could potentially be misclassified as anomalies, emphasizing the need for fine-tuned threshold calibration in future iterations.

Another key limitation stems from the use of **SHAP (SHapley Additive exPlanations)** for model interpretability. Although SHAP is one of the most powerful tools available for explaining complex machine learning predictions, it comes with **considerable computational overhead**, especially in scenarios involving large volumes of transactions. Generating local SHAP explanations for thousands of records in real-time environments can strain system resources and delay response times. This restricts its practical use in high-throughput production settings such as fraud detection in payment gateways or banking APIs. Future implementations could explore **approximation strategies** such as SHAP value caching, dimensionality reduction techniques, or using **TreeExplainer in summary mode** to optimize for latency without compromising interpretability.

The reliance on **anonymized features**, particularly in the Kaggle credit card fraud dataset used for experimentation, also presents limitations. While anonymization preserves user privacy, it severely limits the semantic interpretability of feature importance outputs. For example, knowing that "V14" is a top contributor to a fraud prediction does not provide actionable insight unless the meaning of "V14" is understood. Access to **domain-specific feature names**—such as transaction type, time of day, merchant category, or device type—would greatly improve the **usability of SHAP visualizations** and strengthen model trust among business users and compliance teams. Future deployments should aim to incorporate real-world datasets with known, interpretable features, subject to appropriate data privacy agreements.

From a systems architecture standpoint, the current implementation is based on **batch processing**, making it more suitable for offline or periodic fraud analysis. However, in high-stakes environments such as real-time credit card processing or online banking, decisions must be made within milliseconds. To bridge this gap, future versions of the system should adopt a **streaming architecture** using technologies such as **Apache Kafka**, **Apache Flink**, or **Spark Streaming**. These platforms would allow the model to process continuous transaction flows, detect fraud in real time, and respond instantly to suspicious activity.

Moreover, the existing model primarily leverages **static transaction features**. Incorporating **temporal and behavioral context**—such as user spending history, session patterns, geolocation anomalies, and device



fingerprinting—could vastly improve detection accuracy. Integrating these dimensions would allow the system to move beyond transaction-level analysis toward **user-level fraud modeling**, a critical advancement for fraud prevention in digital ecosystems.

Lastly, enabling **continuous learning via feedback loops**—where the system updates itself based on investigator confirmations or user feedback—would help the model evolve with emerging fraud strategies. This could be achieved through reinforcement learning mechanisms or by periodically retraining the model on newly labeled data, allowing it to stay current with dynamic fraud trends.

## REFERENCES

- [1] K. Koo, M. Park, and B. Yoon, “A suspicious financial transaction detection model using autoencoder and risk-based approach,” *IEEE Access*, vol. **12**, pp. **68926–68939**, **2024**. <https://doi.org/10.1109/ACCESS.2024.3399824>
- [2] S. Bisht, S. Sengupta, I. Tewari, N. Bisht, K. Pandey, and A. Upadhyay, “AI-Driven Tools Transforming The Banking Landscape: Revolutionizing Finance,” *In the Proceedings of the 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*, IEEE, pp. **934–938**, **2024**.
- [3] P. Sharma, A. S. Prakash, and A. Malhotra, “Application of Advanced AI Algorithms for Fintech Crime Detection,” *In the Proceedings of the 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, pp.**1–6**, **2024**.
- [4] G. Konstantinidis and A. Gegov, “Deep Neural Networks for Anti Money Laundering Using Explainable Artificial Intelligence,” *In the Proceedings of the 2024 IEEE 12th International Conference on Intelligent Systems (IS)*, IEEE, pp.**1–6**, **2024**.
- [5] T. H. Phyu and S. Uttama, “Enhancing Money Laundering Detection Addressing Imbalanced Data and Leveraging Typological Features Analysis,” *In the Proceedings of the 2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, pp. **330–336**, **2024**.
- [6] T. H. Phyu and S. Uttama, “Improving Classification Performance of Money Laundering Transactions Using Typological Features,” *In the Proceedings of the 2023 7th International Conference on Information Technology (InCIT)*, IEEE, pp. **520–525**, **2023**.
- [7] C. Maree, J. E. Modal, and C. W. Omlin, “Towards responsible AI for financial transactions,” *In the Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, pp. **16–21**, **2020**.
- [8] Z. Chen, W. M. Soliman, A. Nazir, and M. Shorfuzzaman, “Variational autoencoders and Wasserstein generative adversarial networks for improving the anti-money laundering process,” *IEEE Access*, vol. **9**, pp. **83762–83785**, **2021**. <https://doi.org/10.1109/ACCESS.2021.3086359>
- [9] R. Desrousseaux, G. Bernard, and J. J. Mariage, “Profiling money laundering with neural networks: A case study on environmental crime detection,” *In the Proceedings of the 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, pp. **364–369**, **2021**.
- [10] Z. Ereiz, “Predicting default loans using machine learning (OptiML),” *In the Proceedings of the 2019 27th Telecommunications Forum (TELFOR)*, IEEE, pp. **1–4**, **2019**. <https://doi.org/10.1109/TELFOR48224.2019.8971110>
- [11] H. N. Mohammed, N. S. Malami, S. Thomas, F. A. Aiyelabegan, F. A. Imam, and H. H. Ginsau, “Machine learning approach to anti-money laundering: A review,” *In the Proceedings of the 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*, IEEE, pp. **1–5**, **2022**.
- [12] A. F. Mhammad, R. Agarwal, T. Columbo, and J. Vigorito, “Generative & responsible AI-LLMs use in differential governance,” *In the Proceedings of the 2023 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, pp. **291–295**, **2023**.
- [13] E. Kurshan, H. Shen, and H. Yu, “Financial crime & fraud detection using graph computing: Application considerations & outlook,” *In the Proceedings of the 2020 Second International Conference on Transdisciplinary AI (TransAI)*, IEEE, pp. **125–130**, **2020**.
- [14] A. El-Kilany, A. M. Ayoub, and H. M. El Kadi, “Detecting Suspicious Customers in Money Laundering Activities Using Weighted HITS Algorithm,” *In the Proceedings of the 2024 5th International Conference on Artificial Intelligence, Robotics and Control (AIRC)*, IEEE, pp. **112–117**, **2024**.



- [15] K. Balaji, "Artificial Intelligence for Enhanced Anti-Money Laundering and Asset Recovery: A New Frontier in Financial Crime Prevention," *In the Proceedings of the 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*, IEEE, pp. **1010–1016**, **2024**.
- [16] D. Cheng, Y. Ye, S. Xiang, Z. Ma, Y. Zhang, and C. Jiang, "Anti-money laundering by group-aware deep graph learning," *IEEE Trans. Knowl. Data Eng.*, vol. **35**, no. **12**, pp. **12444–12457**, **2023**. <https://doi.org/10.1109/TKDE.2023.3272396>
- [17] D. V. Kute, B. Pradhan, N. Shukla, and A. Alamri, "Deep learning and explainable artificial intelligence techniques applied for detecting money laundering – A critical review," *IEEE Access*, vol. **9**, pp. **82300–82317**, **2021**. <https://doi.org/10.1109/ACCESS.2021.3086230>
- [18] D. V. Kute, B. Pradhan, N. Shukla, and A. Alamri, "Explainable deep learning model for predicting money laundering transactions," *Int. J. Smart Sens. Intell. Syst.*, vol. **17**, no. **1**, **2024**. <https://doi.org/10.2478/ijssis-2024-0027>
- [19] O. Kuiper, M. van den Berg, J. van der Burgt, and S. Leijnen, "Exploring explainable AI in the financial sector: Perspectives of banks and supervisory authorities," *In the Proceedings of the Artificial Intelligence and Machine Learning: 33rd Benelux Conference on Artificial Intelligence, BNAIC/Benelearn 2021, Esch-sur-Alzette, Luxembourg, Nov. 10–12, 2021, Revised Selected Papers*, vol. **33**, pp. **105–119**, Springer International Publishing, **2022**.
- [20] D. Vijayanand and G. S. Smrithy, "Explainable AI-enhanced ensemble learning for financial fraud detection in mobile money transactions," *Intelligent Decision Technologies*, **2024**, Art. no. **18724981241289751**.
- [21] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, and J. Zhu, "Explainable AI: A brief survey on history, research areas, approaches and challenges," *In the Proceedings of the Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, Oct. 9–14, 2019, Part II*, vol. **8**, pp. **563–574**, Springer International Publishing, **2019**.
- [22] R. Alhajeri and A. Alhashem, "Using Artificial Intelligence to Combat Money Laundering," *Intelligent Information Management*, vol. **15**, no. **4**, pp. **284–305**, **2023**. <https://doi.org/10.4236/iim.2023.154014>
- [23] S. K. Hashemi, S. L. Mirtaheri, and S. Greco, "Fraud detection in banking data by machine learning techniques," *IEEE Access*, vol. **11**, pp. **3034–3043**, **2022**. <https://doi.org/10.1109/ACCESS.2022.3232287>
- [24] E. R. Mill, W. Garn, N. F. Ryman-Tubb, and C. Turner, "Opportunities in real time fraud detection: An explainable artificial intelligence (XAI) research agenda," *International Journal of Advanced Computer Science and Applications*, vol. **14**, no. **5**, pp. **1172–1186**, **2023**. <https://doi.org/10.14569/IJACSA.2023.01405121>
- [25] I. Psychoula, A. Gutmann, P. Mainali, S. H. Lee, P. Dunphy, and F. Petitcolas, "Explainable machine learning for fraud detection," *Computer*, vol. **54**, no. **10**, pp. **49–59**, **2021**. <https://doi.org/10.1109/MC.2021.3081249>
- [26] J. Vidya Sagar and S. Aquter Babu, "A Hybrid Machine Learning Approach for Real-Time Fraud Detection in Online Payment Transactions," *Library Progress International*, vol. **44**, no. **3**, pp. **26067–26090**, **2024**.
- [27] Anirban Majumder, "Intelligent AI Agents for Fraud and Abuse Detection", *International Journal of Computer Science and Engineering*, Vol.**12**, Issue.**4**, pp.**1-7**, **2025**.
- [28] Avinash Malladhi, "Artificial Intelligence and Machine Learning in Forensic Accounting", *International Journal of Computer Science and Engineering*, Vol.**10**, Issue.**7**, pp.**15-21**, **2023**.
- [29] Latha N. R., Shyamala G, Pallavi G. B., Sneha Santhosh Bhat, Archit Mehrotra, Ashish Seru, Tanisha Gotadke, "A Machine Learning Framework for Financial Fraud Detection Using Explainable Artificial Intelligence Techniques", *International Journal of Computer Sciences and Engineering*, Vol.**13**, Issue.**5**, pp.**01-05**, **May 2025**.