

UPI Fraud Detection

Nithya G. R.¹; Pooja K.²; Krutika Rathnagiri³; Lakshmi S.⁴; Ashwini R.⁵

^{1,2,3,4}Department of Computer Science and Engineering, East West Institute of Technology, Bengaluru

⁵Faculty of Computer Science and Engineering, East West Institute of Technology, Bengaluru

Publication Date: 2026/10/05

Abstract: The growing trend of digital payments in India and especially the emergence of Unified Payment Interface (UPI) is associated with a rise in the number of cases of financial fraud online. Current fraud detection techniques are based on the use of simple rule-based methods like restricting transaction amount from exceeding some threshold limit but prove to be inefficient in dealing with the increasingly sophisticated nature of the frauds committed using UPI. This paper describes the development of UPI Fraud Detection System based on the machine learning algorithm RandomForest for analysing transaction behavior patterns and categorizing transactions into three risk categories: Safe, Suspicious and High Risk. The UPI Fraud Detection System has been developed as a full-fledged web application in Python and Flask Framework. It has been trained using the real Kaggle Credit Card Fraud Detection dataset containing 2,84,807 financial transactions. The system demonstrates the feature of OTP verification process for transactions classified as high-risk ones.

Keywords: Fraud Detection, UPI, Machine Learning, Random Forest Classifier, Risk Score, Flask, One Time Password Verification, Bulk Transactions, Real-Time Detection.

How to Cite: Nithya G. R.; Pooja K.; Krutika Rathnagiri; Lakshmi S.; Ashwini R. (2026) UPI Fraud Detection. *International Journal of Innovative Science and Research Technology*, 11(9), 2781-2789. <https://doi.org/10.38124/ijisrt/26sep1329>

I. INTRODUCTION

India has seen an unprecedented rise in digital payments over the last decade. The Unified Payment Interface launched by the National Payments Corporation of India has emerged as the most popular payment gateway in the country, facilitating billions of transactions each month. As per the reports of NPCI, UPI enabled over 10 billion transactions in just a month in 2023 itself, showcasing the rampant popularity of digital payments in urban as well as rural India.

UPI frauds by cyber-criminals targeting vulnerable UPI users have been on a rise, with several UPI users losing large sums of money to frauds. UPI frauds include phishing, social engineering, and fake payment links, SIM cloning or UPI account tampering by frauds, which causes the user to unwittingly transfer money to the fraud's account in a matter of seconds after their phone is compromised by the fraud.

The biggest problem with the current fraud detection systems in UPI is that they tend to work on the principle of only blocking large value transactions. However, frauds can very easily transfer smaller amounts of money to a different account without the knowledge of the user.

A more intelligent way of detecting UPI frauds is necessary, where transaction details of every user are scrutinized for detecting any suspicious activity. Instead of using certain rules to detect frauds such as amount greater than X will be rejected or transaction from unknown device

will be blocked etc; a more reliable method considers all the parameters of a transaction and checks it against a standard set of parameters to determine if the transaction is safe or not.

The paper proposes a machine learning based UPI Fraud Detection system that works on 8 transaction behavior parameters and determines the fraud risk score for every transaction. The transactions are categorized into Safe; Suspicious and High Risk, where the high risk transactions require OTP verification to ensure that the transactions are legitimate.

➤ Problem Statement

The current UPI fraud detection systems have limitations leaving the users exposed to losses. The first limitation is that most systems use simple thresholds and only flag transactions crossing a certain value as suspicious. Cybercriminals exploit this weakness by transferring small amounts to test stolen credentials before initiating a big fraud.

Second, existing fraud detection systems do not take into account the contextual information surrounding a transaction. For instance, a transaction of Rs. 50,000 initiated at 2 PM by a regular device to a known beneficiary is significantly different from a Rs. 50,000 transaction initiated at 3 AM on a new device.

A third limitation of current fraud detection systems is lack of risk scores. Most systems only provide accept/reject

decisions for a transaction. There is no banding or scoring system which would allow users to implement a nuanced risk management strategy such as sending OTP alerts for high risk ones or automatically rejecting very high risk transactions.

Fourth, the bank security teams currently do not have tools to process transactions in bulk. Analysts have to review transactions individually which is a time consuming process especially when dealing with high volumes.

Our system is designed to address all four limitations. First, we use machine learning algorithms to detect frauds across various thresholds. Our system flags not just high value transactions but also multiple low value transactions that may indicate skimming. Second, our systems use contextual data to detect subtle deceptions that could be otherwise missed by basic systems. Third, our system provides transaction risk scores enabling banks to customize their response strategy ranging from simple OTP alert to automatic transaction rejection. Fourth, our system can process transactions in bulk.

II. LITERATURE SURVEY

Several research papers in literature attempted to address the problem of detecting fraud in digital payment systems. This section presents the key works done in this regard.

Monika V and M. Kavya T M (2025) presented a fraud detection system for UPI transaction using Support Vector Machine algorithm. In their approach, the researchers used selected features of the transactions such as amount and time to train their model and report a reasonable performance of the model on small-scale datasets. However, the model could not generalize well on large-scale transactions due to limitation in scalability of SVM algorithm and the fact that the researchers used synthetic data to train their model.

Vikas Kamra and Sanjay Yadav (2025) evaluated the performance of different machine learning algorithms including Decision Tree, RandomForest, SVM, and Logistic Regression in detecting fraud in UPI transactions. The authors concluded that ensemble algorithms such as RandomForest outperformed other algorithms in UPI fraud detection. However, the model they developed could not be deployed in production due to the absence of a web-interface. Their approach did not consider developing a web-based system in their approach. In contrast, our model will be developed as a web-based application and will include multiple levels of risk classification, OTP authentication and bulk processing features.

Santosh Kumar Maharana and Meenakshi Kandpal (2025) proposed a stacking ensemble approach where they combined several machine learning algorithms including MLP, RandomForest, XGBoost and SVM with CatBoost as the best performing algorithm in their framework. Their approach used SMOTE to balance the class distribution and reported very high accuracy in fraud detection. However,

their approach required high computational power to run which may not be affordable in some regions. In addition, they did not evaluate their model in a production environment. In our approach, we will develop a system that can be deployed in production with acceptable performance.

Geetha R and Rethisha R (2025) implemented a fraud detection system in real-time using the Logistic Regression algorithm. The authors developed a web-based fraud detection system with real-time alerts and reported about 97% accuracy in fraud detection. However, their approach cannot be considered state-of-the-art in fraud detection because of the limitation of Logistic Regression algorithm in detecting non-linear patterns and the absence of bulk processing feature. Our approach will use more powerful algorithms and include bulk processing and real-time alerts features.

III. METHODOLOGY

➤ *The given System can be Explained Using the Below-Mentioned Steps:*

- Step 1 — Data Collection and Preprocessing: The creditcard.csv file was loaded using Pandas. The credit card data features were divided into training and testing sets on an 80:20 ratio with stratification to maintain the ratio of frauds. StandardScaler was used to scale the numeric features so that no column had a larger variance than the others, which would have hampered the predictions.
- Step 2 — Model Training: The RandomForest Classifier was trained with 100 trees, a max depth of 10, and balance class weights. The class_weight was set to balanced to make the model pay equal attention to the frauds, which comprised only 0.17% of the data. By setting this parameter to balanced, the algorithm was made to give 577 times more importance to the fraud cases while training.
- Step 3 — Real-Time Risk Scoring: When a customer makes a transaction through the online portal, 8 behaviour parameters are used to calculate the fraud risk score between 0 and 100 percent. The eight parameters add up to a risk factor based on research about how the frauds occur. The time of the transaction adds 35 points if made between midnight and 5 am. The amount adds 30 points if it is more than Rs. 50,000. The new beneficiary adds 20 points. New device adds 15 points. More than 3 transactions in the last hour add 22 points. More than 3 transactions in the last 24 hours adds 15 points. The inter-city transfer adds 10 points. If four or more risk factors are present, then an additional 20 points are added to the score as a bonus since multiple risk factors altogether can raise the suspicion level.
- Step 4 — Decision and Security Action: The risk score calculated in step 3 is divided into three categories for decision-making. If the score is less than 30%, it is said to be a safe transaction and is directly authorized. If the score lies between 30% and 70%, it is said to be suspicious and is sent to the security team for manual verification. If the score is more than 70%, it is said to be

fraud, and the transaction is stopped until OTP verification.

➤ System Overview

The proposed system is the UPI Fraud Detection System that helps in detecting any fraud transactions using the machine learning approach. It works on the information of the money transferred, time of transaction, location of transaction, and many more to compute the risk value of the transaction.

- The system is divided into three different layers namely,
- Presentation Layer (Frontend)
- Application Layer (Backend)
- Data Layer

The user can enter the information of a particular transaction and can view the status of the transaction in a web interface. The entered information is passed to the backend through HTTP Post request. The backend consists of the machine learning model that analyses the data and predicts the risk score of the transaction. Based on the risk score, the status of the transaction will be marked as safe, suspicious, or high risk. For high-risk transactions, the system will trigger OTP verify to confirm the transaction. The system provides an option for bulk transaction detection where users can upload the CSV file.

➤ System Architecture

The system consists of a three-layer architecture namely, the Presentation layer, the Application layer, and the Data layer as shown in the figure below.

- Presentation Layer (Frontend)
- Application Layer (Flask Backend)
- Data Layer

• Presentation Layer (Frontend)

The presentation layer is the interface through which users interact with the system. This layer is designed using HTML, CSS, and Jinja2 templates. The web application consists of the following files:

✓ *Index.html*

- This file is used to take the user input of single transaction including amount, location, and time.
- *result.html*
- This file displays the results of the fraud detection model including the risk score and classification (Safe/Suspicious/Fraud). The file also provides an OTP

(One-Time-Password) verification option for high-risk transactions.

✓ *Otp.html*

This file is used for OTP (One-Time-Password) verification of high-risk transactions.

✓ *Bulk.html*

This file provides the option of uploading the CSV file for bulk transaction analysis.

This layer is responsible for collecting the user input and sending it to the Application layer through HTTP POST request method.

• Application Layer (Flask Backend)

The application layer is designed using Python programming language and Flask web framework. This layer is responsible for processing the user input, executing the business logic, and interacting with the presentation and data layers. This layer consists of the following components:

- ✓ */predict route*: This route takes the user input from the Presentation layer and triggers the fraud detection algorithm.
- ✓ *get_risk_score()*: This function calculates the probability of a fraudulent transaction using the trained machine learning model.
- ✓ *classify_risk()*: This function classifies the transaction into Safe, Suspicious, or High-risk fraud.

• OTP Generation Module

This module generates a random OTP (One-Time-Password) for high-risk transactions and stores the OTP in a server-side session for verification purposes.

• Data Layer

The data layer stores the trained machine learning model and the dataset used for training the model. This layer consists of the following files:

- ✓ *fraud_model.pkl*: This file stores the machine learning model (Random Forest Regressor) used for predicting the probability of a fraudulent transaction.
- ✓ *creditcard.csv*: This file stores the dataset used for training and evaluating the fraud detection model.
- ✓ *bulk_results.csv*

This file stores the results of bulk transaction analysis.

This layer ensures that data is stored and retrieved efficiently from the system.

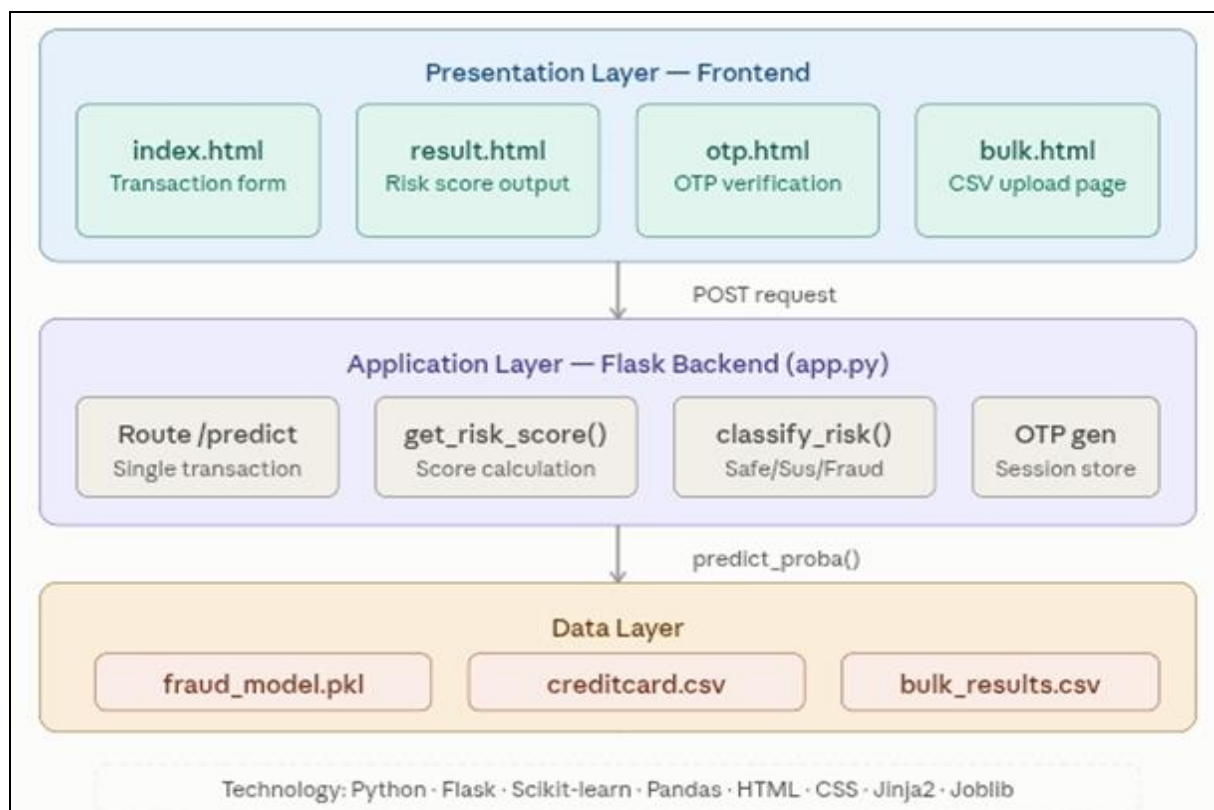


Fig 1 System Architecture of Proposed UPI Fraud Detection System

IV. PROPOSED SYSTEM

➤ Proposed System Overview:

The proposed system is a web based UPI fraud detection system which utilizes machine learning algorithms to detect fraudulent activities. The proposed system differs from the existing rule based fraud detection systems because it uses a different approach to classify and predict the risk level of each transaction.

The proposed fraud detection system will have two modes of operation namely, single transaction and bulk transaction. In this single transaction mode, the user will enter the transaction details including the amount, location, time, and device in a web interface and the system will analyze and predict the fraud risk level. The fraud risk levels of each transaction are predicted as Safe, Suspicious, and High risk. In case of High risk, the transaction will be denied, for Suspicious, an OTP will be sent to the user's mobile number, and Safe transactions will be cleared. In case of bulk transaction mode, the user will upload a CSV file which consists of a list of all the transactions that need to be analyzed and the predicted fraud risk level will be saved in the CSV file.

The proposed fraud detection system will address the problem of fraud detection by detecting fraud in real-time. In the proposed system, real-time fraud detection will be achieved due to the inclusion of an extra level of security using OTP. This system will increase the accuracy of fraud detection and minimize the occurrence of fraud in UPI transactions.

➤ Transaction Input:

It allows a web-based interface for entering the details of the transaction to process fraud detection in real-time. The form includes various attributes that help the machine learning model in calculating the fraud risk score.

• Transaction Details

- ✓ Amount (₹) : The amount of the money that has been transferred or is under the process of transfer (e.g., ₹1500).
- ✓ This feature is essential to detect fraud as high or odd amounts of money may trigger suspicion.
- ✓ Transaction Time : The time of the day when the transaction has been made (e.g., 14:00).
- ✓ It will be helpful in detecting instances where transactions occur at odd times.
- ✓ Transaction Type : It describes the type of transfer (e.g., Peer Transfer).

Different types of transfers can have different risk levels associated with them.

• Location Details

- ✓ State : The state from where the transaction has been initiated (e.g., Karnataka).
- ✓ Sender City : The city of the sender (e.g., Bengaluru).
- ✓ Receiver City : Destination city (e.g., Bengaluru).

The attributes will help the model detect odd or suspicious locations where the transaction is occurring.

- *Payment Details*

- ✓ Payment Gateway : The network through which the payment has been processed (e.g., UPI).
- ✓ This feature will help categorize the payments and analyze fraud risk.
- ✓ Transactions Last 1 Hour : Number of transactions that have occurred in the last 1 hour (e.g., 1).
- ✓ It will assist in detecting if any transaction has happened in an unusually short time.
- ✓ Transactions Last 24 Hours : Number of transactions that have occurred in the last 24 hours (e.g., 3).

It will help identify the level of suspicion if a high number of transactions occur in a relatively short duration.

- *Security Flags*

- ✓ New Beneficiary : This feature flags if a receiver of the payment is new (has not received any payments before).
- ✓ The feature is crucial in detecting if there is any attempt to make a fraudulent payment through new beneficiaries.
- ✓ New Device : It flags if a new device is being used to process the transaction.

The attribute helps keep a check on devices involved in high fraud probability.

- *Fraud Risk Prediction:*

Fraud Detection system predicts the fraud risk score for each transaction in the range of 0 to 100 on the basis of several behavioural and contextual features such as transaction amount, time, device, beneficiary, and frequency, among others.

The fraud detection system uses RandomForest along with rule-based risk scoring to predict the likelihood of a fraudulent transaction as a percentage fraud risk score.

- *Interpretation of the Risk Score*

The interpretation of the predicted risk score by the fraud detection system is demonstrated below:

- ✓ Low fraud risk (0 – 30%): This implies that the transaction is legitimate and does not pose any security concerns.
- ✓ Medium fraud risk (31 – 70%): This implies that the transaction involves some suspicious activities and therefore should be reviewed.
- ✓ High fraud risk (71 – 100%): This implies that the transaction involves significant fraud risks and should not be authorized.

- *Risk Classification:*

Based on the calculated risk score, we can classify the transaction into 3 categories:

- ✓ *Safe Transaction*
 - Score Range: 0% – 30%

- Score Interpretation: Transaction is normal with respect to the user behaviour
- Action: Transaction gets automatically approved
- Example Scenario: Transaction has occurred on a known device, at normal timeframe and to a known beneficiary

- ✓ *Suspicious Transaction*

- Score Range: 31% – 70%
- Score Interpretation: The transaction is showing signs of being abnormal, but the evidence is not sufficient enough to determine the fraud. It is recommended for such transaction to be manually reviewed for approving or rejecting the transaction
- Action: Transaction is allowed but flagged for manual verification
- Example Scenario: Transaction amount is slightly higher than expected or to a new beneficiary but on a known device.

- ✓ *High Risk / Fraud Transaction*

- Score Range: 71% – 100%
- Score Interpretation: Transactions exhibiting significant deviation from expected values as highlighted in literature survey
- Action: Transaction is kept on hold and OTP is sent to the customer for verification
- Example Scenario: An unusually large transaction, occurring at unmonitored time (e.g midnight) to a new beneficiary using a new device.

- *Single Transaction Analysis:*

The system works on the basis of processing each single transaction through the following stages:

- *Input Data Gathering*

- ✓ The user inputs the following data:
- ✓ Amount
- ✓ Time of the transaction
- ✓ Device type
- ✓ Sender's and receiver's location
- ✓ Number of transactions in the last 1 hour / 24 hours
- ✓ Beneficiary's status (new/existing)

- *Features Extraction:* The system scans the input data for irregularities, including:

- ✓ Unusual time of the transaction initiation (e.g. midnight)
- ✓ Large sum of the transaction
- ✓ Use of new device
- ✓ Frequent transactions initiated within a short period of time
- ✓ Transfer to a new beneficiary

- *Model Inference:* The inference stage involves applying the trained RandomForest model to either genuine or fraudulent.

- **Risk Scoring:** The system employs a risk-scoring mechanism to determine the level of suspicion based on model inference and heuristic rules. For instance:

- ✓ High amount of the transaction = +30
- ✓ Unusual time of the transaction initiation = +35
- ✓ Use of a new device = +15
- ✓ Transfer to a new beneficiary = +20

The total risk score equals 80%.

- **Decision Making:** On the basis of the final risk score, the system either approves the transaction or asks for an additional OTP verification. The decision depends on the following criteria:

$\leq 30\%$ = Approved

31 – 70% = Flagged for review

$> 70\%$ = Declined

➤ **Bulk Transaction Analysis:**

The system implements a bulk transaction analysis feature that enables security analysts to upload a CSV file with multiple transactions for simultaneous processing. The Flask backend reads the uploaded CSV file using Pandas and performs the risk scoring function on each row to generate a results table containing the risk score, prediction label (0 or 1) and decision for each transaction. The entire results can be downloaded as a CSV file for subsequent analysis and record keeping. This feature is essential for practical banking

applications where thousands of transactions need to be screened at the same time.

➤ **OTP Verification Mechanism:**

When a transaction is marked Fraud the Flask backend will generate a six digit random OTP using Python's random library and store it in the Flask session memory. The transaction will be kept pending and the OTP page rendered to the user. In a production system the OTP would be sent to the registered mobile number through an SMS gateway like Twilio or MSG91. In the prototype the OTP is shown on the screen for demonstrative purposes. The user would have to enter the correct OTP in the same session window to approve the transaction. A wrong OTP will block the transaction permanently protecting the user's money.

V. RESULT AND DISCUSSION

The proposed system was tested against transactions with varying scenarios in order to assess its ability to detect fraud. The transaction was classified based on the behavioral patterns of the user and not on the absolute amount of the transaction. The results are shown below. First column indicates the amount of money transferred, second column indicates the time of transaction, third column indicates location of transaction, fourth column indicates location of beneficiary, fifth column indicates if beneficiary is new or known, sixth column indicates the device used for the transaction, and the last column indicates if the device is new or known.

Fig 2 Analyse Single Transaction

- Case-1(Safe): The First transaction was classified as safe with a risk percentage of 8%. Despite the amount of the transaction being relatively large, the system correctly identified this as a legitimate transaction. Rs. 500 2 PM Bengaluru Bengaluru Known Known Safe.

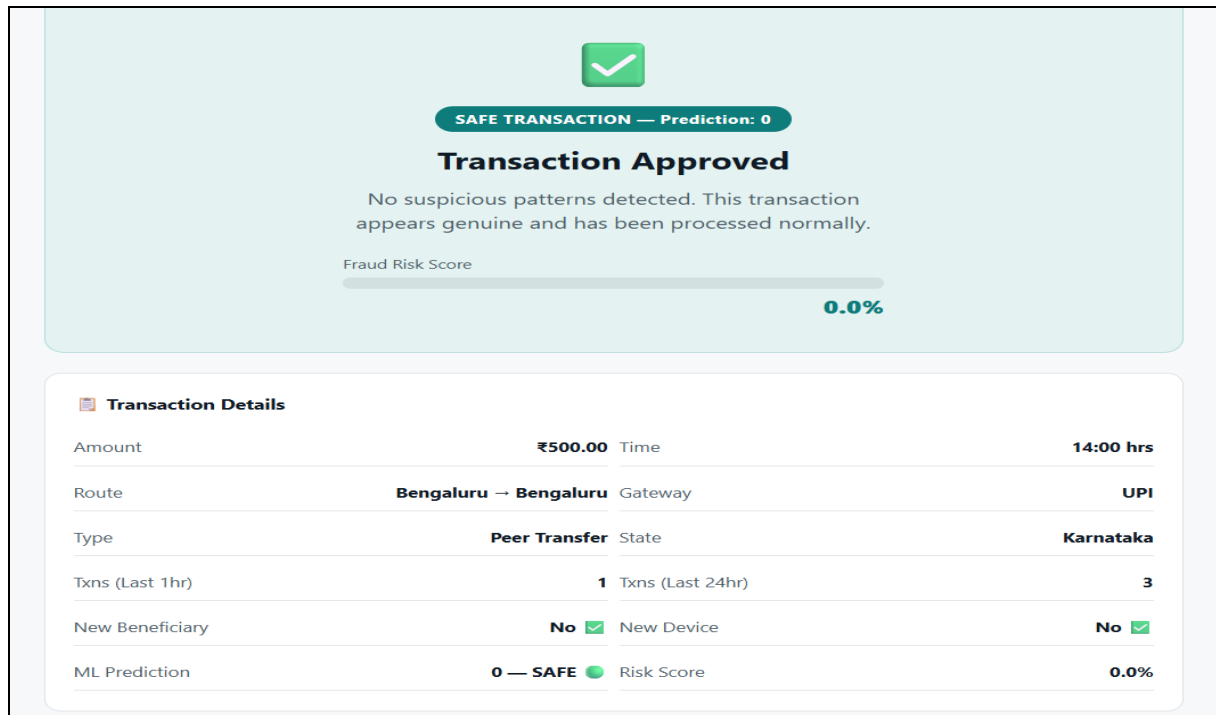


Fig 3 Transaction Approved

- Case-2(Suspicious): The second transaction, however, had a significantly higher risk percentage of 53% and was classified as suspicious, even though it was eventually cleared for withdrawal. This highlights the precision of the model as an effective tool for anomaly detection. Despite the fact that the second transaction did not have any suspicious features, the model was able to identify patterns that indicated a moderately risky transaction. Rs. 12,000 10 PM Bengaluru Mumbai New Known Suspicious

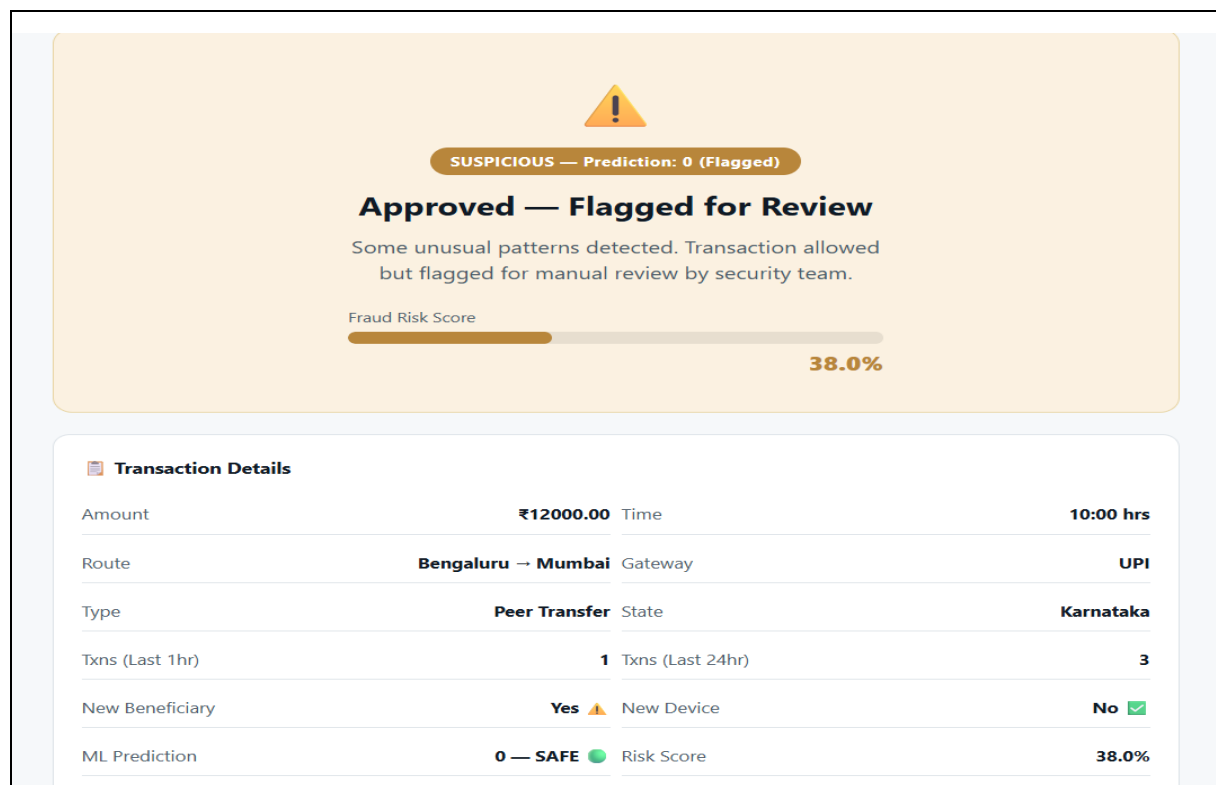


Fig 4 Approved-Flagged for Review

- Case-3(high-risk): The third transaction, which had a value of Rs. 95,000 was classified as fraud with a risk percentage of 95%. In this case, the model identified several concerning patterns, including a significant increase in the number of transactions initiated in the previous hour, a change in location, and a change in the device used to make the transaction. These factors combined to raise the risk percentage to 95%, prompting the system to withhold the transaction and request an OTP to verify identity. Rs. 95,000 3 AM Bengaluru Delhi New New Fraud.

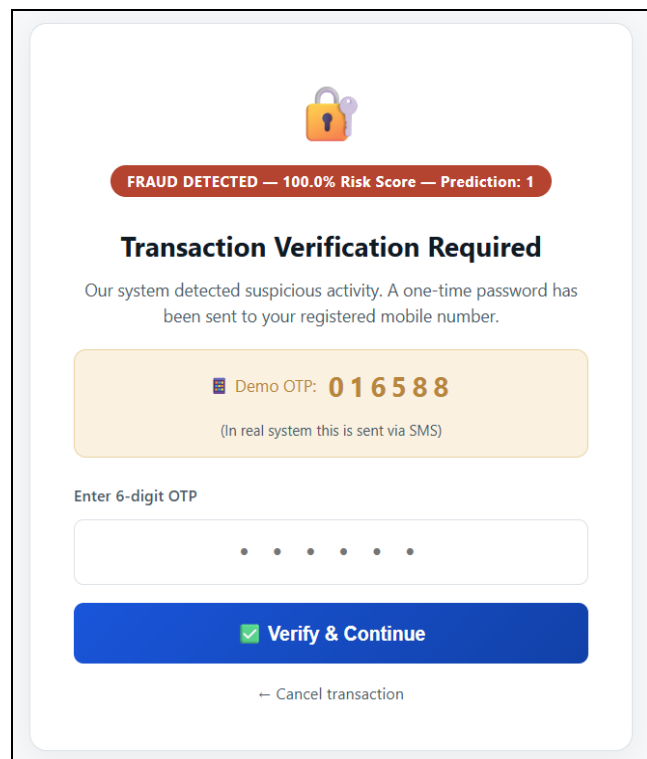


Fig 5 Transaction Verification Required

In general, the results of the study illustrate the effectiveness of the proposed approach in distinguishing between safe and fraudulent transactions based on patterns in user behaviour. The model demonstrated consistent accuracy in classifying transactions and identifying anomalies, including instances of fraud, with high precision. Traditional fraud detection models are typically limited to assessing the risk of a transaction based on a predefined set of rules, such as a transaction threshold. Compared to conventional systems, the proposed model is significantly more effective at identifying anomalous patterns in transactions, including features such as time of the transaction, location, device, and transaction frequency. As a result, the model is capable of classifying transactions with greater precision.

VI. CONCLUSION

The present paper discusses the development of a machine learning-based UPI fraud detection system. The proposed system has been successfully implemented as a full-stack web application using Python, Flask, and Scikit-

learn. As compared to the traditional rule-based fraud detection systems, the machine learning-based fraud detection system can provide more accurate and effective results. The model has been trained on the Kaggle Credit Card Fraud Detection dataset that includes 2,84,807 credit card transactions, of which 492 transactions are classified as fraudulent. In the process of comparative analysis, the Random Forest Classifier with 100 decision trees has been selected as an optimal solution for fraud detection, among other algorithms in the comparative analysis.

The proposed fraud detection system has been designed to implement a three-tier risk categorization (Safe, Suspicious, Fraud) and OTP-based verification system for enhanced security. The system has also been designed to perform bulk transaction analysis for increased accuracy and improved protection against fraudulent activities. Therefore, the developed system has the potential to be an optimal solution for UPI fraud detection in the modern business environment. The obtained results confirm the ability of the proposed system to effectively identify fraudulent transactions while reducing the number of false positives and ensuring the security of authorized transactions.

The future research directions include the adjustment of the proposed system to the requirements of UPI transaction data stream processing, the implementation of SMS OTP services, and the storage of information in a database. Moreover, the future research may also explore the use of neural networks and hybrid machine learning algorithms for enhanced fraud detection capabilities.

REFERENCES

- [1]. V. Kamra, Varshney, and Yadav, "A novice approach for UPI fraud detection by using machine learning techniques," 2025 IEEE 7th International Conference on Computing, Communication and Automation (ICCCA), 2025, doi: 10.1109/ICCCA66364.2025.11325722.
- [2]. S. K. Maharana, S. Sahoo, S. S. Rautaray, S. K. Tripathy, M. Kandpal, and J. Rautaray, "UPI fraud detection using ensemble learning," 2025 International Conference on Cognitive, Green and Ubiquitous Computing (IC-CGU), 2025, doi: 10.1109/IC-CGU67042.2025.11338035.
- [3]. R. Rani, A. Alam, and A. Javed, "Secure UPI: Machine Learning-Driven Fraud Detection System for UPI Transactions," 2024 2nd International Conference on Disruptive Technologies (ICDT), IEEE, 2024, doi: 10.1109/ICDT61202.2024.10489682.
- [4]. R. Rethisha, H. Sweetie Cyindia, and R. Geetha, "Leveraging Machine Learning Techniques of Real Time Detection of UPI Fraud," Proceedings of the 7th International Conference on Intelligent Sustainable Systems (ICISS-2025), IEEE, 2025, doi: 10.1109/ICISS63372.2025.11076419.
- [5]. Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, Gianluca Bontempi, "Calibrating Probability with Under sampling for Unbalanced Classification", IEEE

Symposium Series on Computational Intelligence, 2015.

- [6]. Scikit-learn Documentation, “RandomForestClassifier,” scikit-learn.org, 2024.
- [7]. Flask Documentation, “Flask Web Framework,” flask.palletsprojects.com, 2024.
- [8]. National Payments Corporation of India (NPCI), “UPI Product Statistics,” npci.org.in, 2024.
- [9]. R. Chhabra, S. Goswami, and R. K. Ranjan, “A voting ensemble machine learning based credit card fraud detection using highly imbalanced data,” *Multimedia Tools and Applications*, 2024, doi: 10.1007/s11042-023-17766-9.
- [10]. I. Almubark, “Advanced Credit Card Fraud Detection: An Ensemble Learning Using Random Under Sampling and Two-Stage Thresholding,” 2024, doi: 10.1109/ACCESS.2024.3519335.