



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IX **Month of publication:** September 2026

DOI: <https://doi.org/10.22214/ijraset.2026.84855>

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

A Hybrid Machine Learning Framework for Real-Time Financial Transaction Fraud Detection

Dr. T. Rajesh¹, I. Nikshitha Raj², R. Manaswini³, Puram Ananya⁴

¹Associate Professor, CSE, GNITS, Hyderabad, India

^{2, 3, 4}Student, CSE, GNITS, Hyderabad, India

Abstract: Digital Transactions have certainly made our life easier, but at the same time it makes us susceptible to many threats including misuse of UPI, fraudulent refund, phishing, account hacking, and many others. The traditional rule-based system works according to predefined rules and is unable to cope with changing fraud trends, large number of transactions, false positives, and delay in the detection process. In this paper, we present Transaction Guardian, a real-time domain independent digital fraud detection system, where rule-based fraud engines, machine learning, behavioral analysis, and real time monitoring dashboard are employed. The designed system leverages Random Forests for supervised fraud detection, Isolation Forest for unsupervised anomaly detection along with velocity and geospatial analysis for the detection of burst transactions and impossible travel. Machine Learning module is exposed as a Flask service integrated with the Node.js/Express backend, MySQL for persistent storage, and React/TypeScript frontend. Real time communication is handled using Socket.IO library. Experiments carried out using 30,000 anonymized historical transaction dataset and live transaction feed prove that the proposed solution is capable of operating in real time, has reliable performance, high fraud-detection efficiency, and lower false positives.

I. INTRODUCTION

Today, thanks to the emergence of different types of digital financial services, such as online banking, e-commerce and mobile payments, the money transfer industry is going through revolutionary changes. However, along with the growth in the amount and speed of digital transactions, the risk of being involved in fraudulent actions grows as well. Criminals exploit vulnerabilities in payment systems in order to conduct various kinds of fraud that involve unauthorized access to the accounts, suspicious transactions, phishing, fraudulent refunds, account breaches and so on. Therefore, as follows from the provided project documentation, fraud detection is an essential process that would allow avoiding dependency on post-factum investigation.

Traditional fraud detection techniques are based on pre-developed rules that operate according to the criteria of the transaction amount, location and other necessary parameters. Such systems are both easy and effective when it comes to known patterns of fraud, however, they do not solve the problem of prevention of newly appeared frauds. Moreover, such solutions tend to generate false positive alerts that negatively affect the users' experience and the work of analysts.[1], [2] The solution presented in this project solves all the mentioned problems by means of hybrid architecture that includes rule-based detection, supervised

II. LITERATURE REVIEW

Fraud detection process evolves over time from the expert defined rules to statistical learning, anomaly detection and hybrid methods. The directions considered in project literature review include rule based methods, machine learning approaches, hybrid fraud detection, real time processing and privacy preserving techniques.

A. Rule-Based Fraud Detection

A rule based approach is one that considers transaction attributes based on the pre-existing set of rules. The velocity rules will help detect high velocity transactional activities while the location rules will help find any sudden and unexpected changes in transaction location. It makes the calculations easy but may get complex as time goes by. [2]

B. Machine Learning Methods

Machine learning methods include techniques that learn patterns from the transactions rather than using pre-defined manual rules. In the project we will focus on supervised machine learning algorithms such as Random Forests for prediction and Isolation Forests for discovering non-standard behaviors without defining fraud label for each behavior. The anomaly detection is particularly useful when there is need to identify some unknown activity. [3], [4]

The information contained in financial transactions is very private. In this regard, project documents pay attention to the importance of anonymization, masking, tokenization, authentication, and encryption. [5]

The architecture of the system represents a multi-tier approach with React/TypeScript frontend, Node.js/Express backend, Flask machine-learning engine, MySQL database, hyped fraud engine, and Socket.IO communications book. The architecture diagram of the project illustrates how transactions get received by the front-end/back-end, and how machine-learning engine and database assist the fraud engine.



The process starts with creation of the transaction from the dataset or the simulator engine. The backend receives the transaction and performs a decision according to the rule-based process using the fraud engine. After that, the transaction proceeds to the machine-learning module for detecting anomalies. Both results of rule-based and machine-learning modules are then aggregated to compute the risk score. As a final step, the transaction, the score, and the actions are saved into the MySQL database and the result is delivered to the frontend using Socket.IO. An analyst may choose whether to accept, confirm, or block the transaction.

D. Fault-Tolerant Detection

The machine-learning module has been implemented as a separate Flask app. In case of the failure of the service, the backend can apply the rule-based scoring module to make sure that the transaction will be processed.

IV. METHODOLOGY

E. Dataset Description

The developed solution works with the historical_transactions_30k.csv file, which comprises approximately 30,000 records. The data set is anonymized and is intended to imitate the process of receiving a stream of transactions in a constant manner. The data set includes user identity, transaction amount, timestamp, location, merchant category, transaction frequency, risk score, and a fraud label. The system reads the data into the memory and creates a stream of records by repeating the loop.

F. Feature Selection

The main features that are selected for detection of fraud are transaction amount, timestamp, location, transaction frequency, and merchant category. The feature transaction frequency allows performing velocity analysis, whereas timestamp and location features allow detecting suspicious movement. Transaction amount and merchant category features give some data on deviations in transactions.

G. Transaction Simulation

At the start-up of the system, the data set is read into the memory. A constant loop generates transactions every 2.4 seconds. The WebSocket/Socket.IO events are used to send a new transaction into the application.

H. Velocity Analysis

This analysis is used to detect the rate at which the transactions occur within a certain period of time. A quick succession of transactions may suggest that the transactions are automated or that there are some abnormalities. The fraud detection engine raises the risk score assigned to transactions based on such activities.

I. Geospatial Validation

In this case, the geographic location of transactions serves as a factor for calculation of the distance between these two geographic locations. In the case when this distance provides some unreasonable speed of traveling, then this transaction pattern will be considered and will be used in the risk scoring algorithm.

1) Random Forest

This algorithm belongs to the supervised learning part of the project. Random forest predicts the probability of fraud on the basis of transaction history data. In this case, the prediction process is based on multiple decision trees.

2) Isolation Forest

The isolation forest algorithm is used for anomaly detection without any prior training of the algorithm. Contrary to other methods where all anomalous transaction cases are marked, the algorithm finds those cases that differ from other transaction cases. Results of this algorithm will be another feature for defining risk of the transaction. [8]

3) Risk Scoring and Adaptive Response

Transaction risk scoring is based on outputs of machine learning algorithms and behavioral analysis. Risk score is converted into certain actions on the basis of threat level. As an example of this procedure, there are three major threat levels in the system: low risk transactions are accepted, medium risk transactions are

Threat Level	System Action
Low	Approval
Medium	OTP Verification
High	Block

Table I. Mapping of threat levels to system actions.

V. IMPLEMENTATION

A. Frontend

Frontend was built using React.js and TypeScript. Also, for the interface we used Tailwind CSS and shadcn/ui. The dashboard has a continuous feed of transactions, risks, alerts, system status, and investigation controls. The interface is aimed at analysts who should monitor the transactions and take action if needed.

B. Backend and APIs

Transaction requests, simulation, application logic, interaction with the machine learning service, and persistence are handled by Node.js and Express.js.

GET /api/transactions retrieves the transactions while POST /api/action performs actions such as approving or blocking. Another API is POST/api/control to control the system operations like pausing and resuming.

C. Machine-Learning Service

The ML module is built in Python using the Flask framework. It contains an endpoint that receives the transaction data and gives back the data extracted from the model and needed to calculate the risk score. Machine learning algorithms are written using scikit-learn, while Pandas and NumPy libraries are used for data manipulation.

D. Database and Audit Logging

Transaction, audit trail, user activities, and session data are stored in MySQL. Audit trail captures activities like approval, OTP, step-up authentication, and block. Persistent logs help in analysis of past incidents and also make the process transparent.

E. Real-Time Communication

The Socket.IO provides possibilities to make real-time communication between the backend and the frontend. Consequently, every new transaction, risk result, alert, or any manually performed activities will be updated right on the client-side.

F. Secure Access and Multi-Organization Support

The interface developed incorporates the implementation of a login module that is used to authenticate and choose an organization. As per the documentation, some of the important capabilities included in the project are multitenancy, authentication, and session connection.

VI. RESULTS AND DISCUSSION

A. Secure Login Module

The deployed login interface limits access only to authorized users and enables the choice of the organization. According to project documentation, the principal functionalities comprise multitenancy, authentication, and session management.



Fig. 2. Secure login module for analyst access.

B. Live Transaction Monitoring

The live dashboard provides an ongoing overview of transactions as well as gives risk ratings automatically. The types of data monitored include: merchant, amount of transaction, place of transaction, total number of transactions, number of high-risk transactions, number of blocked transactions and average risk score. The screenshot of the dashboard in the project documentation illustrates the interface for live transaction monitoring.



Fig. 3. Live transaction monitoring dashboard.

C. Adaptive Response

The system possesses an adaptive response that is deployed depending on the evaluated threat level. The evidence shows that the regular transactions are cleared and any suspected transactions may call for OTP verification, while transactions with high-risk features get blocked.

D. Transaction Feed and Alerts

The transaction feed is designed to keep an audit view of recent activities as well as the responses that were provided. The recorded responses include Approved, OTP Required, Step-up Authentication, and Blocked. Additionally, alerts are generated instantly and every transaction is marked with corresponding risk level.

E. Fraud Intelligence Network

The system has fraud intelligence network which makes it possible to share fraud patterns among connected organizations. The project materials state that thanks to of this functionality, the system can revolutionize the approach to fraud detection.

F. Investigation and Manual Control

The analyst is responsible for the background of the suspicious transaction and can take manual actions in case the information of the automated process is not enough or when the transaction is flagged incorrectly.

G. System Performance Evaluation

The tool is evaluated with the help of different transaction cases. The evaluation includes testing the tool through transaction flows, frequent transaction patterns, location-based patterns and machine learning anomaly detection.

Metric	Observed Evaluation
Response Time	Real-time
Detection Accuracy	High
False Alarms	Minimized
System Stability	Stable under continuous load

Table II. System performance evaluation reported in the project documentation.

H. Case-Based Analysis

Case 1 - Ordinary transaction. Such transaction is recognized as low risk, and moreover, the indicator shows that there is no suspicious behavior.

Case 2 – Suspected Transaction. The risk in this case is medium. However, the one-time password (OTP) has to be provided to authorize this transaction.

Case 3 - Fraudulent Transaction. The risk in this case is high - a fraud transaction is blocked after the system detected several indicators (i.e., location, frequency, etc.).

I. Discussion

The results of the experiments show how fraud detection and real-time application monitoring are incorporated. One can observe a hybrid design of combining automated model-based evidence along with the deterministic behavior rules and the project tool used to make the decisions visible.

VII. CONCLUSION AND FUTURE ENHANCEMENTS

Transaction Guardian shows a way of detecting digital fraud in real-time that is not exclusive to particular areas of expertise thanks to the use of machine learning, behavioral analysis, rules based detection, continuous recording, and visual representation techniques that solve the problem at hand.

Random Forest algorithm is responsible for the correct prediction of possible fraud cases, while Isolation Forest detects odd behavior.

Velocity and geolocation insights lend support to the data provided by the transaction level model.

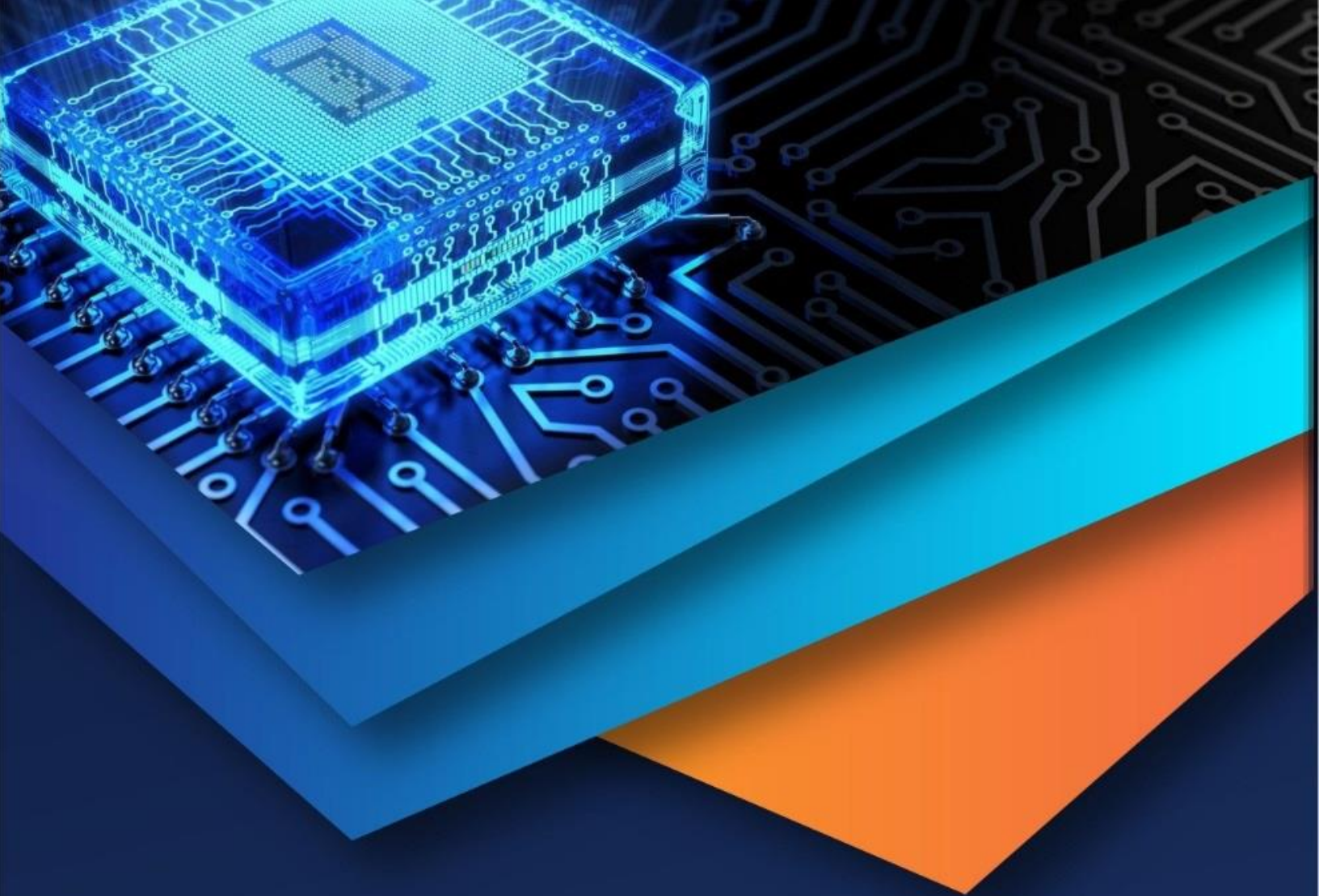
The application runs on a client-server architecture with separated frontend, backend, database, and machine learning service, which allows for independent servicing of the components.

Real-time data transfer is made possible with the help of Socket.IO and MySQL registry, improving transparency and effectiveness. The project shows the capabilities of machine learning technology in its practical use, instead of just offline examples of classification.

The future improvements include use of deep learning methods, cloud solutions, balanced datasets, targeted customers, other signals such as biometrics and IP addresses, stronger encryption, and explainable AI.

REFERENCES

- [1] A. C. Bahnsen, D. Aouada, and B. Ottersten, "Example-Dependent Cost-Sensitive Logistic Regression for Credit Card Fraud Detection," IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 263–269, 2014.
- [2] R. J. Bolton and D. J. Hand, "Statistical Fraud Detection: A Review", Statistical Science, pp. 235–255, 2002.
- [3] F. Carcillo, A. Dal Pozzolo, Y. Le Borgne, O. Caelen, and G. Bontempi, "Combining Unsupervised and Supervised Approaches to Detect Fraudulent Credit Card Usage," Information Sciences, pp. 448–466.
- [4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," ACM Computing Surveys, pp. 1–58, 2009.
- [5] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Privacy-Preserving Data Mining," SIGKDD Explorations, pp. 12–19, 2002.
- [6] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit Card Fraud Detection: Realistic Modeling and a Novel Learning Approach," IEEE Transactions on Neural Networks and Learning Systems, pp. 3784–3797, 2018.
- [7] S. Jha, M. Guillen, and J. Westland, "Using Transaction Aggregation Technique for Credit Card Fraud Detection," Expert Systems with Applications, pp. 12650–12657, 2012.
- [8] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," Proceedings of the IEEE International Conference on Data Mining (ICDM), 2008.
- [9] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and. Sun, "The Application of Data Mining Techniques in Financial Fraud Detection," Decision Support Systems, pp. 559–569, 2011.
- [10] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit Card Fraud Detection using Hidden Markov Model," IEEE Transactions on Dependable and Secure Computing, pp. 37–48, 2008.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)