iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** IV **Month of publication:** April 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.80792

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue IV Apr 2026- Available at www.ijraset.com*

# FraudShield: Fraud Detection in Financial Transactions Using Machine Learning

Dhara Abhinav[2], Chivarla Bhuvan[2], Metta Hemanth[3], V. Alekhya[4]
[1, 2, 3]*U.G. Student, Department of CSE-Data Science, Institute of Aeronautical Engineering College, Hyderabad, India*
[4]*Associate Professor, Department of CSE-Data Science, Institute of Aeronautical Engineering, Hyderabad, India*

***Abstract:*** ***Financial Fraud has become a critical problem as there is a continuous diverse increase in digital payment services, this makes the 0users as well as the financial institutions to get exposed to fraudulent attacks. Traditional approaches often fail to respond to new and evolving patterns of fraud in transactions. This leads to drop in accuracy and faster response. This study addresses that limitation, by using confidence based machine learning for fraud detection system which is designed to detect suspicious activity in real time. This model uses a combination of SMOTE which is used for data balance and Gradient Boosting a classification algorithm to improve detection performance. Using metrics like transaction behaviour and patterns this system can classify legitimate and fraudulent activities with better reliability. And the Experimental Results show the decrease in False positives and a better accuracy of 0.97, precision of 0.89 for fraudulent type transactions, recall of 0.74 and ROC score of 0.932, this analysis gave good results and better findings. And also the model has proved to perform better at detecting Fraudulent Transactions. The model was implemented on a Flask based backend which uses MongoDB Atlas as it's Data Base. This solution has a real time monitoring, providing security measures like OTP based authentication, device blocking and email alerts which prevents unnecessary classification based on confidence. All of these are integrated into a web-interface, where features like transaction history, account analytics and notifications are included. This solution provides a way to integrate software and machine learning methods for real-world systems to increase in efficiency for detection and prevention of financial frauds.***
***Keywords: Fraud Detection, Machine Learning, Flask, MongoDB, Cybersecurity, SMOTE, Gradient Boosting***

## I. INTRODUCTION

The increase in the number of financial transactions using digital methods and e-Commerce services has changed the way consumers usually perform transactions. This is happening because of new digital services like wallets, online payment and online money transfer which are convenient to the consumers. This ultimate lead to increase in electronic financial transactions. But this rapid expansion also had its

consequences, called fraudulent activities like phishing, unlawful access and synthetic identity fraud. According to International CyberSecurity Statistics(ICS), these financial frauds has increased to the point, where its now causing billion dollar loss annually, effecting both Financial Institutions and consumers. Therefore real time Fraud Detection and prevention has become a critical aspect and research challenge.

The Traditional Fraud Detection Systems rely on predetermined patterns, or thresholds. These algorithms can detect some frequent fraud patterns, they fail to learn constantly evolving strategies used by fraudsters. When faced with automated scripts, proxy networks, or other advanced methods traditional systems are less effective in detecting or preventing such fraudulent transactions. Not only that traditional systems often have high false positives as well. To compensate these problems faced by traditional systems, machine learning along with data driven decisions in real-time prove to be useful. These kind of systems where machine learning is integrated, analyse human behaviour during payments and try to find new patterns for fraudulent actions. And overtime these get better as more data is generated in real time and is available for analysis by capturing statistical data and behavioral patterns. Although, explainability, latency constraints,intrgration with secure systems,class imbalance are some challenges related to intergrating Machine Learning in such practical contexts of digital payments. Many academic studies or research focus on model accuracy.

This work introduces FraudShield, which is a full-stack fraud detection and prevention system, which has confident based machine learning techniques. This solution has a combination of fraud prediction pipeline with a confident metric and end to end protection through security protocols. This work uses feature engineer techniques which identifies the user patterns and user transactional statistics like variation in geolat location, transactional frequency, recipient diversity, transaction timing. And Uses the HISTGRADIENTBOOSING CLASSIFER to give a confidence score of classification which flags or sends the score to the decision engine for further verification, the model is trained on a synthetic dataset

which generates real transactional data with also equal number of fraudulent points for training the model to avoid underfit. The SMOTE is layered upon this to reduce the class imbalance, using this the model learns patterns associated with fraudulent markers.
Fraud Shield has a end to end software development architecture including a web based interface, a backend based on Flask, and a cloud based database. This allows real time transaction capture and can provide fraud score through API before the transaction could take place. It has security features like Security PIN, OTP based verification and automatic device blocking method. This work also implements Flask-Mail in backend to send OTP and device logging alerts in real-time to prevent risky events, and this keeps a timely response to user. The system also allows for administrative access on role base, and this allows for reviewing fraud alerts and providing feedback so the model can get better a predictions. This makes the interface both interactive and practical.
Experimental results show that FraudShield works well in classification on synthetic data, with an accuracy of 0.97, precision of 0.89, recall of 0.74, and ROC AUC score of 0.932. The system's modular design supports scalability and flexibility across banking, fintech, and e-commerce applications.
Upon experiments on the synthetic data the results show that the system works well and proves that the design supports for scalability and shows flexibility for banking and e-commerce applications. The results were with an accuracy of 0.97, a precision of 0.89, recall of 0.74, and ROC AUC score of 0.932.
The milestones of this work goes around feature engineering, model development, security protocols. This makes this research a complete end to end system development.

- The feature engineering which maps the geographical location, user behavioural data like Average transaction amount, recipient patterns, transaction frequency, device usage and Time based data like time of the day, day of the week.
- The gradient boosting and SMOTE handling imbalance of class, which enhances the models effectiveness.
- The Implementation of security measures like OTP verification, device fingerprinting, and email notification to tackle a fraud response.

## II. RELATED WORK

This work uses the Synthetic Minority Oversampling Technique (SMOTE) which handles the class balance and the Gradient Boosting for efficient learning and classification.
And implements end-to-end security layers. On the other hand the other studies related to financial fraud detection uses various other strategies like Random Forest, Neural Network and Logistic Regression. But the problem with these techniques is that, there is no class balance on training data on fraud, they have less data points to train on fraud cases compared to legitimate ones which leads to a poor minority class learning.

## III. ARCHITECTURE AND SYSTEM DESIGN

Three main parts make up FraudShield's architecture: a machine learning model pipeline, a Flask-powered backend API, and a web-based interface. The Flask backend manages all requests, processes inputs, and provides predictions in real time, acting as the main conduit between the deployed fraud detection model and the user interface.
The system effectively stores and manages user profiles, transaction records, OTP data, and fraud warnings using MongoDB Atlas, a cloud-hosted NoSQL database. This makes it possible to handle data in a secure and scalable manner, enabling rapid updates and retrieval during transactions and authentication procedures.
Both administrators and users can access interactive dashboards through the frontend interface. While administrators can upload datasets, train models, and examine transactions that have been flagged, users can conduct transactions, check their account balance, and keep an eye on recent activity. Additionally, the dashboards improve usability and decision-making by displaying visual statistics like total transactions, fraud counts, and summaries of recent activity.
In order to examine transaction parameters including transaction amount, time, device ID, IP address, and user behavior patterns, the system also incorporates a trained machine learning pipeline that is loaded via joblib. The program determines a fraud probability score based on these characteristics and instantly categorizes transactions as either fraudulent or lawful.
Lastly, a variety of contemporary technologies, including as Flask, Scikit-learn, Pandas, and MongoDB, are used in the construction of FraudShield, guaranteeing a reliable, scalable, and effective fraud detection system that can manage real-time financial transactions.

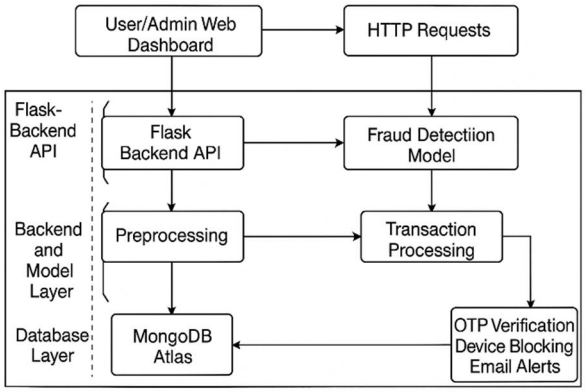



Fig. System Architecture of FraudShield

Fig1. System Architecture of FraudShield.

as the primary link between the user interface and the deployed fraud detection model. MongoDB Atlas is a cloud-hosted database used to maintain user profiles, transaction records, and alert data. The frontend interface's dashboards let both users and administrators to keep an eye on transactions, review fraud alerts, and access visual statistics.

## IV. FEATURE ENGINEERING AND DATA GENERATION

The dataset used to train the model was constructed using Python and the package Faker to replicate real user activity and transaction patterns. These records have information such as the amount, transaction type, geographic distance in kilometers, and an IP change indication. To maintain a expected precision, the script repeated transactions over multiple days and devices.

Feature engineering added new indicators such like the hour of the transaction encoded, the average amount of the preceding ten transactions, the number of unique recipients, the distance from the previous known location, and the time difference since the last transaction. This enhanced the model performance. These artificial characteristics improved the model's ability to distinguish between legitimate and fraudulent behavior, enabling it to capture complex, non-linear transaction processes.

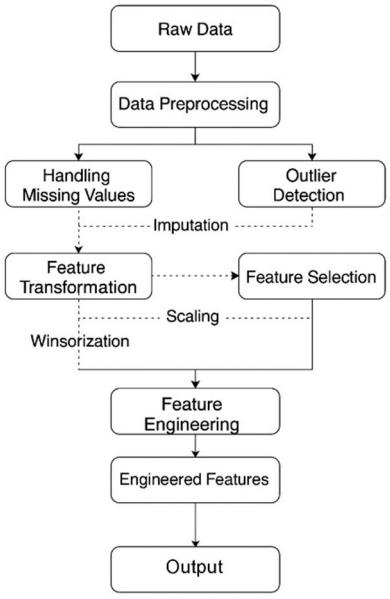



Fig 2: Feature Engineering Workflow for FraudShield.

## V. ML MODEL AND METHODOLOGY

The FraudShield Feature Engineering Workflow shows how transactional data that hasn't been handled can be transformed into valuable model inputs.

### A. Model and Methodology for Machine Learning

The foundation of FraudShield is a supervised learning pipeline that differentiates between legitimate and fraudulent transactions using a mix of statistical, behavioral, and contex-tual factors. Because raw transaction logs vary widely in type and structure, the initial step is to convert them into a uniform, useful feature set appropriate for training and inference.

### B. Overview of the Pipeline

The machine learning workflow is implemented as a single pipeline and includes preprocessing, class imbalance management, and classification. A ColumnTransformer allows for the simultaneous processing of numerical and category properties. Z-score normalization is used to standardize numerical data, including amount, time of day, rolling averages, and geo-graphic distance, in order to improve convergence and prevent features with large scales from dominating learning.

Both categorical data like transaction type and binary markers like known recipient and IP change status are encoded using one-hot encoding. The classifier can identify the influence of each category without assuming anything about ordinal relationships by transforming each category into a binary vector.

The encoded categorical attributes are combined with the processed numerical data to generate a single feature matrix that is used as the classifier's input. When every step is contained within a scikit-learn pipeline, the same transformations are carried out consistently during training and deployment.

### C. Managing Class Inequality

In actual financial databases, fraudulent transactions usually make up fewer than 1%to 5# of all transactions. If a classifier is trained on such uneven data, it can favor the majority class and ignore actual fraud instances. To address this, FraudShield uses the Synthetic Minority Over-sampling Technique (SMOTE) throughout the pipeline. SMOTE generates synthetic minority samples by interpolating between existing fraud samples and their nearest neighbors. By extending the fraud class representation without duplicating data, this allows the classifier to learn a more expansive decision boundary. SMOTE is only used on training data in order to prevent data leaks and preserve the integrity of test results.

### D. Choosing a Classifier: HistGradientBoosting

Fraud-Shield's main classification model is the HistGradientBoostingClassifier, a very effective variant of gradient-boosted decision trees. The foundation of gradient boosting is the sequential, additive creation of a group of decision trees, each of which is trained to fix the lingering defects of its predecessors. Gradient boosting is very good at gathering intricate, non-linear correlations within transactional data because of its iterative refining. Histogram-based optimization of the classifier, which initially groups features Another performance benefit is the division of data into discrete histogram bins rather than analyzing each potential split point for continuous features.

By doing this, computational overhead is greatly reduced without compromising forecast quality. Because it can analyze massive amounts of data with little memory usage and quicker training cycles, the model is particularly well-suited for use in settings requiring high throughput or near-real-time fraud detection. A reliable and scalable method for recognizing complex fraud patterns across several transaction streams is provided by combining the excellent learning capabilities of gradient boosting with the histogram-based efficiency benefits.

HistGradientBoosting can capture complex non-linear correlations and feature interactions, it was chosen.

- It can handle both numerical and category (en-coded) data.
- It connects to scikit-learn pipelines with ease for deployment.

The model is trained using a stratified train-test split in order to maintain the original fraud-to-legitimate ratio. The classifier produces a probability score that indicates the possibility of fraud during prediction. To better balance fraud recall and false alarm rates, FraudShield employs an enhanced threshold of 0.3 rather than the conventional 0.5 level.

### E. Methods of Evaluation

A number of indicators, including accuracy, precision, recall, F1-score, and ROC AUC, are used to assess the FraudShield model's performance. These steps are crucial in financial fraud detection scenarios since unnoticed fraudulent transactions can result in large financial and operational losses. While precision concentrates on the percentage of anticipated fraud episodes that are actually fraudulent, helping to minimize needless system warnings, accuracy offers a high-level measure of total correctness.

Recall measures the model's ability to identify real fraud, which is crucial for reducing fraudulent activity that remains unreported. The F1-score is especially helpful in highly unbalanced datasets since it provides a fair representation of precision and recall. Additionally, the ROC curve assesses model behavior at various threshold levels, demonstrating its ability to distinguish between legitimate and fraudulent behavior.

By clearly separating true positives, false positives, true negatives, and false negatives, the confusion matrix enhances these metrics and enables more thorough diagnostic investigation of model outcomes.

With a ROC AUC score of 0.932, FraudShield showed excellent discrimination between fraudulent and legitimate transactions. Overall findings verify that a dependable system for real-time fraud detection is produced by combining specific characteristics, SMOTE balance, and histogram-based gradient boosting.

Table -1
Performance METRICS ON SYNTHETIC DATASET

| Metric | Value |
|---|---|
| Accuracy | 0.97 |
| Precision (Fraud) | 0.89 |
| Recall (Fraud) | 0.74 |
| ROC AUC | 0.932 |

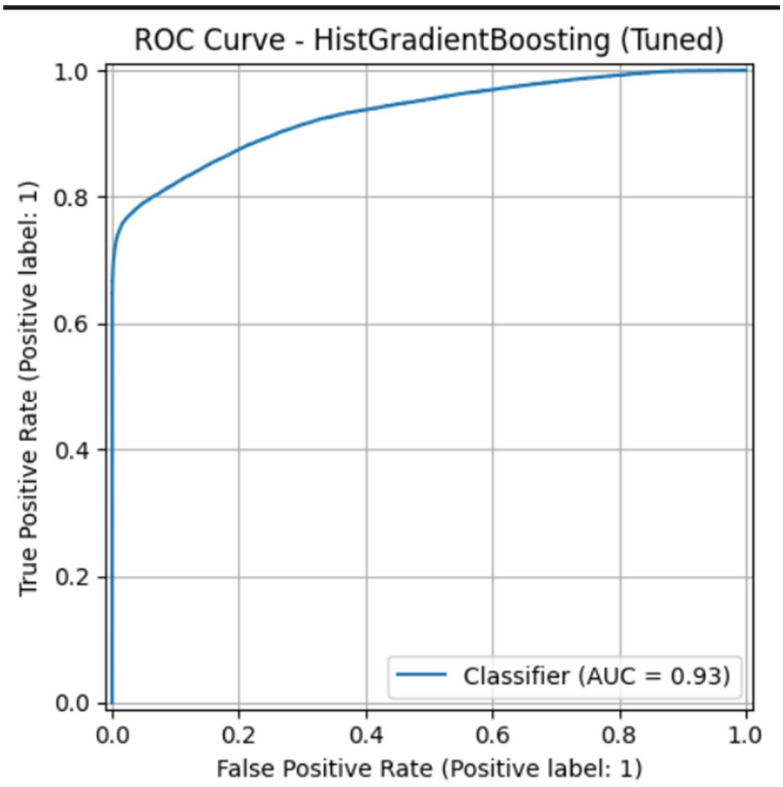



Fig. 3. ROC Curve showing model discrimination performance.

## VI. SECURITY AND INTEGRATION LAYER

To guarantee secure financial transactions, FraudShield integrates a number of helpful security mechanisms. User registration, authentication, and transaction processing are handled by the Flask backend. Before any changes are made to the MongoDB balance data, the machine learning model assesses every transaction.

Security Features:

- OTP Verification: implemented with Flask-Mail to confirm PIN updates and sensitive user actions.
- Device Fingerprinting: This reduces the likelihood of unwanted access by momentarily preventing.
- Automated Email Delivery: Instant notifications are sent for high-risk transactions, new device logins, or unexpected user activity to guarantee prompt user awareness.

To lower risk and guarantee system integrity, a specific security workflow synchronizes the detection, user validation, and remediation operations.

## VII. SECURITY AND INTEGRATION LAYER

The User interface has several HTML pages with modules for administrative supervision, transaction processing, alarm monitoring, login, and registration.

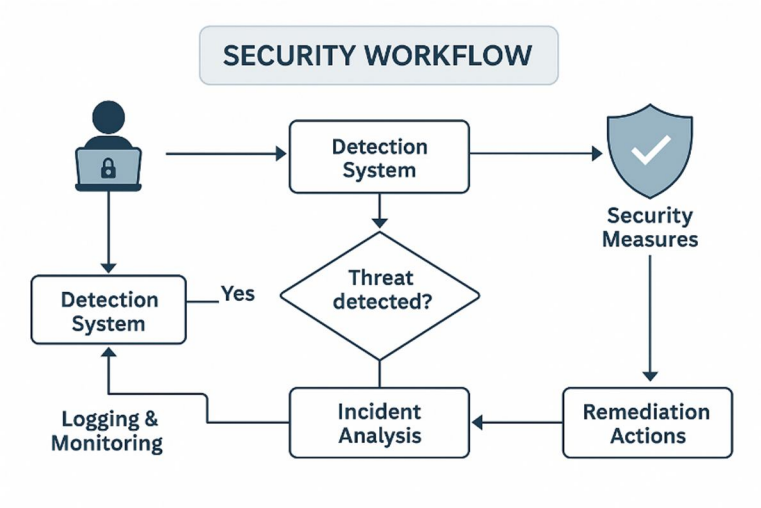



Fig. 4. Security and Integration Workflow in FraudShield.

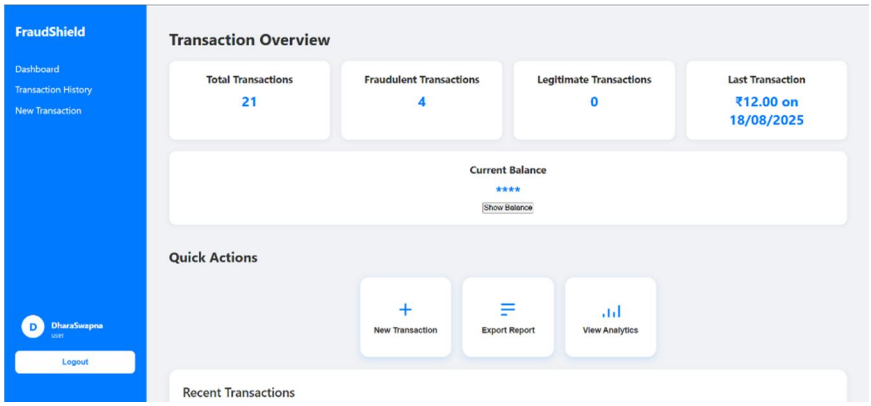



Fig. 5. FraudShield Web Dashboard for Users and Admins.

The dashboard makes it easier for clients to analyze their financial actions and comprehend model est mates by showing transaction history, fraud alerts, and system activity.

## VIII. RESULTS AND DISCUSSION

The model showed excellent predictive ability and high precision on the minority (fraudulent) class. The ROC AUC of 0.932 indicates robust discrimination abilities. The precision recall trade-off shows balanced sensitivity for real-time fraud detection, where false positives must be minimized.

A comparative analysis revealed that the HistGradientBoostingClassifier outperformed Random Forest and Logistic Regression baselines in recall and ROC AUC while maintaining computational efficiency suitable for Flask deployment.

## IX. CONCLUSIONS AND FUTURE SCOPE

This work demonstrates that it is possible to combine machine learning with modern web technology for secure, real-time fraud detection. FraudShield showed excellent accuracy and scalability on synthetic datasets, and its architecture allows for future application to real-world financial APIs.

Future enhancements consist of:

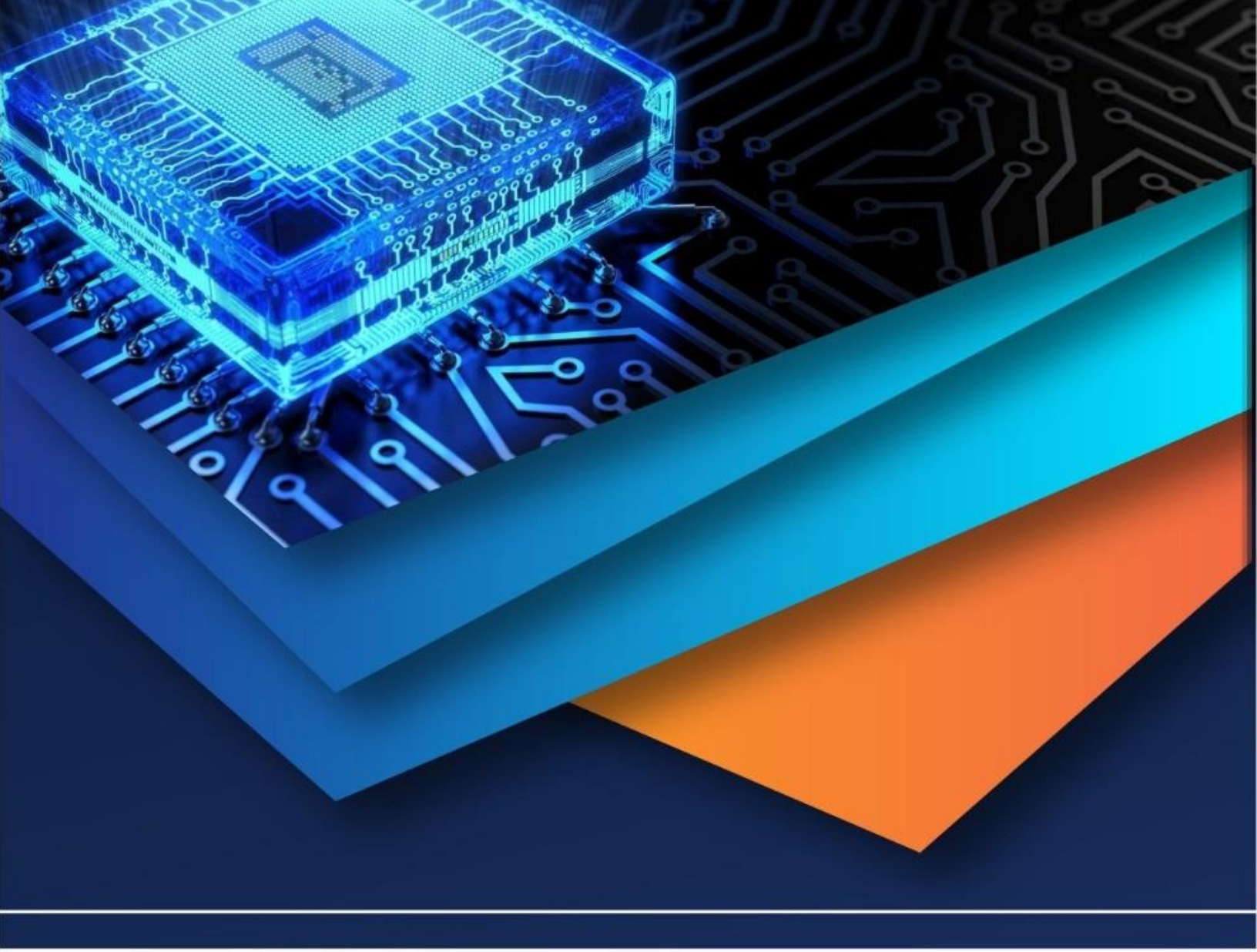
- Blockchain technology for transaction records that cannot be changed,
- Sequential fraud analysis using deep learning modelse such as LSTM,
- The development of a federated learning version for training models that protect privacy across banks.

## X. ACKNOWLEDGEMENT

The authors would like to sincerely thank Assistant Professor Ms. V. Alekhya of the Department of CSE (Data Science), Institute of Aeronautical Engineering, for her valuable advice, mentorship, and support during this study.

## REFERENCES

[1] J. West and M. Bhattacharya, “Intelligent financial fraud detection: a comprehensive review,” Computers & Security, vol. 57, pp. 47–66, 2016.
[2] P. Dal Pozzolo et al., “Calibrating Probability with Undersampling for Unbalanced Classification,” in IEEE Symposium Series on Computational Intelligence, 2015.
[3] R. Chalapathy and S. Chawla, “Deep Learning for Anomaly Detection: A Survey,” arXiv preprint arXiv:1901.03407, 2019.
[4] Scikit-learn Developers, “HistGradientBoostingClassifier Documentation,” https://scikit-learn.org/.
[5] M. Lemaˆıtre, F. Nogueira, and C. Aridas, “Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning,” Journal of ML, 2017
[6] S. J. Russell and P. Norvig, “Artificial Intelligence: A Modern Ap- proach,” 3rd ed., Pearson, 2010.
[7] MongoDB Inc., “MongoDB Atlas Cloud Database Service,”https://www.mongodb.com/cloud/atlas.
[8] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, “The Great Time Series Classification Bake Off: A Review and Experimental Evaluation of Recent Algorithmic Advances,” Data Mining and Knowledge Discovery, vol. 31, no. 3, pp. 606–660, 2017.
[9] S. J. Russell and P. Norvig, “Artificial Intelligence: A Modern Ap-proach,” 3rd ed., Pearson, 2010.
[10] Y. Zhou, G. A. Kumar, and V. R. Vemuri, “Detecting DDoS Attacks Using Machine Learning Techniques,” in IEEE International Conference on Machine Learning and Applications, 2014, pp. 389–394



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY