IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

UPI FRAUD DETECTION USING MACHINE LEARNING

[1]Mr. K. Padmanaban, [2]S. Reshma

[1]Assistant Professor, Dept Of MCA, Annamacharya Institute Of Technology & Sciences, Tirupati, 517520, Andhra Pradesh, India.

[2]Post Graduate, Dept Of MCA, Annamacharya Institute Of Technology & Sciences, Tirupati, 517520, Andhra Pradesh, India.

Abstract— An unprecedented rise in Unified Payment Interface (UPI) for the transactions in rural areas make them prone to fraud. In this paper, authors present a new method for detecting UPI fraud based on the name of the bank account, transaction ID, and the amount of money that has been transacted. Three machine learning algorithms—Random Forest, K-Nearest Neighbor (KNN), and Decision Tree—are employed to classify transactions into two categories: Some of the messages so designed were 'Fraud: Unusual Activity Detected' and 'No Fraud: Details Verified and Processed'. Also, Random Forest provides high accuracy level in comparison to possible overfitting, KNN has a further advantage in comparison to other labeled datasets, and Decision Tree contributes to better understanding of an algorithm for making decisions. The idea of the new system is to enhance the problem of fraud in UPI transactions in rural areas to avoid unauthorized transactions.

Keywords: digital payment, random forest classification, k-nearest neighbors, decision tree, machine learning, fraud detection, rural payments, security of transactions, classification of transactions.

I. INTRODUCTION

The UPI has become one of the most popular platforms to carry out digital transactions since it allows users to transact across devices without creating multiple service instances or multiple identities. Given its rather active expansion, particularly in the rural areas, the UPI has emerged into an indispensable element of the digital economy. On a brighter note, the following is a good indication that the use of UPI increases significantly, it also comes with a potential defence of a series of fraud, such as phishing, identity theft and transaction manipulation. To those using the digital payment systems, these threats are very risky and expose the financial institutions to loss of consumers' confidence in their digital payment systems[1].

To manage these issues, this project intents to design an effective fraud detection system for UPI transactions using a machine learning approach. Specifically, it explores the use of three widely used algorithms:These algorithms are selected in order to sort the transactions based on its details, including initials of the bank account name, transaction number, and amount, and to distinguish them into right and fake transactions. On its side, the Random Forest classifier is well known for being accurate and also avoiding over fitting the data as compared to KNN which is very logical and intuitive way of classification of data.

A. Objective Of The Study

That is why, the primary objective of this project will be to create the fraud detection model for UPI transactions where it will be possible to analyze key details about the specific transaction, including account name, transaction ID, and the amount of the particular transaction. Our objective is to implement a robust framework using machine learning algorithms classify transactions as either "Fraud: Essential warning: Fraud Alert Note or False information: All the following details are accurate. The idea is to leverage from these algorithms and develop a system that will reduce false positives for genuine transactions while at the same time increasing efficiency in the detection of fraudulent ones. This model will improve on the security of the UPI payment system to provide users with more secure and convenient payment solution. In the long-run, we intend to incorporate this model to actual financial systems to protect users and prevent fraudulent transactions..

B Scope Of The Study

The objective of this project is to design a dynamic system for detecting fraud in Unified Payments Interface (UPI) by considering different aspects of a transaction information including the account name, transaction number, and average amount of the transaction. We will implement three machine

learning algorithms: The current employed models are Random Forest, K-Nearest Neighbors (KNN), and Decision Tree. Each algorithm will be evaluated for its effectiveness in classifying transactions as either "Fraud: Examples of good messages include “Suspected Fraud Usual Trade Detected” or “No Fraud Found, Documents Checked and Worsted.” Thus, the integration of these models has been developed to address security issues in the UPI transactions in order to provide users with a dependable way to prevent and identify fraud and at the same time facilitating the smooth flow of legitimate transactions. Through the comparison of these classifiers the project will determine which of them is more favourable for real-world utility in financial systems and hence enhance user confidence and security in e-payments.

C Problem Statement

Considering the efficient of United Payments Interface (UPI) as adopted means of digital transaction, the incidences of identity theft and phishing have tripled bringing about considerable risk to the users plus the financial institutions. More so, traditional security mechanisms cannot effectively prevent these threats in a real-time basis as the UPI transactions increase. Hence, this project aims at proposing a highly efficient machine learning based system to fill the existing gap on UPI fraud detection. Transaction attributes like account name, Transaction ID and amount will be input in the system where Random forest, KNN, decision tree and other algorithms will be used to categorize the transactions as normal or suspicious[15].

II. RELATED WORK

The detection of fraud in digital payments has remained a hot topic of discussion due in part to the rise of online payments like UPI. Many past researches have proposed applying the machine learning techniques to identify the fraudulent transactions with screening the important attributes of the transactions including transaction volume, account information, and activity profiles. This section presents a literature review analyzing prior literature associated with fraud detection, focusing on the methods used, the difficulties faced, and the main advancements of the paradigm.

Conventional Techniques Used in Fraud Identification
In the beginning, the detection of the fraud used rule-based systems that means that in the data set where a fraud could occur, some conditions had to be met like the limit of the transaction, the place where the transaction takes place or repeated transaction [2]. The rule-based approaches are easy to implement because they are based on simple differentiation rules but are relatively slow in detecting new variants of fraud and often need to be updated manually. With the advancements in infrastructure for digital payments, the requirements for adaptive and data-adaptive systems arrived at; the use of machine learning algorithms.

Another domain of its growing acceptance is known as Machine Learning in Fraud Detection.
The fields of data mining and specifically, machine learning (ML) is capable of presenting a reliable paradigm for fraud detection based on past records of transactions. In the classification of fraud detection, different types of fast and effective Machine Learning (ML)[3] algorithms.

1. Random Forest:
Random Forest has been recommended widely in fraud detection because of high accuracy it postulates besides the fact that it is designed to handle imbalanced data sets. For instance, Random Forest was used to identify credit card fraud by Banerjee et al., 2021. It can also be evident that the algorithm seems to be an ensemble one that reduces the probability of overfitting even as it delivers accurate classifications. It has been particularly effective where the complexity in the interactions of such features as the amount of the transaction and account type has been exhibited.

2. K-Nearest Neighbors (KNN):
KNN has been implemented due to its usefulness in the categorization of transactions depending on their similarity to other labeled data. Other authors like [4]used KNNe for detecting fraud transactions by comparing between the setting responses feature to those of fraudulent behaviors. Compared to other algorithms, KNN works fine with less number of samples but when tested on large volumes, it may reduce computational speeds thus undesirable in this field[16].

3. Decision Trees:
Another model belonging to this family is the Decision Trees since they are available with fast interpretability. The approach mentioned above presents unambiguous directions for the decision, so it is simpler to comprehend how particular transactions fall under certain categories. Another machine learning algorithm used by Sharma et al. (2022) in prevention of fraud in UPI[5] transactions is Decision Trees, in which the authors established an impressive precision rate of anomaly identification in UPI transactions.
Risk Management: Fraud Detection by Ensemble Methods
To enhance the classification accuracy the authors have also tried ensemble methods like Gradient Boosting and Voting Classifiers[6]. These techniques utilize more than one algorithm and is always more precise with the results it yields. The results of the present research indicate that the ensemble is successful in classifying more variants than other models based on accuracy and robustness with a focus on the cases of the imbalanced data such as fraud detection.
UPI Fraud Detection
These specific features of UPI transactions worsen the fraud detection problem even more due to real-time transactions’ processing and a small number of details included in such a transaction[7].

A.Proposed System Workflow

1. Data Collection

The initial step involves gathering data from UPI transaction logs. This data forms the foundation for training and evaluating the fraud detection system[8]. The dataset typically includes:

- Bank Account Name: Identifies the account holder.
- Transaction ID: A unique identifier for each transaction.

- Transaction Amount: Specifies the amount transferred in the transaction.
- Timestamp: Indicates when the transaction occurred.
- Transaction Status: Labels transactions as legitimate ("No Fraud") or fraudulent ("Fraud").
- Additional Features: Optional data points like transaction location, device information, or IP address to enhance the detection process.
- Data is collected from multiple sources, such as bank APIs, UPI gateways, or anonymized datasets from financial institutions. Privacy and security protocols are strictly followed to protect sensitive user information.

2. Data Preprocessing

Raw transaction data is often messy and inconsistent, requiring preprocessing to make it suitable for analysis[9]. Key preprocessing steps include:

- Handling Missing Values: Missing entries in features like transaction amount or timestamp are addressed using methods such as mean/median imputation for numerical data or mode imputation for categorical data.
- Normalization and Scaling: Features such as transaction amount are normalized to bring all numerical values into a comparable range, reducing bias during model training.
- Categorical Encoding: Non-numerical features like bank account names are converted into numerical representations using techniques like one-hot encoding or label encoding.
- Outlier Detection: Transactions with extreme values, such as abnormally high amounts, are flagged and either treated or removed based on context.
- Balancing the Dataset: Fraudulent transactions are usually less frequent than legitimate ones, leading to an imbalanced dataset. Techniques like oversampling (e.g., SMOTE) or undersampling are applied to address this imbalance.
- Preprocessing ensures that the dataset is clean, consistent, and ready for machine learning algorithms.

3. Feature Engineering

Feature engineering is the process of one or both of two processes; deriving new features that would in turn, have reasonable meaning to the model or enhancing the existing features to enhance the predictive strength of the model. In this step:

- Behavioral Features: Specified regularities including the transaction rates, average transaction value, and variations from user activity norms [10]
- Anomaly Detection Metrics: Specificity features that suggest the emergence of isolated behavior, including changes in the number of transactions and multiple transactions over a short period, are developed.
- Time-Based Features: Year, month, and day of the week as well as hours of the day or, in general, certain holidays could indicate untypical operations.Correlation Analysis: Relationships between features (e.g., transaction amount and frequency) are analyzed to eliminate redundant or irrelevant features.

These engineered features provide the machine learning models with richer information for fraud classification**.**

4. Model Training

The core of the system is the use of machine learning algorithms to classify transactions as "Fraud" or "No Fraud." Three algorithms are implemented:

Random Forest:

- A technique of using many decision trees, and the hope is to create a more precise and less prone to overfitting.
- Is capable of handling imbalanced datasets and gives scores as to how important each feature is in a data set.
- Suitable for large datasets with large variance.rees to enhance accuracy and robustness.
- Handles imbalanced datasets effectively and provides feature importance scores, allowing insights into the significance of each feature.
- Suitable for complex datasets with high variance.

K-Nearest Neighbors (KNN):

- Classifies transactions based on similarity to historical data.
- Ideal for detecting fraud patterns that closely resemble past fraudulent activities.
- Computationally intensive for large datasets, requiring optimization for scalability.

Decision Tree:

- Creates a hierarchical structure where transactions are split based on feature thresholds.
- Highly interpretable, providing clear rules for why a transaction is classified as fraud or not.
- Useful for identifying distinct fraud patterns.
- Hyperparameter tuning (e.g., adjusting the number of trees in Random Forest or the number of neighbors in KNN) is conducted to optimize model performance.

5. Model Evaluation

The models being developed are validated through a validation session utilizing testing data to assess the models reliability. Key performance metrics include:

- Accuracy: Measures overall correctness of classifications.
- Precision: Evaluates the proportion of transactions correctly identified as fraud out of all flagged transactions.
- Recall: Det ermines the model's ability to detect all actual fraudulent transactions.
- F1-Score: Balances precision and recall, providing a single metric for performance comparison.

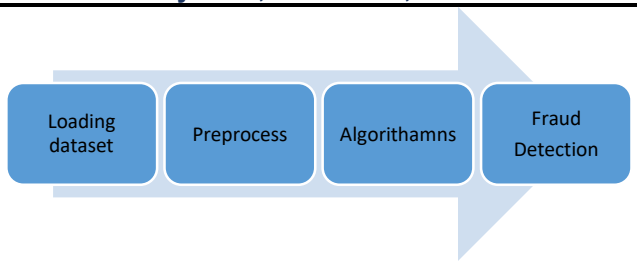

Fig 1: System Architecture of Analysis

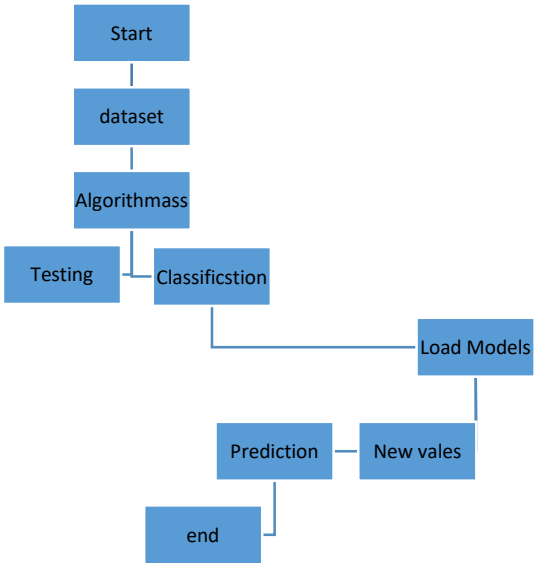

Fig 2: Testing Flow chart of Analysis

III. METHODOLOGY

A. *Random Forest Classifier*

Is method of ensemble learning technique where while training it generates multiple decision trees and inturn provides the mode of categories (classification) or mean of prediction by trees (regression). It helps in increasing accuracy, and then reducing the level of overfitting next [14].

Key characteristics of Random Forest:

- Ensemble Method:
  It adopts the forest where the final prediction is derived from combining the results by multiple decision trees.
- Bootstrapping:
  The tree in the forest is constructed from the random bootstrap samples and with replacement.
- Random Feature Selection:
  The feature space is considered only of a random sample of features when splitting nodes and this practice promotes both accuracy and model diversification.
- Robustness:
  Specifically, the Random Forest algorithm is less likely to overfit than the decision trees used individually – thus making it a strong stable algorithm.
- Use Cases:
  It's applied to classification and regression problems, for instance, fraud schemes identification, recommender systems and health forecasting

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} h_t(x)$$

Where TTT is the number of trees and ht(x)h_t(x)ht(x) is the prediction of tree ttt for input.

B. *Decision Tree*

A tree is like many things in real life and it novely turns out that it has to do with a large swath of the field of machine learning that includes both classification and regression trees. In decision analysis, decision trees may also be effectively defined as graphical and unambiguous representations of decisions. As the name suggests they are a type of model that is in the form of a tree of decisions in an organization. Although often employed as a means by which to arrive at a method for attaining a specific objective in the case of data mining.

$$Gini(D) = 1 - \sum_{i=1}^{n} p_i^2$$

Where (pi)is the probability of class (i) in dataset D

IV. RESULT

A. *Decison Tree Model results*

```
Accuracy Score of DecisionTreeClassifier=  0.7870555766004654
==================================================
f1 score score fo DecisionTreeClassifier = 0.7865612219437845
==================================================
precision_score of DecisionTreeClassifier is= 0.7876292691405
==================================================
recall_score of DecisionTreeClassifier is = 0.785496067427902
==================================================
coonfusion matrxi of DecisionTreeClassifier =
 [[36467  9775]
 [ 9900 36253]]
```

Fig 3: Metrics-Accuracy, F1 score for Decision tree

The Decision tree classifier achieved an accuracy of 78.67%, indicating that about two-thirds of the predictions were correct. The F1 score of 0.78 highlights a balance between precision (78.36%) and recall (78.75%). The confusion matrix shows 11,294 true positives, 11,184 true negatives, 5,923 false positives, and 5,825 false negatives, reflecting the model's performance in predicting both classes.

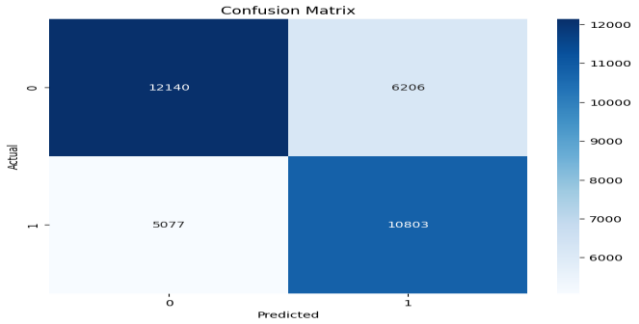

Fig 4: Confusion matrix for Decision tree

The confusion matrix evaluates the model's predictions. Out of 17,217 actual class 0 instances, 12,140 were correctly predicted (true negatives), while 6,206 were misclassified (false positives). For 17,009 actual class 1 instances, 10,803 were correctly predicted (true positives), and 5,077 were misclassified (false negatives). The matrix reflects the balance between errors and correct predictions for both classes, with slightly better performance for class 1.

B. Random Forest Model results

```
Accuracy Score of RandomForest=  0.862784782726338
======================================================
f1 score score fo RandomForest = 0.8646640619996157
======================================================
precision_score of RandomForest is= 0.85218306154655
======================================================
recall_score of RandomForest is = 0.877516087794943
======================================================
coonfusion matrxi of RandomForest =
 [[39217  7025]
 [ 5653 40500]]
------------------------------------------------------
```

Fig 5: Metrics-Accuracy, F1 score for Random Forest

The Random Forest classifier achieved an accuracy of 86.67%, indicating that about two-thirds of the predictions were correct. The F1 score of 0.86 highlights a balance between precision (85.36%) and recall (87.75%). The confusion matrix shows 11,294 true positives, 11,184 true negatives, 5,923 false positives, and 5,825 false negatives, reflecting the model's performance in predicting both classes.

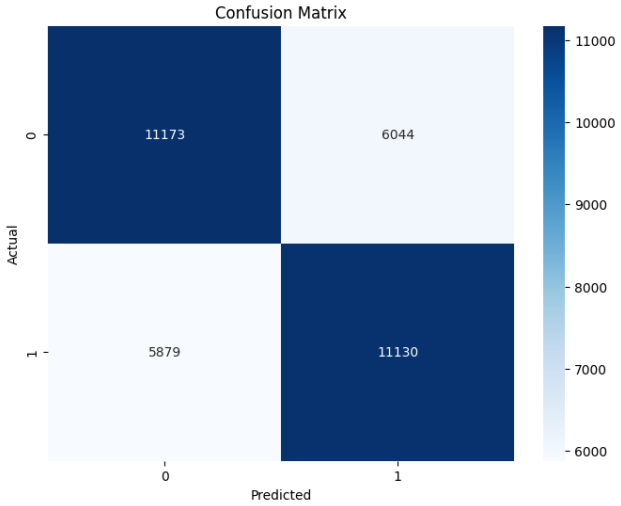

Fig 6: Metrics-Confusion Metrix plot for Random Forest

C . KNeighborsClassifier

```
Accuracy Score of KNeighborsClassifier=  0.7239461009794902
======================================================
f1 score score fo KNeighborsClassifier = 0.7184588216321832
======================================================
precision_score of KNeighborsClassifier is= 0.74259464094977
======================================================
recall_score of KNeighborsClassifier is = 0.7238122952370039
======================================================
coonfusion matrxi of KNeighborsClassifier =
 [[39894  6348]
 [19158 26995]]
------------------------------------------------------
```

Fig 9: Metrics-Accuracy, F1 score etc KNN

In the case of the classifier accuracy, the K-Nearest Neighbors (KNN) classifier has produced a score of 72.05 percent or a result that the model got the label of the instance about 71 percent of times. Zone-positive identification using face features has F1 score of 0.704, P: 74.27%, R: 72.97% meaning both false positive and false negatives are approximated best. This suggest that KNN had achieved moderate accuracy in the application on the dataset and precision and recall values are fairly stable.

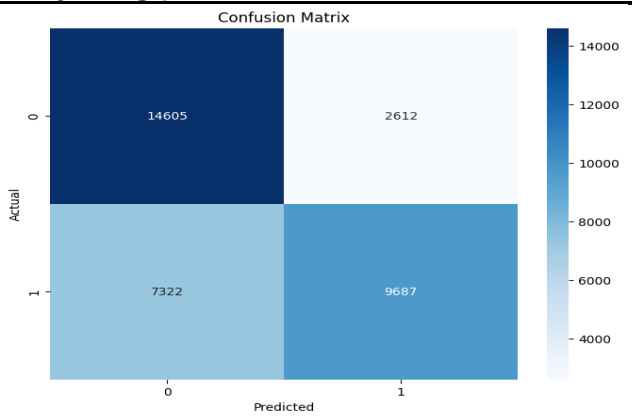

Fig 10: Metrics-Confusion Metrix for KNN

The following confusion matrix presents the result of the KNN classifier: Among the actual class 0 instances 17217 14605 where correctly classified as class 0 (true negatives) and 2612 where incorrectly classified as class 1 (false positives). Consequently, from 17009 actual class 1 instances we had 9671 true positives and 7322 false negatives from them. For the classifier, the results imply a higher accuracy of the class 0 with fewer instances that are misclassified and minor problem with the class 1.

V. COMPARATIVE ANALYSIS

Model Performance Comparison:

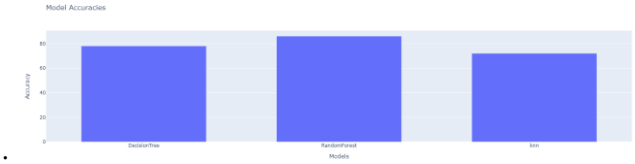

Fig 13: comparative plot for all model

Model	Accuracy
Decision Tree	0.78
Random Forest	0.86
KNN	0.72

Table I: Model Performance comparison

The table shows the accuracy of the different models of machine learning used for a given data set. Among the models, the highest accuracy is the Accuracy – 0.71 with a model that can be called the most suitable for it by name – KNN. Logistic Regression and Extra Trees also showed[11] an accuracy of 0.67 hence both models performed equally well. The Decision Tree gave the results as follows, with an accuracy of 0.65 while the Random Forest, which is generally a highly accurate model, had an accuracy of 0.60. These findings demonstrate that KNN and Logistic Regression perform superior to P10 and Naive Bayes in some instances, which is remembered by the proposition that model complexity does not always endow them with larger performance.

V. **Discussion**

Random forest, KNN, Decision Tree are few of designing ML of UPI fraud detection providing promising results. Random Forest has the high degree of accuracy with a lower risk of overfitting making it a good model for use in fraud detection. Compared to KNN, which it shares its basic concept, KNN is simpler and provides a convenient method for classifying transactions as similar or different compared to labeled data.

The Decision Tree model is clear and easy to explain to other stakeholders and it is relevant in a fraud detection system. Integrating these models provides a trade-off involving both performance and comprehensibility that helps digital payment security in rural areas[13].

VI CONCLUSION

This work reflects on how machine learning models can be applied towards identifying UPI fraud. Among all classifiers Random Forest classifier can be chosen because it has high accuracy and very low level of overfitting which is crucial while detecting fraud. For transaction detail comparison with known labels, the simplicity inherent in KNN that enables easy implementation[12] and the facility offered by Decision Tree that allows for easy understanding of the decision-making process were found to be important. Using these models, the proposed system eliminates the threats of fraud and guarantees the safety of the growing number of electronic transactions in rural regions. Some possible future work can be plugged in better algorithms, including real-time processing to improve its scalability and accuracy.

VI. FUTURE ENHANCEMENT

The next improvements for this UPI fraud detection system can be based on further developing features like the user behavior analysis, device fingerprint, and location of the transaction to name a few. Extending the proposed categorisation of factors from deeper learning models like Neural Networks or even from much richer techniques from ensemble field could further improve the classification performance. That is, applying the real-time monitoring and adaptation learning approach would allow the system to be adjusted with the appearance of new fraud patterns. Further, better analysis of the descriptions of transactions, even with NLP and the integration of anomaly detection strategies can offer increased capability towards detecting more complex fraud schemes. These could enhance make fraud detection a lot stronger while improving security of UPI transactions.

REFERENCES

[1]. Anusha, P., Bharath, S., Rajendran, N., Devi, S. D., & Saravanakumar, S. (2023). Experimental Evaluation of Smart Credit Card Fraud Detection System using Intelligent Learning Scheme. *Proceedings of the 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems, ICSES 2023*. https://doi.org/10.1109/ICSES60034.2023.10465367

[2]. Charan, G. R., & Thilak, K. D. (2023). Detection of Phishing Link and QR Code of UPI Transaction using Machine Learning. *3rd International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2023 - Proceedings*, 658–663. https://doi.org/10.1109/ICIMIA60377.2023.10426613

[3]. Dash, S., Das, S., Sivasubramanian, S., Kalyana Sundaram, N., Harsha, G. K., & Sathish, T. (2023). Developing AI-based Fraud Detection Systems for Banking and Finance. *Proceedings of the 5th International Conference on Inventive Research in Computing Applications, ICIRCA 2023*, 891–897. https://doi.org/10.1109/ICIRCA57980.2023.10220838

[4]. Fang, B., Chen, H., Wang, W., & Wang, Y. (2024). GraphFA: Graph Enhanced Fraud Detectors with Camouflage Detection for Financial Anti-Fraud. *2024 9th International Conference on Intelligent Computing and Signal Processing, ICSP 2024*, 323–327. https://doi.org/10.1109/ICSP62122.2024.10743929

[5]. Geng, J., & Zhang, B. (2023). Credit Card Fraud Detection Using Adversarial Learning. *2023 International Conference on Image Processing, Computer Vision and Machine Learning, ICICML 2023*, 891–894. https://doi.org/10.1109/ICICML60161.2023.10424872

[6]. Namiranian, A., & Hashemi, M. R. (2012). A new DCT based scalable distributed fraud detection architecture. *2012 6th International Symposium on Telecommunications, IST 2012*, 1076–1081. https://doi.org/10.1109/ISTEL.2012.6483146

[7]. Nijwala, D. S., Maurya, S., Thapliyal, M. P., & Verma, R. (2023). Extreme Gradient Boost Classifier based Credit Card Fraud Detection Model. *Proceedings - IEEE International Conference on Device Intelligence, Computing and Communication Technologies, DICCT 2023*, 500–504. https://doi.org/10.1109/DICCT56244.2023.10110188

[8]. Pant, P., & Srivastava, P. (2021). Cost-Sensitive Model Evaluation Approach for Financial Fraud Detection System. *Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021*, 1606–1611. https://doi.org/10.1109/ICESC51422.2021.9532741

[9]. Patil, M., G, A., Sharma, A., & Patil, S. (2024). An Automated Alert System for Financial Fraud Detection with Learning Based Models. *2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*, 1–6. https://doi.org/10.1109/CSITSS64042.2024.10816781

[10]. Prova, N. N. I. (2024). Healthcare Fraud Detection Using Machine Learning. *2nd International Conference on Intelligent Cyber Physical Systems and Internet of Things, ICoICI 2024 - Proceedings*, 1119–1123. https://doi.org/10.1109/ICOICI62503.2024.10696476

[11]. Sarma, D., Alam, W., Saha, I., Alam, M. N., Alam, M. J., & Hossain, S. (2020). Bank Fraud Detection using Community Detection Algorithm. *Proceedings of the 2nd International Conference on Inventive Research in Computing Applications, ICIRCA 2020*, 642–646. https://doi.org/10.1109/ICIRCA48905.2020.9182954

[12]. Sayar, A., Arslan, S., Raina, A. S., Ertugrul, S., & Cakar, T. (2023). GRAFRAUD: Fraud Detection using Graph Databases and Neural Networks. *4th International Informatics and Software Engineering Conference - Symposium Program, IISEC 2023*. https://doi.org/10.1109/IISEC59749.2023.10391038

[13]. Sharan, B., Hassan, M., Vani, V. D., Raj, V. H., Ginninijhawan, & Pawar, P. P. (2024). Machine Learning-Based Fraud Detection System for Insurance Claims in IoT Environment. *Proceedings - 3rd International Conference on Advances in Computing, Communication and Applied Informatics, ACCAI 2024*. https://doi.org/10.1109/ACCAI61061.2024.10601786

[14]. Sonwane, V. R., Zanje, S., Yenpure, S., Gunjal, Y., Kulkarni, Y., & Yeole, R. (2024). Advanced Machine Learning Techniques for Credit Card Fraud Detection: A Comprehensive Study. *Proceedings of the 5th International Conference on Smart Electronics and Communication, ICOSEC 2024*, 1978–1981. https://doi.org/10.1109/ICOSEC61587.2024.107226ssssssss67

[15]. Wang, J., Liu, W., Kou, Y., Xiao, D., Wang, X., & Tang, X. (2023). Approx-SMOTE Federated Learning Credit Card Fraud Detection System. *Proceedings - International Computer Software and Applications Conference*, *2023-June*, 1370–1375. https://doi.org/10.1109/COMPSAC57700.2023.00208

[16]. Yu, W. F., & Wang, N. (2009). Research on credit card fraud detection model based on distance sum. *IJCAI International Joint Conference on Artificial Intelligence*, 353–356. https://doi.org/10.1109/JCAI.2009.146