

# UPI Fraud Detection Using Machine Learning and Ensemble Learning Techniques

Syed Raya<sup>1</sup>, Chaitra A L<sup>2</sup>

<sup>1</sup>Student, Artificial Intelligence and Machine Learning, Sri Siddhartha Institute of Technology, Tumkur

<sup>2</sup>Assistant Professor, Electronics and Communication Engineering, Akshaya Institute of Technology, Tumkur

## Abstract

The application of Unified Payments Interface (UPI) has grown fast mostly because it gives a fast and easy way to move money as needed. UPI transactions include things like: (PhonePe, Google Pay, Paytm, Cred and other similar transactions). With much use of these online transactions there has been a rise in bad activities. Before these online transactions were popular there was a digital payments system that was used and there were no activities. Now financial losses are caused by these transactions.

This creates a need for a fraud detection system that uses Machine Learning (ML) methods, which is meant to find UPI transactions accurately. Before training the original data had steps like filling in missing values turning data into numbers and organizing the data for the models, for this project many of the algorithms were Tested like: Logistic Regression, Decision Tree, K-Nearest Neighbors, Naïve Bayes, XG Boost and Cat Boost. And their results were compared. To make things better two ways of combining models. Voting and Stacking Classifiers. Were used. Evaluation was conducted using Accuracy, Precision, F1-Score, the Confusion Matrix and the Classification Report. The results Demonstrated that integrating models with these techniques gave better fraud detection which is a reliable way to protect digital payment systems.

**Index Terms:** UPI transaction fraud, supervised AI models, Logistic Regression, Decision Tree, K-Nearest Neighbors, Naïve Bayes, gradient boosting (XG Boost, Cat Boost), ensemble voting, stacked generalization.

## 1. Introduction

Digital payments in India have changed a lot because of the Unified Payments Interface or UPI. UPI provides a safe and easy way to move money based on what people need. So more and more people choose transactions are used more than before. On the otherhand fake activities are also increasing. Individuals are experiencing financial losses due to of these activities. Some examples are:

- QR code scams,
- Fake payment requests,
- Phishing links or apps,
- Calls pretending to be someone,
- Fake customer service numbers,
- SIM swap fraud,

- Apps that manipulate users into disclosing their screen,
- Job, Loan or investment scams.

The damage caused by these tricks reveals that there is a demand for systems that can find suspicious behavior on their own. Machine Learning is a way forward because it can learn from historical transactions data and find patterns that are not easy to see when checking by hand.

In this project we used smart learning systems like: Logistic Classification model, Decision Tree-Based Classification model, K-Nearest Neighbors, Naïve Bayes, XG Boost and Cat Boost to find fake UPI transactions. Also Voting Classifier and Stacking Classifier are used to make predictions more accurate. The models can be checked using Accuracy, Precision, F1-Score, Confusion Matrix and Classification Report. The goal is to use ML algorithms with group methods then check them completely in order to make digital payment systems safer.

## 2. Literature Review

Many studies have looked within machine learning and Neural Network ways to spot fake financial activity.

**Pawade [1] (2023):** Created a fraud detection system that gathers details from how people spend money how they. What devices they use. Then provide this data into ML classifiers to catch activity as it happens.

**Meena, Harshitha & Kavitha [2] (2025): A Proactive Approach to Secure Digital Transactions:**

They tested tree-based models and anomaly-detection models like Random Forest, Isolation Forest and One-Class SVM on payment data. They used ROC curve analysis to see which model was best at telling the difference between strange transactions.

**Bappan [3] (2024):** Tried something by adding a Hidden Markov Model into the fraud detection pipeline. This helped to look at how transaction sequences change over time of just looking at one single transaction at a time.

**Research Gate [4] (2025): A Machine Learning Strategy Using Stacked Generalization:**

They put classifiers together using stacked generalization. The method layers models on top of each other. They then saved the fraud verdict on a private blockchain using a smart contract. This was done to make sure the fraud decision was accurate and could not be changed.

System that only use fixed rule can catch fraud patterns that the rules were made for. However these systems fail when criminals use clever tricks. This problem is why researches and banks are moving toward machine learning. Machine learning can change as fraud behavior changes so it does not have to rely on a rulebook.

Previous research has used different classifiers like Logistic Classification model, Decision Tree-Based Classification model, Naïve Bayes, Random Forest and XG Boost to decide if transactions are real or fraud. By learning from transaction records these techniques can identify hidden and flag suspicious activity much better than simple rules. Users are becoming increasingly interested in methods. Ensemble methods mix methods together to enhance the predictions better and prevent errors.

Newer studies show that boosting algorithms like XG Boost and Cat Boost work well on datasets where fraud is rare but important. Using approaches like Voting and Stacking Classifiers helps even more by using the best parts of many different models.

Using these ideas this study applies machine algorithms and ensemble techniques to build a strong and useful system. For finding UPI fraud.

### 3. Methodology

This study presents a framework based on machine learning to detect UPI transactions. The process goes through steps-collecting data preparing and cleaning it training models using combined methods and checking results-all designed to increase the chances of finding fraud while reducing wrong alerts.

#### A. Dataset

This study uses a group of UPI transaction records each with a set of details that describe the transaction. Each record is marked as either real or fake. This mark is what the models learn to predict. Before training the models the data was prepared in different steps.

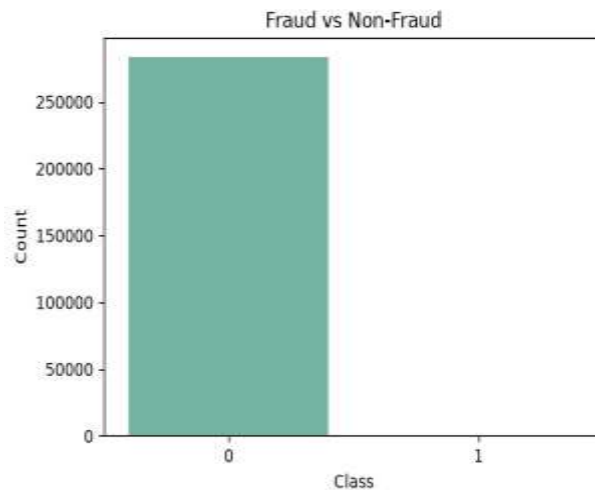


**Fig1. Distribution of Fraudulent VS Non- Fraudulent Transactions**

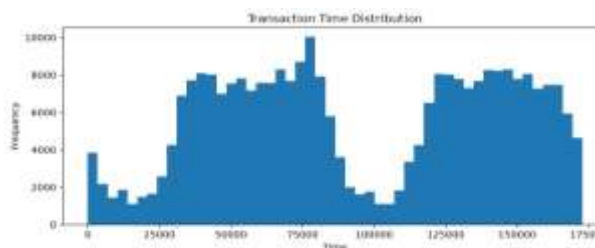


**Fig2. Transaction Time Distribution**

#### B. Data Preprocessing

Cleaning the data included filling in missing parts removing repeated rows and changing categories into numbers that models can use. When distance-based algorithms were used the characteristics were scaled to keep the performance the same. After cleaning the data was split into 80 percent for training and 20 percent for testing using the `train_test_split()` function from Scikit-Learn.

To deal with the fact that there were fake transactions Cost-Sensitive Learning was used with the `class_weight = "balanced"` parameter. This gives attention to fake transactions during training and improves the ability to detect fraud.



**Fig3. Class distribution in Training Data After SMOTE Balancing line**

### C. Machine Learning Models

The classifiers used in this study a comparison was made between are listed below:

- Logical Regression
- Decision Tree Classifier
- K-Nearest Neighbors (KNN)
- Naïve Bayes
- Support Vector Classifier (SVC)
- XG Boost Classifier
- Cat Boost Classifier

In addition to the single classifiers a Weighed Voting Ensemble was also created. Combining Logistic Classification, Decision Tree-Based Classification, XG Boost and Cat Boost. Each model was assigned a weight based on how it performed on its own, which helped make the final prediction more accurate.

The threshold for the model was adjusted from 0.50 to 0.30. This change made the model much better at finding transactions without reducing the overall quality of the results.

Each model was trained on the training part of the data. Then evaluated on the separate part to see how it would work in real situations.

### D. Ensemble Learning

Two methods were used to make the predictions more accurate.

**Voting Classifier:** Combines the results of Logistic Regression, Random Forest, XG Boost and Cat Boost to make the decision.

**Stacking Classifier:** Uses Random Forest, XG Boost and Cat Boost as base models and a final model combines their results, which improves the classification.

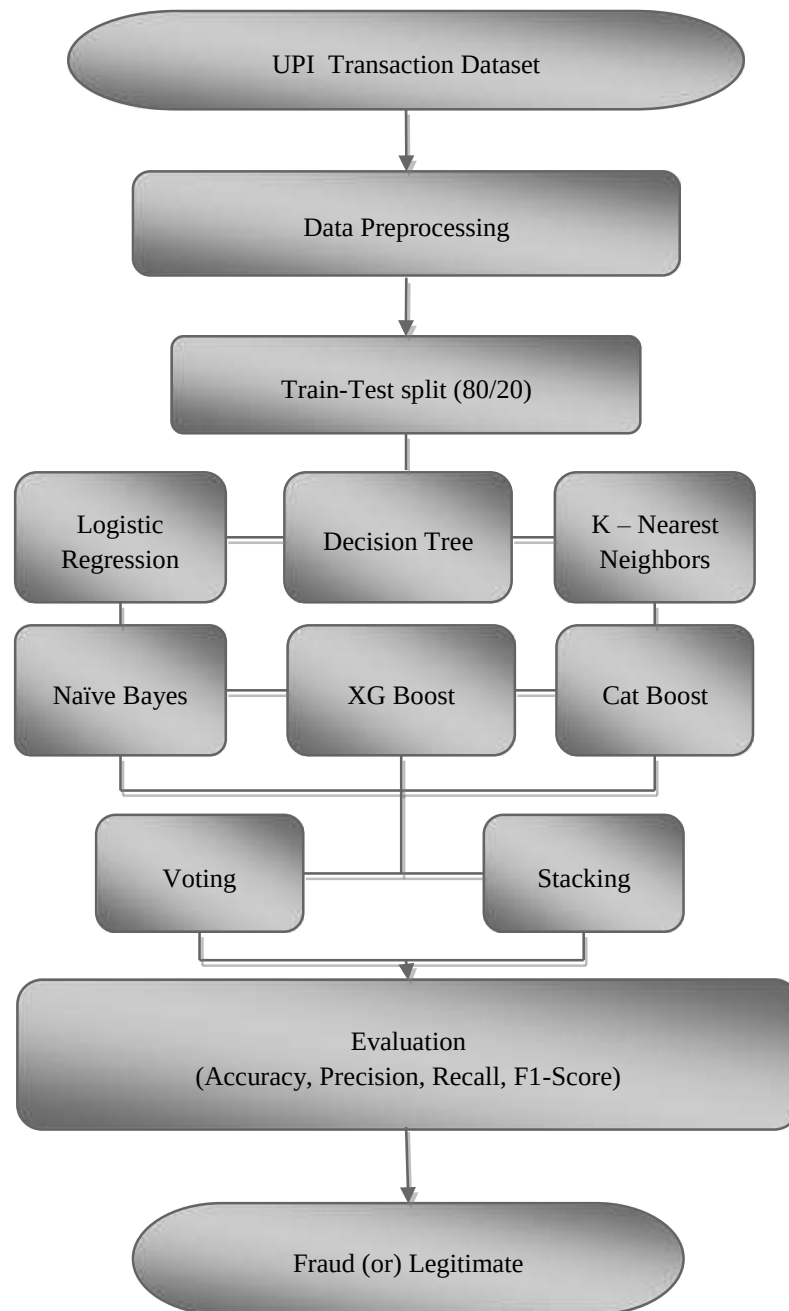


**Fig3. Methodology Flowchart for UPI Fraud Detection System.**

### E. Explainable AI using SHAP

To understand how the models make decisions SHAP (Shapley Additive Explanations) was used in the analysis. This method shows which parts of the data push a prediction toward being marked as fraud or real. Since XG Boost and Cat Boost had the results SHAP was used on these two models. The plots created show how each part of the data affects the prediction making the models choices clearer and more trustworthy.

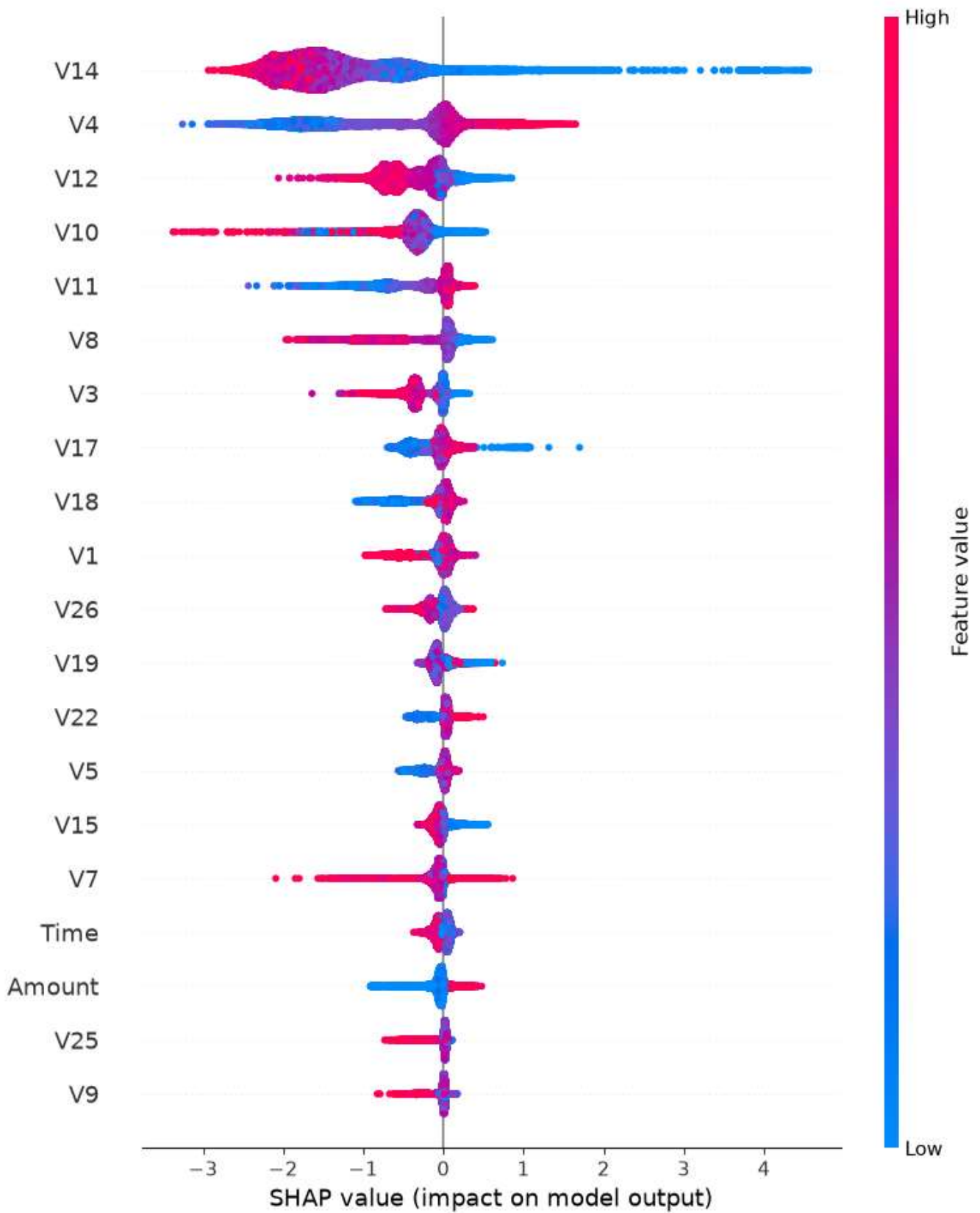


**Fig 4. SHAP Summary Plot for XG Boost**

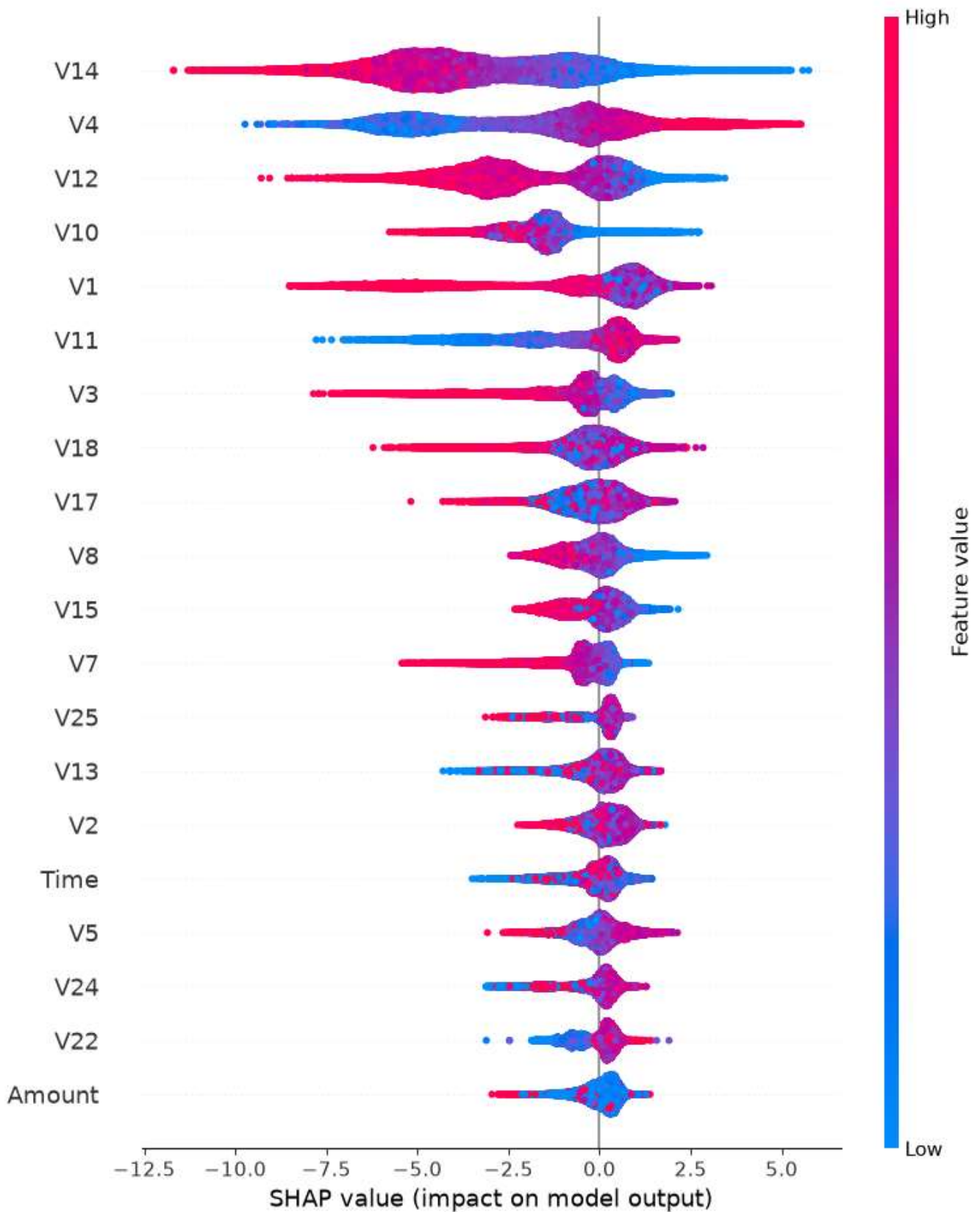


**Fig 5. SHAP Summary Plot for Cat Boost**

## F. Performance Evaluation

To assess the performance of each model worked the following measures were used:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix
- Classification Report

By looking at the results from these measures it was possible to find out which model-among the combined ones-was best, at finding fake UPI transactions.

## 4. Results and Evaluation

This section discusses the results of testing seven classifiers along with two ensemble approaches for detecting UPI fraud. The classifiers tested were Logistic Classification, Decision Tree-Based Classification, K-Nearest Neighbors, Naïve Bayes, XG Boost, Random Forest Classifier and Cat Boost. In addition Weighted Voting Classifier, Weighted Voting Ensemble and Stacking Classifier were used to improve the classification results.

| Model                      | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
|----------------------------|----------|-----------|--------|----------|---------|
| Random Forest              | 0.9962   | 0.2851    | 0.8315 | 0.4247   | 0.9753  |
| Stacking Classifier        | 0.9988   | 0.6147    | 0.7894 | 0.6912   | 0.9779  |
| Weighted Voting Classifier | 0.9984   | 0.5310    | 0.8105 | 0.6416   | 0.9779  |
| XG Boost                   | 0.9952   | 0.2342    | 0.8210 | 0.3644   | 0.9765  |
| Weighted Voting Ensemble   | 0.9979   | 0.4457    | 0.8210 | 0.5777   | 0.9753  |
| Logistic Regression        | 0.9736   | 0.0530    | 0.8736 | 0.1000   | 0.9626  |
| Cat Boost                  | 0.9885   | 0.5434    | 0.7894 | 0.6437   | 0.9390  |
| Naïve Bayes                | 0.9760   | 0.0543    | 0.8105 | 0.1017   | 0.9342  |
| K Nearest Neighbors        | 0.9979   | 0.4425    | 0.8105 | 0.5724   | 0.9099  |
| Decision Tree              | 0.9855   | 0.0829    | 0.7578 | 0.1495   | 0.8163  |

**Table 1. Comparative analysis of machine learning and ensemble learning models for UPI fraud detection.**

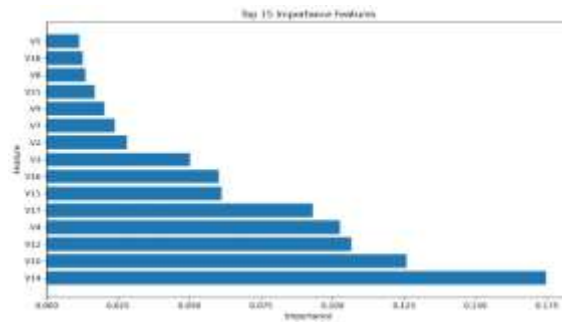


**Fig6.Important Features for Fraud Detection**

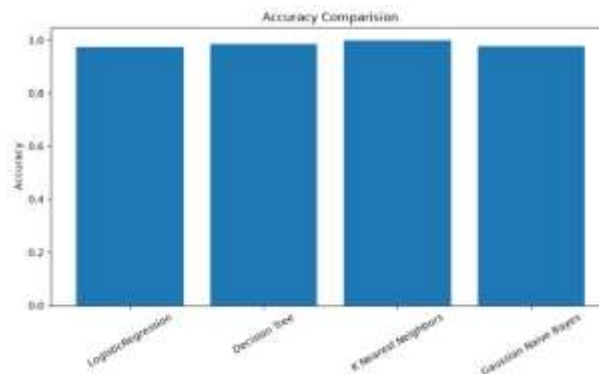


**Fig7.Accuracy Comparison Across Machine Learning Models**

Performance was measured using a collection performance metrics such as Accuracy, Precision, Recall, F1-Score, Confusion Matrix and Classification Report. What came out of the testing was a patterns: models that used ensemble methods consistently performed better than each model on its own. Voting Classifier improved the stability of predictions by combining multiple classifiers while the Stacking Classifier pushed performance even further by learning directly from the results generated by the base models.

Among all the models tested the Stacking Classifier achieved the accuracy and F1-Score. The Weighted Voting Classifier obtained the ROC-AUC among the models that were analyzed. The Confusion Matrix and Classification Report showed that the system was able to detect transactions effectively while keeping false predictions to a minimum.

Overall the framework developed here which integrates various machine learning models with ensemble techniques offers a practical and reliable way to catch fraudulent UPI activity. This approach has the potential to improve the safety of digital payment platforms secure.

The application of Weighted Voting Ensemble, Threshold Optimization, Cost-Sensitive Learning and SHAP Explainable AI improved the fraud detection system. Cost-Sensitive Learning focused more on minority fraud cases, Threshold Optimization improved recall. Weighted Voting improved the prediction performance. SHAP provided explanations for the model results.

## 5. Conclusion

This work aimed to build a machine learning-based approach to identify UPI transactions. A range of classification algorithms including Logistic Regression, Decision-Tree, K-Nearest Neighbors Classifier, Naïve Bayes, Support Vector Classifier, XG Boost and Cat Boost were Compared. Ensemble learning techniques such as Voting Classifier and Stacking Classifier helped improve the prediction performance.

Evaluation was based on Accuracy, Precision, Recall, F1-Score along with the Confusion Matrix and Classification Report. The results clearly showed that ensemble techniques increased both accuracy and the ability to catch fraud more any standalone model could achieve on its own. This kind of approach could offer banks and digital payment providers a tool, for identifying fraud and improving financial security, in the future work can focus on Neural Network-based learning models real-time fraud detection and explainable AI techniques to further improve the system.

## 6. References

1. Pedregosa, F., et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2011.
2. Chen, T., & Guestrin, C. XGBoost: A Scalable Tree Boosting System. Proceedings of KDD, 2016.
3. Prokhorenkova, L. et al. CatBoost: Unbiased Boosting with Categorical Features. NeurIPS, 2018.
4. Breiman, L. Random Forests. Machine Learning, 2001.
5. Cortes, C., & Vapnik, V. Support-Vector Networks. Machine Learning, 1995.
6. Cover, T., & Hart, P. Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 1997.
7. Kaggle. UPI Fraud Detection Dataset.
8. Official Python Documentation.
9. Official Scikit-Learn Documentation.
10. Research papers on fraud detection from IEEE Xplore.