UPI FRAUD DETECTION USING MACHINE LEARNING

-VATSHAYAN

UPI FRAUD
DETECTION
SYSTEM

Introduction to UPI Fraud Detection

- **What is UPI?**
  - A real-time payment system in India enabling instant money transfers.
- **Why UPI Fraud Detection is Important?**
  - Growth in digital payments has increased the risk of fraud.
  - Need for effective fraud detection to protect users and build trust.

Introduction to UPI Fraud Detection

- UPI is a popular real-time payment system in India, essential for its fast transactions but prone to fraud.
- With digital payments rising, fraud detection is crucial to protect users and maintain system integrity.
- This project uses machine learning to detect UPI fraud, with models that identify fraud patterns to minimize false positives and maximize accuracy.

Project Objectives

- Our goal is to detect UPI fraud in real-time with high accuracy, minimizing false positives to avoid blocking legitimate transactions.
- Key metrics include Accuracy, Precision, Recall, F1 Score, and ROC-AUC, each important for evaluating how well our system balances fraud detection and user experience.

Existing System

Existing fraud detection systems in digital payments are generally rule-based, relying on static thresholds or signatures. They primarily detect fraud based on pre-defined rules and known fraud patterns, which are less effective against new or adaptive fraud methods. Additionally, traditional systems often have latency in flagging suspicious activity, leading to delayed responses and potential financial losses for users. These systems lack the flexibility and intelligence to handle the complex and evolving nature of fraud.

Existing System Drawbacks

The major drawbacks of existing UPI fraud detection systems include:

- **Limited Adaptability**: Rule-based systems struggle to detect new, sophisticated fraud techniques.
- **High False Positives/Negatives**: Rigid rules can lead to inaccurate detection.
- **Latency**: Detection may be delayed, reducing the chance of preventing fraudulent transactions in real-time.
- **Scalability Issues**: Traditional systems struggle to handle large volumes of UPI transactions effectively.

Our project aims to address these limitations by implementing a machine learning model that adapts to evolving fraud patterns.

Proposed System

Our proposed system leverages machine learning to enhance UPI fraud detection by learning from transaction patterns and adapting to new fraud types. Using Python and Flask, our system processes transaction data in real-time, analyzing multiple features, such as transaction frequency, amounts, and geographic locations, to detect anomalies. The system provides a web-based interface, allowing users to monitor flagged transactions instantly. This approach aims to minimize false positives, improve adaptability, and provide more accurate fraud detection for UPI transactions.

Literature Survey

Our literature survey explores recent studies on machine learning models for fraud detection, including Random Forest, Decision Trees, and Neural Networks. Studies indicate that machine learning approaches can significantly reduce fraud detection times and improve accuracy compared to rule-based systems. Techniques such as anomaly detection and supervised learning have proven effective in identifying abnormal transaction patterns. The survey underscores the need for flexible models capable of real-time processing, particularly with UPI transactions growing in complexity and volume.

Architecture

Our system architecture includes the following components:

1. **Frontend Interface**: HTML/CSS-based interface for user interaction.
2. **Backend**: Flask server handling transaction data input and communication with the machine learning model.
3. **Database**: Stores historical transaction data for model training and evaluation.
4. **Machine Learning Model**: Processes transaction features and classifies them as legitimate or suspicious.

Machine Learning Algorithms Overview

- We implemented four algorithms:
  - **Random Forest**: Uses multiple decision trees for stable, accurate predictions.
  - **Logistic Regression**: A simple yet powerful binary classifier that calculates probabilities.
  - **Decision Tree**: A transparent model making decisions based on data features.
  - **Support Vector Machine (SVM)**: Separates classes by maximizing the margin, useful for complex data.

Why Random Forest for Fraud Detection?

- Random Forest is effective in handling large, complex datasets with multiple features.
- Its ensemble method reduces overfitting and improves accuracy, detecting patterns even in imbalanced data.
- Feature importance in Random Forest helps identify key fraud indicators, supporting transparency and deeper analysis.

Dataset Overview

- The dataset contains labeled UPI transactions, with features like Transaction Amount, Timestamp, User Location, and Device ID.
- It’s highly imbalanced, as fraud cases are rare.
- The binary target variable indicates whether each transaction is fraudulent or legitimate, providing the foundation for supervised learning.

Data Preprocessing Steps

- Data cleaning handles missing values and outliers.
- Feature engineering adds new variables like transaction intervals for deeper insights.
- Data scaling ensures consistency across algorithms, especially important for models like SVM and Logistic Regression.

Train-Test Split & Cross-Validation

- We split the data 80-20 for training and testing, ensuring enough data to learn and validate.
- K-fold cross-validation was used to check model robustness, improving accuracy by preventing overfitting on any single data partition.

Random Forest Classifier Implementation

- Implemented using `sklearn.ensemble.RandomForestClassifier` with parameters like `n_estimators` and `max_depth`.
- Chosen for its high accuracy and capability in detecting complex fraud patterns.
- Helps in quick decision-making for real-time fraud detection.

Logistic Regression Implementation

- Implemented using `sklearn.linear_model.LogisticRegression`.
- Chosen for its efficiency in binary classification and interpretability in terms of probability scores.
- Enables setting probability thresholds, allowing customization of fraud sensitivity levels.

Decision Tree Classifier Implementation

- Implemented with `sklearn.tree.DecisionTreeClassifier`.
- Known for easy visualization and interpretability, helping understand the decision-making process.
- Handles non-linear relationships well, which is useful for spotting diverse fraud patterns.

Support Vector Machine (SVM) Implementation

- Implemented with `sklearn.svm.SVC` with options for kernel, regularization, and gamma tuning.
- Effective for binary classification with clear boundaries.
- Ideal for complex data with well-separated classes, though less interpretable than trees.

Comparison of All Models

- **Summary Table**:
  - A table summarizes accuracy, precision, recall, F1, and ROC-AUC for each model.
  - Random Forest may emerge as the preferred model due to balanced performance across metrics.
  - Each model has strengths; Random Forest generally offers the best trade-off for fraud detection.

Fraud Detection Pipeline

- **Pipeline**:
  - Data Preprocessing → Feature Engineering → Model Training → Evaluation → Real-Time Deployment.
  - Real-time deployment integrates the model with UPI systems to detect fraud before transactions complete.

Challenges & Limitations

- **Challenges**:
  - Imbalanced datasets make fraud rare, complicating detection.
  - Real-time needs fast model inference.
- **Limitations**:
  - False positives inconvenience users; continuous model updates are needed to counter evolving fraud tactics.

Future Scope

- **Enhancements**:
  - Potential to explore neural networks or hybrid models.
  - Continuous updates and new fraud features to improve long-term efficacy.
- **Real-Time Optimization**:
  - Improve detection speed for seamless UPI integration.

Conclusion

- **Key Points**:
  - Random Forest shows strong performance, balancing accuracy with interpretability.
  - Machine learning models bring scalability and real-time potential to fraud detection.
- **Final Thought**:
  - The project underscores the value of AI in securing digital financial systems like UPI.

Q&A

- **Thank You!**
- Questions and discussion