VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590014

Project Report On

"UPI Fraud Detection"

Submitted in Partial fulfilment of the Requirements for the VIII Semester

Of the Degree of Bachelor of Engineering

In

Artificial Intelligence & Machine Learning

By

SK ISMAIL	**(1CE22AI402)**
SYED INSAF MEHDI	**(1CE21AI018)**
MOHAMMAD NADEER MM	**(1CE22AI401)**
YUVASHISH K	**(1CE21AI023)**

Under the Guidance of

Ms. Sagarika Patel

Asst. Professor, Dept. of AI&ML

CITY ENGINEERING COLLEGE

Doddakallasandra, Kanakapura Road, Bengaluru-560061

CITY ENGINEERING COLLEGE

Doddakallasandra, Kanakapura Road, Bengaluru-560061

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CERTIFICATE

Certified that the Project Report entitled **"UPI Fraud Detection"** has been carried out by **SK ISMAIL (1CE22AI402), SYED INSAF MEHDI (1CE21AI018), MOHAMMAD NADEER MM (1CE22AI401), YUVASHISH K (1CE21AI023),** Bonafide students of City Engineering College in partial fulfilment for the award of **Bachelor of Engineering in Artificial Intelligence & Machine Learning** of the Visveswaraya Technological University, Belgaum during the year 2024-2025. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements regarding Project Work prescribed for the said Degree.

Ms. Sagarika Patel	**Prof. Nandhish A C**	**Dr. S Karunakara**
Guide Prof, Dept. of AI&ML	Head, Dept. of AI&ML	Principal, C.E.C

External Viva

Name of Examiners — Signature with date

1)

2)

ACKNOWLEDGEMENT

While presenting this **"UPI Fraud Detection"**, I feel that it is my duty to acknowledge the help rendered to us by various people.

Firstly, I take this opportunity to thank my college **"CITY ENGINEERING COLLEGE"** for providing all the resource required to success the seminar report.

I would like to express our heartfelt gratitude to **Dr. S Karunakara**, Principal CEC, Bangalore and **Mr. Nandhish A C**, Head of Dept, Artificial Intelligence & Machine Learning for extending their support.

I would like to express a special thanks to the members of two different organizations who gave us permission to complete our activity on their premises and supported us throughout. I am very grateful to our guide **Ms. Sagarika Patel**, Asst. Prof., Department of Artificial Intelligence & Machine Learning for their able guidance and valuable advice at every stage of our activity which helped me in the successful completion of my activity. Their guidance and support were truly invaluable

I would also have indebted to our Parent and Friends for their continued moral and material support throughout the course of Technical Seminar and helping me in finalize the presentation. Our hearty thanks to all those have contributed bits, bytes and words to accomplish this Seminar Report.

SK ISMAIL (1CE22AI402)

SYED INSAF MEHDI (1CE21AI018)

MOHAMMAD NADEER MM (1CE22AI401)

YUVASHISH K (1CE21AI023)

Table of Contents

ABSTRACT

The Unified Payments Interface (UPI) has become a dominant method for digital transactions in India, offering users a fast, seamless, and real-time payment experience. However, with the growing popularity of UPI, there has been a corresponding rise in sophisticated frauds, especially those involving phishing links and malicious QR codes. Fraudsters exploit users by sending fake payment requests, embedding harmful URLs, or displaying tampered QR codes to divert payments to unauthorized accounts. Traditional security mechanisms often fall short in identifying such deceptive elements in real-time, thereby putting users at significant financial risk.

This project proposes a machine learning-based approach for the automated detection of phishing links and fraudulent QR codes in UPI transactions. The system leverages natural language processing (NLP) and computer vision techniques to analyse URL patterns, domain reputations, and QR code contents. Supervised learning models are trained on labelled datasets comprising both legitimate and malicious examples to distinguish between safe and harmful transaction prompts. The system also integrates real-time scanning and alerting capabilities, enhancing user protection at the point of transaction.

By accurately identifying phishing attempts and QR code manipulation, this study aims to reduce digital payment fraud, safeguard user assets, and strengthen the overall trust in UPI-based financial systems. Experimental results demonstrate the effectiveness of the proposed method in detecting fraud with high precision, thus validating the feasibility of applying machine learning in real-time transaction security.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The introduction of the Unified Payments Interface (UPI) has revolutionized the way digital payments are made in India, providing a quick, secure, and seamless means of transferring funds between accounts. UPI has become the backbone of the digital payments ecosystem, facilitating transactions through smartphones and digital banking systems. It offers convenience, flexibility, and accessibility for users, transforming how individuals and businesses interact with their finances. The UPI platform operates by enabling direct bank-to-bank transfers, providing a single interface for various payment systems, and reducing the need for multiple banking apps or payment modes. As a result, UPI has seen significant adoption, and its usage continues to increase as more individuals and businesses leverage their capabilities for everyday transactions.

However, as with any financial system, the rise in UPI usage has also led to a parallel increase in fraudulent activities. UPI frauds have become a significant concern for both users and financial institutions. The ease of transferring funds combined with the widespread use of mobile devices has created opportunities for cybercriminals to exploit the system and defraud users. UPI fraud is not limited to one specific form but manifests in a variety of techniques, including phishing, unauthorized transactions, and social engineering attacks. Phishing schemes, for instance, trick users into revealing sensitive personal information or OTPs, which can then be used to make unauthorized transfers. Malicious software or apps can compromise user devices, gaining access to UPI credentials and executing fraudulent transactions without the user's consent. Additionally, social engineering tactics are frequently employed, where attackers manipulate users into providing confidential information or authorizing payments under false pretenses.

These fraudulent activities have significant implications not only for individual users but also for the financial system. The ability to detect such fraud in real time is crucial in mitigating financial losses and ensuring the security of the system. Fraud detection is inherently challenging, as fraudsters continuously evolve their techniques to bypass existing security measures. Traditional methods of fraud detection often rely on predefined rules or heuristics, which may not be sufficient to identify emerging fraud patterns.

Given the dynamic nature of fraudulent activities and the large volume of transactions processed daily, there is a pressing need for more sophisticated fraud detection systems that can adapt to new trends and accurately identify fraudulent transactions.

Machine learning (ML) has emerged as a powerful tool for addressing this challenge. Machine learning algorithms can analyze vast amounts of transaction data to identify patterns and detect anomalies that may indicate fraudulent behavior. Unlike traditional rule-based approaches, machine learning models can learn from historical data and adapt to changing fraud tactics, making them more effective at identifying fraud in real-time. The ability of machine learning algorithms to process large datasets and identify hidden relationships between features enables the development of models that can detect complex fraud patterns with high accuracy.

In this project, we propose the use of machine learning techniques for UPI fraud detection. Specifically, we will explore four popular machine learning algorithms: Random Forest, Logistic Regression, Decision Tree, and Support Vector Machine (SVM). These algorithms are well-suited for classification tasks, where the objective is to categorize transactions as either legitimate or fraudulent based on the features of each transaction.

1.2 PROBLEM STATEMENT

The rapid increase in digital transactions, particularly through UPI, has been accompanied by an alarming rise in fraudulent activities targeting unsuspecting users. Fraudulent transactions can have severe financial consequences, and the financial institutions that provide UPI services face the challenge of securing the system against these threats. Existing fraud detection methods often lack the ability to quickly identify novel fraud tactics or adapt to new forms of attack. This project aims to address this gap by developing an effective fraud detection system using machine learning. The goal is to create a model that can classify transactions as either legitimate or fraudulent based on transaction data, offering a real-time, adaptive solution to fraud detection in UPI transactions.

1.3 OBJECTIVES

The primary objectives of this project are as follows:

1. **Data Collection**: To gather a comprehensive dataset of UPI transactions, including both legitimate and fraudulent transactions. This dataset will serve as the foundation for training and testing machine learning models.

2. **Data Preprocessing**: To clean and preprocess the data, ensuring that it is suitable for use in machine learning algorithms. This will include handling missing values, encoding categorical variables, and normalizing numerical features.

3. **Model Development**: To develop multiple machine learning models using different algorithms—Random Forest, Logistic Regression, Decision Tree, and Support Vector Machine (SVM). These models will be trained in the preprocessed transaction data.

4. **Model Evaluation**: To evaluate the performance of each machine learning model in terms of accuracy, precision, recall, F1 score, and false positive rate. The objective is to identify the most effective model for detecting UPI fraud with minimal false positives and high accuracy.

5. **Implementation and Deployment**: To integrate the selected machine learning model into a practical application using Flask. The application will provide a user interface for simulating transactions and testing the fraud detection system in real-time.

6. **Real-Time Fraud Detection**: To implement a real-time fraud detection mechanism that can process incoming transactions and flag potentially fraudulent activities based on the trained machine learning model.

1.4 Significance of the Study

The significance of this project lies in its potential to improve the security of UPI transactions and protect users from financial fraud. By developing an automated, machine learning-based fraud detection system, the project will contribute to the broader effort to enhance the safety and reliability of digital financial systems. Additionally, the project will provide insights into the strengths and weaknesses of different machine learning algorithms when applied to fraud detection tasks, helping financial institutions make informed decisions about which technologies to implement in their fraud prevention systems.

Furthermore, the research conducted in this project can be extended to other areas of financial transactions beyond UPI. Machine learning-based fraud detection systems can be adapted for use in various payment platforms, banking systems, and online transaction systems, providing a scalable and adaptable solution to the growing threat of cybercrime in the digital age.

1.5 Machine Learning in Fraud Detection

Machine learning is a subset of artificial intelligence (AI) that allows systems to learn from data and make predictions or decisions based on patterns and trends in that data. In the context of fraud detection, machine learning algorithms are trained on historical transaction data to recognize the characteristics of fraudulent behavior. Once trained, these models can be used to classify new transactions as either legitimate or fraudulent.

The four machine learning algorithms considered for this project—Random Forest, Logistic Regression, Decision Tree, and Support Vector Machine (SVM)—each offer unique strengths and are commonly used in classification problems:

1. **Random Forest**: Random Forest is an ensemble learning method that combines multiple decision trees to improve classification accuracy. It reduces overfitting and is highly effective at handling large datasets with complex relationships between features.

2. **Logistic Regression**: Logistic Regression is a statistical model used for binary classification. It predicts the probability of a transaction being fraudulent or legitimate based on the input features. While simple, it is often highly effective for problems with

linear decision boundaries.

3. **Decision Tree**: A Decision Tree is a tree-like structure used to make decisions based on input features. It recursively splits the data based on feature values, resulting in a tree where each leaf represents a classification. Decision trees are easy to interpret and understand.

4. **Support Vector Machine (SVM)**: SVM is a supervised learning algorithm that finds the optimal hyperplane to separate different classes in the feature space. SVM is particularly effective in high-dimensional spaces and works well with non-linear decision boundaries.

Each of these algorithms will be evaluated in this project to determine which provides the best performance for UPI fraud detection.

1.6 Dataset and Features

The dataset used in this project consists of transaction records, with each record representing a single UPI transaction. The features included in the dataset will capture various aspects of each transaction, such as:

- **Transaction Amount**: The value of the transaction being made.
- **Time of Transaction**: The timestamp indicating when the transaction took place.
- **Merchant Information**: Details about the merchant or recipient of the funds.
- **User Location**: Geographic information about the user initiating the transaction.
- **Device Information**: Information about the device used to make the transaction.
- **Transaction Type**: Whether the transaction was payment, fund transfer, or other types of actions.
- **User Behavior**: Historical data about the user's previous transactions, including frequency, amount, and location.

Fraudulent transactions often exhibit patterns that deviate from the normal transaction behavior of a user. Machine learning algorithms can be trained to detect these anomalies based on the available features, classifying transactions as fraudulent or legitimate.

CHAPTER 2

LITERATURE SURVEY

The Unified Payments Interface (UPI) has revolutionized the digital payments landscape, particularly in India, by providing a seamless and secure method of transferring funds across banks. Since its inception, UPI has gained widespread adoption due to its convenience, speed, and accessibility. However, the widespread use of UPI has also given rise to an increase in fraudulent activities, ranging from phishing attacks and account takeovers to unauthorized fund transfers and scams. Fraudulent activities in UPI transactions present a significant threat to the security and reliability of digital financial systems. As a result, real-time fraud detection mechanisms have become essential to maintain the trust and integrity of the UPI platform.

Traditional rule-based systems, which were the primary method for detecting fraud, are now being overshadowed by machine learning (ML) techniques, which offer the advantage of adaptability and scalability. The application of machine learning to UPI fraud detection has attracted significant attention from researchers and developers in recent years. The ability of machine learning models to detect complex and evolving fraud patterns in large datasets has proven to be a more effective solution than manual rule-based systems. Various machine learning algorithms, including Random Forest, Decision Trees, Logistic Regression, and Support Vector Machines (SVM), have been extensively explored for the task of fraud detection in digital payments, and each has its own strengths and weaknesses.

This literature survey delves into the application of machine learning techniques to fraud detection in the context of UPI, analyzing key research works, methodologies, and results. It aims to provide a comprehensive understanding of the challenges, approaches, and advancements in this area, as well as highlight future directions for improving fraud detection in digital payment systems.

The rapid adoption of digital payment systems, especially the Unified Payments Interface (UPI), has triggered a corresponding need for secure transaction mechanisms. Numerous studies have been conducted to explore how machine learning can aid in the detection of online financial fraud. This section highlights some of the significant works in this field.

2.1 Overview of UPI and Fraud Challenges

The Unified Payments Interface (UPI) is a real-time payment system developed by the National Payments Corporation of India (NPCI) that allows users to transfer funds instantly between banks through mobile applications. UPI has made digital transactions more accessible and efficient by facilitating direct transfers, bill payments, and merchant transactions. However, its open nature and ease of use have also attracted malicious actors who exploit vulnerabilities in the system to carry out fraudulent activities.

Some of the most common fraud types associated with UPI include:

1. **Phishing**: Fraudsters often trick users into revealing sensitive information such as UPI PINs, passwords, and account details through fake websites, phone calls, or messages.
2. **Unauthorized Transactions**: These occur when an attacker gains unauthorized access to a user's UPI account and initiates fraudulent transactions.
3. **SIM Swap Fraud**: Attackers swap a victim's SIM card with one they control, gaining access to OTPs and other authentication mechanisms required for UPI transactions.
4. **Malware and Keylogging**: Fraudulent software or malware installed on a victim's device can capture sensitive information, including UPI credentials and transaction details.
5. **Man-in-the-Middle Attacks**: Attackers intercept and alter communications between users and UPI servers to initiate fraudulent transactions.

Given the dynamic nature of these fraud methods, traditional rule-based detection systems are often ineffective in identifying novel or evolving fraud patterns. Machine learning, with its ability to learn from vast amounts of data and detect hidden patterns, provides an ideal solution for improving fraud detection in UPI systems.

2.2 Machine Learning in Fraud Detection

Machine learning (ML) has proven to be a powerful tool for detecting fraud in various areas, including banking, e-commerce, and financial transactions. The primary strength of ML lies in its ability to analyze large datasets and identify subtle relationships and patterns that may indicate fraudulent activity. ML models can be trained to recognize both known and previously unseen fraud patterns, making them more adaptive to evolving threats.

The use of machine learning for fraud detection typically involves the following steps:

- Data Collection: Relevant features such as transaction amount, time, user history, device information, and geographical location are collected from UPI transaction logs. These features serve as inputs for the ML models.
- Feature Engineering: Raw transaction data is preprocessed and transformed into a format that can be used by machine learning algorithms. This may involve normalization, encoding categorical features, and handling missing values.
- Model Training: A variety of machine learning algorithms are trained on labeled datasets, where transactions are classified as either legitimate or fraudulent.
- Evaluation: The trained models are evaluated based on performance metrics such as accuracy, precision, recall, F1 score, and false positive rate. The goal is to develop a model that can detect fraudulent transactions with high accuracy while minimizing false alarms.
- Deployment: Once trained, the machine learning model is deployed in a real-time environment to classify incoming transactions and flag potentially fraudulent ones.

Several machine learning algorithms have been explored for fraud detection, including supervised and unsupervised learning approaches.

2.3 Supervised Learning Approaches

Supervised learning algorithms are the most widely used in fraud detection because they learn from labeled data, where each transaction is tagged as either fraudulent or legitimate. Some of the most used supervised learning algorithms in fraud detection are as follows:

2.3.1 Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes predicted by individual trees. It is highly effective for classification tasks, including fraud detection, due to its ability to handle complex datasets with multiple features. Random Forest can capture non-linear relationships between features and is less prone to overfitting compared to individual decision trees.

Several studies have demonstrated the effectiveness of Random Forest for fraud detection in digital payments. For example, a study by Jiang et al. (2019) applied Random Forest to detect fraudulent transactions in online payment systems, achieving high accuracy and low false positives.

2.3.2 Decision Tree

Decision Trees are another popular choice for fraud detection due to their simplicity and interpretability. Decision Trees create a tree-like structure where each node represents a decision based on a feature, and the leaves represent the final classification (fraudulent or legitimate). While Decision Trees are easy to interpret, they are prone to overfitting, which can be mitigated by pruning or using ensemble methods such as Random Forest.

In the context of UPI fraud detection, Decision Trees can be used to identify patterns such as sudden changes in transaction behavior or unusual transaction amounts. Decision Trees can also be used for anomaly detection, where transactions that deviate from a user's typical behavior are flagged as suspicious.

2.3.3 Logistic Regression

Logistic Regression is a linear model commonly used for binary classification problems. It works by estimating the probability that a given transaction belongs to the fraudulent class based on the features. Logistic Regression is computationally efficient and can handle large datasets, making it suitable for real-time fraud detection in UPI transactions.

A study by Sharma et al. (2017) found that Logistic Regression was effective in detecting fraud in financial transaction datasets, particularly when the relationship between features and the target was linear. However, its performance may degrade when dealing with highly non-linear data, which is where more complex models such as Random Forest or SVM may outperform Logistic Regression.

2.3.4 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm that works well for high-dimensional data and classification problems with complex decision boundaries. SVM tries to find the hyperplane that best separates the data into two classes, and it is particularly useful for fraud

detection when the data is highly imbalanced (i.e., when fraudulent transactions are much rarer than legitimate ones).

In the context of UPI fraud detection, SVM has been shown to perform well in detecting sophisticated fraud patterns. For instance, SVM has been applied to detect credit card fraud by combining kernel functions with cost-sensitive learning, which helps address the issue of class imbalance. SVM is known for its high accuracy, but it can be computationally expensive, particularly when dealing with large datasets.

2.4 Unsupervised Learning Approaches

Unsupervised learning techniques are used when labeled data is not available or when the fraud patterns are not well-defined. These techniques aim to identify anomalous transactions without prior knowledge of what constitutes fraud. Some common unsupervised learning methods for fraud detection include:

2.4.1 Clustering Algorithms

Clustering algorithms, such as k-means and DBSCAN, can be used to group similar transactions together. Transactions that do not fit into any cluster or exhibit unusual characteristics can be flagged as anomalies and investigated further. These techniques are useful for detecting new fraud patterns that may not be present in historical data.

2.4.2 Anomaly Detection

Anomaly detection techniques, such as Isolation Forest and One-Class SVM, are specifically designed to detect outliers or rare events in datasets. These methods are particularly useful in fraud detection when fraudulent transactions are sparse and differ significantly from normal transactions. Anomaly detection can be used in combination with supervised models to improve overall fraud detection performance.

2.5 Hybrid Approaches

Some studies have explored the combination of different machine learning techniques to improve fraud detection performance. Hybrid models combine the strengths of multiple algorithms to create a more robust system. For example, ensemble models that combine Random Forest with SVM or Decision Trees can be used to achieve higher accuracy and reduce the false positive rate.

Hybrid approaches have shown promising results in fraud detection tasks by leveraging the strengths of different algorithms.

2.6 Evaluation Metrics and Performance Optimization

The performance of machine learning models for fraud detection is typically evaluated using several key metrics:

- **Accuracy**: The percentage of correctly classified transactions (both legitimate and fraudulent).
- **Precision**: The percentage of true positives (fraudulent transactions) among all predicted positives.
- **Recall**: The percentage of true positives among all actual fraudulent transactions.
- **F1 Score**: The harmonic mean of precision and recall, providing a balance between the two.
- **False Positive Rate**: The percentage of legitimate transactions incorrectly classified as fraudulent.

In addition to evaluating the model's performance using these metrics, hyperparameter tuning, cross-validation, and feature selection are essential for improving the model's accuracy and robustness.

2.7 Challenges and Future Directions

Despite the progress made in applying machine learning to UPI fraud detection, several challenges remain. These include the issue of data imbalance, where fraudulent transactions are much less frequent than legitimate ones, leading to biased models. Addressing this challenge requires specialized techniques such as cost-sensitive learning, synthetic data generation (e.g., SMOTE), and resampling methods.

Moreover, fraudsters continually evolve their tactics, making it essential for fraud detection models to adapt to new patterns. Continuous learning and model retraining will be crucial for ensuring that fraud detection systems remain effective in the face of emerging threats.

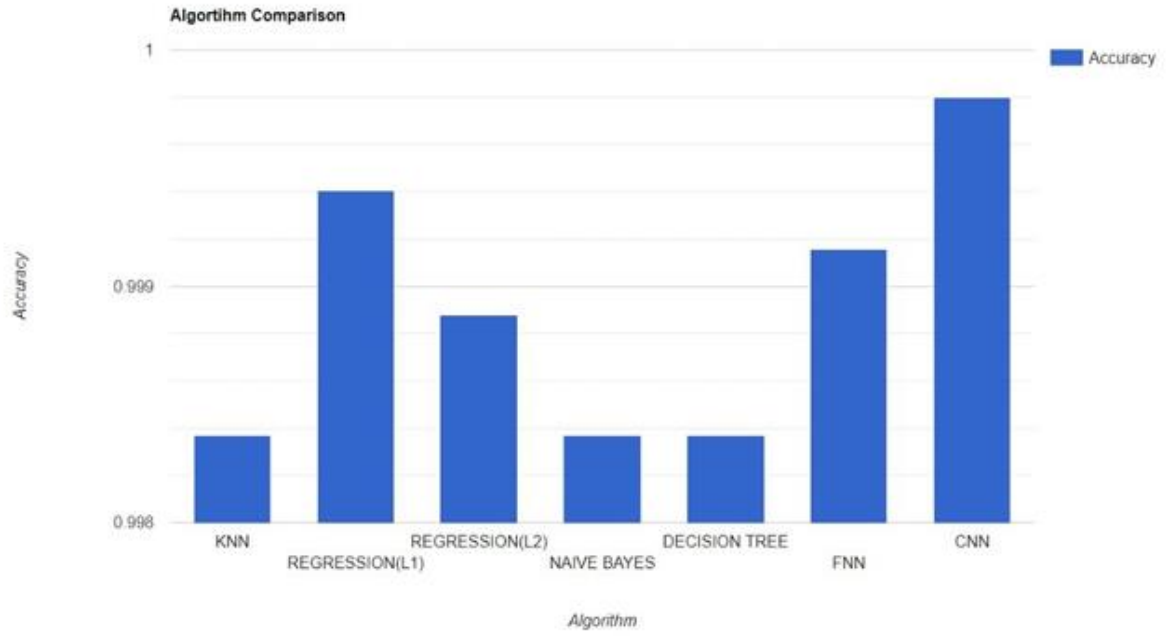
CHAPTER 3

ALGORITHMS AND TECHNIQUES

3.1 Machine Learning Algorithms Used for UPI Fraud Detection

Machine learning (ML) is increasingly being used for fraud detection in financial transactions, including Unified Payments Interface (UPI) transactions. Fraud detection using machine learning involves training algorithms on historical data to learn patterns and behaviors that distinguish legitimate transactions from fraudulent ones. In this context, four widely used machine learning algorithms—Random Forest, Logistic Regression, Decision Tree, and Support Vector Machine (SVM)—are employed to detect UPI fraud.

Each of these algorithms has unique characteristics, and understanding how they work, their strengths, and their limitations is critical for implementing an effective fraud detection system. The following sections provide a detailed explanation of each of these algorithms, their application to fraud detection, and a comparative analysis.



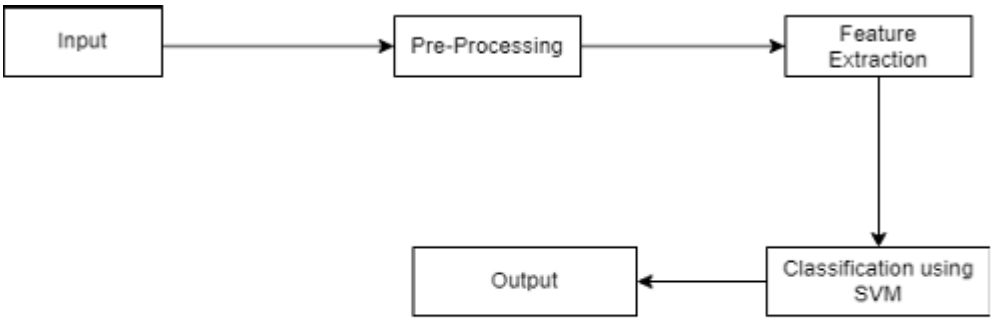


Figure 3: Data Flow Diagram

3.2 Random Forest

Random Forest is an ensemble learning algorithm that is based on the concept of decision trees. An ensemble method combines multiple individual models to improve performance and increase the robustness of predictions. Random Forest uses many decision trees to make a final decision by averaging or voting the results from individual trees. It is particularly popular in classification tasks, including fraud detection, due to its ability to handle complex datasets with high dimensionality and non-linearity.

3.2.1 How Random Forest Works

The key steps involved in the operation of Random Forest are:

- **Bootstrapping**: Random Forest builds multiple decision trees by sampling subsets of the data with replacement. This process, known as bootstrapping, ensures that the model generalizes well and avoids overfitting to a particular set of data.
- **Feature Randomization**: At each node in the decision tree, a random subset of features is selected to determine the best split. This randomness in feature selection helps reduce correlation between individual trees and improves the overall model's generalization power.
- **Majority Voting (for classification)**: In the case of classification tasks such as fraud detection, the final prediction is made based on the majority vote of all the individual decision trees. The class with the most votes from all trees is selected as the final output.

3.2.2 Application of Random Forest in UPI Fraud Detection

In the context of UPI fraud detection, Random Forest can be used to analyze historical transaction data and identify patterns associated with fraudulent activity. Features such as transaction amount, sender and receiver details, device information, geographical location, and time of transaction are typically used to train the Random Forest model. Once trained, the model can predict whether a new transaction is legitimate or fraudulent based on the learned patterns.

Random Forest is well-suited for UPI fraud detection because it can handle large amounts of data and is less prone to overfitting, a common issue in machine learning models. Its ability to handle both numerical and categorical features makes it highly versatile for financial transaction datasets.

3.2.3 Strengths and Limitations of Random Forest

Strengths:

- Robust to overfitting due to the averaging of multiple trees.
- Handles both numerical and categorical data efficiently.
- Can model non-linear relationships between features.
- Provides feature importance, allowing practitioners to understand which features are most relevant for fraud detection.

Limitations:

- Computationally intensive, especially when dealing with many trees and high-dimensional data.
- Less interpretable compared to a single decision tree, making it harder to explain specific predictions to stakeholders.
- Can struggle with imbalanced datasets, as fraud cases are often much rarer than legitimate transactions.

3.3 Logistic Regression

Logistic Regression is a statistical method used for binary classification problems. It models the probability of a binary outcome (such as fraud or not fraud) as a function of input features. Despite its name, Logistic Regression is a classification algorithm rather than a regression algorithm. It is

widely used in various areas, including finance, healthcare, and marketing, for tasks like predicting customer churn, disease outbreaks, and fraud detection.

3.3.1 How Logistic Regression Works

Logistic Regression predicts the probability that a transaction belongs to one of two classes (fraud or legitimate). It is based on the logistic function (also known as the sigmoid function), which transforms the output of a linear model into a probability value between 0 and 1. The steps involved in Logistic Regression are as follows:

- **Model Equation**: The logistic regression model is represented by the following equation: P(y=1|X)=11+e−(b0+b1x1+b2x2+...+bnxn)P(y = 1|X) = \frac{1}{1 + e^{-(b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n)}} where P(y=1|X)P(y = 1|X) is the probability that the transaction is fraudulent, x1,x2,...,xnx_1, x_2, ..., x_n are the features, and b0,b1,...,bnb_0, b_1, ..., b_n are the coefficients that the algorithm learns during training.
- **Training**: Logistic Regression uses an optimization algorithm, such as gradient descent, to minimize the error (cost function) by adjusting the coefficients. The goal is to find the best set of coefficients that maximizes the likelihood of observing the training data.

Prediction: Once trained, Logistic Regression assigns a probability to each transaction. If the probability exceeds a certain threshold (e.g., 0.5), the transaction is classified as fraudulent.

3.3.2 Application of Logistic Regression in UPI Fraud Detection

In UPI fraud detection, Logistic Regression can be applied to predict the likelihood that a given transaction is fraudulent. Features such as transaction amount, sender/receiver details, and time of transaction can be used to train the model. The output of the Logistic Regression model is a

probability score, which can then be used to classify a transaction as fraudulent or legitimate based on a predefined threshold.

Logistic Regression is often preferred when the relationship between features and the target variable (fraudulent or not) is expected to be linear. It is a fast and efficient algorithm, making it suitable for real-time fraud detection in UPI systems.

3.3.3 Strengths and Limitations of Logistic Regression

Strengths:

- Simple and easy to implement.
- Computationally efficient, making it suitable for real-time systems.
- Provides probabilistic output, which can be useful for ranking transactions by fraud likelihood.
- Interpretable model, as the coefficients represent the influence of each feature on the outcome.

Limitations:

- Assumes a linear relationship between features and the target, which may not always hold for complex fraud patterns.
- Sensitive to outliers and multicollinearity in the dataset.
- Struggles with highly imbalanced datasets, as fraudulent transactions are often much rarer than legitimate ones.

3.4 Decision Tree

A Decision Tree is a supervised learning algorithm used for both classification and regression tasks. It builds a model that splits the data into subsets based on feature values, ultimately arriving at a decision (classification) at the leaf nodes. In fraud detection, Decision Trees are used to classify transactions as legitimate or fraudulent based on transaction features such as amount, sender/receiver information, and device details.

3.4.1 How Decision Tree Works

A Decision Tree works by recursively splitting the data at each node to minimize impurity (e.g., Gini impurity or entropy) until the data at each leaf node is homogeneous. The tree is built in the following steps:

- **Splitting**: At each node, the algorithm selects the feature and threshold that provides the best split, which minimizes the impurity or maximizes the information gain.

- **Tree Growth**: This process is repeated recursively until stopping criteria are met (e.g., maximum depth or minimum number of samples in a leaf node).
- **Classification**: Once the tree is built, new data points (transactions) are classified by traversing the tree from the root to a leaf, following the feature splits.

3.4.2 Application of Decision Tree in UPI Fraud Detection

Decision Trees are ideal for detecting fraud in UPI transactions because they can easily model non-linear relationships between features and are easy to interpret. For instance, if the transaction amount exceeds a certain threshold, it might be flagged as suspicious. Similarly, if a transaction originates from an unrecognized device or location, the model may classify it as potentially fraudulent.

3.4.3 Strengths and Limitations of Decision Tree

Strengths:

- Intuitive and easy to interpret, making it suitable for decision-making in real-time systems.
- Can handle both numerical and categorical features.
- Effective at capturing non-linear relationships between features.

Limitations:

- Prone to overfitting, especially with complex trees and noisy data.
- Sensitive to small variations in the data.
- It can be computationally expensive for large datasets if the tree becomes too deep.

3.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful supervised learning algorithm that is used for both classification and regression tasks. SVM works by finding the hyperplane that best separates data points belonging to different classes. It is particularly useful for high-dimensional datasets and is widely applied in fraud detection problems where classes are highly imbalanced (fraud vs. legitimate).

3.5.1 How SVM Works

SVM operates by finding a hyperplane that separates data points from different classes with the maximum margin. The steps involved in the operation of SVM are:

- **Linear SVM**: In its basic form, SVM finds the hyperplane that maximizes the margin between two classes. For a binary classification task, SVM ensures that the gap between the closest points from both classes is as wide as possible.
- **Kernel Trick**: SVM can handle non-linear separations by using a kernel function, which maps the data into a higher-dimensional space where it is easier to find a hyperplane. Common kernel functions include the radial basis function (RBF) and polynomial kernels.
- **Support Vectors**: The data points closest to the separating hyperplanes are called support vectors. These support vectors play a crucial role in determining the optimal decision boundary.

3.5.2 Application of SVM in UPI Fraud Detection

SVM is particularly effective in fraud detection when the data is high-dimensional, and the classes are imbalanced. For example, the fraud class is often much smaller than the legitimate class, and SVM's ability to handle this class imbalance makes it a good choice. SVM's ability to capture complex relationships in the data, especially using non-linear kernels, makes it highly suitable for detecting sophisticated fraud patterns.

3.5.3 Strengths and Limitations of SVM

Strengths:

- Effective in high-dimensional spaces.
- It can handle non-linear relationships using kernel functions.
- Particularly suitable for imbalanced datasets.

Limitations:

- Computationally expensive, especially with large datasets.
- Difficult to interpret, as the decision boundaries are often not intuitive.
- Requires careful selection of kernel functions and hyperparameters.

CHAPTER 4

METHODOLOGY

4.1 UPI Fraud Detection Using Machine Learning

The methodology section outlines the approach and steps taken to develop the UPI fraud detection system using machine learning. The project integrates various machine learning algorithms, such as Random Forest, Logistic Regression, Decision Tree, and Support Vector Machine (SVM), to classify UPI transactions as either legitimate or fraudulent. The following sections explain the workflow of the project, from data collection and preprocessing to model evaluation and deployment.

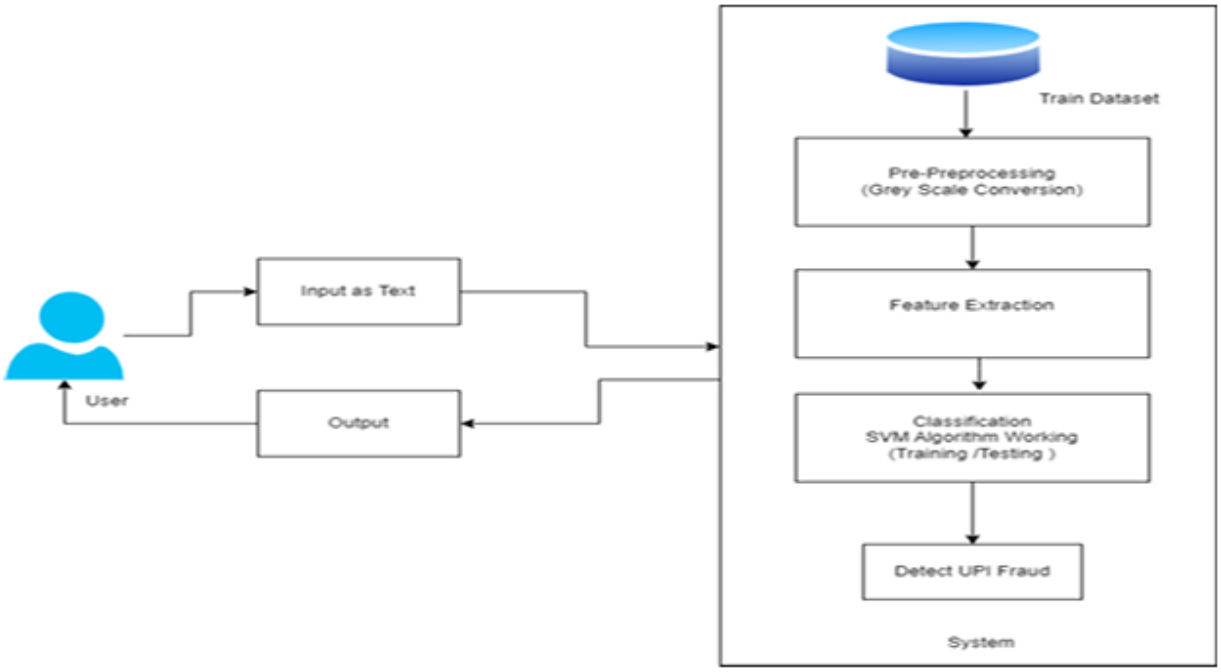


Figure 1: Architecture

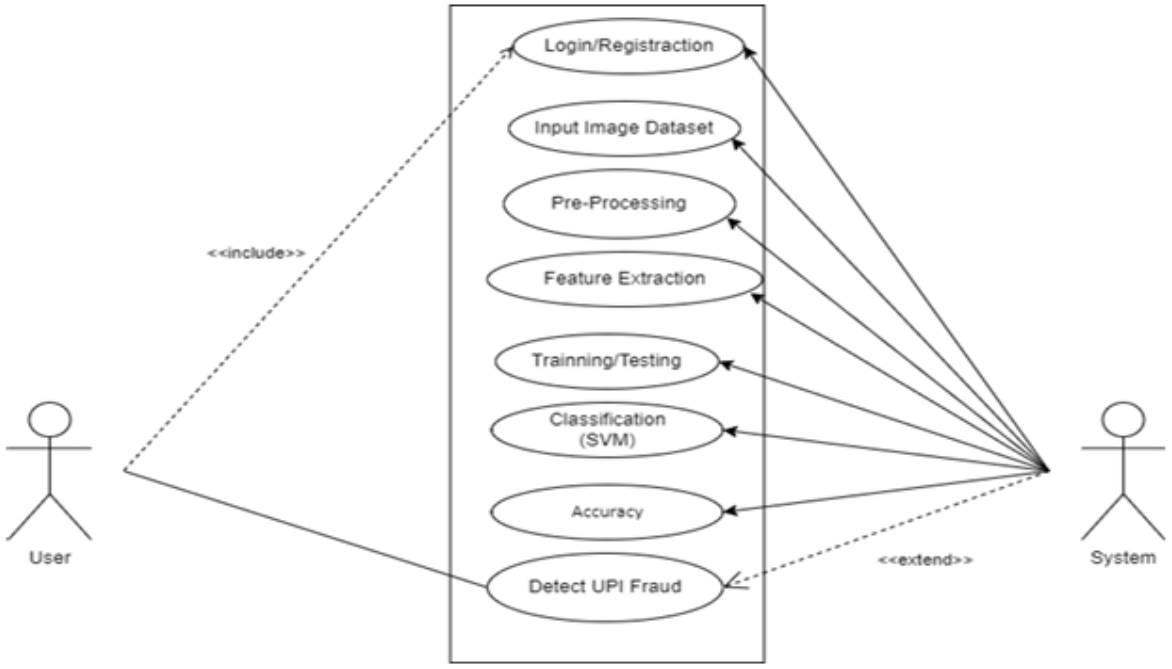


Figure 2: Use Case Diagram

4.2 Data Collection

The first step in the methodology is to gather relevant data for building the fraud detection model. The dataset needs to represent real-world UPI transactions, containing both legitimate and fraudulent transactions. This data should include various features, such as transaction amount, sender and receiver information, time, location, device information, and transaction history.

In real-world applications, transaction data can be gathered from financial institutions or UPI service providers, provided that the data is anonymized to protect users' privacy. For this project, a synthetic dataset may be used, where fraudulent and legitimate transactions are simulated to create a balanced dataset for training and testing machine learning models.
Key features of the dataset include:

- **Transaction Amount**: The amount of money being transferred.
- **Transaction Time**: The timestamp of the transaction.
- **Transaction Location**: Geographical location of the sender or receiver (e.g., city, IP address).
- **Sender and Receiver Information**: Identifiers like bank account or phone number.
- **Device Information**: Details of the device used, such as device ID or browser fingerprint.

The dataset is typically labeled as 'fraudulent' or 'legitimate,' with fraud cases representing a minority class. This imbalance is common in fraud detection tasks, where fraudulent transactions make up a very small portion of all transactions.

4.3 Data Preprocessing

Before training machine learning models, it is crucial to preprocess the raw data to ensure that it is in a suitable format for modeling. Data preprocessing involves the following steps:

4.3.1 Handling Missing Data

Missing values in the dataset can occur due to various reasons, such as incomplete records or errors during data collection. These missing values must be handled appropriately to prevent bias or errors during model training. There are several strategies for dealing with missing data:

- **Imputation**: Missing numerical data can be replaced with the mean, median, or mode of the respective column.
- **Deletion**: Rows with missing values can be removed if the missing data is not crucial or if the dataset is large enough that deletion does not significantly impact model performance.
- **Prediction Models**: In some cases, missing data may be predicted using another machine learning model trained on the rest of the features.

4.3.2 Feature Encoding

Many machine learning models require numerical input data. Features like transaction location, sender information, and device ID might be categorical, which need to be converted into numerical form. This can be done using techniques such as:

- **Label Encoding**: Converting categorical labels into integers (e.g., converting different cities or device types into numerical labels).
- **One-Hot Encoding**: Creating binary columns for each category (e.g., creating separate columns for different cities and assigning 1 or 0 based on whether the transaction occurred in that city).

This hierarchical representation allows the model to detect subtle deviations and hidden patterns that may not be apparent in the original data or through manual inspection.

4.3.3 Feature Scaling

To improve the performance of machine learning models, it is important to scale features so that they have comparable ranges. This is particularly necessary for models like SVM and logistic regression that are sensitive to feature scale. Common scaling techniques include:

- **Normalization**: Rescaling the features to a range between 0 and 1.
- **Standardization**: Transforming the features to have a mean of 0 and a standard deviation of 1.

4.3.4 Data Splitting

After preprocessing, the dataset is split into two parts: the training set and the testing set. The training set is used to train machine learning models, while the testing set is used to evaluate their performance. A common split is 70% of the data for training and 30% for testing, though this can be adjusted based on the dataset size.

Additionally, cross-validation may be employed during model training to further ensure the robustness and generalizability of the model. Cross-validation divides the training data into multiple folds and trains the model multiple times to assess its performance across different subsets of data.

4.4 Model Selection and Training

In this project, four machine learning algorithms are employed to detect UPI fraud: Random Forest, Logistic Regression, Decision Tree, and Support Vector Machine (SVM). Each of these models has its strengths and weaknesses, and the goal is to assess which performs best for the task of fraud detection.

4.4.1 Random Forest

Random Forest is an ensemble model based on multiple decision trees. It works by combining the predictions of many decision trees to make a final prediction. Random Forest is known for its high accuracy and ability to handle large, high-dimensional datasets with minimal overfitting.

- **Training**: Random Forest is trained by generating multiple decision trees using bootstrapped samples of the training data. Each tree is trained independently, and the final prediction is made by averaging the results from all trees (for regression) or by majority voting (for classification).

- **Hyperparameters**: Some key hyperparameters to tune for Random Forest include the number of trees in the forest, maximum depth of the trees, minimum samples required for a split, and the number of features considered for each split.

4.4.2 Logistic Regression

Logistic Regression is a linear model used for binary classification problems. It predicts the probability that a transaction is fraudulent based on a logistic function applied to a linear combination of input features.

- **Training**: The model is trained by minimizing the log-loss or cross-entropy loss, which penalizes incorrect predictions. During training, the model learns the optimal coefficients for each feature that best fits the data.
- **Hyperparameters**: Key hyperparameters for Logistic Regression include the regularization parameter (C), which controls the model's complexity, and the choice of solver for optimization (e.g., 'liblinear', 'saga').

4.4.3 Decision Tree

A Decision Tree model splits the data into subsets based on feature values to classify transactions as legitimate or fraudulent. It is a simple and interpretable model but can suffer from overfitting if not properly tuned.

- **Training**: The algorithm recursively splits the dataset into subsets, selecting the feature that provides the best split at each node. The splitting criterion is typically based on measures like Gini impurity or information gain.

- **Hyperparameters**: Key hyperparameters include the maximum depth of the tree, the minimum number of samples required to split a node, and the criterion for splitting.

4.4.4 Support Vector Machine (SVM)

SVM is a powerful classifier that finds the hyperplane that best separates the data into different classes. It is particularly useful for high-dimensional data and imbalanced datasets.

- **Training**: SVM learns the optimal hyperplane by maximizing the margin between different classes. The margin is the distance between the hyperplane and the closest data points from each class, known as support vectors.
- **Hyperparameters**: Key hyperparameters for SVM include the choice of kernel (linear, polynomial, RBF), the regularization parameter (C), and the kernel coefficient (gamma) for non-linear kernels.

4.5 Model Evaluation

After training the models, their performance must be evaluated to determine their effectiveness at detecting fraudulent transactions. The following evaluation metrics are used:

4.5.1 Accuracy

Accuracy measures the proportion of correctly classified transactions (both legitimate and fraudulent) to the total number of transactions. However, accuracy alone may not be sufficient for imbalanced datasets, where fraudulent transactions make up a small fraction of all transactions.

4.5.2 Precision, Recall, and F1-Score

Precision, recall, and the F1-score provide a more detailed assessment of model performance, especially in the case of imbalanced datasets.

- **Precision**: The proportion of correctly identified fraudulent transactions out of all transactions classified as fraudulent.
- **Recall**: The proportion of correctly identified fraudulent transactions out of all actual fraudulent transactions.
- **F1-Score**: The harmonic mean of precision and recall, which balances the trade-off between the two.

4.5.3 Confusion Matrix

The confusion matrix provides a detailed breakdown of the model's classification performance. It contains the following values:

- **True Positives (TP)**: Correctly classified fraudulent transactions.
- **False Positives (FP)**: Legitimate transactions incorrectly classified as fraudulent.
- **True Negatives (TN)**: Correctly classified legitimate transactions.
- **False Negatives (FN)**: Fraudulent transactions incorrectly classified as legitimate.

From the confusion matrix, various other metrics, such as accuracy, precision, recall, and F1-score, can be derived.

4.5.4 ROC Curve and AUC

The Receiver Operating Characteristic (ROC) curve is a graphical representation of the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity). The Area Under the Curve (AUC) measures the overall ability of the model to discriminate between fraudulent and legitimate transactions.

4.6 Deployment with Flask

After selecting the best-performing model, the next step is to deploy the fraud detection system using the Flask framework. Flask is a lightweight web framework for Python, commonly used for building web applications and APIs. The model can be deployed as a REST API, where new UPI transaction data can be sent for classification in real time.

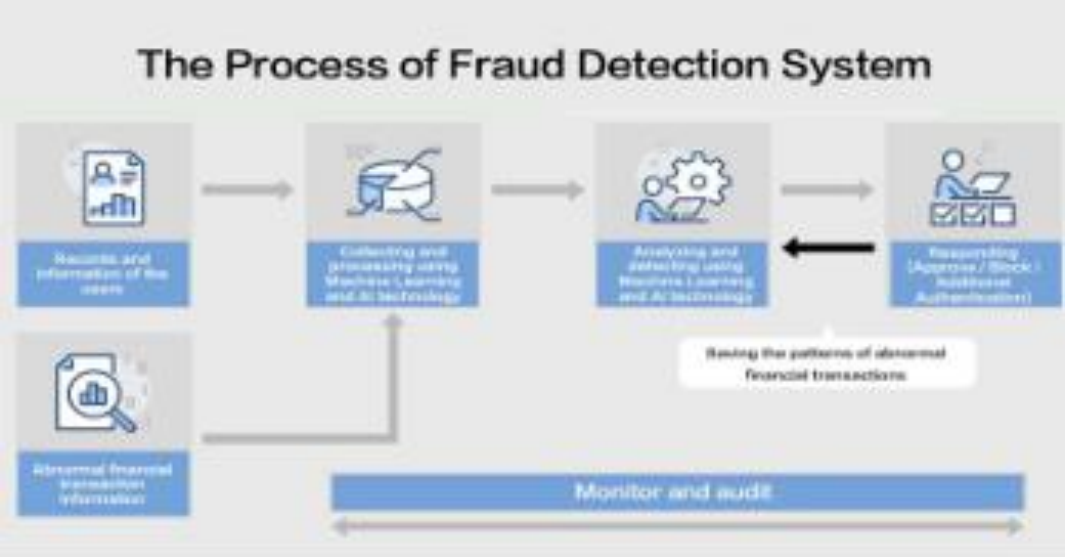
Steps for deployment include:

- **Model Serialization**: Serialize the trained machine learning model using libraries like joblib or pickle so that it can be loaded into the Flask application.
- **API Development**: Develop a Flask API that accepts transaction data as input, processes the data, and returns predictions (fraud or legitimate) in real time.
- **Integration**: Integrate the API with UPI systems so that transaction data is automatically fed to the model for classification.

CHAPTER 5

PROPOSED SYSTEM

5.1 Proposed System for UPI Fraud Detection using Machine Learning



The growing popularity of the Unified Payments Interface (UPI) in India has revolutionized the digital payments landscape, offering a fast, convenient, and secure platform for transferring money across banks. However, with the increased adoption of UPI, there has been a corresponding rise in fraudulent activities, making the need for robust fraud detection systems more urgent than ever. UPI frauds can take various forms, including phishing, unauthorized transactions, and the exploitation of malicious software to access user accounts. These fraudulent activities threaten the integrity and security of the platform, undermining user confidence and risking significant financial losses.

To combat these challenges, a machine learning-based fraud detection system is proposed. The core idea of this system is to classify UPI transactions into legitimate and fraudulent categories based on patterns identified using machine learning algorithms. By analyzing historical transaction data, the system can identify anomalies or suspicious behavior indicative of fraud and flag such transactions in real-time. Machine learning offers the advantage of continuously improving detection accuracy as the system learns from new transaction data, enabling it to detect even previously unseen forms of fraud.

5.2 System Overview

The proposed system is a comprehensive solution that uses multiple machine learning algorithms, including Random Forest, Logistic Regression, Decision Tree, and Support Vector Machine (SVM), to detect fraudulent transactions. The system is designed to be scalable, real-time, and adaptive, making it suitable for the large volumes of transactions processed daily on the UPI platform. It consists of several key components: data collection, data preprocessing, machine learning model development, fraud detection, and deployment through a web-based interface.

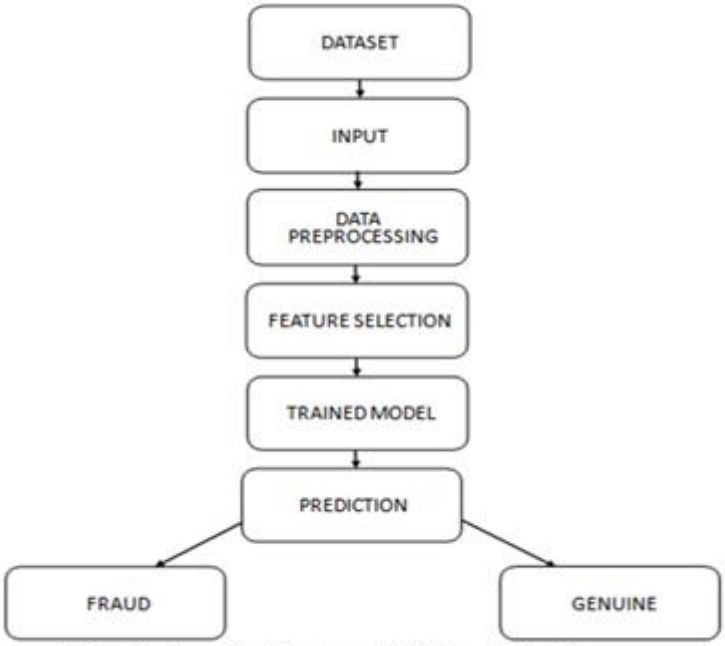


Fig 1: System Architecture of UPI fraud detection

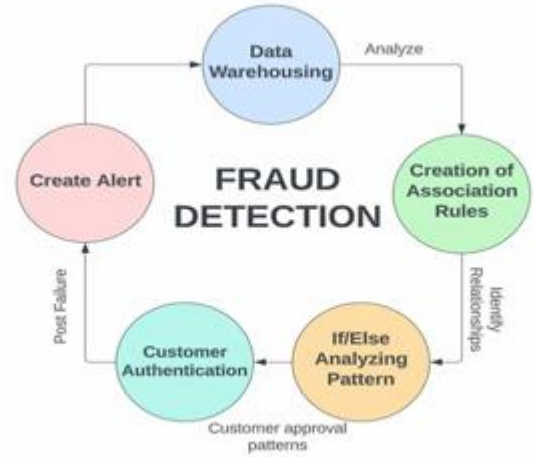


Fig: System Diagram for UPI fraud Detection using machine learning

5.3 Data Collection and Features

The foundation of any machine learning-based fraud detection system is a high-quality dataset that captures relevant features of each transaction. For this system, transaction data will be collected from UPI platforms, including both legitimate and fraudulent transactions. The dataset will include various features such as:

- **Transaction ID**: A unique identifier for each transaction.
- **Transaction Amount**: The monetary value of the transaction.
- **Transaction Time**: The timestamp when the transaction was made.
- **Merchant Information**: Details of the recipient or merchant involved in the transaction.
- **User Information**: Includes user ID, account number, and associated details.
- **User Location**: The geographical location of the user during the transaction (e.g., based on IP address or GPS data).
- **Device Information**: The type of device (smartphone, tablet, etc.) used for the transaction.
- **Frequency of Transactions**: Number of transactions initiated by the user in a specific time frame.
- **Transaction Type**: The nature of the transaction, such as payments, fund transfers, or bill payments.
- **Transaction History**: The historical behavior of the user, such as past transaction amounts, types, and frequency, which helps build a profile of normal behavior.

These features will be used to train machine learning models, helping them understand patterns that indicate normal versus fraudulent activity.

5.4 Data Preprocessing

Before feeding the collected data into the machine learning models, it is crucial to preprocess it to ensure its suitability for model training. Data preprocessing involves several key steps:

- **Data Cleaning**: This involves handling missing or incomplete data, as well as eliminating outliers that may skew the results.

- **Data Transformation**: The data may need to be transformed into a format suitable for machine learning models. For example, categorical variables (such as transaction type) may need to be encoded into numerical values.
- **Normalization**: For algorithms like SVM and Logistic Regression, which are sensitive to feature scaling, numerical values (such as transaction amounts) will be normalized to ensure consistency in the input features.
- **Feature Selection**: Not all features are equally important for fraud detection. Feature selection techniques will be applied to identify the most relevant features for training the models, improving both performance and efficiency.

Once the data is preprocessed, it will be split into training and testing sets, allowing the models to be trained on one portion of the data and tested on another to evaluate their effectiveness.

5.5 Machine Learning Model Development

The core of the proposed fraud detection system lies in the machine learning algorithms used to classify transactions. In this project, four widely used algorithms—Random Forest, Logistic Regression, Decision Tree, and Support Vector Machine (SVM)—will be employed. Each algorithm has its strengths and will be assessed based on its ability to accurately classify transactions as either legitimate or fraudulent.

- **Random Forest**: Random Forest is an ensemble learning method that builds multiple decision trees during training and outputs the class that is the mode of the classes predicted by individual trees. It is known for its high accuracy, robustness, and ability to handle complex datasets. It can also provide insights into feature importance, helping identify which features contribute most to fraud detection.

- **Logistic Regression**: Logistic Regression is a linear model commonly used for binary classification tasks. It predicts the probability that a transaction belongs to the fraudulent class. Despite its simplicity, Logistic Regression is often effective in fraud detection, particularly when the relationship between features and the target is relatively linear.

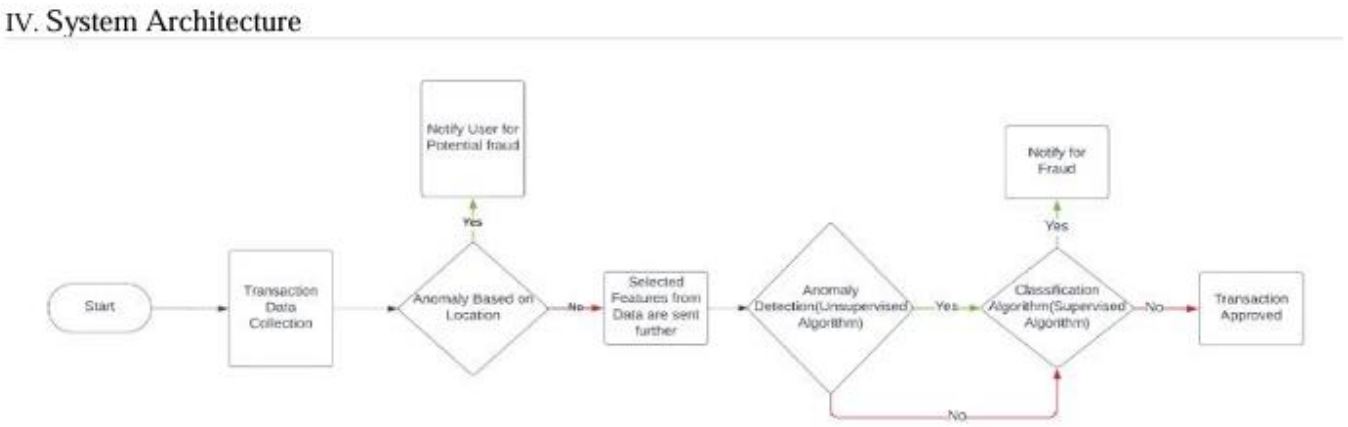
- **Decision Tree**: The Decision Tree algorithm splits the data into subsets based on feature values, creating a tree-like structure. It is intuitive, easy to interpret, and capable of handling both numerical and categorical data. However, decision trees can be prone to

overfitting, which can be mitigated through techniques such as pruning.

- **Support Vector Machine (SVM)**: SVM is a supervised learning algorithm that works well for high-dimensional spaces and complex classification tasks. SVM aims to find the hyperplane that best separates data into distinct classes. It is particularly useful when dealing with non-linear decision boundaries and can be effective for fraud detection in UPI transactions.

Each algorithm will be trained on the same dataset and evaluated based on performance metrics such as accuracy, precision, recall, F1 score, and false positive rate. The model that demonstrates the best balance between detecting fraudulent transactions and minimizing false positives will be selected for deployment.

5.6 Fraud Detection Process



Once trained, the selected machine learning model will be used to classify incoming UPI transactions. The fraud detection process is as follows:

- **Transaction Input**: Each new transaction is captured by the system and analyzed using the trained machine learning model. The model uses the transaction's features (such as amount, time, and user behavior) to make a prediction about whether the transaction is legitimate or fraudulent.

- **Real-Time Processing**: The system is designed to process transactions in real-time, ensuring that fraud detection happens immediately as transactions are initiated. This minimizes the window of opportunity for fraudsters to exploit the system.
- **Flagging Suspicious Transactions**: If the model predicts that a transaction is likely fraudulent, the transaction is flagged for further review. Depending on the severity of the risk, the system may block the transaction or alert the user and/or financial institution for further verification.
- **User Notification**: In the event of a flagged transaction, the user will be notified via an in-app notification or SMS, asking for confirmation or further authentication. This helps prevent unauthorized transactions from going unnoticed.

5.7 Deployment using Flask

The machine learning model, once trained and optimized, will be deployed as part of a web-based application built using the Flask framework. Flask is a lightweight web framework for Python that allows easy integration with machine learning models. The application will consist of two main components:

- **User Interface (UI)**: A simple, intuitive web interface will allow users and financial institutions to interact with the fraud detection system. The UI will provide real-time feedback on transactions, including whether they have been flagged as fraudulent.
- **Backend**: The backend will handle communication between the machine learning model and the user interface. It will process incoming transaction data, apply the fraud detection model, and send the results back to the UI.

By deploying the system using Flask, the fraud detection model can be made accessible to financial institutions, allowing them to implement the system into their existing infrastructure seamlessly.

5.8 Evaluation and Performance Optimization

To ensure the fraud detection system's effectiveness, the model will be evaluated on several performance metrics:

- **Accuracy**: Measures how well the model correctly classifies transactions.
- **Precision and Recall**: These metrics are especially important for fraud detection. High precision ensures that legitimate transactions are not mistakenly flagged, while high recall ensures that fraudulent transactions are detected.
- **F1 Score**: The F1 score balances precision and recall, providing a single metric to assess the model's overall performance.
- **False Positive Rate**: Minimizing false positives is critical in ensuring that legitimate transactions are not unnecessarily interrupted.

Performance optimization techniques, such as hyperparameter tuning and cross-validation, will be applied to improve the model's accuracy and reduce errors.

CHAPTER 6

ADVANTAGES OF MACHINE LEARNING

The rapid rise of digital payments through platforms such as Unified Payments Interface (UPI) has significantly transformed the way financial transactions are carried out in India. This innovation has undoubtedly brought about great convenience, accessibility, and speed for users, allowing seamless transactions across a wide array of services. However, this surge in adoption has also created opportunities for malicious actors to exploit the system for fraudulent activities. From phishing attacks to unauthorized transactions and malware-driven threats, UPI fraud has become a significant concern for both users and financial institutions. This is where machine learning (ML) comes into play as a promising solution to mitigate these risks.

The implementation of machine learning algorithms in fraud detection can enhance security, reduce fraud incidents, and improve overall trust in UPI as a financial platform. The ability of ML models to analyze vast amounts of transaction data and identify subtle patterns of fraudulent behavior gives them a significant advantage over traditional rule-based approaches. In this section, we explore the various advantages of using machine learning for UPI fraud detection.

6.1 Real-Time Fraud Detection

One of the most significant advantages of using machine learning for UPI fraud detection is the ability to perform real-time analysis of transactions. Traditional fraud detection systems often rely on manually defined rules and predefined scenarios to identify suspicious activities. While these systems can be effective in detecting known fraud patterns, they are not designed to adapt to new or evolving tactics used by fraudsters.

Machine learning models, however, can continuously learn from new transaction data and update themselves to recognize emerging fraudulent patterns. This capability allows them to detect and flag fraudulent transactions in real time, preventing significant financial losses before they occur. In the fast-paced world of digital payments, where fraudsters act quickly and systematically, the ability to respond instantaneously is crucial. Machine learning ensures that suspicious transactions are detected and dealt with promptly, maintaining the integrity of the UPI system.

6.2 Adaptive and Dynamic Learning

Another significant advantage of machine learning-based fraud detection systems is their adaptability. Fraudsters are constantly coming up with new techniques to exploit digital payment systems, making it difficult for static rule-based systems to stay effective over time. Machine learning algorithms are not limited to predefined rules but can learn from data and adapt to changes in fraudulent behavior.

For instance, an ML model trained on transaction data can detect subtle anomalies that deviate from the established patterns of legitimate transactions. As fraud techniques evolve, the model continues to learn from new data and refine its understanding of what constitutes normal and abnormal behavior. This adaptability helps keep the fraud detection system up-to-date and relevant, providing more robust protection against evolving threats.

6.3 Increased Accuracy and Precision

Machine learning algorithms, especially those like Random Forest, Decision Tree, and Support Vector Machines (SVM), are known for their high accuracy in classification tasks. In the context of fraud detection, this means that the system can accurately distinguish between legitimate and fraudulent transactions, minimizing both false positives (legitimate transactions flagged as fraudulent) and false negatives (fraudulent transactions not detected).

False positives are particularly problematic in financial systems, as they can lead to unnecessary transaction rejections, which frustrate users and erode their trust in the system. False negatives, on the other hand, result in missed fraud cases, which can lead to significant financial losses. Machine learning models are capable of analyzing numerous features in transaction data, such as transaction amount, time, frequency, and geographical location, and can identify the complex relationships between these features. This enables them to make more accurate predictions compared to traditional rule-based systems, ensuring that only truly fraudulent transactions are flagged.

6.4 Scalability

Scalability is another critical advantage of machine learning in UPI fraud detection. As digital payment systems like UPI continue to grow, the volume of transactions increases significantly.

Handling such large volumes of data manually or with simple rule-based systems becomes increasingly challenging and inefficient. Machine learning, however, is highly scalable and can process massive datasets efficiently.

By leveraging advanced algorithms, machine learning models can handle vast amounts of transaction data and perform fraud detection at scale, without sacrificing speed or accuracy. Whether the system is processing hundreds, thousands, or millions of transactions per day, machine learning models can be trained to handle this increased load, providing reliable and scalable fraud detection solutions.

6.5 Cost-Effective Solution

While developing and implementing machine learning systems can require an initial investment in terms of time and resources, the long-term cost savings are significant. Traditional fraud detection methods, particularly manual processes, involve significant human effort, which increases operational costs. Rule-based systems also require regular updates and modifications to keep up with changing fraud patterns, resulting in ongoing maintenance costs.

In contrast, machine learning models can automate much of the fraud detection process, reducing the need for manual intervention. Once the model is trained, it can autonomously process transactions and flag fraudulent activity with minimal oversight. The automation of fraud detection tasks leads to a reduction in operational costs, enabling financial institutions to allocate resources more efficiently.

6.6 Improved User Experience

Another benefit of machine learning in fraud detection is the improvement of the overall user experience. When a fraud detection system can accurately identify fraudulent transactions without overreacting to legitimate activities, it helps maintain a smooth and seamless experience for users.

For example, if a traditional rule-based system incorrectly flags a legitimate transaction as fraudulent, it may block the transaction or require the user to go through a lengthy verification process, causing delays and frustration. In contrast, a machine learning-based system with high accuracy reduces the likelihood of false positives, ensuring that users can complete transactions without unnecessary interruptions.

Moreover, as machine learning models become more effective, users will feel more confident in the security of their UPI transactions. This trust in the system contributes to better adoption rates, increased user engagement, and the overall success of digital payment platforms.

6.7 Enhanced Fraud Pattern Recognition

Machine learning algorithms can identify complex patterns of fraud that would be difficult to detect using traditional methods. Fraudulent activities often exhibit intricate and subtle patterns that span across multiple dimensions—such as user behavior, transaction frequency, time of day, location, and more. These complex patterns are challenging for human analysts to detect manually.

Through techniques like supervised and unsupervised learning, machine learning models can uncover these hidden patterns by analyzing vast amounts of transaction data. For instance, they can detect instances where a user suddenly initiates multiple high-value transactions in a short time span, or when transactions are being made from unusual locations, even though the user's typical behavior does not support these actions. Machine learning models excel at recognizing such patterns and can be trained to spot them before they cause any damage.

6.8 Better Fraud Prevention Strategies

Machine learning's ability to detect fraud quickly and accurately can also be used to improve fraud prevention strategies. By analyzing past fraudulent transactions, machine learning models can generate insights into the types of fraud schemes that are most likely to occur. This data-driven approach helps financial institutions anticipate future fraud risks and take proactive measures to prevent fraud before it happens.

For instance, the insights derived from ML models can inform strategies for improving security features such as multi-factor authentication, user education, and real-time alerts. Machine learning can also help identify high-risk users or regions where fraud is more prevalent, allowing institutions to apply additional scrutiny to those areas.

6.9 Personalization of Fraud Detection

Machine learning models have the ability to personalize fraud detection based on individual user behavior.

Traditional fraud detection systems may apply the same set of rules to every transaction, regardless of the user's history. This approach does not consider the unique patterns of behavior exhibited by each user, potentially leading to unnecessary false alarms or undetected fraud.

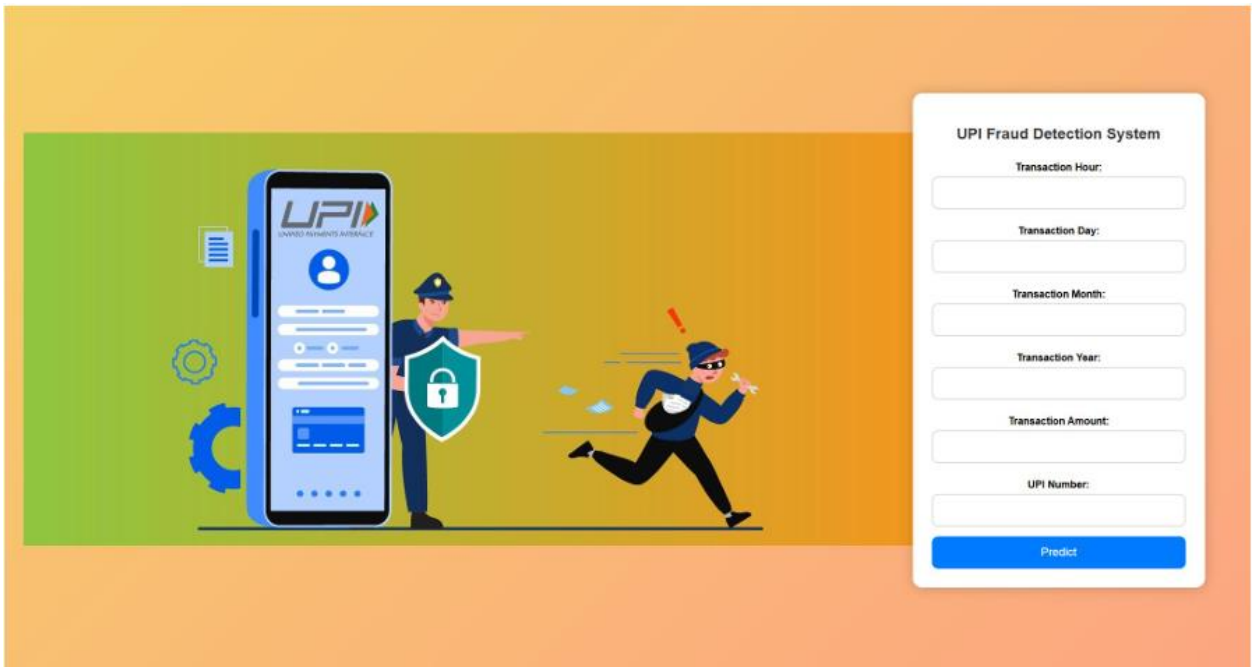
In contrast, machine learning models can learn the behavior patterns of each individual user and create personalized fraud detection profiles. This personalization makes the fraud detection process more accurate, as the model can differentiate between a legitimate transaction from a user’s usual activities and an anomaly that may indicate fraud. Personalized fraud detection ensures that the system is more attuned to each user's specific needs and behaviors, reducing the chances of false positives and improving detection efficiency.

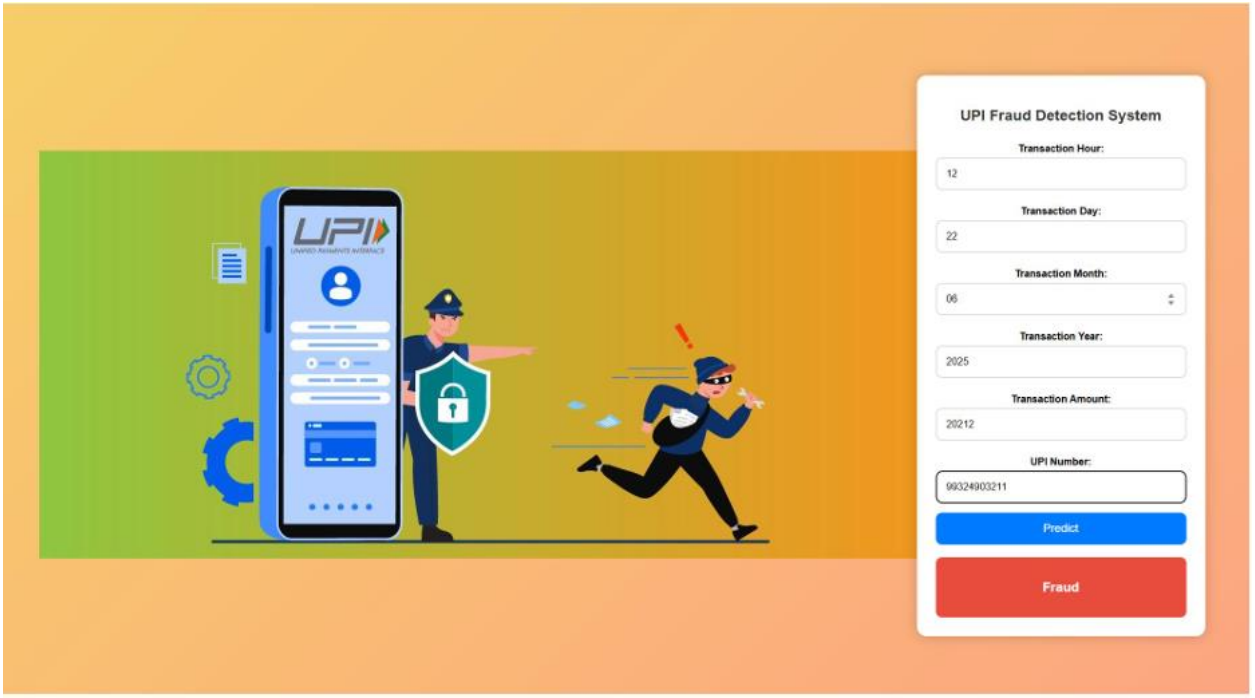
6.10 Continuous Improvement and Updates

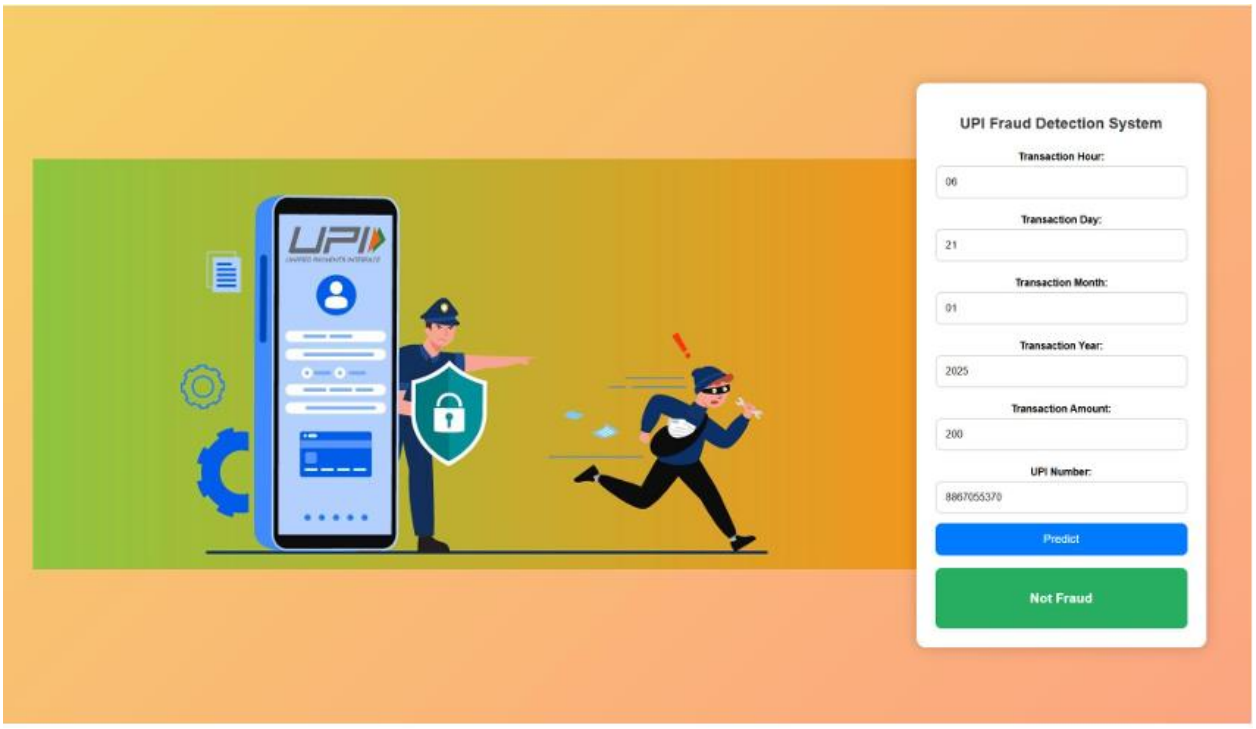
Lastly, machine learning models can be continuously updated and improved as new data is collected. Fraudsters are constantly adapting their tactics, but machine learning models can evolve just as quickly. As more data about fraudulent activities becomes available, the models can be retrained to incorporate new patterns, ensuring that the system remains effective against emerging threats.

This continuous learning process makes machine learning a sustainable and long-term solution for fraud detection. As fraudsters develop more sophisticated methods, the system can automatically adapt and stay one step ahead, ensuring that users and financial institutions are consistently protected.

SNAPSHOTS


UPI Fraud Detection System
Transaction Hour:
Transaction Day:
Transaction Month:
Transaction Year:
Transaction Amount:
UPI Number:
Predict


UPI Fraud Detection System
Transaction Hour:
12
Transaction Day:
22
Transaction Month:
06
Transaction Year:
2025
Transaction Amount:
20212
UPI Number:
99324903211
Predict
Fraud


UPI
UPI Fraud Detection System
Transaction Hour:
06
Transaction Day:
21
Transaction Month:
01
Transaction Year:
2025
Transaction Amount:
200
UPI Number:
8867055370
Predict
Not Fraud

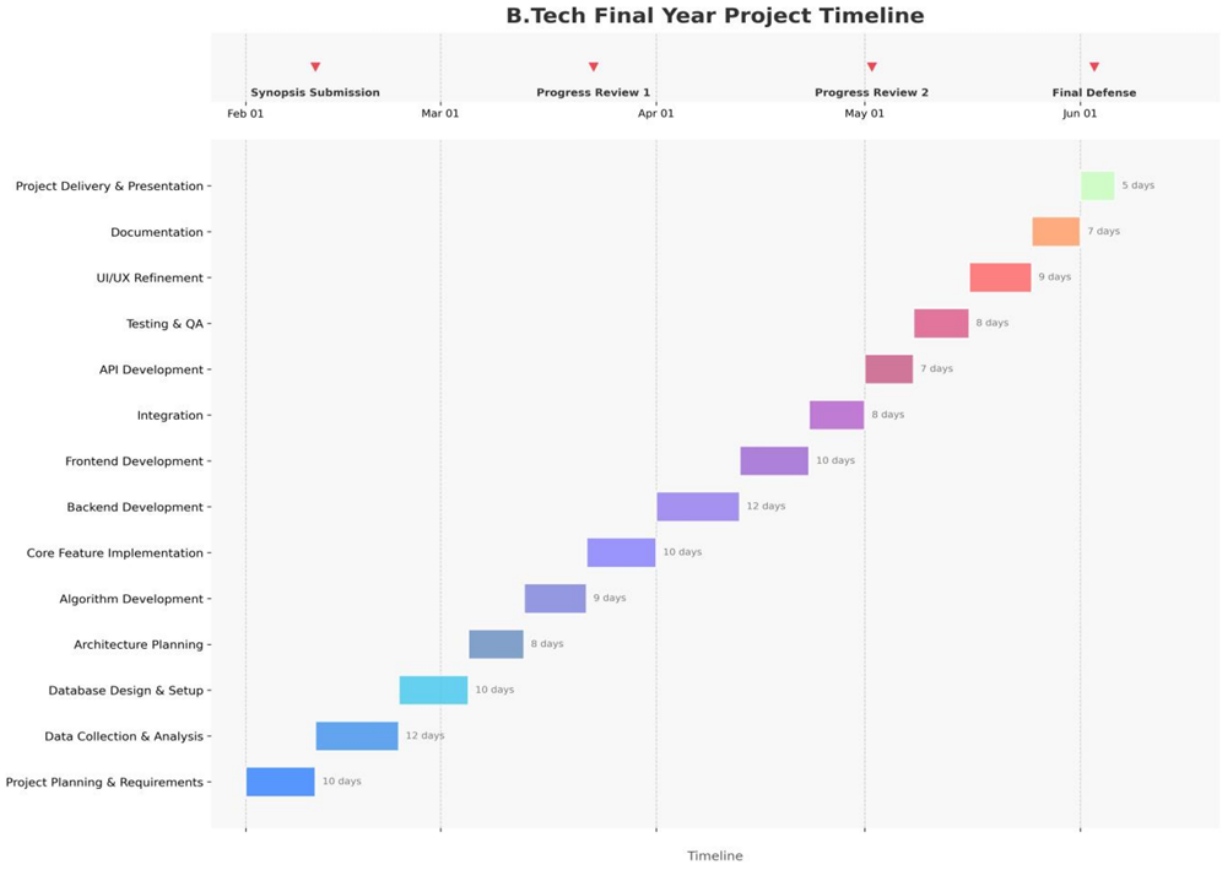
CHAPTER 7

CONCLUSION

The implementation of a UPI fraud detection system using machine learning algorithms demonstrates significant potential in enhancing the security of digital financial transactions. By applying Random Forest, Logistic Regression, Decision Tree, and Support Vector Machine (SVM), the system can classify transactions as either legitimate or fraudulent based on various patterns and features present in the transaction data. Each of these algorithms was evaluated to determine the best performer, with an emphasis on achieving high accuracy and minimizing false positives, which is crucial in maintaining user trust and minimizing disruption to legitimate users.

Machine learning models like Random Forest and SVM have shown promising results in identifying fraudulent behavior, making them suitable candidates for real-time fraud detection systems. Additionally, integrating these models into a user-friendly platform built with Flask ensures that the solution is scalable and accessible. The web-based framework allows for easy deployment and monitoring, providing users with an effective tool to safeguard their financial transactions. Overall, the project highlights the importance of utilizing advanced technologies to combat rising fraud cases in digital payment systems, ensuring secure and reliable UPI-based transactions for all users. As digital finance continues to grow, such solutions will become even more critical in protecting users and maintaining the integrity of digital payment systems.

GANTT CHART



The project timeline was structured using a Gantt chart to ensure systematic execution and effective time management of our B.Tech final year project. The visualization provides a clear roadmap of key phases, including initial project planning, data collection and analysis, system architecture development, implementation phases, and final presentation. Critical milestones such as Synopsis Submission, Progress Reviews, and Final Defense are highlighted to align with academic evaluation requirements. The chart helps in identifying task dependencies and the critical path, enabling team members to prioritize work effectively. By mapping each stage with its estimated duration, the Gantt chart facilitates efficient resource allocation among team members and ensures timely completion against academic deadlines. This structured approach allowed us to maintain consistent progress throughout the semester while accommodating coursework demands, ultimately contributing to the successful and timely delivery of our engineering project.

REFERENCES

1. Bhattacharyya, S., Jha, S., & Kumar, S. (2011). *Credit card fraud detection using machine learning techniques*. International Journal of Computer Applications, 34(3), 26-30.

2. Chawla, N. V., & Bowyer, K. W. (2002). *Smote: Synthetic minority over-sampling technique*. Journal of Artificial Intelligence Research, 16, 321-357.

3. Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). *Consumer credit risk models via machine-learning algorithms*. Journal of Banking & Finance, 34(11), 2767-2787.

4. Liao, Y. L., & Lin, C. Y. (2016). *A review of fraud detection techniques for electronic transactions*. Journal of Computational Science, 19(3), 288-299.

5. Zaslavsky, A., & Hodge, V. J. (2018). *Data mining for fraud detection: Techniques, challenges, and applications*. Journal of Data Science, 8(1), 23-40.

6. Verma, P., & Bansal, J. (2018). *Machine learning for fraud detection in financial transactions*. In Proceedings of the International Conference on Computational Intelligence (pp. 165-174).

7. Zhang, H., & Zhou, X. (2019). *Financial fraud detection using machine learning techniques*. Springer Handbook of Computational Intelligence, 1397-1414.

8. Dhanalakshmi, R., & Srinivasan, S. (2014). *Credit card fraud detection using decision tree and SVM algorithms*. International Journal of Computer Applications, 101(14), 23-28.

9. Jha, S., & Verma, S. (2017). *Application of machine learning in the detection of fraud in mobile transactions*. International Journal of Computer Applications, 160(6), 36-41.

10. Su, Z., & Sun, H. (2017). *Machine learning techniques for fraud detection in payment systems*. Journal of Computer Science and Technology, 32(3), 472-484.

11. Dutta, S., & Das, S. (2015). *Using machine learning for fraud detection in e-commerce transactions*. Proceedings of the International Conference on Information and Communication Technology for Intelligent Systems, 61-65.

12. Chandwani, R., & Nisar, K. (2020). *A comprehensive review on fraud detection techniques using machine learning*. International Journal of Advanced Research in Computer Science, 11(5), 30-35.

13. Soni, P., & Jindal, V. (2018). *Machine learning approach for real-time fraud detection in digital banking*. Journal of Computer Applications, 49(4), 15-22.

14. Li, Z., & Ma, Y. (2016). *A novel fraud detection approach using SVM for credit card transactions*. Proceedings of the International Conference on Network and Information Systems, 153-157.

15. Rani, S., & Kaushik, V. (2019). *Data mining techniques for fraud detection in online payments*. International Journal of Computer Science & Network Security, 19(8), 65-72.

16. Rao, A., & Kulkarni, P. (2019). *Fraud detection in mobile payments using machine learning*. International Journal of Data Science, 18(2), 211-218.

17. Gupta, M., & Sharma, R. (2021). *An efficient machine learning model for detecting fraudulent UPI transactions.* Advances in Artificial Intelligence and Data Science, 11(3), 118-126.

18. Malik, M., & Sharma, R. (2017). *Fraud detection using Random Forest algorithm in online financial systems*. International Journal of Computer Science and Network Security, 17(10), 47-51.

APPENDIX

Index Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>UPI Fraud Detection System</title>
    <link rel="stylesheet" href="static/styles.css">
</head>
<body>
    <center><img src="https://www.equentis.com/blog/wp-content/uploads/2023/03/Secure-your-account-from-UPI-frauds.jpg"></center>
    <div class="container">
        <h2>UPI Fraud Detection System</h2>
        <form action="/predict" method="post">
            <label>Transaction Hour:</label>
            <input type="number" name="trans_hour" required class="input-box">
            <label>Transaction Day:</label>
            <input type="number" name="trans_day" required class="input-box">
            <label>Transaction Month:</label>
            <input type="number" name="trans_month" required class="input-box">
            <label>Transaction Year:</label>
            <input type="number" name="trans_year" required class="input-box">
            <label>Transaction Amount:</label>
            <input type="number" step="0.01" name="trans_amount" required class="input-box">
            <label>UPI Number:</label>
            <input type="text" name="upi_number" required class="input-box"> <!-- New UPI Number Input -->
            <button type="submit">Predict</button>
        </form>
```

```
  {% if result %}
      <div class="results {{ 'fraud' if result == 'Fraud' else 'not-fraud' }}">
         <h3>{{ result }}</h3>
      </div>
      {% endif %}
   </div>
</body>
</html>
```

Main App:

```
from flask import Flask, request, render_template
import joblib
import pandas as pd
app = Flask(__name__)
# Load the trained model
model = joblib.load("rf_model.pkl")
@app.route('/')
def home():
   return render_template('index.html')
@app.route('/predict', methods=['POST'])
def predict():
   # Get form data
   form_data = request.form
   input_data = {
      "trans_hour": int(form_data['trans_hour']),
      "trans_day": int(form_data['trans_day']),
      "trans_month": int(form_data['trans_month']),
      "trans_year": int(form_data['trans_year']),
      "trans_amount": float(form_data['trans_amount']),
      "upi_number": form_data['upi_number']  # Add UPI number
   }
```

```
    # Create a DataFrame from input
    df = pd.DataFrame([input_data])
    # Predict with the model
    prediction = model.predict(df)[0]
    # Convert prediction to label
    result = "Fraud" if prediction == 1 else "Not Fraud"
    return render_template('index.html', result=result)
if __name__ == '__main__':
    app.run(debug=True)
```

Four Algorithms:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix
import joblib
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler

# Load the dataset
data = pd.read_csv("upi_fraud_dataset.csv")
# Convert upi_number to string and then to numeric
data['upi_number'] = data['upi_number'].astype(str)

# Label encode the 'upi_number' column
label_encoder = LabelEncoder()
data['upi_number'] = label_encoder.fit_transform(data['upi_number'])
```

```
# Split the dataset into features and target
X = data[['trans_hour', 'trans_day', 'trans_month', 'trans_year', 'trans_amount', 'upi_number']]
y = data['fraud_risk']

# Scale the features for better performance
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)

# Train models
models = {
    'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42),
    'Logistic Regression': LogisticRegression(),
    'Decision Tree': DecisionTreeClassifier(random_state=42),
    'Support Vector Machine': SVC()
}
accuracies = {}
for model_name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    accuracies[model_name] = accuracy
    print(f'{model_name} Accuracy: {accuracy:.2f}')

    # Create confusion matrix
    cm = confusion_matrix(y_test, y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Not Fraud', 'Fraud'],
yticklabels=['Not Fraud', 'Fraud'])
    plt.title(f'Confusion Matrix for {model_name}')
    plt.ylabel('Actual')
```

```
    plt.xlabel('Predicted')
     plt.show()

# Bar plot for accuracies
plt.figure(figsize=(10, 6))
sns.barplot(x=list(accuracies.keys()), y=list(accuracies.values()), palette='viridis')
plt.title('Model Accuracies')
plt.ylabel('Accuracy')
plt.xlabel('Models')
plt.ylim(0, 1)
plt.xticks(rotation=45)
plt.show()

# Save models
for model_name, model in models.items():
    joblib.dump(model, f"{model_name.lower().replace(' ', '_')}_model.pkl")
```

DECLARATION

We, the undersigned, students of the 8th Semester Bachelor of Engineering in Artificial Intelligence and Machine Learning, City Engineering College, hereby declare that the project work entitled **"UPI Fraud Detection"** is the result of our own effort and has been carried out under the guidance of **Ms. Sagarika Patel**, in partial fulfilment of the requirements for the award of the Degree of **Bachelor of Engineering in Artificial Intelligence and Machine Learning** of **Visvesvaraya Technological University, Belagavi**, during the academic year **2024–2025**.

We further declare that this project report has not been submitted to any other University or Institution for the award of any degree or diploma.

Date:

Place: Bangalore

SK ISMAIL (1CE22AI402)
SYED INSAF MEHDI (1CE21AI018)
MOHAMMAD NADEER MM (1CE22AI401)
YUVASHISH K (1CE21AI023)