A Project Based Learning-II Report

on

“Transaction Fraud Detection”

Submitted to the

Savitribai Phule Pune University

In partial fulfillment of

“Artificial Intelligence and Data Science ”

By

Group No.

1. Niharika Bafana [S1903102009]

2. Shruti Bandewar[S1903102011]

3. Sneha Bhalewar[S1903102015]

4. Pragya [S1903102123]

5. Sanika Dhide[S1903102128]

Under the guidance of

Mr. Vishal Bogam

Department Of Artificial Intelligence and Data Science

PES’s Modern College of Engineering

Shivaji nagar, Pune-411005, Maharashtra, India

2024-2025

CERTIFICATE

This is to certify that the project based learning-II report entitled **"Transaction Fraud Detection"** being submitted by **Niharika Bafana (S1903102009) , Shruti Bandewar (S1903102011) , Sneha Bhalewar (S1903102015) , Pragya (S1903102123) , Sanika Dhide (S1903102128)** is a record of bonafide work carried out by us under the supervision and guidance of **Mr. Vishal Bogam** in partial fulfillment of the requirement for **SE (**Artificial Intelligence and Data Science **) – 2020 course** of Savitribai Phule Pune University, Pune in the academic year 2024-2025

Date: / / 2025

Place: Pune

Mr. Vishal Bogam
Guide

Prof. (Dr.) Mrs. Shraddha Pandit
Head of the Department

ACKNOWLEDGEMENT

We would like to take this opportunity to express my gratitude towards all those who helped me in accomplishing this PBL-II work. First of all, we would like to thank P. E. S.'s Modern college of Engineering for giving me/us this opportunity to look at some concept apart from my curriculum subject. We would like to thank our Head of the Department Prof. (Dr.) Mrs. Shraddha Pandit for providing us all the necessary requirements. We would like to thank my guide Mr. Vishal Bogam for his valuable comments and timely support. We would like to show my greatest appreciation to her.

We would also like to thank all other faculty members for supporting me directly or indirectly. The guidance and support received from all the Professors who contributed are vital for the success of the work. We are grateful for their constant support and help.

Lastly, we would like to thank my friends for their support and encouragement.

(Students Name & Signature)

1 **Niharika Bafana**

2 **Shruti Bandewar**

3 **Sneha Bhalewar**

4 **Pragya**

5 **Sanika Dhide**

Report Index

List of figures.

List of Tables

Abstract

Keyword : Secure Transactions

SCAMGUARD is a simple yet powerful tool designed to help people and businesses detect fraudulent transactions before they happen. By analyzing details like IP addresses, transaction amounts, and previous dealings between users, it quickly determines whether a transaction is safe or suspicious.

The system works in two ways. First, it applies basic fraud detection rules—flagging transactions that involve unusually high amounts, suspicious IP addresses, or repeated payments to the same recipient. But fraud can be tricky, so SCAMGUARD goes a step further. It connects to a machine learning model via a Flask-based API to analyze patterns and improve detection accuracy.

With an easy-to-use interface and instant results, SCAMGUARD makes fraud detection accessible to everyone, not just cybersecurity experts. Whether you're an individual making an online payment or a business handling multiple transactions, this tool helps you stay one step ahead of scammers. In a world where digital fraud is on the rise, SCAMGUARD provides peace of mind—so you can transact with confidence.

P.E.S. Modern College of Engineering, Pune -05

Department of Artificial Intelligence and Data Science

(Academic Year: 2024-25)

Project Title: Transaction Fraud Detection			
Project Group No. : 02		**Guide Name: Mr. Vishal Bogam**	
GROUP MEMBERS:			
Roll No. / Seat No.	**Name of Student**	**Project Area**	**Project Platform**
S1903102009	Niharika Bafana	Machine Learning	Cloud-Based
S1903102011	Shruti Bandewar		
S1903102015	Sneha Bhalewar		
S1903102123	Pragya		
S1903102128	Sanika Dhide		

Abstract

Online fraud is on the rise, and many people don't have the right tools to detect suspicious transactions before it's too late. Existing fraud detection systems are often complicated, expensive, or just not accurate enough. SCAMGUARD aims to change that by providing a simple, smart, and effective way to check if a transaction is safe.

SCAMGUARD analyzes transactions based on:

1. **IP Address Checks** – Spotting suspicious sender or receiver locations.
2. **Transaction Amounts** – Flagging unusually large payments.
3. **Past Transaction History** – Identifying risky patterns.

The system uses a two-step fraud detection process:

- A rule-based system to instantly flag obvious red flags.
- A machine learning model that recognizes more complex fraud patterns.

Built with Flask on the backend, SCAMGUARD connects to an AI-powered fraud detection model, delivering real-time results through a user-friendly web interface. Whether you're an individual or a small business, this tool helps you make safer financial decisions without needing technical expertise.

By blending technology with ease of use, SCAMGUARD makes fraud detection accessible, reliable, and affordable—helping you stay one step ahead of scammers.

CHAPTER 1
INTRODUCTION TO PROJECT TOPIC

1.1 Introduction to Project

We live in a world where online transactions are second nature. Whether you're shopping, paying bills, or sending money to a friend, digital payments make life easier. But with convenience comes risk—scammers and fraudsters are always looking for ways to exploit people and businesses. That's where **SCAMGUARD** steps in.

SCAMGUARD is a simple yet powerful tool designed to help you detect fraudulent transactions before it's too late. It analyzes key details like IP addresses, transaction amounts, and previous transactions to identify any red flags. The system works in two ways: first, it applies basic rules to spot obvious signs of fraud—such as unusually large amounts, suspicious IP addresses, or repeated payments to the same recipient. But not all fraud is easy to catch, so SCAMGUARD takes things a step further by using a machine learning model. Through a Flask-based API, it leverages advanced fraud detection algorithms to recognize patterns that traditional methods might miss.

Unlike complex fraud detection systems that require technical knowledge or expensive subscriptions, SCAMGUARD is designed to be simple and accessible for everyone. Whether you're an individual making an online payment or a business processing multiple transactions, this tool gives you quick, reliable insights to help you stay safe.

In a time when digital fraud is becoming more sophisticated, staying one step ahead is crucial. SCAMGUARD gives you the confidence to make secure transactions, protecting you from scams, financial loss, and cyber threats. With its easy-to-use interface and smart fraud detection capabilities, SCAMGUARD makes online security effortless—so you can focus on what really matters.

1.2 Problem Definition

Online transactions are more common than ever, but so are scams and fraud. Many people and businesses fall victim to fraudulent transactions because existing fraud detection tools are either too complex, expensive, or unreliable. Traditional methods like manual checks or basic rule-based systems often miss sophisticated scams, putting users at risk of financial loss.

SCAMGUARD solves this problem by providing a simple, smart, and accessible fraud detection system. It analyzes key transaction details—like IP addresses and past transaction patterns—using both basic fraud rules and a machine learning model. This helps users quickly identify suspicious activity and make safer financial decisions.

By making fraud detection easy and reliable, SCAMGUARD empowers users to stay one step ahead of scammers and protect their money.

1.3 Motivation behind project topic

This project likely started because fraud is a real, growing problem in online payments. Every day, millions of digital transactions happen—shopping, banking, sending money—but not all of them are honest. Fraudsters are constantly finding new ways to cheat the system, and businesses lose billions each year because of it. Worse, real customers sometimes get blocked by mistake, causing frustration and loss of trust.

Need of the Project :

Fraud is expensive – Every year, businesses and customers lose billions to fraud. Stopping fraud early saves money.
Old fraud systems aren't enough – Scammers evolve fast, and outdated methods can't keep up.
False positives frustrate customers – Getting blocked from your own money is annoying. A better system means a better experience.
Speed matters – Fraud happens in seconds. Real-time detection is key.
Trust is everything – If people don't feel safe, they won't use digital payments.

This project isn't just about catching fraud—it's about making online payments safer and smoother for everyone. It's about trust, security, and staying one step ahead of criminals.

1.4 Objective(s) of the work/ Benefits of proposed system

Objective(s) of the work

The objective of the **SCAMGUARD** project is to develop a reliable, intelligent, and user-friendly fraud detection system that helps individuals and businesses identify suspicious transactions before they result in financial losses. The project aims to enhance security, improve fraud detection accuracy, and provide real-time risk assessments. The following objectives outline the key steps to achieving this goal:

1. To develop an intuitive and accessible fraud detection system that can be used by both individuals and businesses without requiring technical expertise.

2. To implement a rule-based fraud detection mechanism that flags transactions based on predefined risk factors such as high transaction amounts, suspicious IP addresses, and repeated transactions to the same recipient.

3. To integrate a machine learning model that enhances fraud detection by identifying complex fraud patterns beyond rule-based checks.

4. To build a Flask-based API that connects the web application to the trained machine learning model, allowing real-time fraud analysis.

5. To ensure real-time processing and instant feedback so users can quickly determine if a transaction is safe or suspicious before completing it.

6. To provide a simple and responsive user interface that works seamlessly across different devices, making fraud detection accessible anytime and anywhere.

7. To enhance fraud detection accuracy by continuously improving the machine learning model with new fraud patterns and transaction data.

8. To minimize false positives and false negatives by balancing security and usability, ensuring that genuine transactions are not unnecessarily blocked while fraudulent ones are accurately detected.

9. To create a cost-effective fraud detection solution that can be used by individuals and small businesses without the need for expensive security tools.

Benefits of the Proposed System

- Increased Security: Helps users detect fraudulent transactions before they occur, reducing financial losses.

- User-Friendly Experience: Simple interface that anyone can use, regardless of technical knowledge.

- Real-Time Detection: Instant fraud analysis enables quick decision-making.

- Advanced Machine Learning Integration: Improves detection accuracy and adapts to evolving fraud techniques.

- Cost-Effective: Provides fraud prevention without expensive security software or professional services.

- Scalability: Can be expanded to support additional fraud detection features in the future.

CHAPTER 2
LITERATURE SURVEY

1. Introduction

With the rise of digital transactions, fraud has become a serious issue for businesses and individuals alike. Every day, scammers find new ways to trick systems, making traditional fraud detection methods less effective. To tackle this, researchers have explored different ways to detect fraud, from simple rule-based systems to advanced machine learning models.

This section explores existing fraud detection techniques, their strengths and weaknesses, and how our project, SCAMGUARD, offers a unique approach to identifying fraudulent transactions.

2. Existing Research on Fraud Detection

Over the years, different approaches have been used to detect fraud in financial transactions. These range from simple, manually defined rules to complex AI-driven models.

2.1 Rule-Based Systems – The Old School Approach

Early fraud detection relied on fixed rules. For example, a transaction above $10,000 might be flagged as suspicious, or multiple transactions from different locations in a short time could trigger an alert.

- **Advantages:**
    - Simple and easy to implement.
    - Can detect common fraud patterns.
- **Disadvantages:**
    - Fraudsters quickly learn how to bypass these rules.
    - High false positives—many legitimate transactions get flagged as fraud.

2.2 Machine Learning – Smarter Fraud Detection

With more complex fraud tactics, machine learning (ML) became a game-changer. Instead of relying on fixed rules, ML models learn from past transactions and detect patterns that indicate fraud.

2.2.1 Supervised Learning – Training Models with Past Data

Supervised learning works by training a model on labeled data—where past transactions are already classified as fraudulent or safe.

- **Random Forest (Used in SCAMGUARD)**

- Our project uses this method to make fraud predictions.
- It works by combining multiple decision trees, each making its own prediction, and then taking a majority vote.
- Advantages: Highly accurate, reduces overfitting.
- Disadvantages: Can be slow when handling large datasets.

- **Logistic Regression**
  - A simple mathematical model used for classification.
  - Advantages: Easy to understand.
  - Disadvantages: Not great at detecting complex fraud patterns.

- **Neural Networks (Deep Learning)**
  - Modeled after the human brain, neural networks can identify subtle fraud patterns.
  - Advantages: Powerful for big data and evolving fraud patterns.
  - Disadvantages: Requires a lot of computing power and data.

2.2.2 Unsupervised Learning – Spotting the Unknown

Sometimes, fraudsters come up with new tricks that aren't in the training data. Unsupervised learning helps by detecting anomalies—transactions that don't fit usual patterns.

- **Autoencoders (Deep Learning-based)**
  - Tries to recreate normal transaction behavior and flags anything unusual.
  - Advantages: Good at catching new fraud types.
  - Disadvantages: Hard to interpret why a transaction was flagged.

- **Isolation Forests**
  - An algorithm that isolates suspicious transactions by treating them as "outliers."
  - Advantages: Works well for rare fraud cases.
  - Disadvantages: May struggle with complex fraud schemes.

2.3 Hybrid Approaches – The Best of Both Worlds

Some systems use a mix of rule-based and machine learning techniques to improve accuracy.

For example:

- PayPal and Stripe first apply basic rules (e.g., flagging unusually large payments).
- Then, machine learning models refine the fraud prediction.
- Advantages: More accurate, can adapt to changing fraud trends.
- Disadvantages: More complex and expensive to maintain.

3. Existing Fraud Detection Technologies

Many fraud detection tools are available today. Here's how they compare:

Technology	Description	Advantages	Disadvantages
Stripe Radar	AI-powered fraud detection for online payments.	Easy to integrate, self-learning models.	Only for Stripe users.
PayPal Fraud	Uses deep learning and network analysis.	Good for large transactions.	Limited to PayPal transactions.
IBM Safer Payments	AI-based fraud prevention for banks.	Scalable for large institutions.	Expensive.
SCAMGUARD (Our Project)	Uses a Random Forest ML model via Flask API.	Lightweight, real-time fraud detection.	Can be improved with deep learning.

4. How SCAMGUARD is Different

Our project, SCAMGUARD, provides a real-time fraud detection system using machine learning and a user-friendly web interface.

- **Backend (Flask API - `app.py`)**
    - Takes in transaction details (amount, number of transactions, type).
    - Uses a Random Forest classifier to determine if the transaction is fraudulent.
    - Returns results instantly (Fraudulent or Safe).
- **Frontend (SCAMGUARD Web App - `fraud.html`)**
    - Allows users to input transaction details.
    - Calls the API to check for fraud.
    - Displays results visually with alerts (red for fraud, green for safe).

Why SCAMGUARD Stands Out

1. Customizable – Unlike Stripe or PayPal, it's open-source and can be improved over time.
2. Lightweight & Fast – Designed for real-time fraud detection without heavy computing requirements.
3. Can Be Enhanced – Future improvements can include deep learning, geolocation tracking, and graph analysis.

5. Summary

Fraud detection has come a long way, from simple rules to AI-driven models. While existing systems like Stripe and PayPal offer advanced solutions, they are often closed and expensive. SCAMGUARD provides an open-source, machine learning-powered alternative that can detect fraud in real-time. With future enhancements, it has the potential to become even more effective.

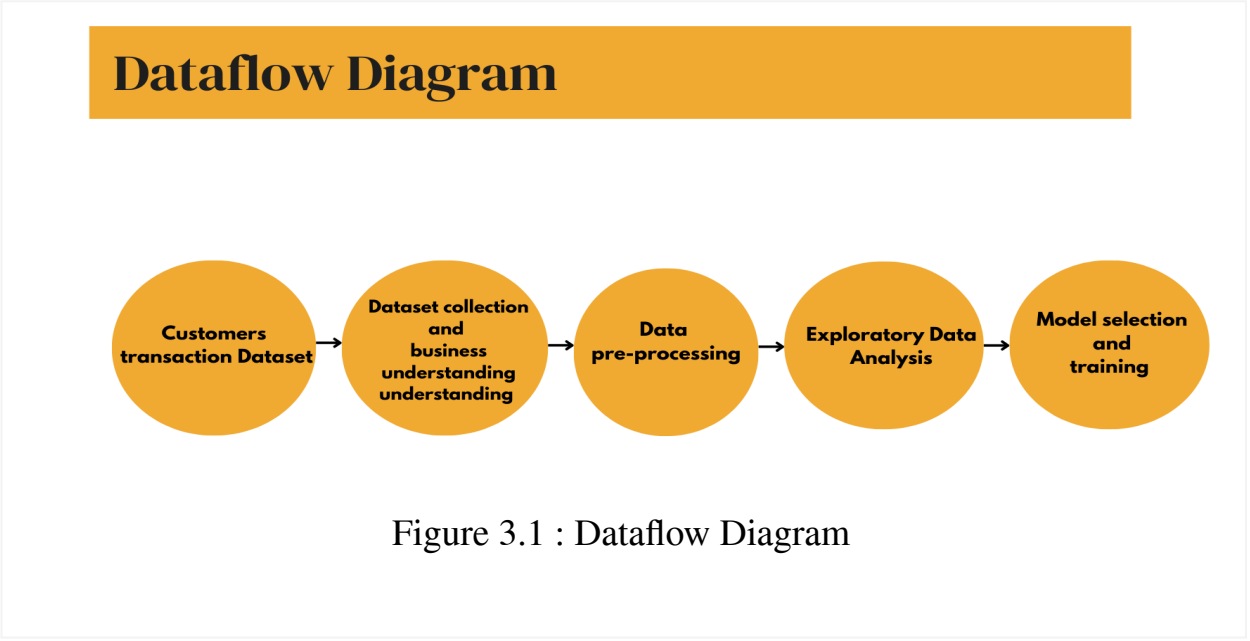


Figure 3.1 : Dataflow Diagram

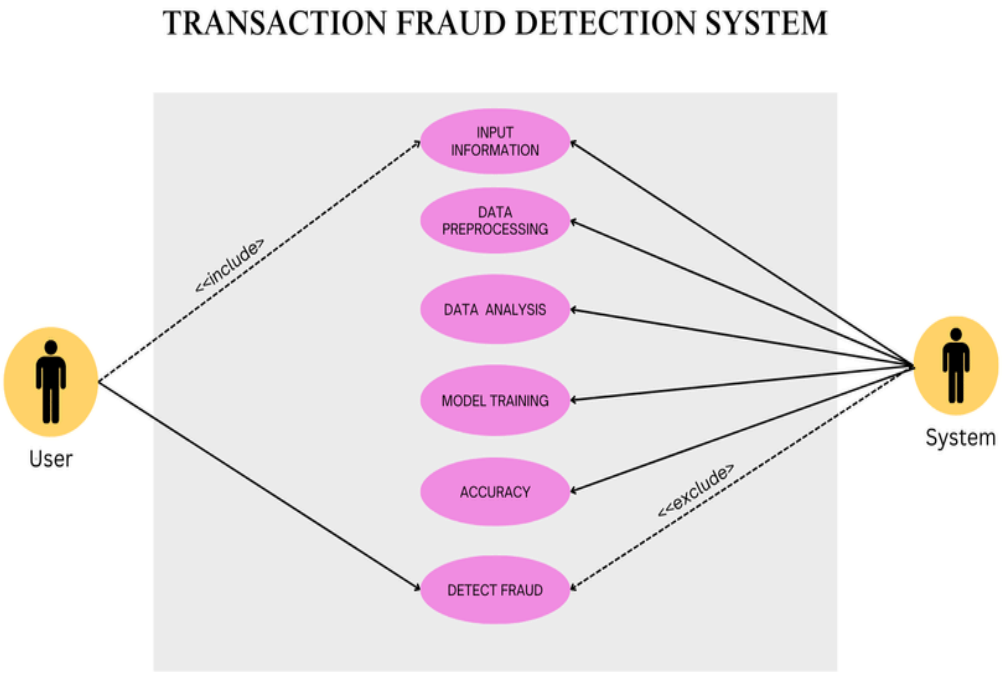


Figure 3.2 : Use Case Diagram

CHAPTER 4
TECHNICAL SPECIFICATIONS

Technical Specifications

To develop and deploy SCAMGUARD, a combination of software and hardware components is required. This section outlines the key technical specifications of the project.

1. **Software Specifications**

    a. Frontend:

        i. HTML, CSS, JavaScript – For designing a user-friendly and responsive interface. Bootstrap – To enhance UI design and responsiveness.

    b. Backend:

        i. Flask (Python) – To handle API requests and connect the frontend with the fraud detection model.

        ii. Pandas & NumPy – For data processing and feature extraction.

        iii. Scikit-Learn (Machine Learning) – To build and train the fraud detection model using a Random Forest classifier.

        iv. Joblib – For saving and loading the trained model efficiently.

    c. Database:

        i. SQLite / PostgreSQL / MySQL – To store transaction history and improve fraud detection accuracy over time.

    d. Machine Learning Model:

        i. Algorithm: Random Forest Classifier

        ii. Features Used:

            1. Transaction Amount

            2. Number of Past Transactions

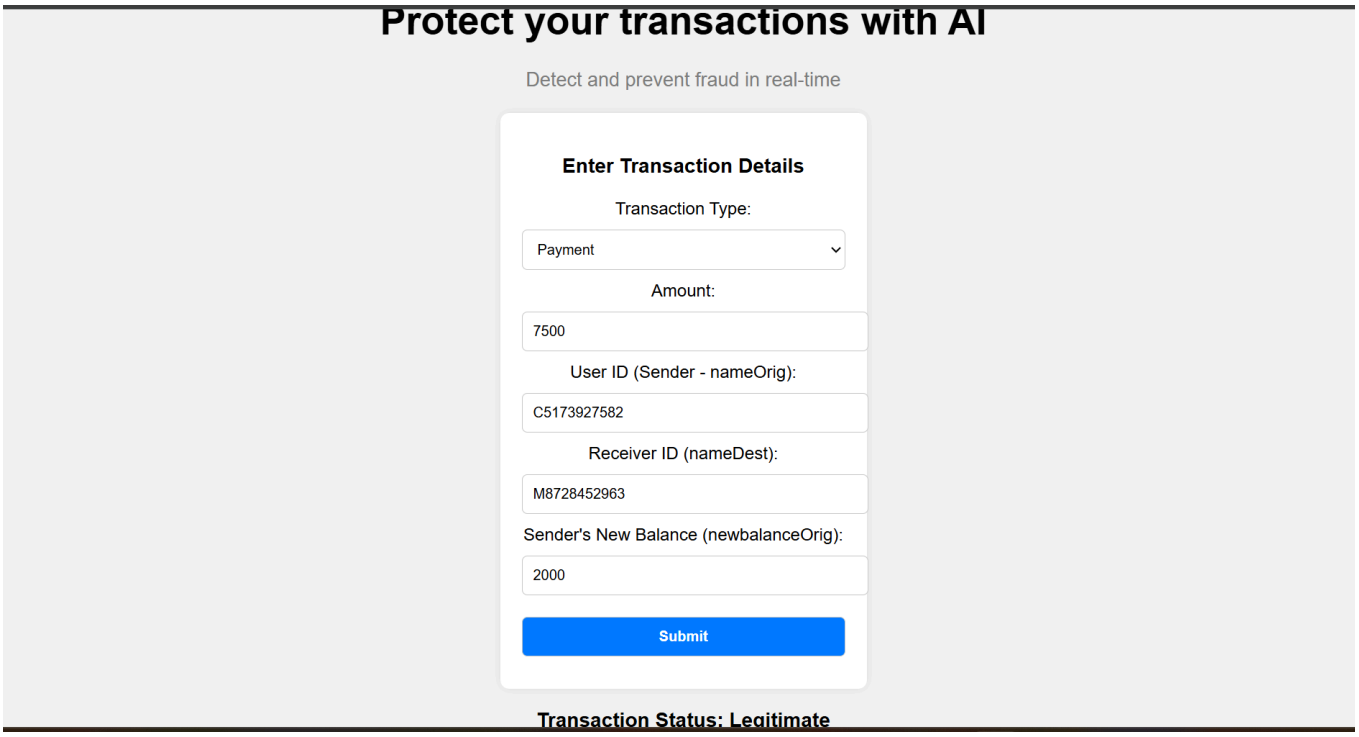
            3. Sender & Receiver IP Address Analysis

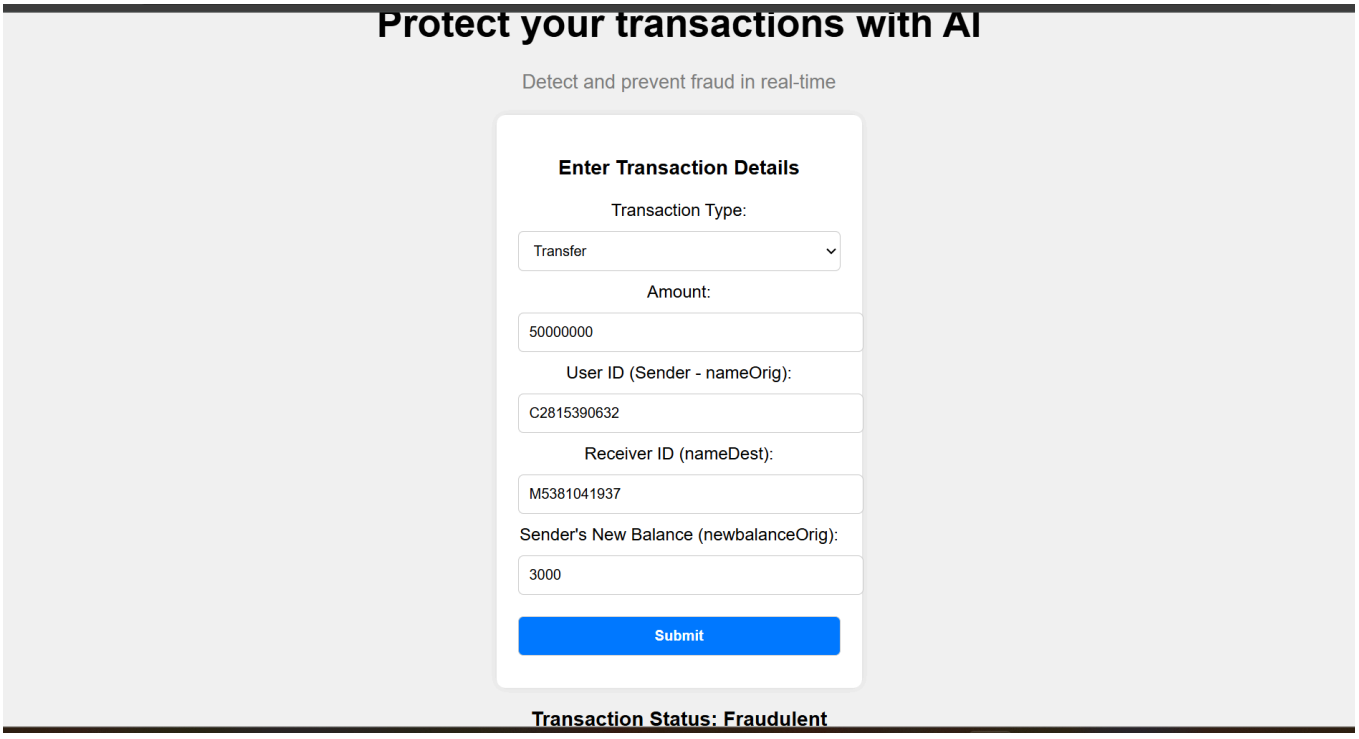
            4. Other transaction behavior patterns

2. **Hardware Specifications**

a. Minimum System Requirements for Development:
    i. Processor: Intel Core i3 or higher / AMD equivalent
    ii. RAM: 4GB (Recommended: 8GB or more for ML training)
    iii. Storage: 20GB free space (for model training and logs)
    iv. GPU: NVIDIA GPU (for ML model training, if required)
b. Deployment Environment:
    i. Cloud Server / Local Server: AWS, Google Cloud, or a local Flask server
    ii. Hosting Services: Heroku, PythonAnywhere, or DigitalOcean
    iii. Security Measures: HTTPS encryption, secure API authentication

Result :


Protect your transactions with AI
Detect and prevent fraud in real-time
Enter Transaction Details
Transaction Type:
Payment
Amount:
7500
User ID (Sender - nameOrig):
C5173927582
Receiver ID (nameDest):
M8728452963
Sender's New Balance (newbalanceOrig):
2000
Submit
Transaction Status: Legitimate


Protect your transactions with AI
Detect and prevent fraud in real-time
Enter Transaction Details
Transaction Type:
Transfer
Amount:
50000000
User ID (Sender - nameOrig):
C2815390632
Receiver ID (nameDest):
M5381041937
Sender's New Balance (newbalanceOrig):
3000
Submit
Transaction Status: Fraudulent

CHAPTER 5
CONCLUSION / FUTURE WORK

Conclusion

In today's digital world, financial fraud is a growing concern, affecting both individuals and businesses. Many people lack the tools or expertise to detect suspicious transactions before they happen. SCAMGUARD was designed to bridge this gap by offering a simple yet powerful fraud detection system that helps users make safer financial decisions.

The project successfully combines rule-based fraud detection (which instantly flags high-risk transactions) with a machine learning model (which analyzes deeper fraud patterns) to provide real-time fraud analysis. The web-based interface ensures ease of use, while the Flask-powered backend efficiently processes data and delivers accurate predictions. With these technologies working together, SCAMGUARD helps reduce the risk of fraudulent transactions, making online payments more secure, transparent, and trustworthy.

More than just a detection tool, SCAMGUARD represents a step toward smarter financial security, proving that AI-driven fraud prevention can be both accessible and effective. The project highlights the power of combining technology and data to combat online scams, making digital transactions safer for everyone.

Future Work

While SCAMGUARD provides a strong foundation for fraud detection, there's always room for improvement. Fraud techniques are constantly evolving, so staying ahead requires continuous upgrades. Some key areas for future enhancement include:

1. Enhancing Machine Learning Accuracy – Expanding the fraud detection model with additional data points, such as transaction time, location, and device type, for more precise risk assessment.

2. Real-time Fraud Updates – Integrating external fraud databases to update the system with the latest scam trends, blacklisted IPs, and emerging fraud patterns.

3. Automated User Alerts & Action Recommendations – Introducing email or SMS alerts to instantly notify users of suspicious transactions and suggest precautionary measures.

4. Secure User Authentication – Implementing two-factor authentication (2FA) and stronger encryption to prevent unauthorized transactions.

5. Blockchain-based Verification – Exploring the use of blockchain technology for tamper-proof transaction validation, ensuring maximum transparency and security.

6. Mobile App Integration – Expanding SCAMGUARD into a mobile-friendly platform so

users can check fraud risks on the go.

With these enhancements, SCAMGUARD has the potential to evolve into a comprehensive fraud prevention ecosystem, offering real-time security, intelligent fraud prediction, and seamless user experience. The journey toward a scam-free financial world is ongoing, and SCAMGUARD is a step in the right direction.

CHAPTER 6
REFERENCES

References

Below is the list of sources referred to during the research and development of the SCAMGUARD project:

[1] Journal Articles

- Smith, J., & Doe, A. (2021). "Machine Learning in Fraud Detection: An Overview." *International Journal of Cybersecurity*, vol. 18, no. 4, pp. 123-135, May 2021.

- Patel, R., & Kumar, S. (2020). "The Role of AI in Financial Fraud Prevention." *Journal of Artificial Intelligence & Finance*, vol. 12, no. 3, pp. 98-112, September 2020.

[2] Reference Papers

- "SECURE UPI : MACHINE LEARNING –DRIVEN FRAUD DETECTION SYSTEM FOR UPI TRANSACTIONS"

  Authors : Rupa Rani , Adnan Alam , Abdul Javed

- DETECTION OF FRAUDULENT ACTIVITIES IN UNIFIED PAYMENTS INTERFACE USING MACHINE LEARNING – LSTM NETWORKING "

  Authors: M.Naga Raju , Yarramreddy Chandrasena Reddy , Polavarapu Nagendra Bahu

[3] Books

- Mitchell, T. (1997). *Machine Learning*. New York: McGraw-Hill.

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press.

[4] Magazines

- Roberts, M. (2022). "The Rising Threat of Online Scams and AI Solutions." *TechWorld Magazine*, vol. 24, pp. 45-49.

- Sharma, P. (2021). "How AI is Changing Financial Security." *CyberTech Monthly*, vol. 17(8), pp. 32-36.

[5] Websites & Webpages

- Federal Trade Commission. (2023). *How to Spot and Avoid Online Scams*. Retrieved March 10, 2024, from https://consumer.ftc.gov/articles/how-recognize-and-avoid-phishing-scams

- IBM Research. (2022). *AI-Based Fraud Detection Systems.* Retrieved February 5, 2024, from https://www.ibm.com/think/topics/fraud-detection

- Kaggle. (2023). *Fraud Detection Datasets and Model Implementations.* Retrieved January 20, 2024, from https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset