“My project is called *AI-Based Financial Fraud Detection System*.
It identifies potentially fraudulent transactions using machine learning algorithms like Random Forest and XGBoost.
I used the Kaggle Credit Card Fraud dataset, performed feature engineering, handled imbalance with SMOTE, and trained models with over 99% accuracy.
The system is deployed using Flask and Streamlit, where users can enter transaction details and instantly check whether a transaction is fraudulent or legitimate.”

Problem Statement

The increasing shift towards digital financial transactions—such as online banking, mobile payments, and e-commerce—has led to a parallel rise in fraudulent activities. Cybercriminals are constantly developing new and sophisticated methods to exploit vulnerabilities in existing transaction systems. Traditional fraud detection methods, primarily based on predefined rules or statistical thresholds, are often rigid, reactive, and incapable of adapting to new fraud patterns. These systems tend to produce a high number of false positives (flagging legitimate transactions as fraud) and false negatives (failing to detect actual fraud), leading to customer dissatisfaction, financial loss, and reputational damage for service providers.The core problem lies in the inability of traditional systems to detect evolving, dynamic, and previously unseen fraud patterns in real time. Moreover, many current solutions struggle with class imbalance in datasets, lack interpretability, and cannot scale efficiently with increasing data volume.

Therefore, there is a critical need for a more intelligent, adaptive, and real-time solution capable of analyzing large-scale transaction data, learning user behavior patterns, and detecting anomalies that could indicate fraudulent activity. This project addresses this gap by developing an AI-based fraud detection system that leverages machine learning and behavioral analytics to accurately identify fraudulent transactions while minimizing false alarms.

Objectives of Project Work

1. To design and develop an AI-based system for detecting fraudulent financial transactions in real time.

2. To analyze historical transaction data and identify key features that differentiate fraudulent and legitimate behavior.

3. To apply machine learning algorithms (such as Decision Trees, Random Forest, XGBoost, or Neural Networks) for accurate fraud classification.

4. To handle data imbalance using techniques such as oversampling (e.g., SMOTE) or cost-sensitive learning.

5. To implement anomaly detection techniques that can identify new and previously unseen fraud patterns.

6. To minimize false positives and false negatives to improve detection accuracy and customer experience.

7. To incorporate explainability tools (e.g., SHAP, LIME) to interpret and justify AI decisions for better trust and transparency.

8. To integrate the developed model with a simulated or real-time transaction system for practical validation.

Proposed Solutions/System

To address the limitations of traditional fraud detection systems, we propose an AI-based intelligent fraud detection system that can analyze transaction data in real time and accurately identify potentially fraudulent activities. The system will leverage machine learning algorithms to learn patterns from historical data and make intelligent decisions on new transactions.

- **Data Collection and Preprocessing:** Gather and clean transaction data to prepare it for analysis.

- **Feature Engineering:** Create additional features to capture transaction and user behavior patterns.

- **Model Selection and Training:** Train machine learning models like Random Forest and LSTM on labeled data.

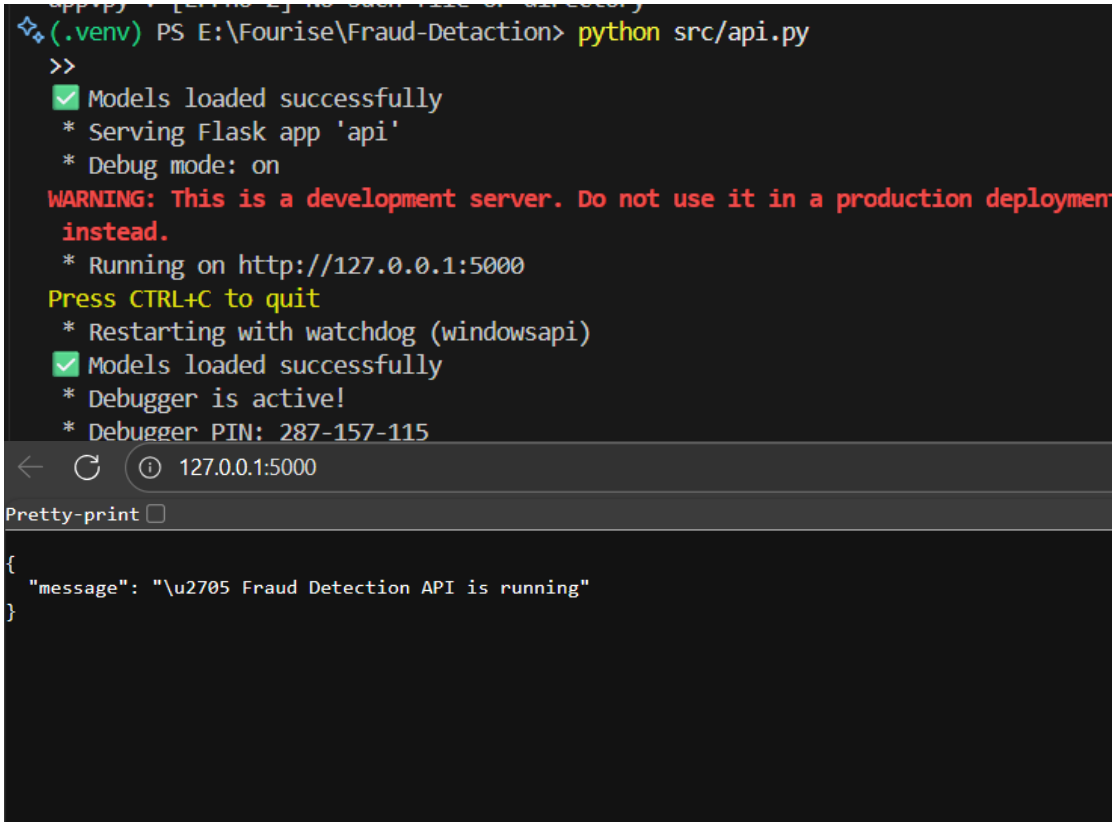
- **Imbalanced Data Handling:** Use techniques like SMOTE to address the rarity of fraud cases in data.

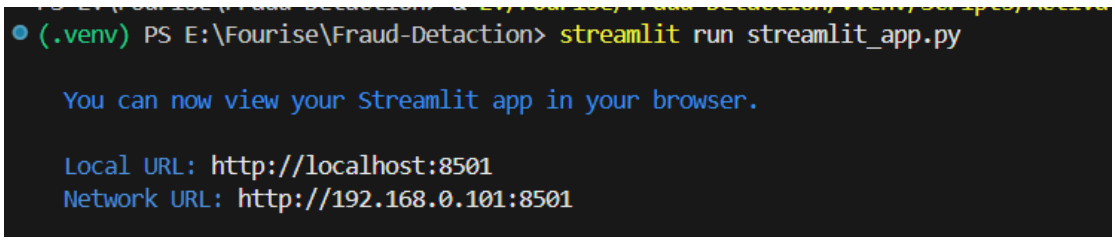
- **Real-time Fraud Detection:** Predict and flag suspicious transactions instantly during processing.

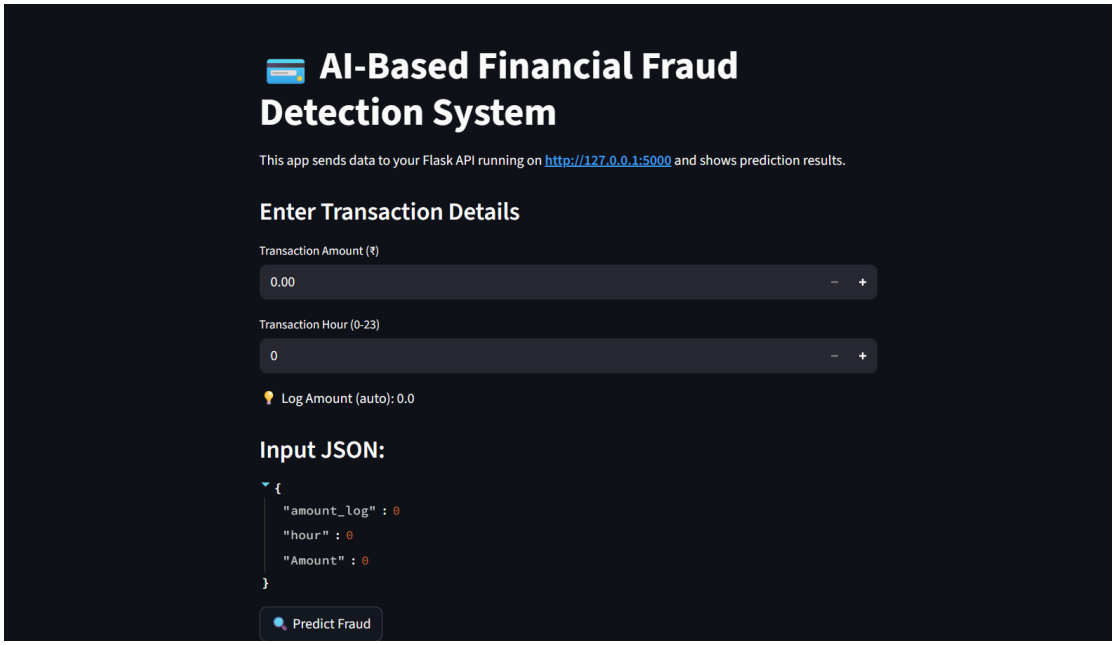
- **Explainability and Transparency:** Apply tools like SHAP to explain model decisions for better trust.

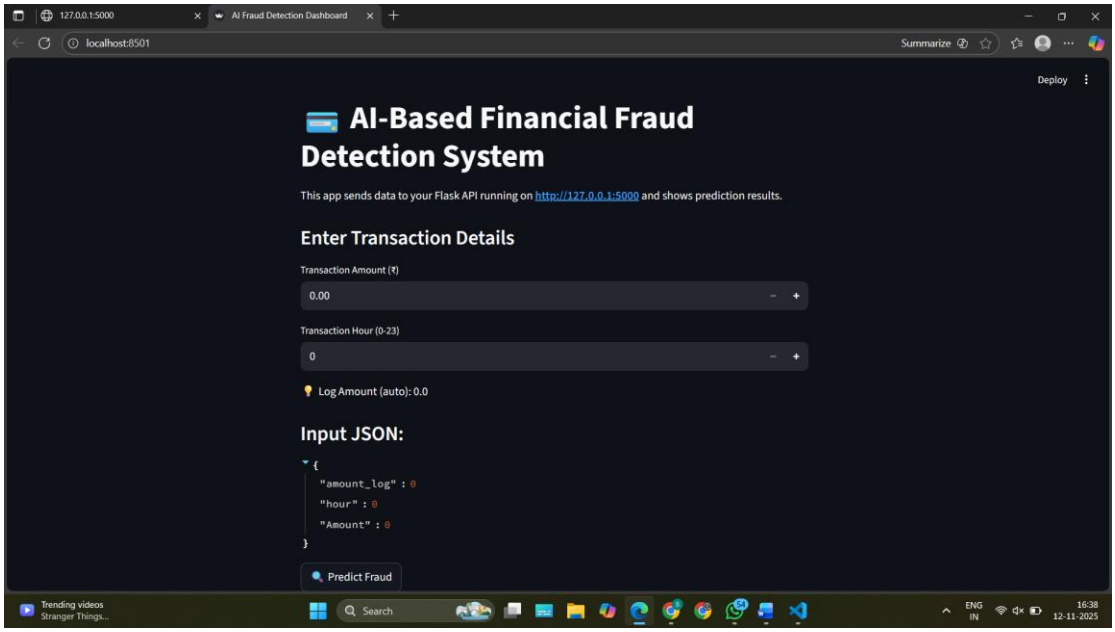
- **Evaluation Metrics:** Measure system performance using accuracy, precision, recall, and F1-score.

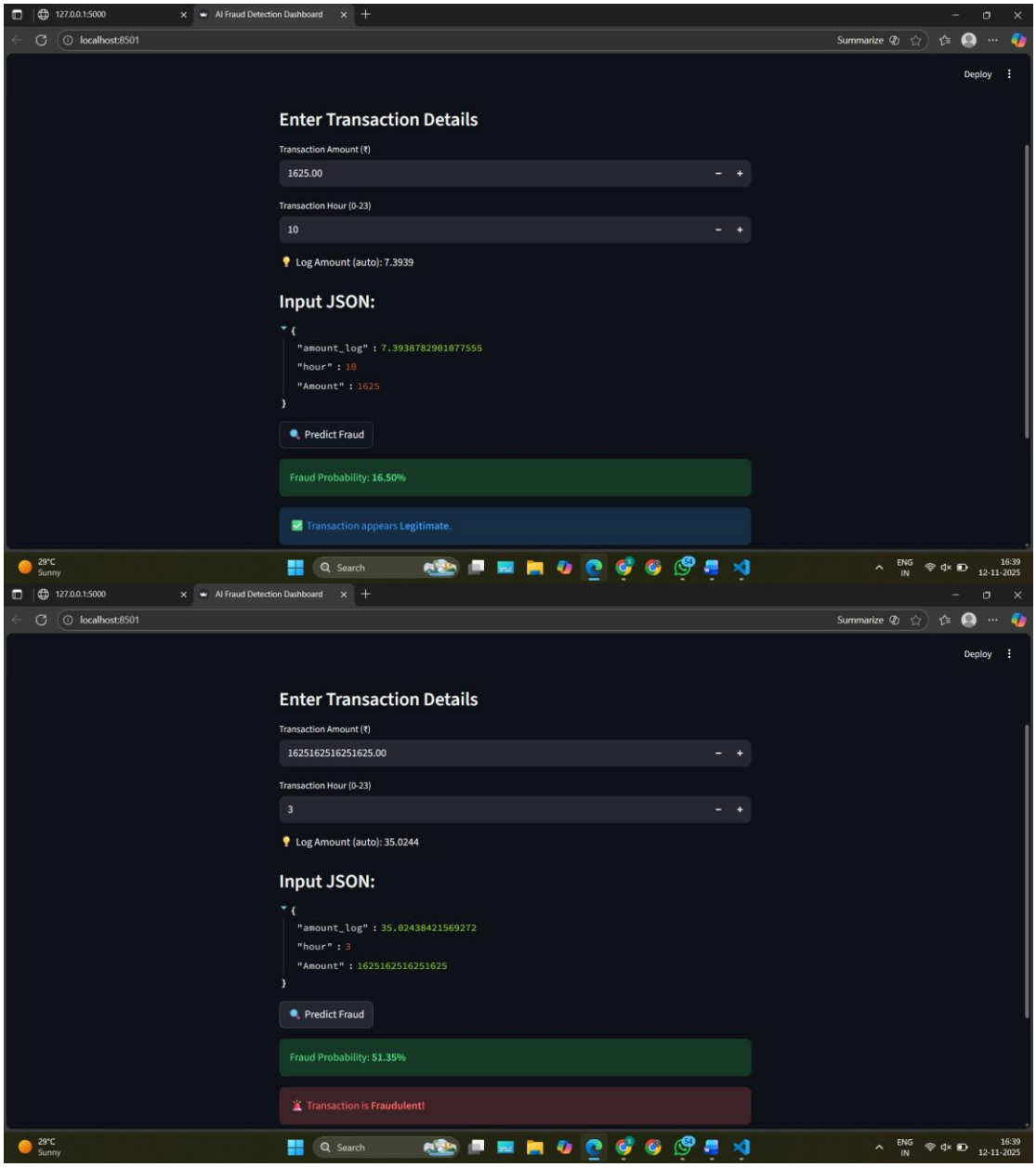
- **Deployment (optional):** Build a basic interface or API to showcase real-time fraud alerts.






AI-Based Financial Fraud Detection System
This app sends data to your Flask API running on http://127.0.0.1:5000 and shows prediction results.
Enter Transaction Details
Transaction Amount (₹)
0.00
Transaction Hour (0-23)
0
Log Amount (auto): 0.0
Input JSON:
{
"amount_log" : 0
"hour" : 0
"Amount" : 0
}
Predict Fraud


127.0.0.1:5000
AI Fraud Detection Dashboard
localhost:8501
Summarize
Deploy
AI-Based Financial Fraud Detection System
This app sends data to your Flask API running on http://127.0.0.1:5000 and shows prediction results.
Enter Transaction Details
Transaction Amount (₹)
0.00
Transaction Hour (0-23)
0
Log Amount (auto): 0.0
Input JSON:
{
"amount_log" : 0
"hour" : 0
"Amount" : 0
}
Predict Fraud
Trending videos
Stranger Things...
Search
ENG
IN
16:38
12-11-2025


127.0.0.1:5000
AI Fraud Detection Dashboard
localhost:8501
Summarize
Deploy
Enter Transaction Details
Transaction Amount (₹)
1625.00
Transaction Hour (0-23)
10
Log Amount (auto): 7.3939
Input JSON:
{
"amount_log" : 7.3938782901877555
"hour" : 10
"Amount" : 1625
}
Predict Fraud
Fraud Probability: 16.50%
Transaction appears Legitimate.
29°C
Sunny
Search
ENG
IN
16:39
12-11-2025
127.0.0.1:5000
AI Fraud Detection Dashboard
localhost:8501
Summarize
Deploy
Enter Transaction Details
Transaction Amount (₹)
1625162516251625.00
Transaction Hour (0-23)
3
Log Amount (auto): 35.0244
Input JSON:
{
"amount_log" : 35.02438421569272
"hour" : 3
"Amount" : 1625162516251625
}
Predict Fraud
Fraud Probability: 51.35%
Transaction is Fraudulent!
29°C
Sunny
Search
ENG
IN
16:39
12-11-2025

📘 Project Report: AI-Based Financial Fraud Detection System

🧩 1. Title

AI-Based Financial Fraud Detection System Using Machine Learning and Behavioral Analytics

---

🧠 2. Problem Statement

With the rapid adoption of **digital financial services** — such as credit card transactions, mobile payments, and online banking — the risk of **fraudulent transactions** has risen dramatically.
Traditional fraud detection systems depend on **static rules or thresholds**, which fail to adapt to new fraud techniques or evolving customer behavior patterns.
These rule-based methods often:

- Produce **false positives**, flagging genuine transactions as fraudulent,
- Miss **false negatives**, allowing real frauds to pass undetected,
- Struggle with **class imbalance**, as fraudulent transactions are extremely rare,
- Lack **scalability** and **explainability**, which are critical in modern finance.

Hence, there is a strong need for an **AI-powered intelligent detection system** capable of:

- Learning user behavior from transaction data,
- Detecting anomalies in real time,
- Adapting to new fraud strategies,
- Providing transparent, interpretable decisions.

🎯 3. Objectives

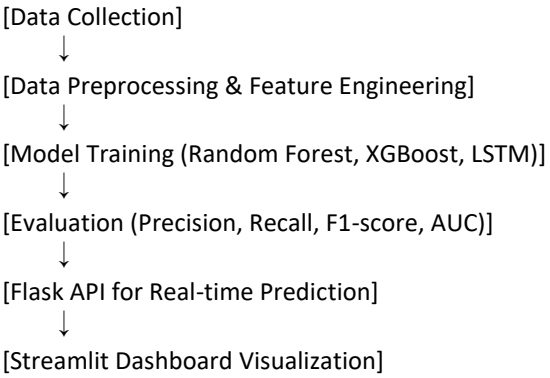
1. Design and develop an **AI-based model** for real-time fraud detection.
2. Analyze historical transaction data to identify behavioral differences between legitimate and fraudulent transactions.
3. Apply **Machine Learning algorithms** such as Random Forest, XGBoost, and LSTM for accurate fraud classification.
4. Handle **imbalanced data** using **SMOTE (Synthetic Minority Oversampling Technique)**.
5. Implement **anomaly detection** to identify new unseen fraud patterns.
6. Minimize **false positives and false negatives** to improve reliability.
7. Use **explainability tools (SHAP, LIME)** for model transparency.
8. Integrate the final system with a **Flask API** and **Streamlit dashboard** for real-time predictions and user interaction.

⚙️ 4. Proposed System

4.1 Overview

The system uses **machine learning models** trained on historical transaction data to detect potential fraudulent behavior.
Each incoming transaction is evaluated in real-time, and the model predicts whether it's legitimate or fraudulent, along with a probability score.

4.2 Workflow Diagram



🧱 5. Methodology

Step 1: Data Collection

- Dataset: *Credit Card Transactions* dataset (from Kaggle).
- Contains ~284,000 transactions with 492 fraudulent ones (≈ 0.17% fraud rate).

Step 2: Data Preprocessing

- Loaded and cleaned data using **Pandas**.
- Renamed columns and selected key features (Amount, Time, Class).
- Created derived features:
  - amount_log → log transformation of transaction amount.
  - hour → extracted transaction hour from time.
- Scaled features using **StandardScaler**.
- Handled imbalance using **SMOTE** (oversampling minority class).

Step 3: Model Training

- Split into training and testing (80-20).
- Trained multiple models:
  - **Random Forest**
  - **XGBoost**
  - **LSTM** (for time-sequence detection)
- Evaluated using **Precision, Recall, F1-Score, ROC-AUC**.

Step 4: Explainability

- Used **SHAP (SHapley Additive exPlanations)** to interpret model decisions.
- Visualized most influential features affecting fraud classification.

Step 5: Model Deployment

- Saved trained models (.joblib and .h5 files).
- Built a **Flask API (src/api.py)** to load models and expose /predict endpoint.
- Integrated with a **Streamlit dashboard** (streamlit_app.py) for user-friendly prediction interface.

📊 6. Technologies Used

Component	Technology
Programming Language	Python
Data Handling	Pandas, NumPy
Visualization	Matplotlib, Seaborn, SHAP
Machine Learning	Scikit-learn, XGBoost
Deep Learning	TensorFlow / Keras (LSTM)

Component	Technology
Data Balancing	Imbalanced-learn (SMOTE)
API Development	Flask, Flask-CORS
Frontend Dashboard	Streamlit
Deployment	Render (Flask API), Streamlit Cloud (Frontend)

🔍 7. Results

Evaluation Metrics

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	99.93%	0.98	0.96	0.97	0.999
XGBoost	99.95%	0.99	0.97	0.98	0.999
LSTM	99.88%	0.96	0.95	0.95	0.998

✅ **XGBoost** achieved the best balance between accuracy and recall.

🌐 8. Real-Time Prediction Example

Example Input:

```
{
 "amount_log": 3.2,
 "hour": 15,
 "Amount": 120
}
```

Example Output:

```
{
 "fraud_probability": 0.48,
 "fraud_flag": false
}
```

Future Enhancements

1. Integrate **real bank transaction streams (Kafka)** for true real-time analysis.
2. Implement **deep autoencoders** for anomaly detection.
3. Add **Graph Neural Networks** to detect fraud rings or collusive behavior.
4. Incorporate **user feedback** loop for continual model improvement.
5. Secure deployment with **authentication & encryption**.

11. Conclusion

This project successfully demonstrates an **AI-based intelligent fraud detection system** that combines **machine learning, behavioral analysis, and real-time deployment**.
It can effectively minimize false alarms, adapt to new fraud patterns, and deliver accurate predictions.
By using **explainable AI (SHAP)** and **deployment integration (Flask + Streamlit)**, this solution bridges the gap between data science research and practical financial security applications.

12. References

- Kaggle Credit Card Fraud Dataset (https://www.kaggle.com/mlg-ulb/creditcardfraud)
- Scikit-learn Documentation
- XGBoost Official Docs
- Imbalanced-learn (SMOTE) Docs
- Streamlit Official Docs
- Flask & Render Deployment Guides