ISSN 2277-2685
IJESR/June. 2025/ Vol-15/Issue-3s/523-529
A.Kranthi *et. al.*, / International Journal of Engineering & Science Research

Credit Card Fraud Detection using Machine Learning

Mohd Basit Mohiuddin, A.Kranthi, S.Pranavi, U.Sowmya, MVS.Darshitha

[1]Assistant Professor, Department Of Cse (AI&Ml), Bhoj Reddy Engineering College For Women, India.

[2,3,4,5]B. Tech Students, Department Of Cse (AI&Ml), Bhoj Reddy Engineering College For Women, India.

ABSTRACT

Credit card fraud has become a growing concern with the rapid expansion of digital payments and online transactions. Detecting fraudulent activities in real-time is critical to minimize financial losses and ensure customer trust. This project aims to develop a machine learning-based system for detecting credit card fraud by analyzing transaction patterns and identifying anomalies that may indicate fraudulent behavior.

The system leverages historical transaction data, including features such as transaction amount, time, location, and merchant category, to train supervised learning models like Logistic Regression, Decision Trees, and Random Forests. Advanced techniques such as SMOTE (Synthetic Minority Over-sampling Technique) are used to handle data imbalance, as fraudulent transactions typically represent a small fraction of the total data. The model's performance is evaluated using metrics like accuracy, precision, recall, and F1-score to ensure effective detection with minimal false positives.

This approach enables proactive fraud prevention and supports financial institutions in enhancing their security systems. The project demonstrates how data-driven techniques can be applied to address real-world problems in the financial domain.

1. INTRODUCTION

A credit card is a financial tool provided by banks and financial institutions that allows individuals to borrow money up to a predetermined limit to make purchases or withdraw cash. Unlike debit cards, which directly deduct funds from a bank account, credit cards enable users to access a line of credit that they repay later. Each month, the cardholder receives a statement listing all the transactions, and they can either pay the full amount, which avoids any interest charges, or pay a minimum amount and carry the remaining balance forward, which incurs interest charges based on the card's Annual Percentage Rate (APR). Credit cards offer numerous benefits, including convenience, worldwide acceptance, purchase protection, rewards programs such as cashback or travel points, and the opportunity to build a positive credit history if managed responsibly.

Existing System

The existing credit card fraud detection system relies on rule-based methods, manual reviews, and basic machine learning models. It flags suspicious transactions based on predefined rules and requires human verification, which can be slow and inefficient. While machine learning helps detect fraud patterns, it struggles with new fraud techniques. The system also faces issues like high false positives and delays in detection. Due to these limitations, modern approaches using ML and real-time monitoring are needed for better accuracy and efficiency.

Rule-based systems rely on predefined rules and thresholds, such as flagging transactions that exceed a certain amount or occur in unusual geographic locations, while statistical models analyze historical data to detect patterns that

deviate from normal behavior. Machine learning approaches like decision trees, random forests, support vector machines, neural networks, and deep learning models have become increasingly popular due to their ability to learn complex patterns from large datasets and improve over time.

Proposed System

The proposed system aims to improve fraud detection accuracy using advanced technologies like Machine Learning(ML). It analyzes transaction patterns in real-time, reducing false positives and improving detection speed. The system adapts to new fraud techniques using machine learning models, eliminating reliance on fixed rules. Automated alerts notify users and banks immediately upon detecting suspicious activity, enhancing security. Additionally, it ensures data privacy and scalability to handle large transaction volumes efficiently. This approach minimizes manual intervention while providing a faster, more reliable fraud detection system.

2. REQUIREMENT ANALYSIS

Functional Requirements

In this project, the system is divided into the following modules:

1. **Transaction Upload Module:**
   Users can upload Transaction Data (CSV or input form) through the webpage to determine whether a transaction is legitimate or fraudulent.

2. Trained Model:

- **Feature Extraction:** The system analyses transaction attributes such as payment status, billing statements etc.
- **Classification:** The trained ML model classifies transactions as legitimate or fraudulent based on the extracted features.

Non-Functional Requirements

Usability:

- User interface is web-based, responsive, and easy to navigate.
- Clear feedback is provided to indicate correct or incorrect poses.

Reliability:

- System should operate smoothly during live webcam streaming.
- Keypoints must be reliably detected for correct classification.

Maintainability:

- The modular code structure allows easy updates (e.g., adding new poses)

Performance:

- Pose classification should occur within 1–2 seconds of detection.
- Low latency during camera feed and inference.

Scalability:

- Capable of supporting more poses in future with retraining.
- System structure supports deployment to multiple users.

Sofware Requirements

Operating System: Windows 7, 8, 10, or more
Text Editor : VS Code / Sublime Text
Framework : Flask
Coding Language: Python
Tools: Pandas, Numpy ,Seaborn , Pickle, Scikit-learn

Hardware Requirements

Processor: Intel i3 or higher
RAM: 8GB+ for smooth execution
Storage: 150GB SSD (recommended)

3. DESIGN

Design represents the number of components we are using as a part of the project and the flow of

request processing i.e., what components in processing the request and in which order.

An architecture description is a formal description and representation of a system organized in a way that supports reasoning about the structure of the system.

System Architecture

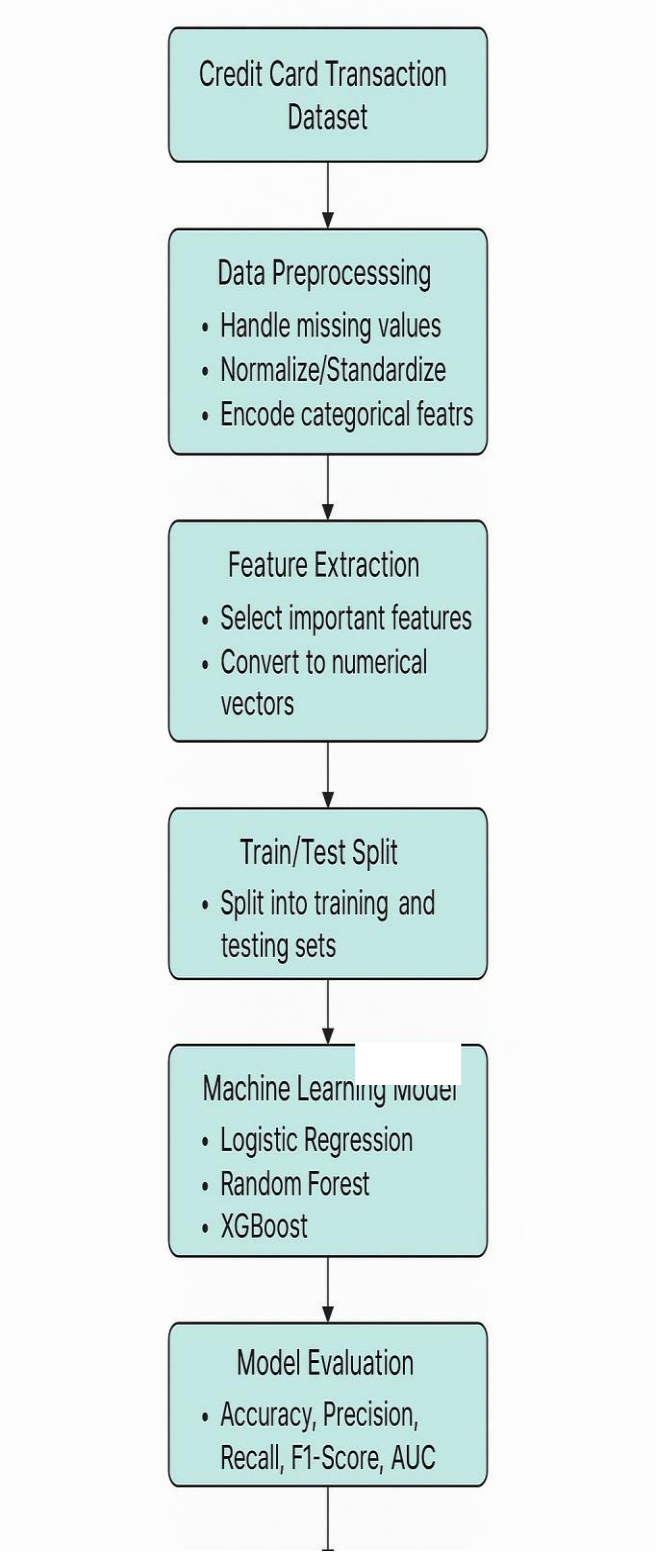

Fig 3.1 System Architecture of Credit Card Fraud Detection

Software Process Model

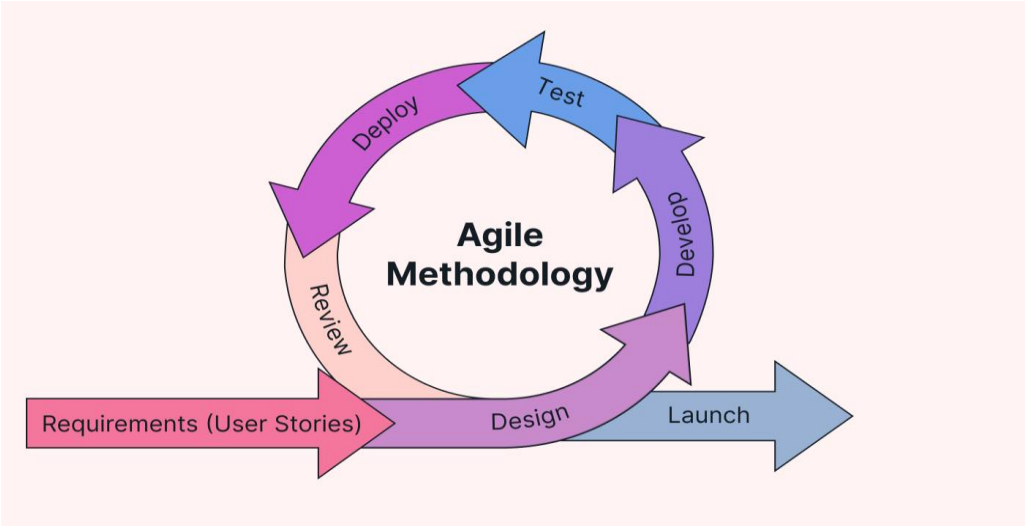

Fig 3.2 Software process model

4. IMPLEMENTATION

Technologies

The proposed system is implemented using Python-based tools and libraries to enable efficient data preprocessing, machine learning model training, and deployment through a web-based interface. Below are the key steps in the implementation, along with the technologies used at each stage:

1. **Environment Setup**

Technology Used: Python, Jupyter Notebook, Streamlit

Set up a Python environment with all necessary libraries for machine learning, data analysis, and visualization.

2. **Data Collection**

Technology Used: CSV files (e.g., Kaggle Credit Card Fraud Dataset)

A labeled dataset of credit card transactions, including both legitimate and fraudulent cases, is used for model training and testing.

3. **Feature Selection**

Technology Used: Scikit-learn

Apply feature importance methods like correlation analysis.

4. **Model Training and Evaluation Technology used:**

Technology Used: Scikit-learn

5. **Model Saving:**

Technology Used: Pickle

Visual overlays and voice instructions guide users to correct their alignment.

6. Web Deployment

Technology Used: Streamlit

A lightweight web interface is created where users can input transaction details.

It displays real-time classification results as Fraud or Legitimate.

5-TESTING

Unit Testing

During This first round of testing, the program is submitted to assessments that focus on specific units or components of the software to determine whether each one is fully functional. The main aim

of this endeavour is to determine whether the application functions as designed.

In this phase, a unit can refer to a function, individual program or even a procedure, and White box testing method is usually used to get the job done. One of the biggest benefits of this testing phase is that it can be run every time a piece of code is changed, allowing issues to be resolved as quickly as possible. It quite common for software developers to perform unit tests before delivering software to testers for formal testing.

Integration Testing

Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. This is particularly beneficial because it determines how efficiently the units are running together. Keep in mind that no matter how efficiently each unit is running, if they properly integrated, it will affect the functionality of the software program. In order to run these types of tests, individuals can make use of various testing methods, but the specific method that will be used to get the job done will depend greatly on the way in which the units are defined

System Testing

System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards. System testing is undertaken by independent testers who haven't played a role in developing the program. This testing is performed in an environment that closely mirrors production. System Testing is very important because it verifies that the application meets the technical, functional, and business requirements that were set by the customer.

Acceptance Testing

The final level, Acceptance testing (or User Acceptance Testing), is conducted to determine whether the system is ready for release. During the Software development life cycle, requirements changes can sometimes be misinterpreted in a fashion that does not meet the intended needs of the users. During this final phase, the user will test the system to find out whether the application meets their business needs. Once this process has been completed and the software has passed, the program will then be delivered to production. The extensiveness of these tests is just another reason why bringing software testers in early is important. When a program is more thoroughly tested, a greater number of bugs will be detected; this ultimately results in higher quality software.

6. SCREENSHOTS

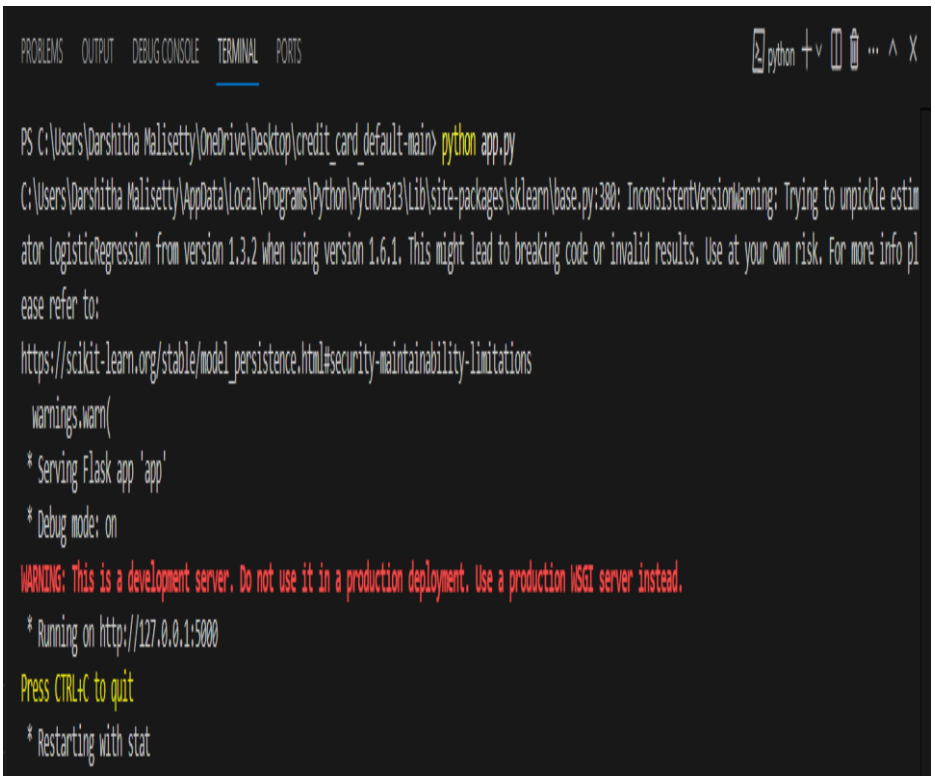

Fig 6.1 Activate Project

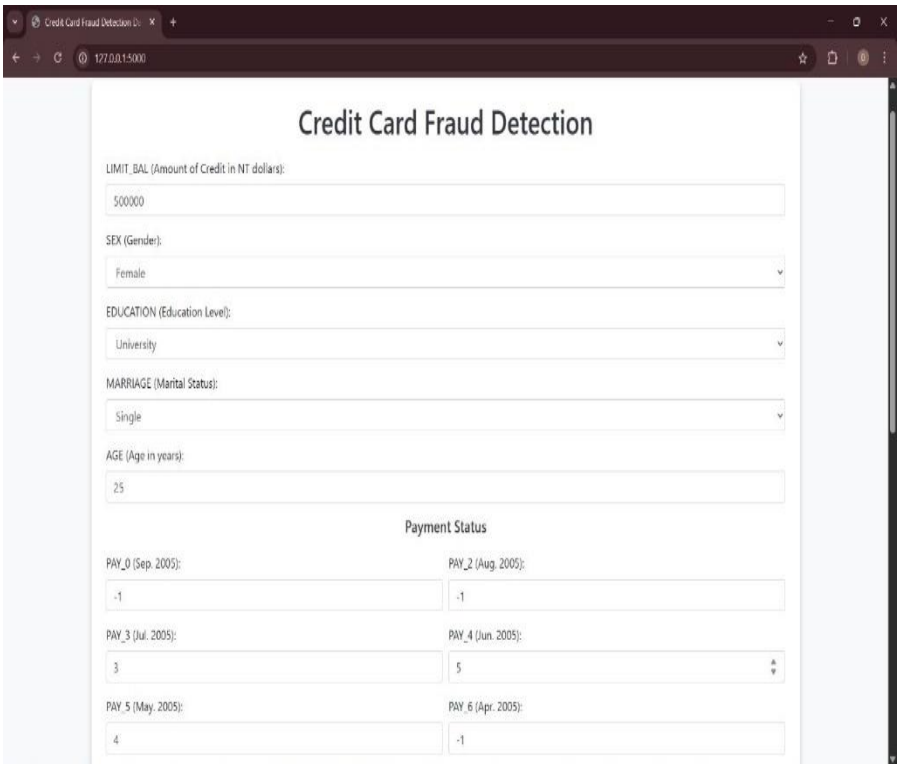

Fig 6.2 Main Page 1

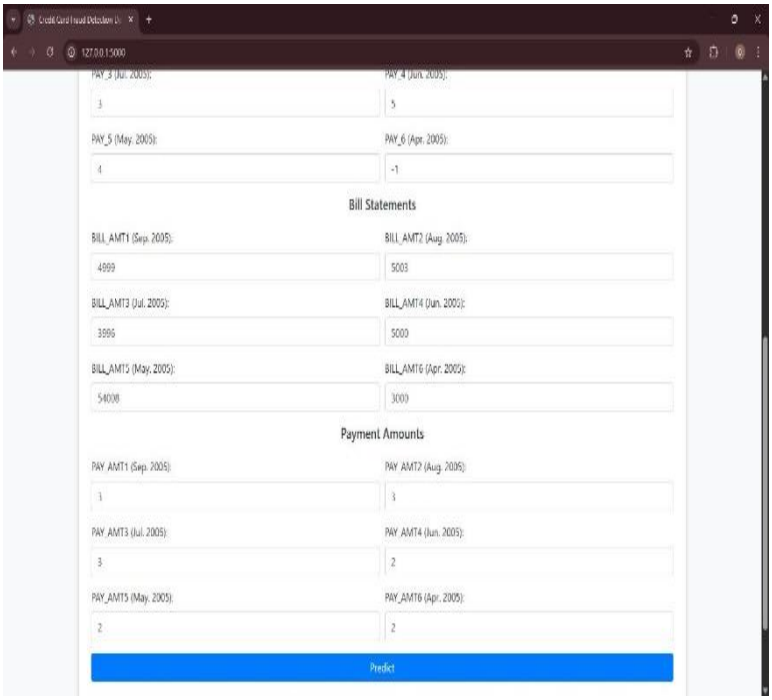

Fig 6.3 main Page 2

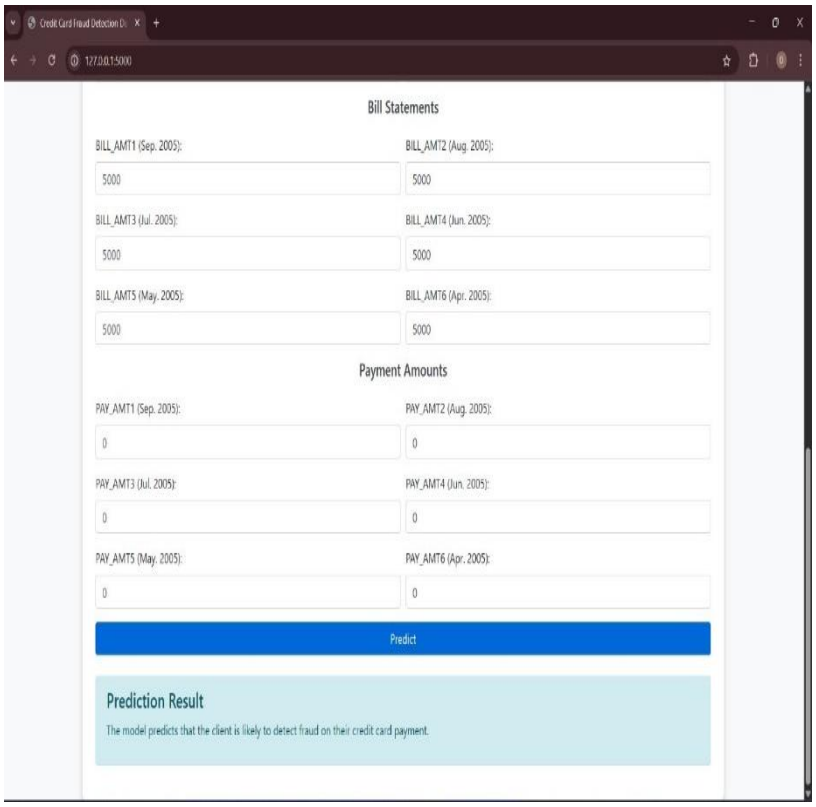

Fig 6.4 Output 1

7-CONCLUSION AND FUTURE SCOPE

Conclusion

Credit card fraud detection system is essential for preventing unauthorized transactions and ensuring financial security. By leveraging machine learning and real-time monitoring, the proposed system enhances fraud detection accuracy while minimizing false positives. It adapts to evolving fraud techniques, providing a more reliable and scalable solution compared to traditional rule-based methods. With automated alerts, improved security, and reduced manual intervention, this system helps protect both customers and financial institutions from financial losses. Implementing such an advanced fraud detection system ensures a safer and more efficient transaction process in the digital payment ecosystem.

Future Scope

- Use smarter models to catch more fraud and reduce mistakes.
- Detect fraud in real-time while the transaction is happening.
- Show clear reasons why a transaction is marked as fraud using explainable tools.
- Make the system learn and improve automatically as new fraud tricks appear.
- Look at connections between accounts and transactions to find hidden fraud.
- Handle unbalanced data better, since fraud cases are much fewer than normal ones.
- Use secure technology like blockchain to make transactions safer.
- Put the system on cloud platforms so it can work faster and handle more data.

REFERENCES

1. L. Zheng, G. Liu, C. Yan, and C. Jiang, ―Transaction fraud detection based on total order relation and behavior diversity,‖ IEEE Trans. Comput. Social Syst., vol. 5, no. 3, pp. 796–806, Sep. 2018.
2. 2.A.Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," *Machine Learning*, vol. 100, no. 2-3, pp. 251–277, Sept. 2015. [Online]. Available: https://link.springer.com/article/10.1007/s10994-015-5526-3
3. S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, Feb. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S095741741100077X
4. M. Zareapoor and P. Shamsolmoali, "Application of credit card fraud detection: Based on bagging ensemble classifier," *Procedia Computer Science*, vol. 48, pp. 679–685, 2015.