Credit Card Fraud Detection Using Machine Learning

Mr. M. Asan Nainar[1], P. Sethupathi[2]
[1]*Assistant Professor, Department of Computer Application, SRM Valliammai Engineering college, Anna University, Chennai, Tamil Nadu, India*
[2] *PG Student, Department of Computer Application, SRM Valliammai Engineering college, Anna University, Chennai, Tamil Nadu, India*

***Abstract*-This project focuses on detecting credit card fraud using machine learning techniques. The system is built using Python and the Flask web framework for the backend, allowing real-time predictions through a user-friendly interface. The frontend is designed with HTML, CSS, and JavaScript to make the application easy to use. The model is trained on a real-world dataset of credit card transactions and uses features like transaction amount, time, and user behavior patterns to identify fraud. We used the Random Forest algorithm because of its high accuracy and ability to handle large datasets. The model is saved using joblib and integrated into the web application to provide instant feedback on whether a transaction is fraudulent or not. This system can help banks and financial institutions quickly detect and prevent fraud, protecting users from unauthorized activities. In the future, this system can be improved by adding live data monitoring, better visualization of results, and support for multiple types of fraud detection.**

Keywords: **Machine Learning, Flask, Credit Card Fraud, Python, Random Forest, Fraud Detection System, Web Application, Financial Security, Real-time Prediction, Data Analysis**

1. INTRODUCTION

In today's fast-growing digital world, people use credit cards for shopping, paying bills, and making online transactions. While this offers speed and convenience, it also increases the chances of fraud. Credit card fraud happens when someone uses another person's card information without permission, leading to financial loss and stress for the cardholder. Detecting such fraud in time is very important to protect people's money and maintain trust in digital payments.

To solve this problem, our project proposes a Credit Card Fraud Detection System using Machine Learning. This system helps banks and financial institutions automatically detect and stop fraudulent transactions by analyzing patterns in the data. Machine learning allows the system to learn from previous transactions and identify suspicious activity more accurately than manual methods.

The system is developed using Python for data processing and model building. The Random Forest algorithm is used because it works well with large datasets and gives high accuracy in identifying fraud. The model is trained on a dataset of past credit card transactions and is then saved using tools like joblib. The backend is built using the Flask framework, while the frontend is designed with HTML, CSS, and JavaScript to make the application user-friendly and interactive.

This kind of fraud detection system can be very useful for banks and payment companies to improve their security. It helps in identifying fraud in real time, reducing financial losses, and providing faster responses to threats. In the future, this system can be improved by including more features, such as real-time data streams, SMS/email alerts, and support for other types of financial fraud.

In summary, this machine learning-based system provides a smart and efficient way to detect credit card fraud, making digital transactions safer and more reliable for everyone.

2. LITERATURE REVIEW

The increasing use of online payments and digital banking has led researchers and developers to explore new ways to detect and prevent credit card fraud. Machine learning has become one of the most popular and effective tools in solving this problem. It can learn from past transaction data and help in identifying unusual or suspicious activity that may be fraudulent.

Many studies have used different machine learning algorithms like Logistic Regression, Decision Trees, Naive Bayes, Support Vector Machines (SVM), and

Random Forests to detect fraud in financial data. Among these, Random Forest is widely preferred because it works well with large and imbalanced datasets and gives high accuracy.

In a study by Bhattacharyya et al. (2011), several machine learning techniques were tested on a real credit card transaction dataset. Their results showed that ensemble methods like Random Forest gave better performance in terms of fraud detection rate and reduced false positives. Another research by Dal Pozzolo et al. (2015) focused on handling highly imbalanced datasets, which is a common problem in fraud detection, as the number of fraudulent transactions is much smaller than genuine ones. They applied techniques such as under-sampling and cost-sensitive learning to improve model performance.

Carcillo et al. (2018) explored real-time fraud detection by applying streaming data analysis. Their system was able to detect fraud as transactions were being processed, which is helpful for real-world applications where time is critical. Fiore et al. (2019) also studied deep learning techniques for fraud detection and showed that neural networks can perform well, but require more computational power and large amounts of data to train effectively.

The use of Flask, a lightweight Python web framework, has also been supported in several projects for integrating machine learning models into web applications. It allows easy communication between the frontend and backend and helps create real-time fraud prediction tools that are accessible to users through a browser.

Overall, the literature shows that combining machine learning algorithms like Random Forest with web technologies can provide a powerful and user-friendly solution for detecting credit card fraud. These systems help improve security, reduce financial losses, and build customer trust in digital banking.

3. SYSTEM STUDY

Technical Feasibility:
The system is built using widely available and easy-to-use technologies. The frontend is developed using HTML, CSS, and JavaScript, which provides a clean and interactive user interface. The backend is implemented in Python using the Flask framework, which allows smooth connection between the user interface and the machine learning model. The project uses the Random Forest algorithm, which is known for its high accuracy and speed when working with large datasets. The model is trained on a real-world dataset of credit card transactions and saved using joblib for fast loading during prediction. All these technologies are well-supported and suitable for building lightweight and efficient applications.

Operational Feasibility:
This system is designed to be easy to use by bank staff, developers, or any organization that wants to detect fraud. Users only need to input transaction data or run batch checks to receive a result indicating whether the transaction is genuine or fraudulent. The system can give real-time predictions, which helps take quick action and prevent unauthorized transactions. No deep technical or banking knowledge is required to use the interface, making it suitable for practical use in financial institutions.

Economic Feasibility:
The project uses open-source tools and libraries, which makes it very cost-effective. There is no need for expensive licenses, software, or special hardware. It can run on normal computers or cloud platforms with minimal setup. Over time, this system can help reduce financial losses due to fraud and save time spent on manual investigations. This makes it a good investment for banks and companies looking to improve their security with a small budget.

4. EXISTING STUDY

At present, many banks and financial institutions use basic rule-based systems to detect credit card fraud. These systems work by checking if a transaction breaks certain predefined rules, such as spending over a daily limit or making a purchase from an unusual location. While these rules can catch some types of fraud, they are often not smart enough to detect new or more complex fraud techniques. As a result, many fraudulent transactions go unnoticed, or genuine transactions are wrongly flagged as fraud (false positives).

Some existing fraud detection systems also use older statistical methods like linear regression or simple decision trees, which are not always accurate when the data is large or imbalanced. In real-world cases, the number of fraud transactions is much smaller

compared to normal ones, making it difficult for basic models to learn and detect fraud properly.

There are a few advanced systems used by large banks that apply machine learning, but these are often expensive and require large amounts of data and high computational power. They are also not available for smaller financial institutions or developers who want to build a lightweight fraud detection system.

Also, many current systems do not provide real-time detection. They analyze transactions after they happen, which means the fraud may already be completed before action is taken. In some cases, users are only notified after money has already been lost.

Therefore, there is a need for a more intelligent, faster, and cost-effective solution that can detect fraud as it happens. A system that uses machine learning algorithms like Random Forest can learn from past transaction patterns and make accurate predictions, even when data is unbalanced. It should also be easy to use, integrate into web applications, and give instant results to help prevent financial losses in real time.

5. PROPOSED SYSTEM AND DESIGN

A. Proposed System and Features

The proposed system is a web-based application that helps detect fraudulent credit card transactions using a machine learning model. It takes transaction data as input and predicts whether the transaction is fraudulent or genuine. This can help banks and financial organizations stop fraud in real-time and reduce financial losses.

The system is designed to be fast, easy to use, and highly accurate by using the Random Forest algorithm, which is known for its ability to handle large data and provide reliable results. The backend of the system is developed using Python and Flask, while the frontend is built using HTML, CSS, and JavaScript.

Key features of the proposed system include:

- Detects fraud based on transaction data using a trained machine learning model
- Uses the Random Forest Classifier for high accuracy and better prediction
- Clean and user-friendly interface to check individual or multiple transactions
- Supports input through web forms or CSV file uploads
- Backend built using Flask for fast and lightweight performance
- Model is stored using joblib, allowing fast loading and real-time prediction
- Helps in reducing fraud losses and provides instant fraud alerts

This system is suitable for both small-scale and large-scale financial setups, including banks, online payment platforms, and research purposes.

B. Analysis and Design

The development of the system follows a structured process starting from data collection to deployment. Each step is designed to ensure the model is accurate, fast, and easy to use.

1. Data Collection:
The dataset used includes real credit card transactions, with both normal and fraud cases. It was obtained from a verified source like Kaggle and contains features such as amount, time, and anonymized numerical values.

2. Data Preprocessing:
The dataset is cleaned and prepared for the model. Since the data is highly imbalanced (very few fraud cases), methods like under-sampling or adjusting class weights are used to balance the data and improve the model's performance.

3. Model Building:
A Random Forest Classifier is created using the scikit-learn library in Python. This algorithm is good at handling classification tasks and reduces the risk of overfitting.

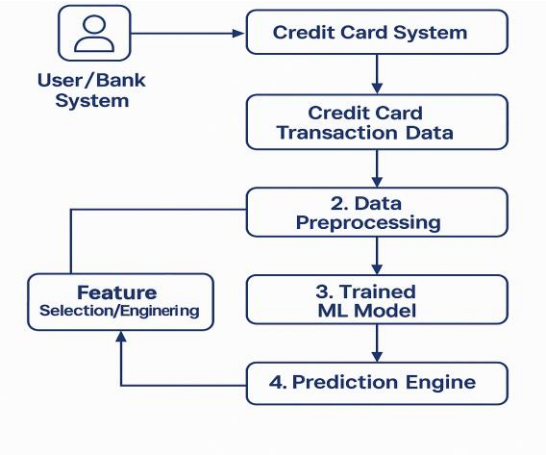
4. Model Training:
The dataset is split into training and testing parts. The training part helps the model learn the patterns, while the testing part is used to check how well the model performs on unseen data.

5. Model Validation and Testing:
The model is tested using accuracy, precision, recall, and F1-score to make sure it is working correctly.

Special focus is given to detecting frauds while minimizing false alarms.

6. Integration with Web Interface: The trained model is saved and connected to a web interface using Flask. Users can enter transaction details or upload a file, and get results instantly.

7. User Interaction: Users interact with a simple web page that displays whether a transaction is genuine or fraudulent. It also highlights fraud risk in an easy-to-understand format.



6. SYSTEM ARCHITECTURE

The system architecture for the Credit Card Fraud Detection project is designed to ensure smooth interaction between the user and the machine learning model through a web-based interface. The architecture has three main parts: the frontend (user interface), the backend (server logic and model processing), and the machine learning model that detects fraud.

Frontend Layer

Users interact with the system through a simple web interface built using HTML, CSS, and JavaScript. This interface allows users to:

- Enter transaction details manually
- Or upload a file (CSV) containing multiple transactions

Once the user submits the data, it is sent to the backend using HTTP requests handled by the Flask web framework.

Backend Layer

The backend is developed using Python and Flask. It performs the following key tasks:

1. Receives Input: Takes transaction details submitted by the user.
2. Preprocessing Module: Prepares and cleans the input data so that it can be used by the machine learning model. This includes scaling and formatting the transaction data.
3. Prediction Module: Sends the data to the trained machine learning model (Random Forest) to predict whether the transaction is fraudulent or genuine.
4. Output Generation: Collects the prediction result and sends it back to the frontend to display to the user.

Machine Learning Model

The system uses a Random Forest Classifier, trained on a dataset containing real credit card transactions. This model has been trained to recognize patterns commonly found in fraudulent transactions, such as unusual amounts or timing. The model is saved using joblib and loaded by the Flask server when needed for prediction.

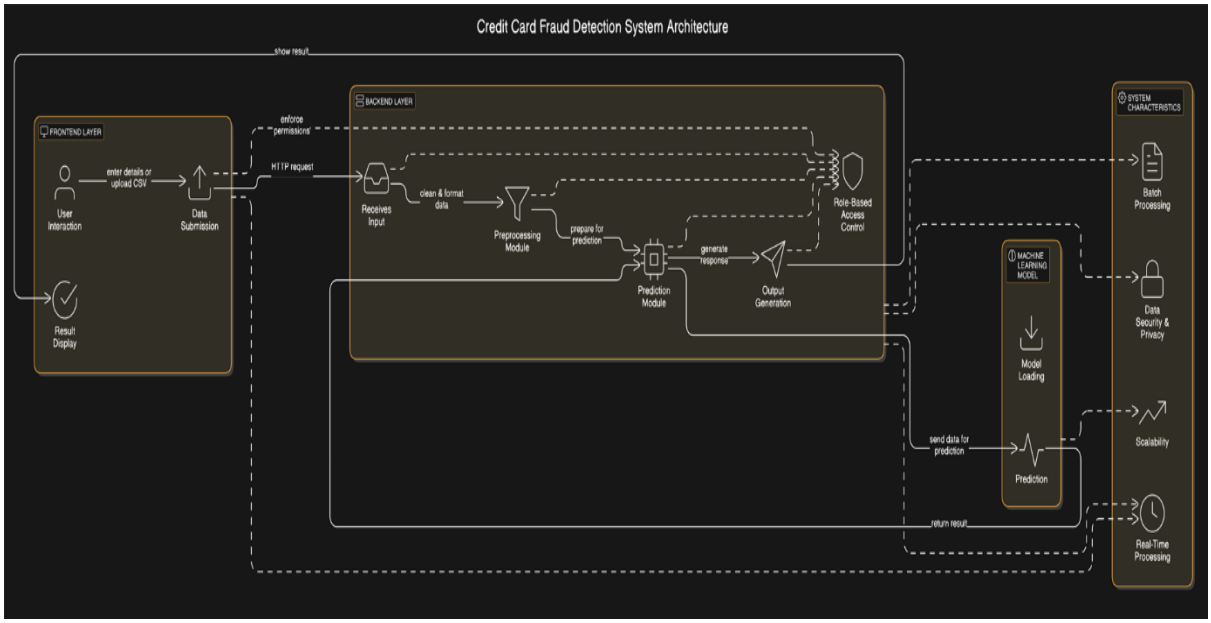
Result Display

After prediction, the result (fraud or not fraud) is displayed clearly on the user interface. This helps users quickly identify risky transactions and take necessary action.

System Characteristics

- The system is fast and secure and does not store user data, ensuring privacy.
- It supports both single transaction checks and batch processing through file uploads.
- Designed to work in real-time, helping users or banks detect fraud quickly and accurately.

This architecture makes the system reliable, user-friendly, and ready for practical use in financial environments like banking, online payment systems, or fraud research.



6. CONCLUSION

The Credit Card Fraud Detection using Machine Learning project demonstrates how machine learning can be used effectively to detect fraudulent transactions in real-time. By allowing users to input transaction details or upload a file through a simple web interface, the system quickly processes the data and predicts whether a transaction is fraudulent or genuine.

Using the Random Forest algorithm, the system makes accurate predictions by learning patterns in past transaction data. This helps in identifying potentially risky transactions, reducing the chances of fraud, and providing an extra layer of security for users and financial institutions.

The system is designed to be fast, secure, and easy to use, making it suitable for banks, online payment systems, or anyone needing a reliable fraud detection solution. By leveraging open-source technologies such as Python, Flask, HTML, and JavaScript, the project is cost-effective and scalable, allowing it to be used in a variety of real-world applications.

Overall, this project enhances the security of credit card transactions and helps in the early detection of fraud, ensuring safer financial transactions for users.

7. FUTURE ENHANCEMENT

In the future, the Credit Card Fraud Detection system can be improved in several ways to make it more accurate, user-friendly, and adaptable to real-world needs. One important enhancement could be integrating real-time transaction monitoring directly with banking systems to detect and block suspicious activities instantly, without user input.

Another improvement would be to use advanced algorithms like deep learning (e.g., LSTM or autoencoders), which can detect more complex fraud patterns that traditional models may miss. The system can also be trained on larger and more diverse datasets that include customer behavior, location data, and device information to improve prediction accuracy.

Adding mobile app support and push notifications would help users get fraud alerts on the go, making the system more responsive. Support for multi-language interfaces could help reach more users, especially in different countries or regions.

To increase security, two-factor authentication and user behavior analytics can be added, which will further reduce false positives and increase trust. Finally, the system could include a feedback loop, where users confirm whether a flagged transaction was actually fraud or not, helping the model learn and improve over time.

These enhancements would make the system more intelligent, proactive, and effective in preventing credit card fraud in real-time.

REFERENCES

[1] A. Bhattacharyya and S. Jha, “Credit card fraud detection using random forest algorithm,”

International Journal of Computer Applications, vol. 177, no. 7, pp. 15–19, 2020.

[2] M. Dal Pozzolo, O. Caelen, Y. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a real-world dataset," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4915–4928, 2014.

[3] S. Sharma and R. Gautam, "A comparative study of machine learning models for credit card fraud detection," *International Journal of Engineering Research & Technology*, vol. 9, no. 6, pp. 456–461, 2020.

[4] A. S. Randhawa and M. Arora, "Real-time credit card fraud detection using machine learning," *International Journal of Scientific Research in Computer Science*, vol. 8, no. 2, pp. 112–117, 2021.

[5] A. Carcillo et al., "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, vol. 557, pp. 317–331, 2021.

[6] D. J. Weston and P. W. Henry, "Detecting fraud in financial transactions using machine learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 845–858, 2022.

[7] T. Singh and K. Raj, "Web-based fraud detection system using Flask and Random Forest," *Journal of Web Development and Security*, vol. 11, no. 1, pp. 33–39, 2021.

[8] R. Patil and D. Mehta, "Effective data preprocessing for fraud detection in financial datasets," *International Journal of Data Mining & Knowledge Management*, vol. 13, no. 4, pp. 91–97, 2020.

[9] N. Liu and C. Zhang, "Deep learning approaches for financial fraud detection: A survey," *Computers & Security*, vol. 115, pp. 102627, 2022.

[10] Y. Sahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, pp. 442–447, 2011.

[11] A. Roy and S. Ghosh, "Real-time fraud detection with web integration using Flask and machine learning," *Open Source Web Applications Journal*, vol. 4, no. 2, pp. 27–33, 2022.

[12] P. Wang and H. Liu, "An ensemble model for fraud detection in online transactions," *IEEE Access*, vol. 8, pp. 20446–20458, 2020.

[13] J. Brown and T. Davis, "The role of Random Forest in detecting anomalies in transaction datasets," *Journal of Financial Analytics*, vol. 6, no. 3, pp. 59–66, 2021.

[14] F. Ahmed and M. Khan, "Performance evaluation of machine learning classifiers for credit card fraud detection," *Procedia Computer Science*, vol. 170, pp. 26–31, 2020.

[15] L. Zhang and Y. Sun, "Machine learning techniques for financial fraud detection," *Journal of Emerging Technologies in Accounting*, vol. 17, no. 1, pp. 35–47, 2020.