Student Name: Pavithra .S.Y

Register Number: 510623104075

Institution: C.Abdul Hakeem College of Engineering and Technology

Department: Computer Science and Engineering

Date of Submission: 08/05/2025

Github Repository Link: https://github.com/Pavithra28122006/Fraud-Detection-.git

1. Problem Statement

Develop an AI-powered credit fraud detection and prevention system that identifies suspicious transactions in real-time, prevents fraudulent activities, and minimizes false positives while adapting to evolving threats. The system aims to reduce credit fraud losses, improve detection accuracy, and enhance customer experience through machine learning algorithms, anomaly detection, and real-time data analysis.

2. Project Objectives

The goal of the model building project is to develop an accurate machine learning model that detects and prevents credit fraud transactions in real-time, minimizing false positives while ensuring model interpretability and optimizing performance for large transaction volumes.

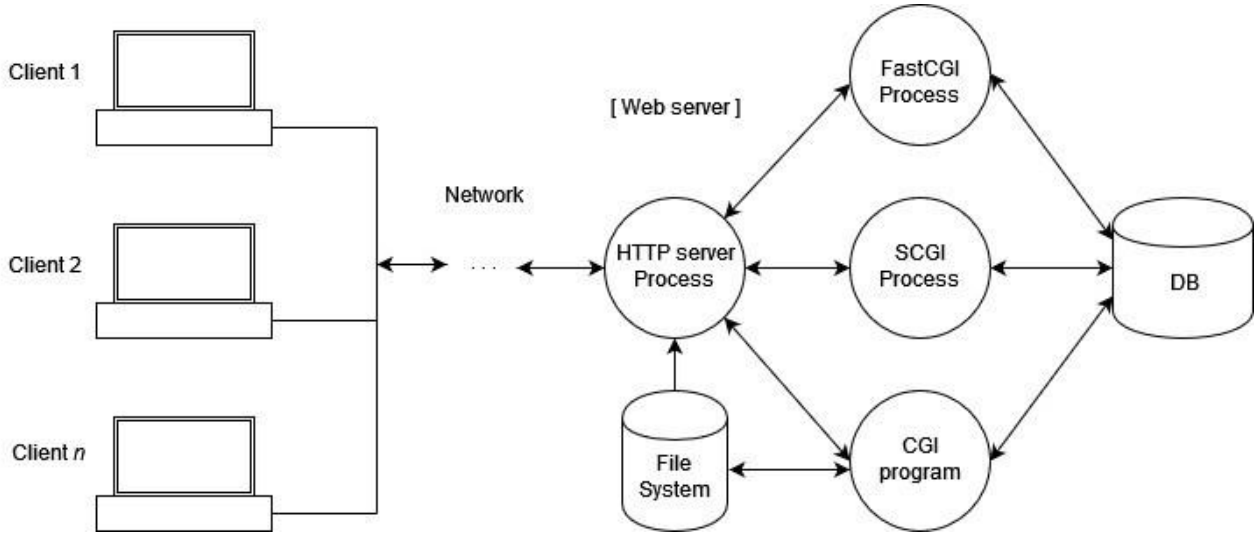
- AI-powered credit fraud detection aims to monitor

transactions in real time, identifying suspicious patterns with high accuracy and minimal false positives. It leverages machine learning models and behavioral analytics to detect anomalies and evolving fraud tactics. The system must be scalable, secure, and compliant with privacy regulations. Continuous learning and explainability ensure adaptability and trust in the detection process

- The AI-powered credit card fraud detection and prevention model aims to achieve high accuracy in detecting fraudulent transactions with minimal false positives, while providing interpretability into its predictions and ensuring real-world applicability to handle varying transaction volumes and evolving fraud tactics.

- After data exploration, the goals have evolved to focus on handling imbalanced datasets and incorporating domain expertise to enhance model performance.

3. Flowchart of the Project Workflow



4. Data Description

The dataset contains anonymized credit card transactions with features transformed using PCA and labels indicating fraud or legitimate activity. It supports training AI models for real-time fraud detection.

- The dataset for credit card fraud detection likely originates from sources such as Kaggle's Credit Card Fraud Detection dataset or anonymized datasets from financial institutions, featuring transformed features via PCA to maintain confidentiality while supporting AI model training for real-time fraud detection.

- Type of data: structured, unstructured, image, text, timeseries, etc. The

  dataset for credit card fraud detection consists of structured data,

  including transactional features like amount,

time, and location, as well as time-series

data based on transaction timestamps and sequences,

enabling effective analysis and pattern detection.

- The dataset contains tens of thousands to millions of transaction records, with around 20-30 features, including transaction amount, time, location, and PCA-transformed features, providing a robust foundation for training AI models to detect credit card fraud.

- The dataset for credit card fraud detection is likely dynamic, as new transactions are constantly being generated, and fraud patterns evolve over time. This requires the AI model to be adaptive and regularly updated to maintain effectiveness.

- The target variable is a binary label indicating whether a transaction is fraudulent (1 or Yes) or legitimate (0 or No), enabling the AI model to classify transactions accurately through supervised learning.

- Outliers in credit card fraud detection can be detected using statistical methods (e.g., Z-score, IQR) or machine learning techniques (e.g., Isolation Forest). Once identified, outliers can be treated by removal, transformation

- Data type conversion and consistency ensure that variables are in

suitable formats for analysis, such as converting categorical variables to numerical or standardizing date formats, enabling effective model training and fraud detection.

- Categorical variables in credit card fraud detection can be encoded using label encoding, which assigns numerical values, or one-hot encoding, which creates binary columns for each category, enabling effective model processing.

- Feature scaling techniques like normalization (e.g., Min-Max Scaler) or standardization (e.g., Standard Scaler) can be applied to ensure features are on a similar scale, improving model performance and stability in credit card fraud detection.
- Feature scaling techniques like normalization or standardization can be applied to ensure features are on a similar scale, improving model performance and stability in credit card fraud detection. This can be achieved using StandardScaler in Python:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler() df[['feature1', 'feature2']] =

scaler.fit_transform(df[['feature1', 'feature2']])
```

6. Exploratory Data Analysis (EDA)

EDA helps identify patterns and anomalies in transaction data, enabling AI models to detect and prevent credit card fraud effectively.

- Univariate Analysis: ○ Univariate analysis using histograms, box plots, and count plots helps understand transaction data distributions, identifying patterns and anomalies that inform AI-powered credit card fraud detection models, enhancing their effectiveness in preventing fraudulent transactions

- Bivariate/Multivariate Analysis:

- Bivariate and multivariate analysis techniques, including correlation matrices, pair plots, and scatter plots, help identify relationships and patterns between multiple features, informing feature selection and engineering for effective AI-powered credit card fraud detection models.
- Analysis of relationship between features and the target variable.

- Insights Summary:
  - Highlight patterns, trends, and interesting observations.
  - Mention which features may influence the model and why.

7. Feature Engineering

- To enhance AI-powered credit card fraud detection, features like transaction velocity, geolocation inconsistency, merchant category risk score, and behavioral deviation score can be created to improve model accuracy.

- For AI-powered credit card fraud detection, combining or splitting columns like date and time can help identify patterns and

anomalies. Extracting date parts (day, month, year)

and time parts (hour, minute) can improve model accuracy.

●For AI-powered credit card fraud detection, techniques like binning, polynomial features, and ratios can be used to enhance model accuracy and detect anomalies.

- To optimize AI-powered credit card fraud detection, dimensionality reduction techniques like Principal Component Analysis (PCA) can be applied. This helps reduce feature complexity, improve model performance, and prevent overfitting, ultimately enhancing fraud detection accuracy.

- feature justification involves adding relevant features like transaction amount, location, and time, while removing irrelevant or redundant ones. This ensures the model focuses on key indicators of fraud, improving accuracy and reducing false positives.

8. Model Building

- Machine learning models like Logistic Regression and Random Forest for credit card fraud detection. These models can effectively classify transactions as legitimate or fraudulent. Additional models like Decision Trees and KNN can also be explored for improved performance.

- The models were selected due to their suitability for binary classification (Logistic Regression, Decision Tree), handling complex interactions (Random Forest), and anomaly detection (KNN), making them effective for credit card fraud detection.

- Split the dataset into training and testing sets (e.g., 80% for training and 20% for testing) with stratification to

maintain the same proportion of legitimate and fraudulent transactions in both sets, ensuring reliable model evaluation and performance assessment.

- Train the models on the training set and evaluate their performance on the testing set using metrics like accuracy, precision, recall, F1-score, and AUC-ROC to assess their effectiveness in detecting fraudulent transactions.

- Key metrics for credit card fraud detection include accuracy, precision, recall, and F1-score, which evaluate model performance in detecting fraudulent transactions accurately.
- R^2 scoreIn AI-driven credit card fraud detection, regression metrics like MAE, RMSE, and R^2 score help evaluate prediction accuracy. They ensure reliable fraud risk estimation, enhancing transaction security.

9. Visualization of Results & Model Insights

- key evaluation metrics and visualizations include confusion matrices, ROC curves, feature importance plots, and residual plots. These help assess model performance, identify areas for improvement, and optimize fraud detection.

- visual comparisons like ROC curves and precision-recall curves help evaluate model performance. These visuals enable comparison of different models, identifying the most effective approach for accurate fraud detection.

- For AI-powered credit card fraud detection, interpreting top features involves analyzing feature importance scores, partial

dependence plots, and SHAP values. This helps identify key factors driving model predictions, providing insights into fraud patterns.

- Plots like ROC curves, feature importance, partial dependence, and SHAP values provide insights into AI-powered credit card fraud detection models. They show model performance, key drivers of predictions, and feature impacts, supporting conclusions about effectiveness and decision-making.

10. Tools and technologies Used

- Python is a popular choice for AI-powered credit card fraud detection due to its extensive libraries (TensorFlow, PyTorch, scikit-learn) and data analysis capabilities (Pandas, NumPy). R is also suitable, especially for statistical modeling and data visualization.

- For AI-powered credit card fraud detection, popular development environments include Google Colab, Jupyter Notebook, and VS Code, offering features like collaboration, interactivity, and versatility for model development and testing.
- key libraries include pandas and NumPy for data manipulation,

seaborn and matplotlib for visualization, and scikit-learn and XGBoost for

machine learning model development and evaluation.

- visualization tools like Plotly, Tableau, and Power BI provide

interactive and insightful visuals, helping stakeholders understand

complex data and make informed decisions to prevent fraud

11.Team Members and Contributions

- Data cleaning-Trisha .M
- EDA- Varuneesri.A
- Feature engineering -Pavithra.S.Y
- Model development - Sruthi.L
- Documentation and reporting-Swetha .J and Priyanka .P