**Guarding transactions with AI powered credit card fraud detection and prevention**

**Student name : Krishnapriya S**

**Register number: 732323106025**

**Institution: SSM College Of Engineering**

**Department: BE. Electronics And Communication Engineering**

**Date Of Submission: 23.04.2025**

**1.Problem statement**

#### Credit card fraud is a serious issue causing huge financial losses and reducing trust in digital transactions. Traditional methods often fail to detect new and evolving fraud techniques. With the rise in online payments, the need for smarter, faster fraud detection is critical. This project aims to use AI to identify and prevent fraudulent transactions in real-time. Solving this problem will protect users and strengthen financial system security.

**2.Objectives of the projects**

#### The project aims to develop an AI system for real-time credit card fraud detection. It will focus on accurately identifying fraudulent transactions while reducing false positives. The system will adapt to evolving fraud patterns and provide actionable insights for financial institutions. Key outcomes include improved security, reduced financial losses, and enhanced customer trust. Ultimately, the project aims to offer an effective tool for preventing fraud in digital payments.

**3.Scope of the project**

#### The project will focus on analyzing features like transaction time, amount, and user behavior to detect fraud. It will build machine learning models to identify fraudulent transactions accurately. Key features include anomaly detection, real-time predictions, and visual dashboards. The project will use the static Kaggle credit card fraud dataset. Limitations include no access to real-time or private financial data and deployment only in a test or demo environment. The project will rely on specific tools like Python, scikit-learn, and Streamlit.

**4.Data sources**

#### The project will use the Kaggle Credit Card Fraud Detection Dataset, which is publicly available. This static dataset contains transaction details, including both legitimate and fraudulent transactions. It includes anonymized features such as transaction amount and time. The dataset will be downloaded once for training and testing the AI model. Since it is not updated in real-time, the system will be developed based on historical data, with future integration possible for live transaction feeds in a production environment.

**5.High-level methodology**

#### The project will begin by collecting data from the Kaggle Credit Card Fraud Detection Dataset. After cleaning the data by handling missing values, duplicates, and encoding categorical variables, we’ll perform Exploratory Data Analysis (EDA) to uncover patterns and relationships. Feature engineering will include creating new features like transaction time or scaling numerical values. For model building, we’ll experiment with algorithms like Logistic Regression, Random Forest, and Neural Networks. Model evaluation will focus on metrics such as precision, recall, and F1 score, followed by visualizations of key findings. Finally, the model will be deployed via a web app or dashboard for real-time fraud detection.

**6.Tools and technologies**

#### For this project, I will use Python as the primary programming language. The development will take place in Jupyter Notebook or Google Colab for an interactive coding environment. Key libraries will include pandas and numpy for data processing, matplotlib and seaborn for visualization, and scikit-learn and XGBoost for machine learning models. TensorFlow or Keras may also be used for neural networks. For deployment, Streamlit or Flask will be considered to create a web app or dashboard for real-time fraud detection.

**7.Team members and roles**

#### [Krishnapriya.S]: Project Lead – Responsible for data collection, model building, and deployment.

#### [Lavanya.R]: Data Analyst – Handles data cleaning, EDA, and visualization.

#### [Kalimuthu.M]: Model Developer – Focuses on feature engineering and model evaluation.

#### [Logesh.K]: UI Developer – Designs and implements the web interface for deployment.

#### 

#### 