****

**An**

**Assessment Report**

on

**“Credit Card Fraud Detection”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSE(AI&ML)**

By

**Group Members** :- Saurabh(202401100400169), Rahul(202401100400150)

Sankalp(202401100400164), Ravi(202401100400154),

Ram Ji(202401100400152)

**Under the supervision of**

“Abhishek Shukla”

**KIET Group of Institutions, Ghaziabad**

## 1. Introduction

Credit card fraud is a major issue in the financial industry, with substantial financial losses incurred annually. This project aims to detect fraudulent transactions using unsupervised machine learning, specifically the Isolation Forest algorithm. By leveraging transaction features such as amount, time, and anonymized inputs, this model attempts to flag anomalies that may indicate fraud.

## 2. Problem Statement

To develop an anomaly detection model using unsupervised learning techniques to identify fraudulent credit card transactions without using labeled fraud data in training.

## 3. Objective

- Preprocess the dataset including scaling of numerical features.  
- Train the Isolation Forest model for anomaly detection.  
- Evaluate model performance using classification metrics.  
- Visualize confusion matrix for better interpretation of results.

## 4. Methodology

Data Collection:  
- The dataset used is 'creditcard.csv', a standard benchmark dataset for fraud detection.  
  
Data Preprocessing:  
- Scaled the 'Amount' and 'Time' columns using StandardScaler.  
- Dropped original unscaled 'Amount' and 'Time' columns.  
- Rearranged the dataset to place 'Class' as the target variable at the end.  
  
Model Building:  
- Used Isolation Forest with a contamination rate of 0.001 (based on the percentage of frauds in the dataset).  
- Converted predictions to binary values: 1 = Fraud, 0 = Genuine.  
  
Model Evaluation:

- Evaluated predictions using accuracy, precision, recall, F1-score.  
- Displayed confusion matrix using seaborn heatmap.

# 5. Data Preprocessing

The 'Amount' and 'Time' columns were standardized using StandardScaler for better model convergence. The original columns were then dropped to avoid scale inconsistencies. The target column 'Class' was moved to the end to improve readability.

# 6. Model Implementation

Isolation Forest was selected as it is well-suited for anomaly detection in high-dimensional datasets. It isolates anomalies by randomly selecting features and then randomly selecting split values. The model was trained using 100 estimators with a contamination threshold of 0.001.

# 7. Evaluation Metrics

The performance of the model was evaluated using:

- Accuracy: Proportion of total predictions that were correct.

- Precision: Ratio of correctly predicted frauds to all predicted frauds.

- Recall: Ratio of correctly predicted frauds to all actual frauds.

- F1 Score: Harmonic mean of precision and recall.

- Confusion Matrix: Visualized using heatmap for interpretability

# 8. Results and Analysis

The model successfully identified fraudulent transactions with reasonable accuracy and F1 score despite being unsupervised. The confusion matrix showed a strong distinction between genuine and fraudulent transactions. However, recall can still be improved for catching more fraudulent cases.

## 9. Code

# Step 1: Import Libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.ensemble import IsolationForest

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score, precision\_score, recall\_score, f1\_score

import warnings

warnings.filterwarnings('ignore')

# Step 2: Load Dataset

df = pd.read\_csv('creditcard.csv')

print("✅ Data Loaded. Shape:", df.shape)

# Step 3: Data Summary

print("\n🔍 Summary Statistics:")

print(df.describe())

print("\n📊 Class Distribution:")

print(df['Class'].value\_counts())

# Step 4: Data Preprocessing

scaler = StandardScaler()

df['scaled\_amount'] = scaler.fit\_transform(df['Amount'].values.reshape(-1, 1))

df['scaled\_time'] = scaler.fit\_transform(df['Time'].values.reshape(-1, 1))

# Drop original 'Time' and 'Amount'

df.drop(['Time', 'Amount'], axis=1, inplace=True)

# Rearrange columns

columns = [col for col in df.columns if col != 'Class'] + ['Class']

df = df[columns]

# Step 5: Isolation Forest Model

X = df.drop('Class', axis=1)

y = df['Class']

model = IsolationForest(n\_estimators=100, contamination=0.001, random\_state=42)

model.fit(X)

# Predict (-1 for fraud, 1 for normal)

pred = model.predict(X)

pred = [1 if x == -1 else 0 for x in pred] # Convert to 1 = Fraud, 0 = Normal

# Step 6: Evaluation

print("\n✅ Evaluation Metrics:\n")

print(confusion\_matrix(y, pred))

print(classification\_report(y, pred, target\_names=["Genuine", "Fraud"]))

print(f"Accuracy : {accuracy\_score(y, pred) \*100:.4f}")

print(f"Precision : {precision\_score(y, pred):.4f}")

print(f"Recall : {recall\_score(y, pred):.4f}")

print(f"F1 Score : {f1\_score(y, pred):.4f}")

# Step 7: Confusion Matrix Heatmap

cm = confusion\_matrix(y, pred)

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=["Genuine", "Fraud"], yticklabels=["Genuine", "Fraud"])

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()

## 10. Results

## 

## 

## 11. Conclusion

This project demonstrates how Isolation Forest, an unsupervised algorithm, can be effectively used for fraud detection in credit card transactions. While the current model offers decent performance, future work can involve ensemble methods or hybrid models for better accuracy and recall.

# 12. References

- scikit-learn documentation  
- pandas and numpy documentation  
- Seaborn visualization library  
- Credit card fraud detection datasets and research papers

- Dataset Link: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud