1. Introduction

With the rise of online financial transactions, fraud detection has become an essential task for banks and financial institutions. Fraudulent activities result in billions of dollars of losses globally each year. Detecting fraudulent transactions in real time is challenging due to the imbalanced nature of transaction data, where fraud cases represent a very small fraction of overall transactions.

Traditional rule-based systems are not sufficient, as fraud patterns evolve rapidly. Machine Learning (ML) provides automated and adaptive techniques to learn hidden patterns and detect anomalies that indicate fraudulent activity. By leveraging ML, institutions can reduce financial losses, improve security, and build trust with customers.

2. Objective

The objective of this project is to:
- Build a machine learning model to classify online transactions into fraudulent and non-fraudulent categories.
- Demonstrate preprocessing, feature encoding, model training, and evaluation steps.
- Evaluate the system using metrics such as confusion matrix, precision, recall, F1-score, and accuracy.
- Highlight the challenges of working with imbalanced datasets.

3. Dataset Description

The dataset used for this project contains synthetic transaction records inspired by real-world online payments. Each row represents a single transaction with the following features:

- step: Unit of time in hours (1 step = 1 hour).
- type: Type of online transaction (CASH_IN, CASH_OUT, DEBIT, PAYMENT, TRANSFER).
- amount: Transaction amount.
- nameOrig: Unique ID of the customer initiating the transaction.
- oldbalanceOrg: Customer balance before the transaction.
- newbalanceOrig: Customer balance after the transaction.
- nameDest: Unique ID of the recipient.
- oldbalanceDest: Recipient balance before the transaction.
- newbalanceDest: Recipient balance after the transaction.
- isFraud: Target variable (1 = Fraudulent transaction, 0 = Normal transaction).

4. Methodology

The project workflow includes:

- Data Preprocessing
- Removed high-cardinality ID columns (nameOrig, nameDest) since they do not provide predictive value.
- Encoded the categorical feature type using Label Encoding.
- Train-Test Split
- Split the dataset into 80% training and 20% testing data.
- Stratified sampling ensured class balance between train and test sets.
- Model Selection
- Used Random Forest Classifier as the main model for its robustness, ability to handle large datasets, and strong performance on imbalanced data.

- Logistic Regression was tested as a baseline.
- Evaluation Metrics
- Accuracy: Overall correctness of predictions.
- Confusion Matrix: Breakdown of correct and incorrect classifications by class.
- Precision: Proportion of predicted frauds that were actual frauds.
- Recall: Proportion of actual frauds detected.
- F1-score: Balanced measure of precision and recall, more suitable for imbalanced data.

5. Model Implementation

Example implementation using scikit-learn:

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
import pandas as pd

# Load dataset
df = pd.read_csv("fraud_dataset.csv")

# Preprocessing
df = df.drop(columns=["nameOrig", "nameDest"])
df["type"] = LabelEncoder().fit_transform(df["type"])

# Split dataset
X = df.drop(columns=["isFraud"])
y = df["isFraud"]
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.2, stratify=y, random_state=42
)

# Train model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Evaluate
y_pred = model.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

6. Results and Observations

The Random Forest Classifier achieved high accuracy, but accuracy alone is not sufficient due to dataset imbalance.

- Confusion Matrix showed most non-fraud transactions were correctly classified, but detecting fraud remains challenging.
- Precision indicated fewer false alarms when predicting fraud.
- Recall reflected the ability to identify fraud correctly.
- F1-score provided a balanced evaluation, making it more suitable for imbalanced datasets.

Key Insights:
- Fraud detection systems should prioritize recall to minimize undetected frauds, while maintaining reasonable precision.
- Random Forest outperformed Logistic Regression on imbalanced data.
- Further improvements can be achieved using SMOTE (Synthetic Minority Oversampling), anomaly detection, or deep learning methods.

7. Conclusion

This project demonstrated the use of machine learning for online fraud detection effectively. The Random Forest Classifier provided a strong baseline with promising results.

Key takeaways:
- Fraud detection is a crucial ML application in the finance sector.
- Imbalanced datasets require tailored handling techniques.
- Ensemble methods, neural networks, and anomaly detection approaches can further enhance system performance.
- Real-time fraud detection systems must balance accuracy, computational efficiency, and interpretability.