CREDIT CARD FRAUD DETECTION

INTRODUCTION

Credit card payments, cardless purchases via Google Pay, PhonePe, Samsung Pay, and PayPal are all common in daily life. The detection of fraud, which results in a significant financial loss each year, is a current issue. The scam is predicted to reach double digits by 2020 if it keeps going in this direction. The fact that the card is no longer physically necessary to complete the exchange has led to an increase in extortion transactions. The economy is impacted emotionally by fraud discovery
[1].

Fraud detection is
essential and necessary in this approach. To combat this issue, financial institutions must use a variety of fraud detection tools. But over time, scammers find ways to get around the strategies put in place by business owners. Fraud detection continues to

increase and it remains a serious worry in society despite all the preventive measures taken by financial institutions, strengthening of law, and government doing their best efforts to eradicate fraud detection. Credit cards are frequently used in the expansion of Internet commerce, mobile applications, and particularly in web-based transactions. The use ofonline transactions and payments is made simpler and easier with the use of credit cards. Fraudulent transactions have a significant impact on businesses. Machine learning techniques are frequently utilised and have grown significantly in significance in many fields where spam classifiers safeguard our mail id. The fraud detection systems pick up on extraction aspects and aid in keeping the fraud detection under control [3].

Researchers have been considering several strategies for detecting fraudulent activity in credit card transactions as they build models based on artificial intelligence, data mining, fuzzy logic, and machine learning. Detecting credit card fraud is a common problem that is also quite tough to tackle. We used machine learning to build the credit card fraud detection in our suggested system. the development of machine learning methods. Fraud detection techniques that successfully use machine learning have been identified. During online transaction operations, a substantial amount of data is sent, resulting in either a true or false conclusion. Features are built inside the example fraudulent datasets. These are data points, including the credit card's country of origin, age and account balance of the customer. There are hundreds of traits, and each one contributes, in varied degrees, to the likelihood of fraud. Be aware that the machine's artificial intelligence, which is fuelled by the training set, determines the extent to which each attribute contributes to the fraud score rather than a fraud analyst. Therefore, in terms of card fraud, the weighting of a transaction that uses a credit card for fraud purposes will be equal if the use of cards to commit fraud is demonstrated to be high. detection of credit card fraud utilising by utilising the classification and regression techniques, machine learning is accomplished [2]. To categorise the fraudulent card transactions made either online or offline, we use supervised learning algorithms like the Random Forest algorithm. An evolved variation of the decision tree is the random forest. The accuracy and efficiency of Random Forest are higher than those of other machine learning techniques. By just selecting a portion of the feature space at each split, random forest seeks to lessen the correlation problem that was previously highlighted.

II. PROPOSED SYSTEM

In proposed System, we are applying random forest algorithm for classification of the credit card dataset. Random Forest is an algorithm for classification and regression. Summarily, it is a collection of decision tree classifiers. Random forest has advantage over decision tree as it corrects the habit of over fitting to their training set. A subset of the training set is sampled randomly so that to train each individual tree and then a decision tree is built, each node then splits on a feature selected from a random subset of the full feature set [5]. Even for large data sets with many features and data instances training is extremely fast in random forest and because each tree is trained independently of the others. The Random Forest algorithm has been found to provide a good estimate of the generalization error and to be resistant to over fitting.

2.1 Implementation of Random Forest Algorithm

The graphic below shows how the random forest method is executed. Figure 1 illustrates the several processes that must be taken. The first step is the obligation to obtain and store the information. The acquired data are in the form of an excel sheet's worth of data. Data exploration involves checking the full data set and eliminating any extraneous information that may be there. However, using both the train and test data sets, the pre-processed data is then further processed using the random forest algorithm. The obtained results are confirmed to be the outcome of an illegal and fraudulent transaction.

2.1.1 Collection of Data Sets

Here, gathering the data sets is the first step. A variety of techniques, such crawling or application programme interfaces, can be used to obtain the data sets. Aspects including the customer's name, email address, card number, payment method, cellphone number, bank account number, and pin number must be included in the data sets [6]. The data set is used for analysis after being collected using the aforementioned qualities. The main difference between the training data set and the test set is that the training set has labels while the test set does not. Regression analysis is used to first train the data set, and the random forest approach is then used to test it.

2.1.2 Analysis of Data

Data preparation is followed by analysis using a variety of techniques. The functions in this data can be used to train the data and build classification prediction models. The data sets are separated into training sets and test sets using the random forest algorithm. It is made up of a variety of techniques, including data splitting and data preparation, both of which are carried out using the resampling approach.

2.1.3 Reporting Results

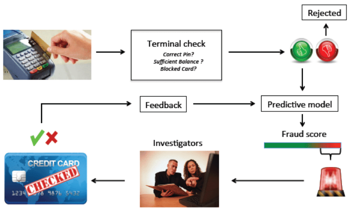
Credit Card Transaction Fraud Detection using 193carrying out analysis. The main difference between the training data set and the test data set is that the test set lacks labels while the training data set has labels. Regression analysis is used to first train the data set, and the random forest approach is then used to test it. Data Analysis, Data preparation is followed by analysis using a variety of techniques. The functions in this data can be used to train predictive classification algorithms. The data sets are separated into training set and remaining test set using the random forest technique. It includes a variety of tools, including those for dividing data and preparing data using the resampling method.

III.Data Processing

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithms of the same type i.e., multiple decision trees, resulting in a forest of trees, hence the

name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.
Use Case Diagram
Upload Credit Card Dataset
Generate Train and Test Model
Run Random Forest Algorithm
Detect Fraud From Test Data
Clean And Fraud Transaction.

Organize your selected data by formatting, cleaning and sampling from it. Three common data pre-processing steps are[7]:
Use



Formatting: The data you've chosen might not be in a form that you can work with; for example, it might be in a relational database but you'd prefer it in a flat file, or it might be in a proprietary file format but you'd prefer it in a relational database or text file.

Cleaning: Data cleaning is the process of removing or replacing missing data. There can be data instances that are insufficient and lack the information you think you need to address the issue.
These
occurrences might need to be eliminated.

Sampling: There can be far more available selected data than you need to deal with. Algorithms may take much longer to perform on bigger amounts of data, and their computational and memory requirements may also increase. Before thinking about the entire dataset, you can take a smaller representative sample of the chosen data that may be much faster for exploring and testing ideas.

IV.RESULTS

Credit Card Fraud Detection Using Random Forest Tree

Random Forest generate 99.78% percent accuracy while building model on train and test data. Now click on 'Detect Fraud From Test Data' button to upload test data and to predict whether test data contains normal or fraud transaction.

Credit Card Fraud Detection Using Random Forest Tree

In above screen beside each test data application will display output as whether transaction contains cleaned or fraud signatures. Now click on 'Clean & Fraud Transaction Detection Graph' button to see total test transaction with clean and fraud signature in graphical format. See below screen.

Credit Card Fraud Detection Using Random Forest Tree

In above graph we can see total test data and number of normal and fraud transaction detected. In above graph x-axis represents type and y-axis represents count of clean and fraud transaction.

Developing a credit card fraud detection system involves multiple steps, and it's a complex process. Here's a simplified example of how you can start with coding the initial part in Python using scikit-learn and pandas. This is part 1 of the development process and covers data preparation and feature engineering:

Coding:

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Load your dataset (replace 'credit_card_data.csv' with your dataset)
data = pd.read_csv('credit_card_data.csv')

# Explore the data (e.g., check for missing values, data types)
print(data.info())

# Feature engineering and preprocessing
# In this step, you might want to create new features or transform existing ones.

# Split the data into training and testing sets
X = data.drop('fraud_label', axis=1)
y = data['fraud_label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train a classifier (Random Forest in this example)
clf = RandomForestClassifier()
clf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(X_test)
```

```
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Classification Report \n{class_report}')
```

This is a basic starting point. In subsequent parts, you can explore more advanced techniques like feature scaling, dimensionality reduction, and fine-tuning your model for better fraud detection performance. Additionally, you'll need to set up a system for real-time monitoring and continuous improvements.

In practice, you might want to explore other algorithms, perform hyperparameter tuning, and deal with imbalanced datasets. Additionally, feature scaling and engineering are crucial steps in building an effective fraud detection model.

Developing a credit card fraud detection system is a comprehensive process that involves several steps. Below is a high-level overview of the development process:

Data Collection: Gather historical credit card transaction data, which includes both legitimate and fraudulent transactions. This dataset will serve as the foundation for training and testing your fraud detection model.

Data Preprocessing:

Handle Missing Values: Check for missing data and decide how to handle it (impute or remove).

Data Transformation: Convert categorical features to numerical format using techniques like one-hot encoding.
Feature Scaling: Normalize or standardize numerical features to have similar scales.
Exploratory Data Analysis (EDA):

Explore the dataset to understand its characteristics and distributions.
Identify class imbalance (fraudulent transactions are usually rare compared to legitimate ones).
Feature Engineering:

Create new features that might help in fraud detection.
Extract relevant information from the data, such as transaction timestamps, location data, etc.
Data Splitting: Divide the dataset into training and testing sets. It's common to use a large portion of the data for training and a smaller part for testing.

Model Selection:

Choose an appropriate machine learning algorithm for fraud detection. Common choices include Random Forest, Logistic Regression, and Gradient Boosting.
Consider ensemble methods for improved performance.
Model Training: Train the selected model on the training data.

Model Evaluation:

Evaluate the model's performance using metrics like accuracy, precision, recall, F1-score, and AUC-ROC.
Due to class imbalance, pay particular attention to the selection of evaluation metrics.
Hyperparameter Tuning:

Optimize the model's hyperparameters using techniques like grid search or random search to improve performance.
Cross-Validation: Perform cross-validation to ensure the model's generalizability and robustness.

Model Deployment:

Deploy the trained model into a production environment where it can continuously monitor credit card transactions.
Set up real-time or batch processing for incoming transaction data.
Monitoring and Alerts:

Implement a system to monitor the model's predictions in real-time.
Set up alert mechanisms for potential fraud cases.
Feedback Loop:

Continuously update and retrain the model with new data to adapt to changing fraud patterns.
Regulatory Compliance:

Ensure compliance with legal and regulatory requirements, including data privacy and security.
Documentation and Reporting:

Document the entire development process, including model details and evaluation results.
Report findings and actions taken in the event of a fraud incident.
Remember that credit card fraud detection is an ongoing process, and the model's performance needs to be continuously monitored and improved to stay ahead of evolving fraud patterns.