High Level Design (HLD)

Fraud Transaction Detection

Nitin kushwaha

Mohammad Danish

Diksha Sharma

Document Version Control

Date Issued	Version	Description	Author
19th sep 2021	1.1	Initial HLD v1.0	Diksha Sharma Nitin kushwaha Mohammad Danish

Contents

Abstract

Credit card frauds are easy and friendly targets. E-commerce and many other online sites have increased the online payment modes, increasing the risk for online frauds. Increase in fraud rates, researchers started using different machine learning methods to detect and analyse frauds in online transactions. The main aim of the paper is to design and develop a novel fraud detection method for Streaming Transaction Data, with an objective, to analyse the past transaction details of the customers and extract the behavioural patterns. Where cardholders are clustered into different groups based on their transaction amount.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like: Security
  Reliability
  Maintainability
  Portability
  Reusability
    - Application
      compatibility o
      Resource utilization
    - Serviceability

1.2 Scope

The HI-D documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HI-D uses non-technical

to mildly-technical terms which should be understandable to the administrators of the system.

.

1.3 Definations

Terms	Description
FTD Fraud	Transcation Detection
IDE	Integrated development system
Heroku	Deployment server

2 General Description

2.1 Product Perspective

The fraud transaction detection based system is a machine learning-based object detection model which will help us to detect the anomalies in the society and take the necessary action.

2.2 PROPOSED SOLUTION

In this we developed a novel method for fraud detection, where customers are grouped based on their transactions and extract behavioural patterns to develop a profile for every cardholder. Then different classifiers are applied on three different groups later rating scores are generated for every type of classifier. This dynamic changes in parameters lead the system to adapt to new cardholder's transaction behaviours timely. Followed by a feedback mechanism to solve the problem of concept drift. We observed that the Matthews Correlation Coefficient was the better parameter to deal with imbalance dataset. MCC was not the only solution. By applying the SMOTE, we tried balancing the dataset, where we found that the classifiers were performing better than before. The

other way of handling imbalance dataset is to use one-class classifiers like one-class SVM. We finally observed that Logistic regression, decision tree and random forest are the algorithms that gave better results.

2.3 Technical Requirements

We should be able to log every activity done by the user.

- The System identifies at every transaction.

2.4 Data Requirements

In the Fraud Transaction detection system we use the kaggle dataset.

.

2.5 Deployment

1. Heroku

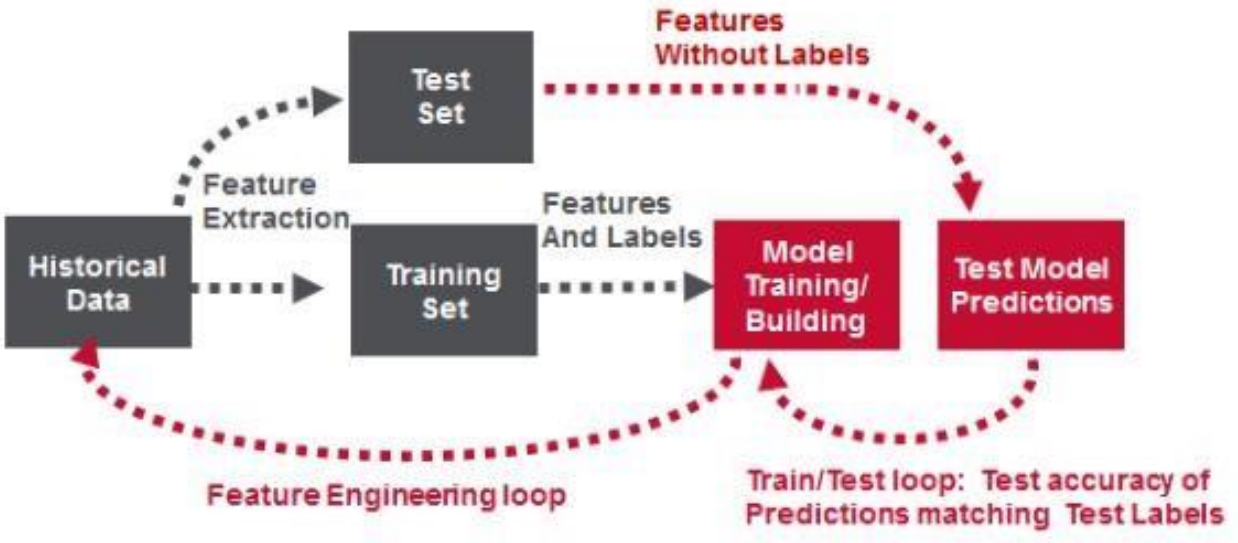
2.6 Tools Used

Python programming language and frameworks such as NumPy, Pandas, Scikitlearn are used to build the whole model.

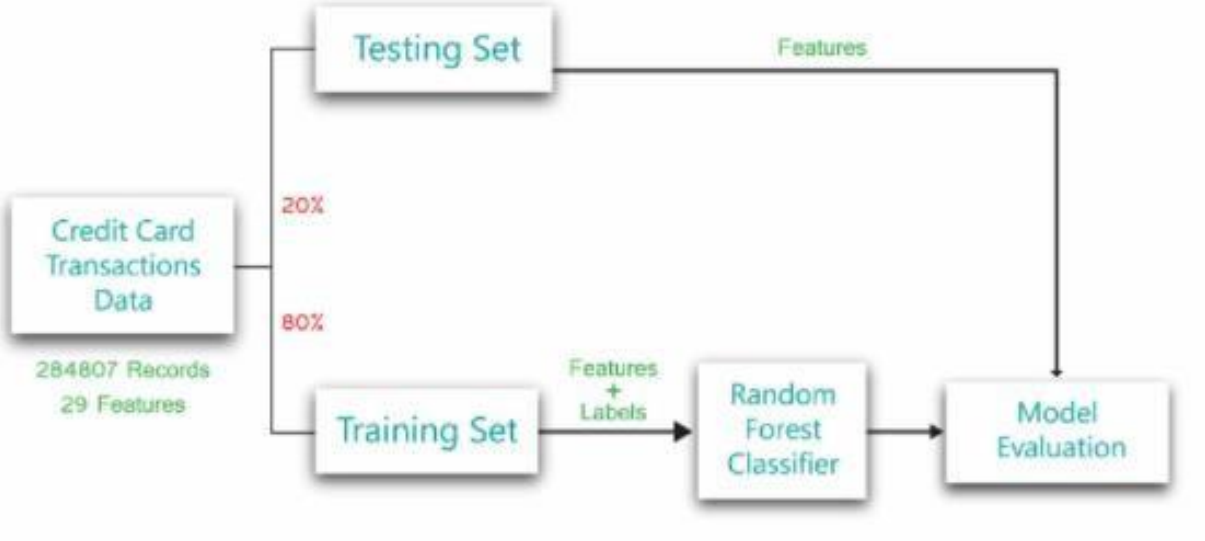
- Spyder is used as IDE
- Heroku is used as model deployment
- For visualization of the plots, Matplotlib, and Plotly are used.

3 Design Details

Front End	HTML/CSS/
Backend	Python Flask
Database	Kaggle
Deployment	Heroku



3.1 Process Flow



3.2 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. Firstly, we use clustering method to divide into different cgroups based on their transaction amount.
2. Extract some features from dataset behavioural patterns.

3. Followed by the average amount in the window and even the time elapsed.

4 Formula

In our proposed system we use the following formulae to evaluate, accuracy and precision are never good parameters for evaluating a model. But accuracy and precision are always considered as the base parameter to evaluate any model.

The Matthews Correlation Coefficient (MCC) is a machine learning measure which is used to check the balance of the binary (two-class) classifiers. It takes into account all the true and false values that is why it is generally regarded as a balanced measure which can be used even if there are different classes,

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

4.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

4.2 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

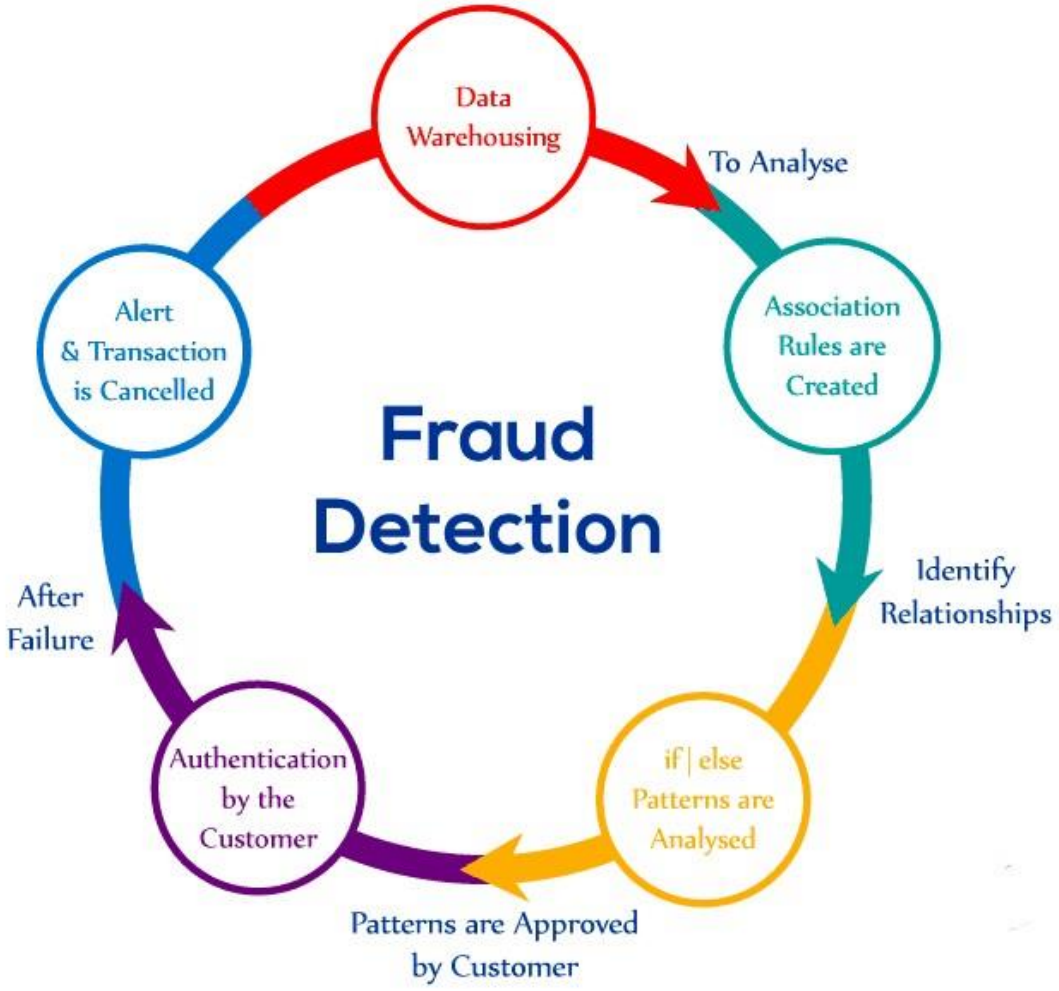
4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

4.3 Deployment

1. Heroku

5 Model training/validation workflow



6 Experimental Results

Out[48]:

	SVM	ksvm	navie bayes	decision tree	random forest	real values
0	1	0	1	1	1	1
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
...	...	...	...	...	...	...
85438	0	0	0	0	0	0
85439	0	0	0	0	0	0
85440	0	0	0	0	0	0
85441	0	0	0	0	0	0
85442	0	0	0	0	0	0

85443 rows × 6 columns

7 Conclusion

In this project we developed a novel method for fraud detection, where customers are grouped based on their transactions and extract behavioural patterns to develop a profile for every cardholder. Then different classifiers are applied on three different groups later rating scores are generated for every type of classifier. This dynamic changes in parameters lead the system to adapt to new cardholder's transaction behaviours timely. Followed by a feedback mechanism to solve the problem of concept drift. We observed that the Matthews Correlation Coefficient was the better parameter to deal with imbalance dataset. MCC was not the only solution. By applying the SMOTE, we tried balancing the dataset, where we found that the classifiers were performing better than before. The other way of handling imbalance dataset is to use one-class classifiers like one-class SVM. We finally observed that Logistic regression, decision tree and random forest are the algorithms that gave better results.