Credit Card Fraud Detection Using Machine Learning

Submitted To:
Mr. Mayur Dev Sewak
General Manager, Operations
Eisystems Services

&

Ms. Mallika Srivastava
Trainer, Programming & Algorithms
Eisystems Services

Submitted By:
Rishi Mani Tripathi
Student, ML Batch

CONTENT TABLE

ABSTRACT

It is vital that credit card companies are able to identify fraudulent credit card transactions so that customers are not charged for items that they did not purchase. Such problems can be tackled with Data Science and its importance, along with Machine Learning, cannot be overstated. This project intends to illustrate the modelling of a data set using machine learning with Credit Card Fraud Detection. The Credit Card Fraud Detection Problem includes modelling past credit card transactions with the data of the ones that turned out to be fraud. This model is then used to recognize whether a new transaction is fraudulent or not. Our objective here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud classifications. Credit Card Fraud Detection is a typical sample of classification. In this process, we have focused on analysing and preprocessing data sets as well as the deployment of multiple anomaly detection algorithms such as Local Outlier Factor and Isolation Forest algorithm on the PCA transformed Credit Card Transaction data.

Keywords — Credit card fraud, applications of machine learning, data science, isolation forest algorithm, local outlier factor, automated fraud detection.

OBJECTIVES
INTRODUCTION

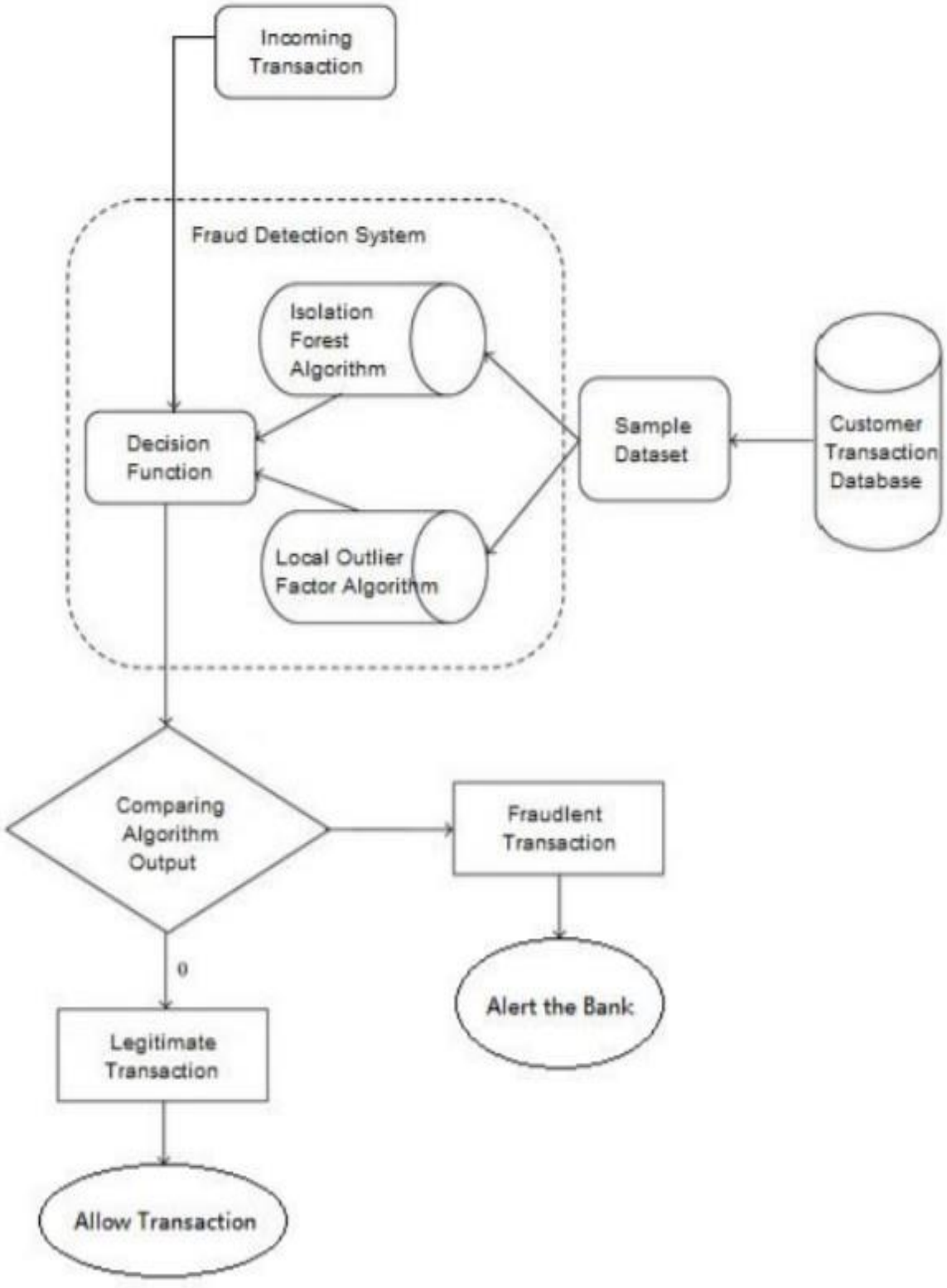

Figure 1 – Flow Chart : Credit Card Fraud Detection System

'Fraud' in credit card transactions is unauthorized and unwanted usage of an account by someone other than the owner of that account. Necessary prevention measures can be taken to stop this abuse and the behaviour of such fraudulent practices can be studied to minimize it and protect against similar occurrences in the future. In other words, Credit Card Fraud can be defined as a case where a person uses someone else's credit card for personal reasons while the owner and thecard issuing authorities are unaware of the fact that the card is being used. Fraud detection involvesmonitoring the activities of populations of users in order to estimate, perceive or avoid objectionable behaviour, which consist of fraud, intrusion, and defaulting. This is a very relevant problem that demands the attention of communities such as machine learning and data science where the solution to this problem can be automated. This problem is particularly challenging from the perspective of learning, as it is characterized by various factors such as class imbalance. The number of valid transactions far outnumber fraudulent ones. Also, the transaction patterns often change their statistical properties over the course of time.

These are not the only challenges in the implementation of a real-world fraud detection system, however. In real world examples, the massive stream of payment requests is quickly scanned by automatic tools that determine which transactions to authorize. Machine learning algorithms are employed to analyse all the authorized transactions and report the suspicious ones. These reports are investigated by professionals who contact the cardholders to confirm if the transaction was genuine or fraudulent. The investigators provide feedback to the automated system which is used totrain and update the algorithm to eventually improve the fraud-detection performance over time.

Fraud detection methods are continuously developed to defend criminals in adapting to theirfraudulent strategies.
These frauds are classified as:

- Credit Card Frauds: Online and Offline
- Card Theft
- Account Bankruptcy
- Device Intrusion
- Application Fraud
- Counterfeit Card
- Telecommunication Fraud

Some of the currently used approaches to detection of such fraud are:

- Artificial Neural Network
- Fuzzy Logic
- Genetic Algorithm
- Logistic Regression
- Decision tree
- Support Vector Machines
- Bayesian Networks
- Hidden Markov Model
- K-Nearest Neighbour

REVIEW

Fraud acts as the unlawful or criminal deception intended to result in financial or personal benefit. It is a deliberate act that is against the law, rule or policy with an aim to attain unauthorized financial benefit. Numerous literatures pertaining to anomaly or fraud detection in this domain have been published already and are available for public usage. A comprehensive survey conducted by Clifton Phua and his associates have revealed that techniques employed in this domain include data mining applications, automated fraud detection, adversarial detection. In another paper, Suman, Research Scholar, GJUS&T at Hisar HCE presented techniques like Supervised and Unsupervised Learning for credit card fraud detection. Even though these methods and algorithms fetched an unexpected success in some areas, they failed to provide a permanent and consistent solution to fraud detection. A similar research domain was presented by Wen-Fang YU and Na Wang where they used Outlier mining, Outlier detection mining and Distance sum algorithms to accurately predict fraudulent transactions in an emulation experiment of credit card transaction data set of one certain commercial bank. Outlier mining is a field of data mining which is basically used in monetary and internet fields. It deals with detecting objects that are detached from the main system i.e. the transactions that aren't genuine. They have taken attributes of the customer's behaviour and based on the value of those attributes they've calculated that distance between the observed value of that attribute and its predetermined value. Unconventional techniques such as hybrid data mining/complex network classification algorithm is able to perceive illegal instances in an actual card transaction data set, based on network reconstruction algorithm that allows creating representations of the deviation of one instance from a reference group have proved efficient typically on medium sized online transactions. There have also been efforts to progress from a completely new aspect. Attempts have been made to improve the alert feedback interaction in case of fraudulent transactions. In case of fraudulent transaction, the authorised system would be alerted and a feedback would be sent to deny the ongoing transaction. Artificial Genetic Algorithm, one of the approaches that shed new light in this domain, countered fraud from a different direction. It proved accurate in finding out the fraudulent transactions and minimizing the number of false alerts. Even though it was accompanied by a classification problem with variable misclassification costs.

SYSTEM REQUIREMENTS

Hardware Requirements:
Processor : Pentium i3 or higher.
RAM : 4 GB or higher.
Hard Disk Drive : 20 GB (free).
Peripheral Devices : Monitor, Mouse and Keyboard.

Software Requirements: Operating system : Windows 8/10.IDE Tool : Jupyter Notebook
Coding Language : Python 3.8
APIs : Numpy, Pandas,PySpark, Matplotlib

DATA FLOW DIAGRAM

The DFD is used as a communication tool between system and user.it is a simple representation ofthe complete project process.

Transaction detection activity follows three phases.

1. Data exploration
2. Data preprocessing
3. Data classifications

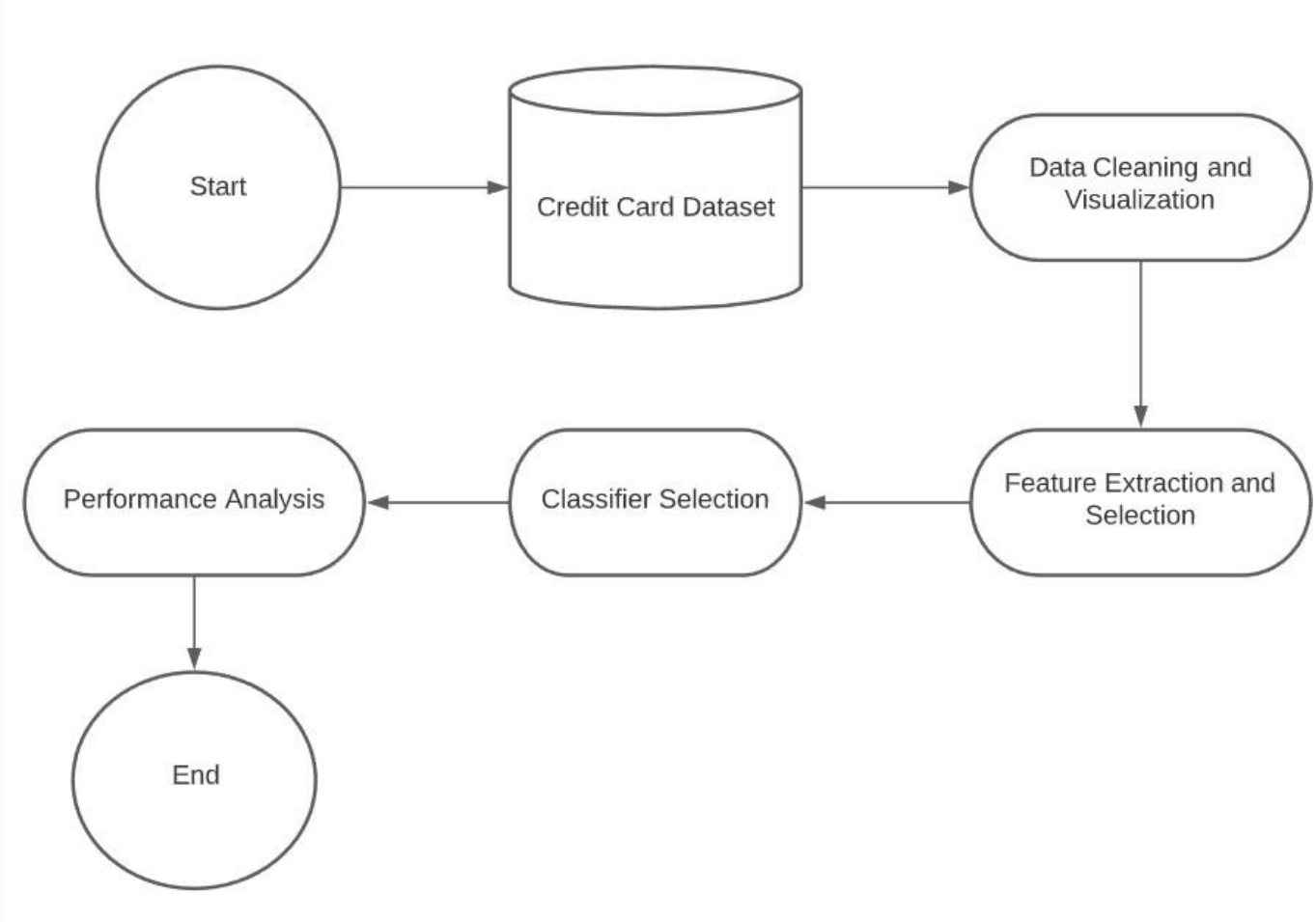

Figure 2 - Data Flow Diagram(DFD)

METHODOLOGY

The approach that this paper proposes, uses the latest machine learning algorithms to detect anomalous activities, called outliers. The basic rough architecture diagram can be represented with the following figure:

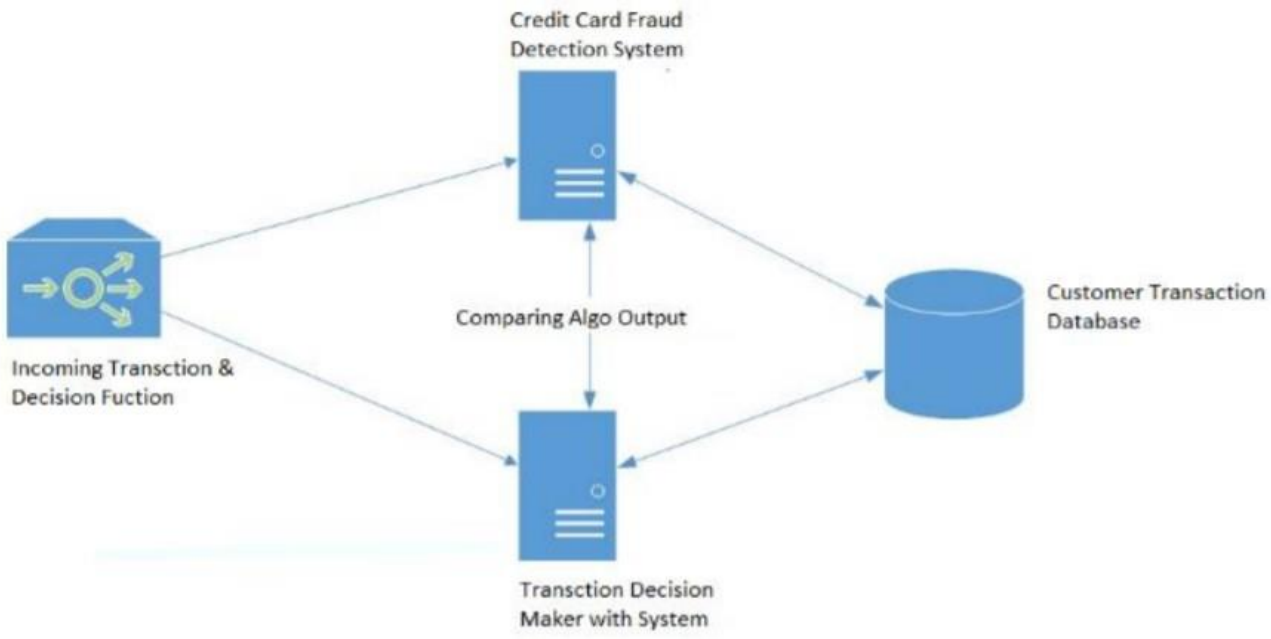

Figure 3 - Basic Architecture Diagram(Credit Card Fraud Detection)

When looked at in detail on a larger scale along with real life elements, the full architecture diagram can be represented as follows:

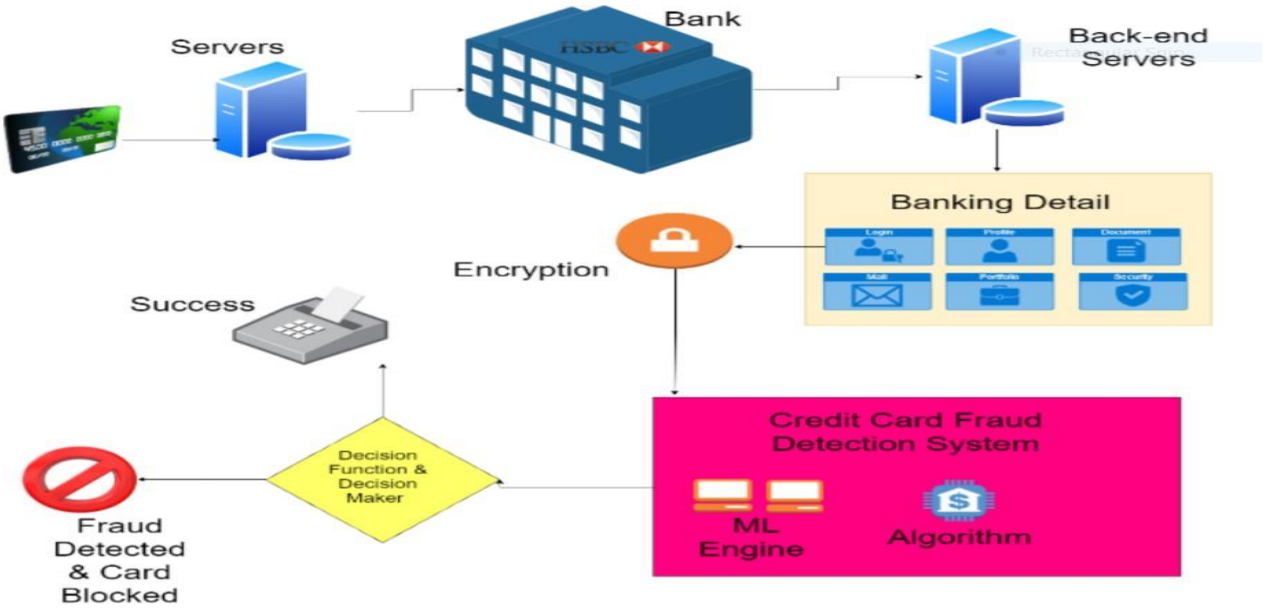

Figure 4 - Full Architecture Diagram(Credit Card Fraud Detection)

First of all, we obtained our dataset from Kaggle, a data analysis website which provides datasets. Inside this dataset, there are 31 columns out of which 28 are named as v1-v28 to protect sensitivedata. The other columns represent Time, Amount and Class. Time shows the time gap between thefirst transaction and the following one. Amount is the amount of money transacted. Class 0represents a valid transaction and 1 represents a fraudulent one. We plot different graphs to checkfor inconsistencies in the dataset and to visually comprehend it:

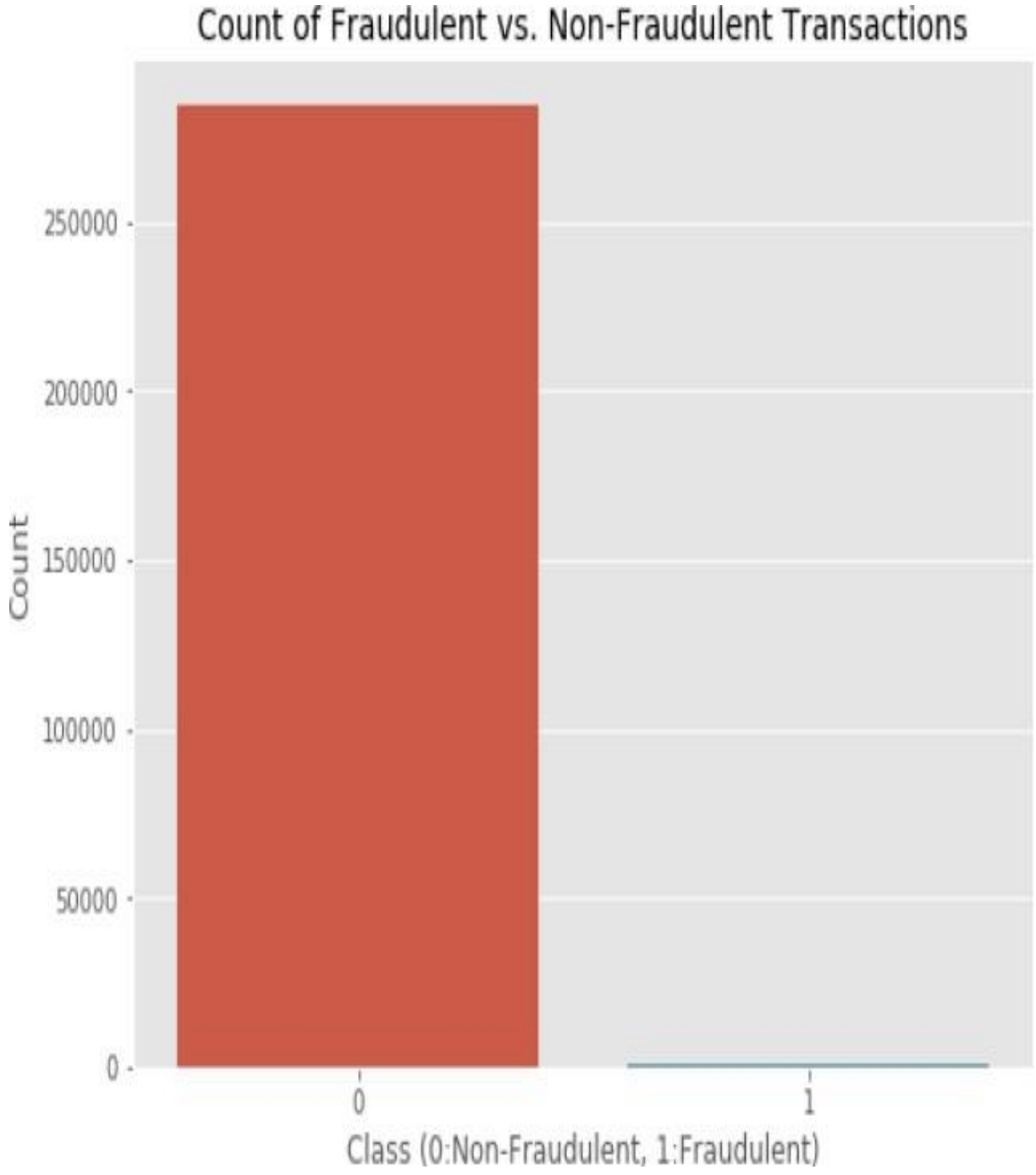

Figure 5 - Fraudulent VS Non-Fraudulent Transactions

This graph shows that the number of fraudulent transactions is much lower than the legitimate ones.

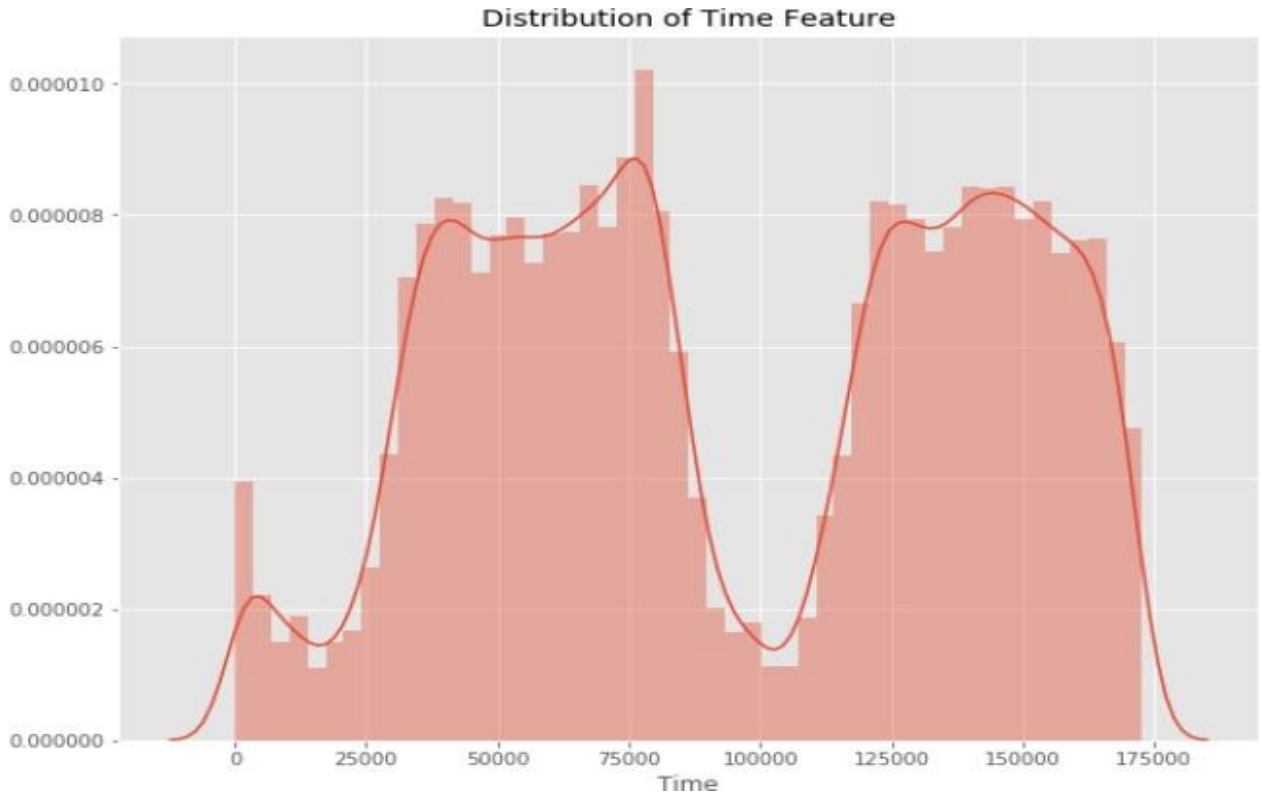

Figure 6 - Time Feature Distribution

This graph shows the times at which transactions were done within two days. It can be seen that theleast number of transactions were made during night time and highest during the days.

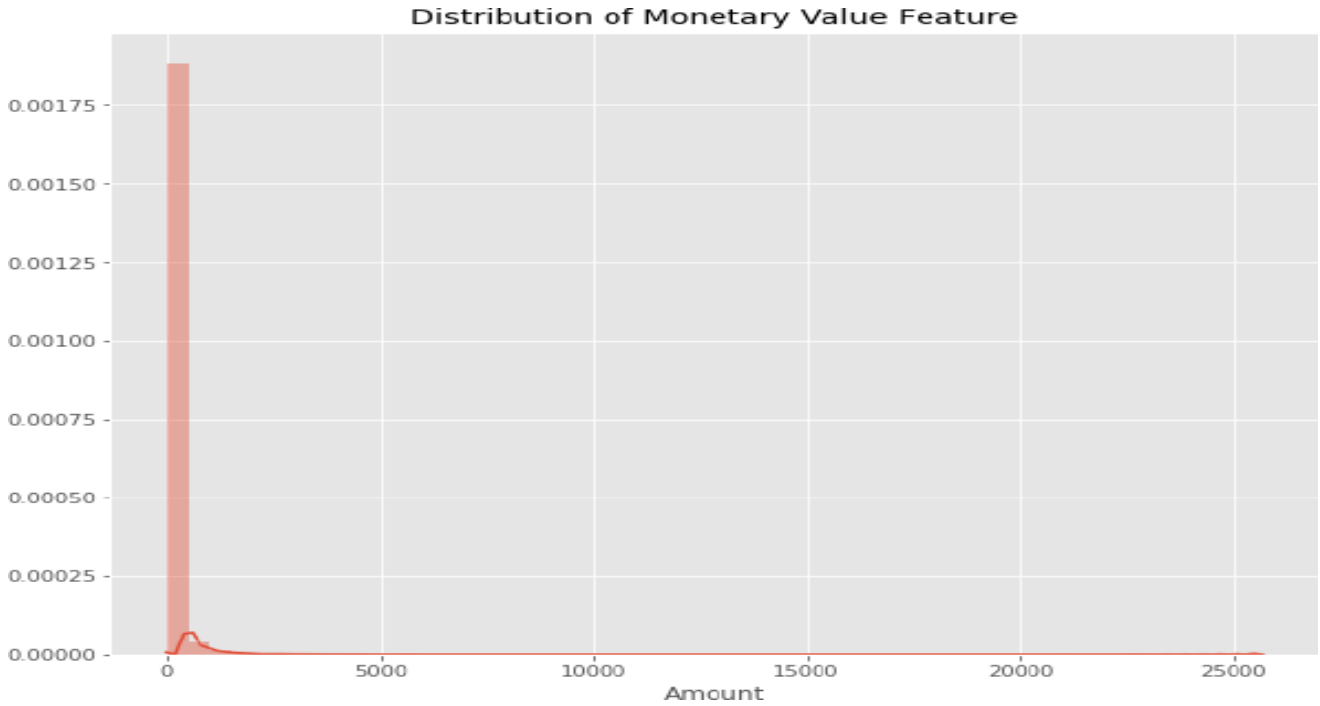

Figure 7 - Monetary Value Distribution

This graph represents the amount that was transacted. A majority of transactions are relativelysmall and only a handful of them come close to the maximum transacted amount.
After checking this dataset, we plot a histogram for every column. This is done to get a graphical representation of the dataset which can be used to verify that there are no missing values in thedataset. This is done to ensure that we don't require any missing value imputation and the machine learning algorithms can process the dataset smoothly.

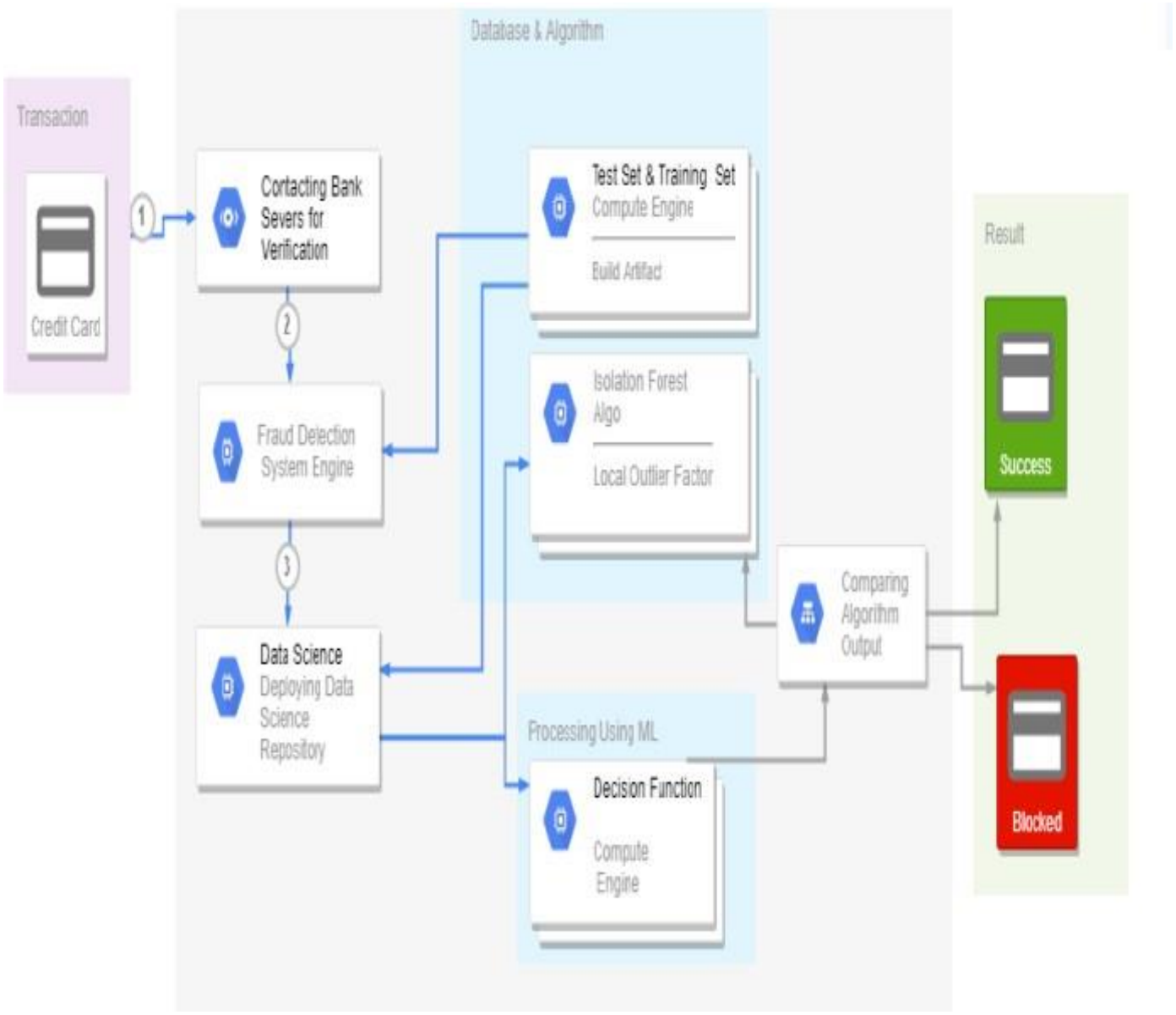

Figure 8 - Flowchart : Credit Card Fraud Detection System

After this analysis, we plot a heatmap to get a coloured representation of the data and to study thecorrelation between our predicting variables and the class variable. This heatmap is shown below:

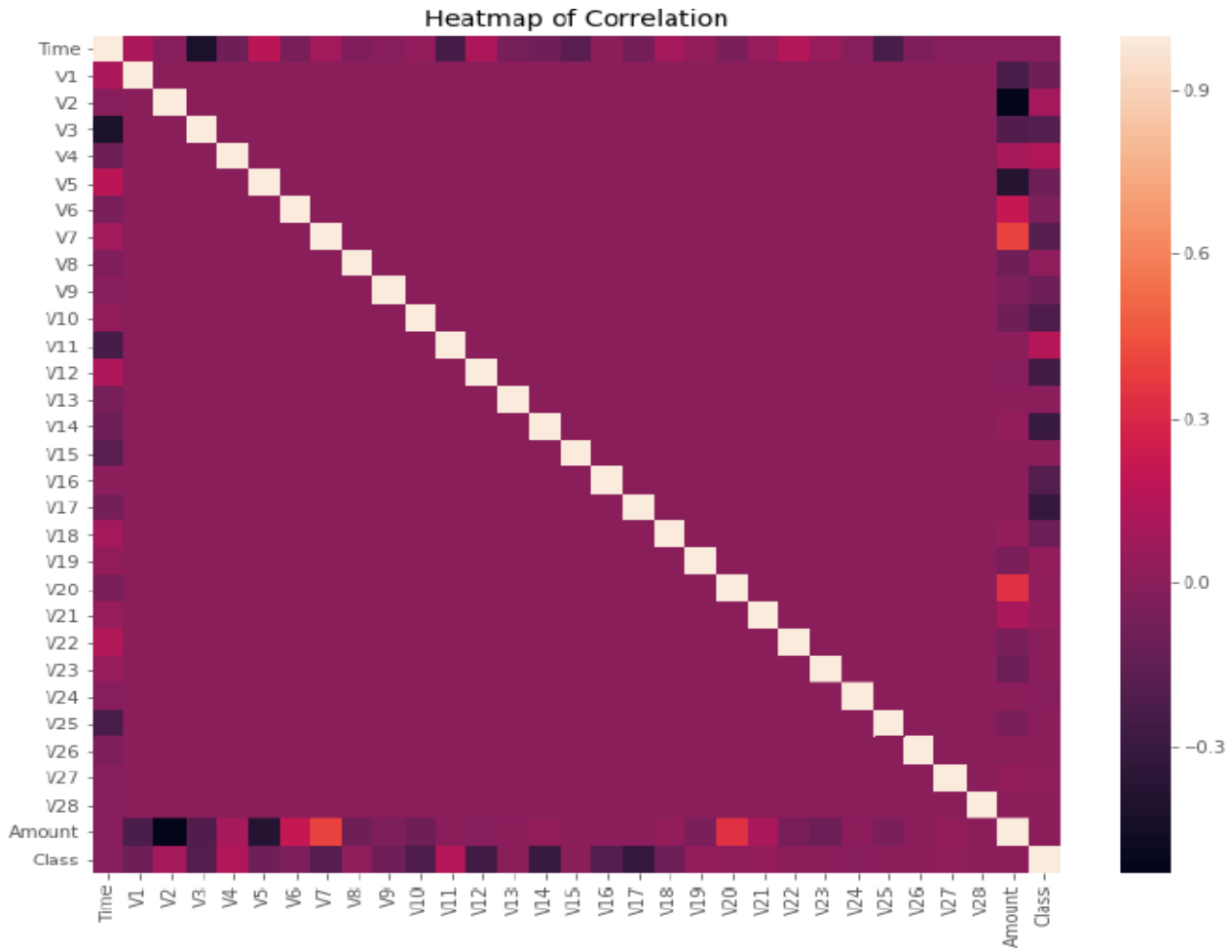

Figure 9 - HeatMap showing correlation

The dataset is now formatted and processed. The time and amount column are standardized and the Class column is removed to ensure fairness of evaluation. The data is processed by a set of algorithms from modules. The following module diagram explains how these algorithms work together: This data is fit into a model and the following outlier detection modules are applied on it:

- Local Outlier Factor
- Isolation Forest Algorithm

These algorithms are a part of sklearn. The ensemble module in the sklearn package includes ensemble-based methods and functions for the classification, regression and outlier detection. This free and open-source Python library is built using NumPy, SciPy and matplotlib modules which provides a lot of simple and efficient tools which can be used for data analysis and machine learning. It features various classification, clustering and regression algorithms and is designed to interoperate with the numerical and scientific libraries. We've used the Jupyter Notebook platform tomake a program in Python to demonstrate the approach that this paper suggests. This program canalso be executed on the cloud using Google Collab platform which supports all python notebook files.Detailed explanations about the modules with pseudocodes for their algorithms and output graphs are given as follows:

ALGORITHMS

A. LOCAL OUTLIER FACTOR

It is an Unsupervised Outlier Detection algorithm. 'Local Outlier Factor' refers to the anomaly scoreof each sample. It measures the local deviation of the sample data with respect to its neighbours.
More precisely, locality is given by k-nearest neighbours, whose distance is used to estimate thelocal data. The pseudocode for this algorithm is written as:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import LocalOutlierFactor

np.random.seed(42)

# Generate train data
X = 0.3 * np.random.randn(100, 2)
# Generate some abnormal novel observations
X_outliers = np.random.uniform(low=-4, high=4, size=(20, 2))
X = np.r_[X + 2, X - 2, X_outliers]

# fit the model
clf = LocalOutlierFactor(n_neighbors=20)
y_pred = clf.fit_predict(X)
y_pred_outliers = y_pred[200:]

# plot the level sets of the decision function
xx, yy = np.meshgrid(np.linspace(-5, 5, 50), np.linspace(-5, 5, 50))
Z = clf._decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
```

On plotting the results of Local Outlier Factor algorithm, we get the following figure:

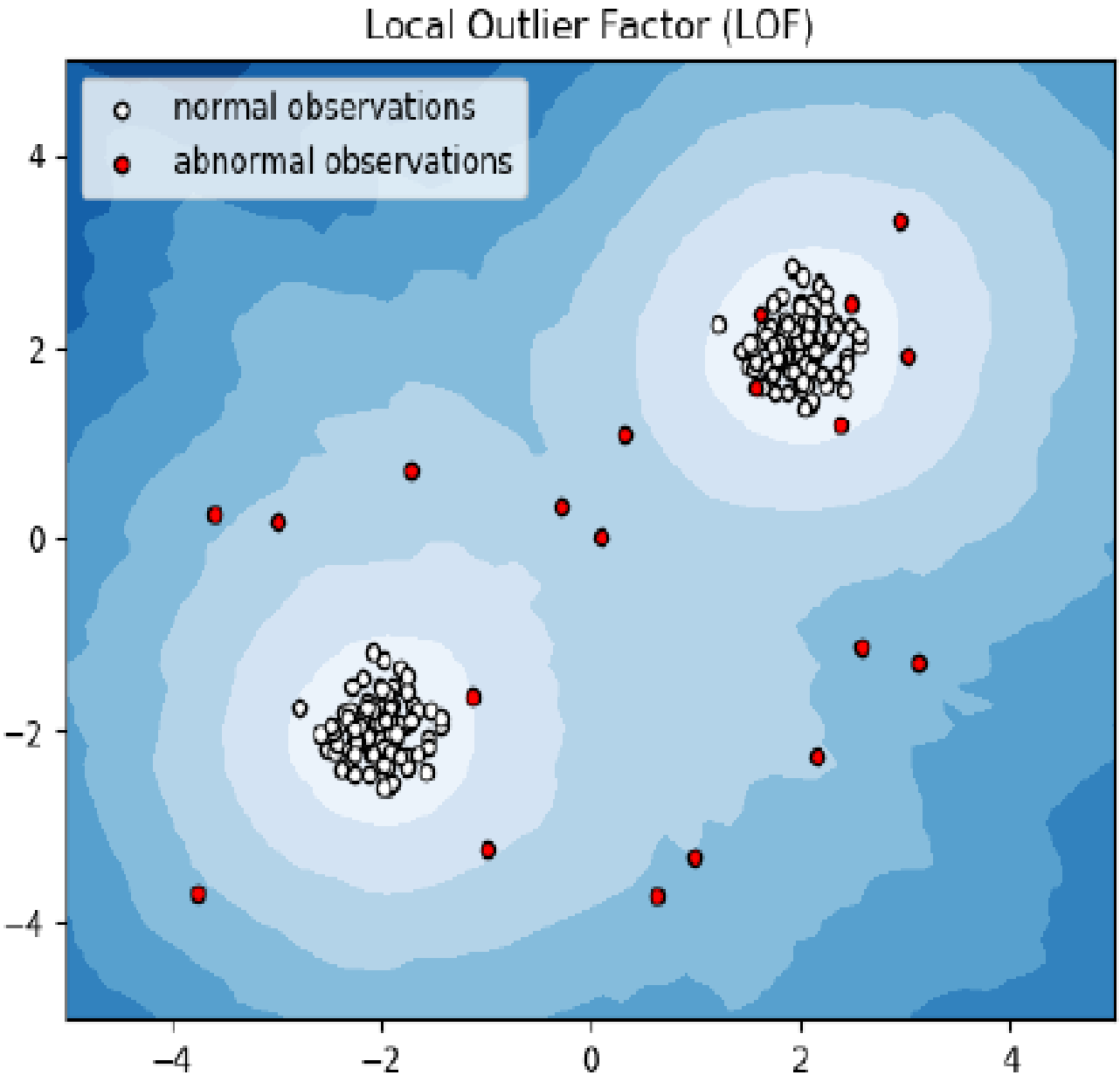

Figure 9 - Local Outlier Algorithm(Results)

By comparing the local values of a sample to that of its neighbours, one can identify samples that are substantially lower than their neighbours. These values are quite amanous and they are considered as outliers. As the dataset is very large, we used only a fraction of it in our tests to reduce processing times. The final result with the complete dataset processed is also determined and is given in the results section of this paper.

B. ISOLATION FOREST ALGORITHM

The Isolation Forest 'isolates' observations by arbitrarily selecting a feature and then randomly selecting a split value between the maximum and minimum values of the designated feature. Recursive partitioning can be represented by a tree, the number of splits required to isolate a sample is equivalent to the path length root node to terminating node. The average of this path length gives a measure of normality and the decision function which we use. The pseudocode for this algorithm can be written as:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import IsolationForest

rng = np.random.RandomState(42)

# Generate train data
X = 0.3 * rng.randn(100, 2)
X_train = np.r_[X + 2, X - 2]
# Generate some regular novel observations
X = 0.3 * rng.randn(20, 2)
X_test = np.r_[X + 2, X - 2]
# Generate some abnormal novel observations
X_outliers = rng.uniform(low=-4, high=4, size=(20, 2))

# fit the model
clf = IsolationForest(behaviour='new', max_samples=100,
                      random_state=rng, contamination='auto')
clf.fit(X_train)
y_pred_train = clf.predict(X_train)
y_pred_test = clf.predict(X_test)
y_pred_outliers = clf.predict(X_outliers)

# plot the line, the samples, and the nearest vectors to the plane
xx, yy = np.meshgrid(np.linspace(-5, 5, 50), np.linspace(-5, 5, 50))
Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
```

On plotting the results of Isolation Forest algorithm, we get the following figure:

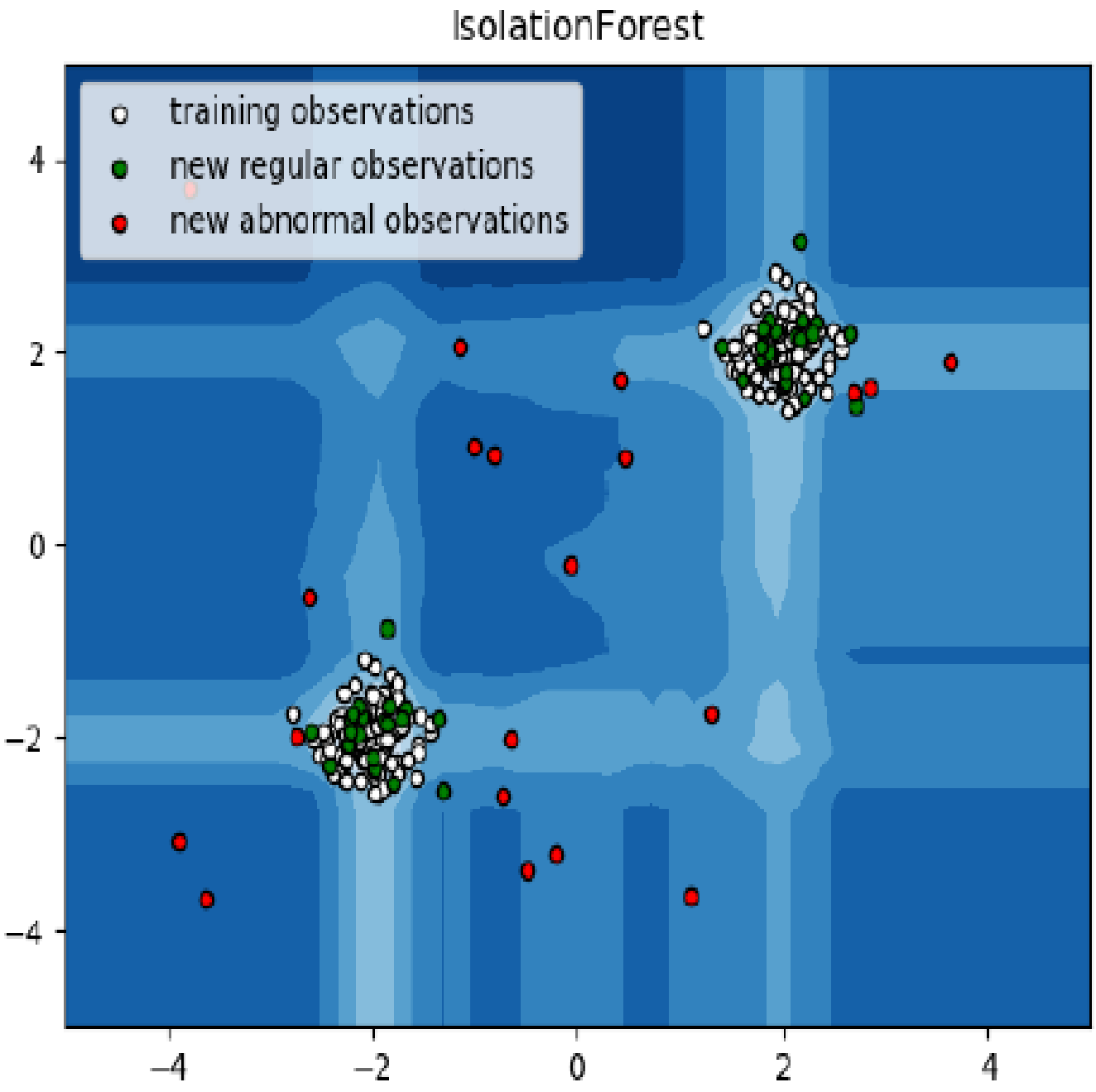

Figure 10 - Isolation Forest Algorithm(Results)

Partitioning them randomly produces shorter paths for anomalies. When a forest of random trees mutually produces shorter path lengths for specific samples, they are extremely likely to be anomalies. Once the anomalies are detected, the system can be used to report them to the concerned authorities. For testing purposes, we are comparing the outputs of these algorithms to determine their accuracy and precision.

IMPLEMENTATION

This idea is difficult to implement in real life because it requires the cooperation from banks, which aren't willing to share information due to their market competition, and also due to legal reasons and protection of data of their users. Therefore, we looked up some reference papers which followed similar approaches and gathered results. As stated in one of these reference papers: "This technique was applied to a full application data set supplied by a German bank in 2006. For banking confidentiality reasons, only a summary of the results obtained is presented below. After applying this technique, the level 1 list encompasses a few cases but with a high probability of being fraudsters. All individuals mentioned in this list had their cards closed to avoid any risk due to their high-risk profile. The condition is more complex for the other list. The level 2 list is still restricted adequately to be checked on a case by case basis. Credit and collection officers considered thathalf of the cases in this list could be considered as suspicious fraudulent behaviour. For the last list and the largest, the work is equitably heavy. Less than a third of them are suspicious. In order to maximize the time efficiency and the overhead charges, a possibility is to include a new element in the query; this element can be the five first digits of the phone numbers, the email address, and the password, for instance, those new queries can be applied to the level 2 list and level 3 list."

PROJECT CODE LINK

https://github.com/Devils-judge/Credit-Card-Fraud-Detection/blob/main/credit_card_detection%20.ipynb

RESULTS

The code prints out the number of false positives it detected and compares it with the actual values. This is used to calculate the accuracy score and precision of the algorithms. The results along with the classification report for each algorithm is given in the output as follows, where class 0 means thetransaction was determined to be valid and 1 means it was determined as a fraud transaction. This result matched against the class values to check for false positives.

```
Classifier IF:
Number of Errors: 173
Accuracy: 99.79752346621 1%

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85306
           1       0.36      0.37      0.37       136

    accuracy                           1.00     85442
   macro avg       0.68      0.68      0.68     85442
weighted avg       1.00      1.00      1.00     85442

Classifier LOF:
Number of Errors: 273
Accuracy: 99.68048500737342%

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85306
           1       0.00      0.00      0.00       136

    accuracy                           1.00     85442
   macro avg       0.50      0.50      0.50     85442
weighted avg       1.00      1.00      1.00     85442
```

CONCLUSION

Credit card fraud is without a doubt an act of criminal dishonesty. This report has listed out the mostcommon methods of fraud along with their detection methods and reviewed recent findings in thisfield. This paper has also explained in detail, how machine learning can be applied to get betterresults in fraud detection along with the algorithm, pseudocode, explanation its implementation and experimentation results. While the algorithm does reach over 99.6% accuracy, its precision remainsonly at 28% when a tenth of the data set is taken into consideration. However, when the entiredataset is fed into the algorithm, the precision rises to 33%. This high percentage of accuracy is tobe expected due to the huge imbalance between the number of valid and number of genuinetransactions. Since the entire dataset consists of only two days' transaction records, it's only afraction of data that can be made available if this project were to be used on a commercial scale.Being based on machine learning algorithms, the program will only increase its efficiency over timeas more data is put into it.

FUTURE ENHANCEMENTS

While we couldn't reach our goal of 100% accuracy in fraud detection, we did end up creating a system that can, with enough time and data, get very close to that goal. As with any such project, there is some room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project. More room for improvement can be found in the dataset. As demonstrated before, the precision of the algorithms increases when the size of the dataset is increased. Hence, more data will surely make the model more accurate in detecting frauds and reduce the number of false positives. However, this requires official support from the banks themselves.

REFERENCES

1. “Credit Card Fraud Detection Based on Transaction Behaviour -by John Richard D. Kho, Larry A. Vea” published by Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017
2. CLIFTON PHUA1, VINCENT LEE1, KATE SMITH1 & ROSS GAYLER2 “ A Comprehensive Survey of Data Mining-based Fraud Detection Research” published by School of Business Systems, Faculty of Information Technology, Monash University, Wellington Road, Clayton, Victoria 3800, Australia
3. “Survey Paper on Credit Card Fraud Detection by Suman” , Research Scholar, GJUS&T Hisar HCE, Sonepat published by International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 3, March 2014
4. “Research on Credit Card Fraud Detection Model Based on Distance Sum – by Wen-Fang YU and Na Wang” published by 2009 International Joint Conference on Artificial Intelligence
5. “Credit Card Fraud Detection through Parenclitic Network Analysis- By Massimiliano Zanin, Miguel Romance, Regino Criado, and SantiagoMoral” published by Hindawi Complexity Volume 2018, Article ID 5764370, 9 pages
6. “Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy” published by IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 29, NO. 8, AUGUST 2018
7. “Credit Card Fraud Detection-by Ishu Trivedi, Monika, Mrigya, Mridushi” published by International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 1, January 2016
8. David J.Wetson,David J.Hand,M Adams,Whitrow and Piotr Jusczak “Plastic Card Fraud Detection using Peer Group Analysis” Springer, Issue 2008. International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERT V8I090031 (This work is licensed under a Creative Commons Attribution 4.0 International License.) Published by : www.ijert.org Vol. 8 Issue 09, September-2019 115