EasyChair Preprint

№ 5616

Credit Card Fraud Detection Using Machine Learning

Harshita Anand, Richa Gautam and Raman Chaudhry

May 26, 2021

Credit Card Fraud Detection using Machine Learning

School of Computer Science and Engineering
Galgotias University

Harshita Anand, Richa Gautam, Raman Chaudary
School of Computer Science and Engineering
Galgotias University

***Abstract*— Whenever we hear the word Credit Card the first thing that pops in our mind is the frauds that are associated with these cards. Credit card has become an indispensable part of our lives. Although a credit card has many advantages when used in a proper manner but damages can be caused to it by many fraudulent activities as well. But in today's advanced world these frauds can be detected with a vast knowledge of machine learning algorithms.**

The Credit Card Anomaly Detection Problem includes modeling past credit card transactions with the ones that turned out to be fraud. After the implementation of this model we can use it further to identify, a new transaction that is occurring as fraudulent or not. Basically our focus here is to detect 100% fraud transactions that is being occur by minimizing the incorrect fraud classification.

This detection process is a typical example of classifications.

This process involve the analysis and the pre-processing of data sets as well as the utilization of multiple Anomaly detection algorithms such as Local Outlier Factor, Super Vector Machine and many such relevant algorithms.

In today's world this is the major concern, which demands the attention of the fields such as Machine Learning, Artificial Intelligence, Deep Learning etc. where the solution of this issue can be automated.

Our aim is to predict the accuracy/precision of the fraud detection through different algorithms. Further this analysis can be used to implement the fraud detection model.

Keywords—Credit Card Fraud Classification, Fraud Detection Techniques, Python, Artificial Intelligence, Machine Learning Algorithm, Data Science, Dataset, Comparative Analysis.

I. INTRODUCTION

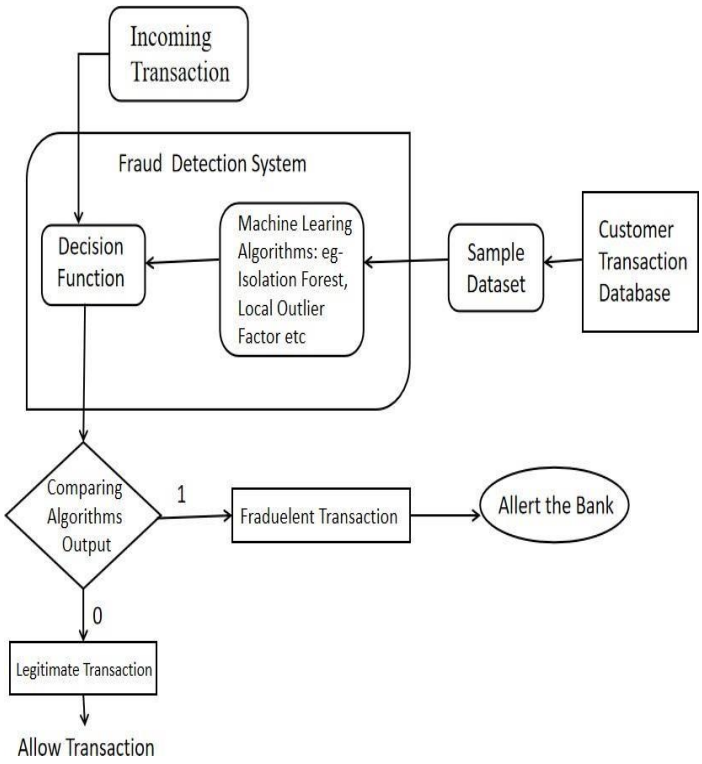
Credit card fraud is an overall term that can be used to define the fraud that may be carry out by any payment card such as credit card or debit card. The basic aim of these fraud is to purchase goodies without paying or to steal money from someone else's account.

The Payment Card Industry Data Security Standard (PCI DSS) is the data security standard created to help businesses process card payments securely and reduce card fraud. There is a rapid growth in the usage of Cards which has led to rise in the fraudulent activities.

The process of credit card fraud detection involve the analysis and the pre-processing of data sets as well as the utilization of multiple Anomaly detection algorithms such as Local Outlier Factor, Super Vector Machine and many such relevant algorithms. In today's world this is the major concern, which demands the attention of the fields such as Machine Learning, Artificial Intelligence, Deep Learning etc. where the solution of this issue can be automated.

Our aim is to predict the accuracy/precision of the fraud detection through different algorithms. Further this analysis can be used to implement the fraud detection model.

This problem is very challenging in terms of learning as it is characterized by many factors that makes it more challenging to solve. Moreover there are many more challenges associated with real-world fraud detection system.



The Anomaly detection methods are being developed to protect cards from criminals in adapting there fraudulent activities. These frauds are classified as:

- Credit Card Frauds can be Online and Offline.
- Now-a-days Card Theft is very common.
- Bankruptcy to accounts.
- Application related Fraud.
- Cloning of Card are very common these days.
- Many Fraudulent are done from Telecommunication.

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

- Fuzzy Logic
- Logistic Regression
- Decision tree
- Support Vector Machines
- Random Forest tree
- Isolation tree

II. LITERATURE REVIEW

Fraud act 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 transaction 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 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 transaction.

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 transaction.

In case of fraudulent transaction, the authorised system would be alerted and a feedback would be sent to deny the ongoing transaction.

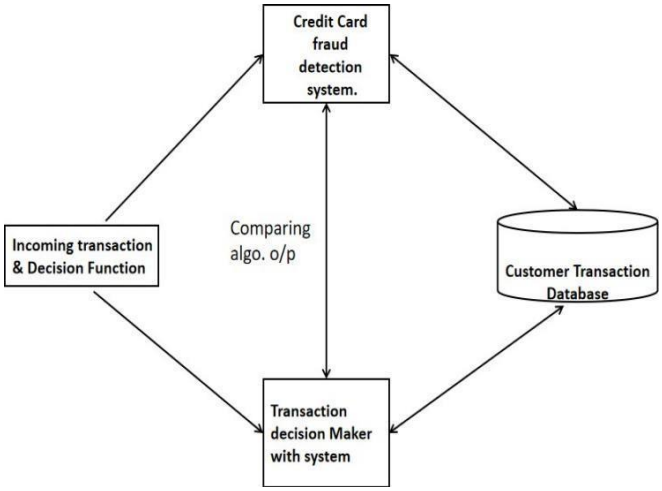
Artificial Genetic Algorithm, one of the approaches that shed new light in this domain, countered fraud from a different direction.

It proved accurate in finding out the fraudulent transactions and minimizing the number of false alerts. Even though, it was accompanied by classification problem with variable misclassification costs.

III. METHODOLOGY

The approach that this paper proposes, uses the latest machine learning algorithms to detect anomalous activities, called outliers.

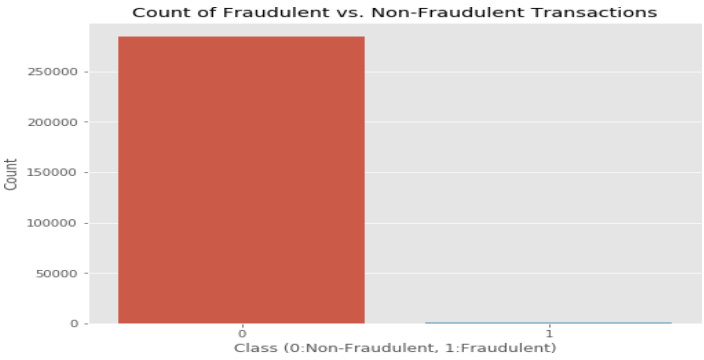
The basic rough architecture diagram can be represented with the following figure:



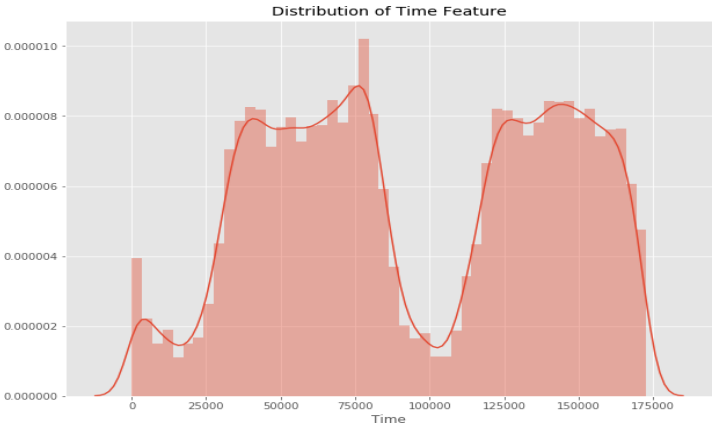
First of all, we obtained our dataset from Kaggle, a data analysis website which provides datasets.

The other columns represent Time, Amount and Class. Time shows the time gap between the first transaction and the following one. Amount is the amount of money transacted. Class 0 represents a valid transaction and 1 represents a fraudulent one.

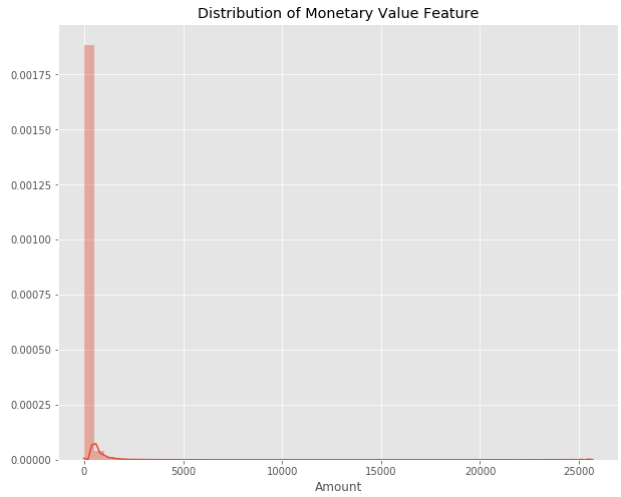
We plot different graphs to check for inconsistencies in the dataset and to visually comprehend it:



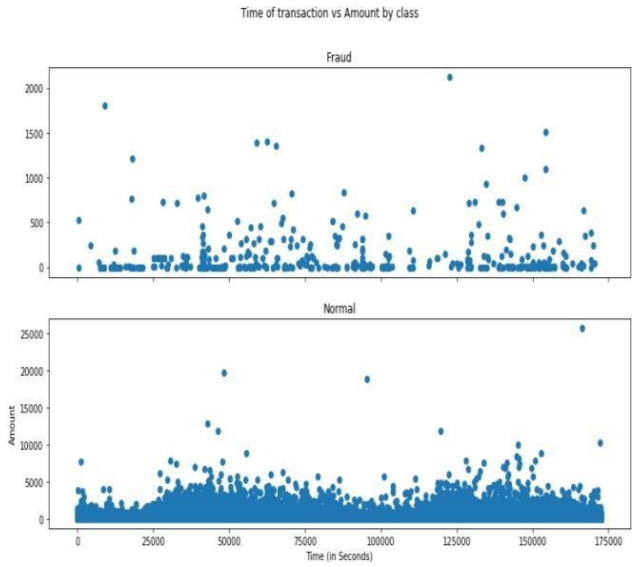
This graph shows that the number of fraudulent transactions is much lower than the legitimate ones:



This graph below depicts the times at which transactions were done in two days. It can be clearly be seen that the least number of transactions were done during night time and highest during the days.



This graph shows the transacted amount. A majority of transactions are comparatively small and only some of them were close to the maximum transacted amount.



Now the dataset is formatted and processed. The Class column is removed to ensure fairness of evaluation and the time and amount column are standardized. A set of algorithms from the modules is used to process the dataset. The subsequent module diagram explains how these algorithms work together: The data is fit into a model and then the following outlier detection modules are put on it:

- Local Outlier Factor
- Isolation Forest Algorithm

These algorithms are a part of sklearn. The grouped module in the sklearn package contains ensemble-based methods and functions for the classification, regression and outlier detection.

NumPy is used to build this free and open-source Python library, SciPy and matplotlib modules which provides a plenty of easy and systematic tools that can be used for data analysis and machine learning are also used. It has various characteristic classification, clustering and regression algorithms and is outlined to interoperate with the numerical and scientific libraries.

We have used Jupyter Notebook platform to make a program in Python to exhibit the approach that this paper suggests. This program can additionally be executed on the cloud using Google Collab platform which supports all python notebook files.

Detailed explanations of the modules with pseudocodes for their algorithms and output graphs are given below:

A. Local Outlier Factor

(Unsupervised Outlier Detection Algorithm)

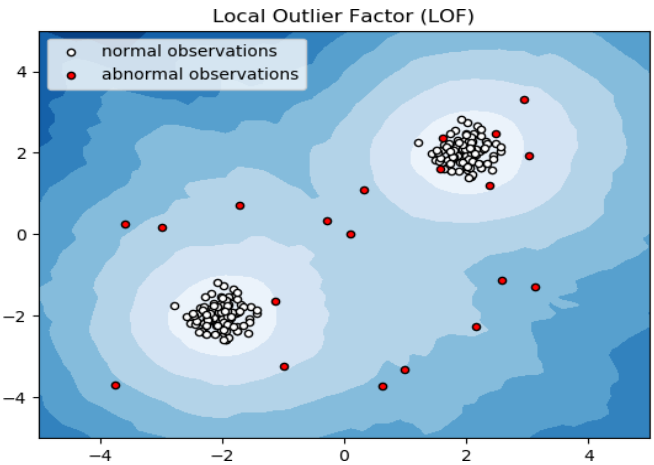
'Local Outlier Factor' denotes the anomaly score of each sample. It calculates the local deviation of the sample data with respect to its neighbor.

More accurately, locality is given by k-nearest neighbors, whose distance is used to estimate the local data.

The pseudo code for this algorithm is written as follows:

```
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)
```

After plotting the results of Local Outlier Factor algorithm, we get a figure as follows:



On comparing the local values of a sample with that of its neighbors, one can easily pick out samples that are considerably lower than their neighbors. These values are quite amanous and they are contemplated as outliers.

B. Isolation Forest Algorithm

The Isolation Forest 'isolates' observations by selecting an offhand feature and then arbitrarily selecting a split value between the highest and lowest values of the chosen feature. The representation of Recursive partitioning can be done by a tree, the number of splits needed to isolate a sample is equal to the path length root node to terminating node.

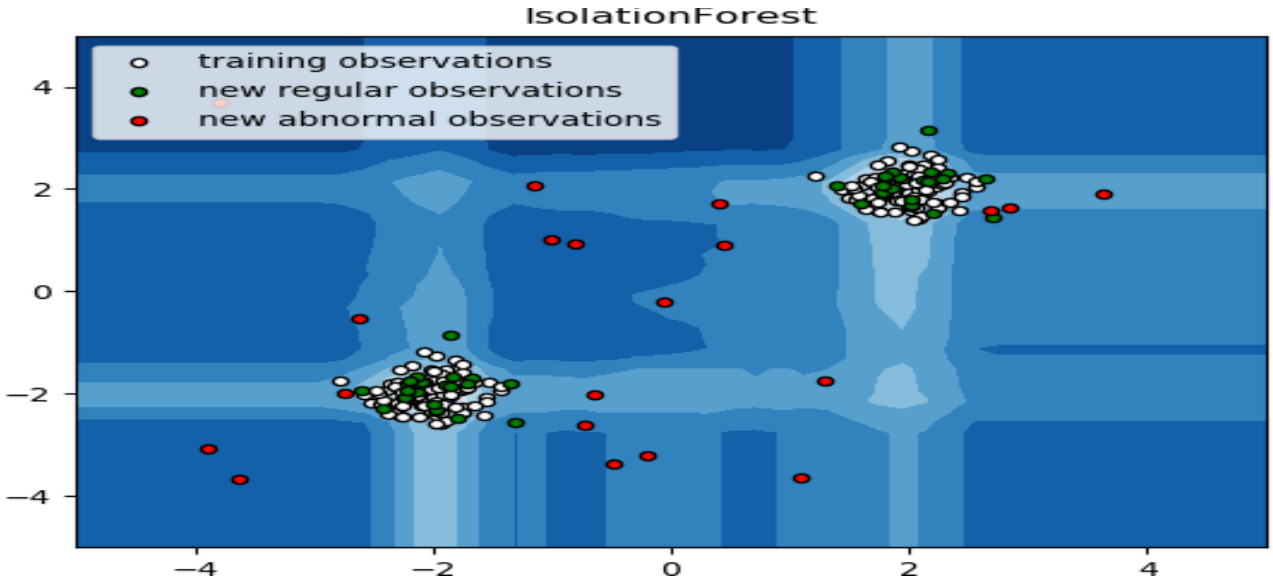
The average of this path length provides a measure of normality and the decision function which we use.

One of the latest techniques to detect anomalies is called **Isolation Forests**. The algorithm is formulated on the fact that anomalies are data points that are few and different. As a result of these properties, anomalies are responsive to a mechanism called isolation.

The pseudocode for this algorithm can be written as follows:

```
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 Isolation Forest algorithm, we get the following figure:



Partitioning them randomly produces shorter paths for anomalies. When a forest of random trees mutually makes shorter path lengths for specified samples, they are extremely likely to be anomalies.

The system can be used to report to the concerned authorities, once the anomalies are detected. For testing purposes, we are comparing the outputs of these algorithms to determine their correctness and exactness.

C. Support Vector Machine Algorithm

Support Vector Machine or SVM is one amongst most favored Supervised Learning algorithms which is used for solving Classification together with Regression problems. But, firstly, it is employed on Classification problems in Machine Learning. The objective of the SVM algorithm is to generate the best line or decision boundary which can separate n-dimensional space into classes so that the new data point can easily be put under the correct category in the future and this best decision boundary is known as a hyperplane.

SVM can be of two types as follows:

Linear SVM: Linear SVM is employed on linearly separable data, that means if a dataset can be classified into two classes by using a single straight line, then such data is called as linearly separable data, and classifier employed is called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is employed on non-linearly separated data, that means if a dataset cannot be classified by using a straight line, then such data is called as non-linear data and classifier employed is called as Non-linear SVM classifier.

```
import numpy as np
import matplotlib.pyplot as plt
from skleam. neighbors import OneClassSVM
np.random.seed(42)

# Generate train data
X = 0.3 np.random.rando(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 = Local Outlier Factor (n_neighbors=20)
y_pred = clf.fit_predict(X)
y_pred outliers = y_pred[200:]

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

IV. IMPLEMENTATION

This idea is hard to implement in real life as it needs the cooperation from banks, which are not at all ready to share information because of their market competition, protection of data of their users as well as due to legal reasons.

Due to this reason, we looked up some reference papers which go with much similar approaches and collected results. As expressed in one of these reference papers:

"In 2006, the same approach was applied to a full application data set supplied by a German bank. For banking clandestinity reasons, only a summary of the results acquired is presented below. The level 1 list encloses a few cases after applying this technique that too with a high probability of being fraudsters.

All the people in this list had shut down their cards to stay away from being a prey to the fraud due to their high-risk profile. The other list has a more complex condition. The level 2 list is still confined appropriately to be checked on a case-by-case basis.

Credit and collection officers contemplate that half of the cases in this list could be considered of a suspicious fraudulent behavior. For the last and the largest list, the work is equally heavy. Less than a third of them are doubtful.

In order to accelerate the time efficiency and the overhead charges, to include a new element in the query is a possibility; this element can be the five initial digits of the phone numbers, the e-mail address, and the password, for example, those new queries can be applied to the level 2 list and level 3 list.".

V. CHALLENGES

The first category which includes the lost or stolen cards, is a comparatively common one and should be reported instantly to avoid any damages.

The second one is "account takeover" which happens when a cardholder accidently gives his/her personal information (such as home address, mother's maiden name, etc.) to a fraudster, who then contacts the cardholder's bank, reports a lost card and requests for change of address, and acquire a new card in the victim's name.

The third is counterfeit cards occurs when a card is "cloned" from another and then used to make purchases.

The fourth is called "never received" it occurs when a new or replacement card is stolen from the email, never reaching its rightful owner.

The fifth is fraudulent application which occurs when a fraudster uses other person's name and information to apply for and get a credit card.

The sixth is called "multiple imprint" which occurs when a single transaction is recorded multiple times on old-fashioned credit card imprint machines known as "knuckle busters".

VI. FUTURE SCOPE

Evolution in technology give criminals progressively powerful tools to commit fraud, specially using credit cards or internet bots. To fight the evolving face of fraud, researchers are developing progressively sophisticated tools, with algorithms and data structures capable of handling large-scale complex data analysis and storage.

So, our research mainly focuses on the analysis of different Machine Learning algorithms that can detect the fraud with accuracy.

VII. RESULTS

The number of false detected by the code is printed out and compared with the authentic values. This is used to calculate the exact score and accuracy of the algorithms. The fragment of data used for faster testing is just 10% of the entire dataset. The complete dataset has also been used at the end and both the results are printed.

These results along with the classification report for every algorithm is given in the output as follows, where class 0 means the transaction was determined to be valid and 1 means it was a fraud transaction.

This result is matched against the class values to check for false positives.

The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the data of the ones that turned out to be fraud. This model is further used to pick out a new fraudulent transaction. Our goal here is to detect 100% of the fraudulent transactions while cut back the incorrect fraud classifications.

Results when 10% of the dataset is used:

Isolation Forest
Number of Errors: 71
Accuracy Score: 0.99750711000316

	precision	recall	f1-score	support
0	1.00	1.00	1.00	28432
1	0.28	0.29	0.28	49
accuracy			1.00	28481
macro avg	0.64	0.64	0.64	28481
weighted avg	1.00	1.00	1.00	28481

Local Outlier Factor
Number of Errors: 97
Accuracy Score: 0.9965942207085425

	precision	recall	f1-score	support
0	1.00	1.00	1.00	28432
1	0.02	0.02	0.02	49
accuracy			1.00	28481
macro avg	0.51	0.51	0.51	28481
weighted avg	1.00	1.00	1.00	28481

Results with the complete dataset is used:

Isolation Forest
Number of Errors: 659
Accuracy Score: 0.9976861523768727

	precision	recall	f1-score	support
0	1.00	1.00	1.00	284315
1	0.33	0.33	0.33	492
accuracy			1.00	284807
macro avg	0.66	0.67	0.66	284807
weighted avg	1.00	1.00	1.00	284807

Local Outlier Factor
Number of Errors: 935
Accuracy Score: 0.9967170750718908

	precision	recall	f1-score	support
0	1.00	1.00	1.00	284315
1	0.05	0.05	0.05	492
accuracy			1.00	284807
macro avg	0.52	0.52	0.52	284807
weighted avg	1.00	1.00	1.00	284807

Support Vector Machine: 8516
Accuracy Score : 0.7009936448860644
Classification Report :

	precision	recall	f1-score	support
0	1.00	0.70	0.82	28432
1	0.00	0.37	0.00	49
accuracy			0.70	28481
macro avg	0.50	0.53	0.41	28481
weighted avg	1.00	0.70	0.82	28481

VIII. CONCLUSION

Credit card fraud is no a doubt a crime. This article contains the most common methods of fraud along with their detection methods and reviews recent findings in this field. This paper has also explained how machine learning can be used to get better results in fraud detection along with the algorithm, pseudocode, explanation along with its implementation and experimentation results.

While the algorithm reaches over 99.6% exactness, its precision remains only at 28% when only a tenth of the data set is taken into consideration. But when the entire dataset is fed into the algorithm, the accuracy increases to 33%. This high percentage of precision is to be expected due to the huge imbalance between the number of valid and number of authentic transactions.

As the entire dataset consists of only two days transaction records, it's only a fraction 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 with time when more data is put into it.

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