Credit Card Fraud Detection System

Shashank Kumar Pandey[1] and Shadab Akhtar Ansari[2]

Abstract— **Throughout the financial sector, machine learning algorithms are being developed to detect fraudulent transactions.That is exactly what we are going to be doing as well. Using a dataset of credit card transactions and multiple unsupervised anomaly detection algorithms, we are going to identify transactions with a high probability of being credit card fraud.The sub-aim is to present, compare and analyze two anomaly detection algorithms Local Outlier Factor and Isolation Forest algorithm.Furthermore, using metrics such as precision, recall, and F1-scores, we will investigate why the classification accuracy for these algorithms can be misleading.**

I. INTRODUCTION

Credit card fraud detection is a relevant problem that draws the attention of machine-learning and computational intelligence communities, where large number of automatic solutions have been proposed. In fact, this problem appears to be particularly challenging from a learning persepctive, since it is characterized at the same time by class imbalance, namely genuine transactions far outnumber frauds, and concept drift [4], [16], namely transactions might change their statistical properties over time. These, however, are not the only challenges characterizing learning problems in a realworld Fraud-Detection System (FDS). In a real-world FDS, the massive stream of payment requests is quickly scanned by automatic tools that determine which transactions to authorize. Classifiers are typically employed to analyze all the authorized transactions and alert the most suspicious ones. Alerts are then inspected by professional investigators that contact the cardholders to determine the true nature (either genuine or fraudulent) of each alerted transaction. By doing this, investigators provide a feedback to the system in the form of labeled transactions, which can be used to train or update the classifier, in order to preserve (or eventually improve) the fraud-detection performance over time. The vast majority of transactions cannot be verified by investigators for obvious time and cost constraints. These transactions remain unlabeled until customers discover and report frauds, or until a sufficient amount of time has elapsed such that non-disputed transactions are considered genuine.

Fraud act as the wrongful/criminal deception intended to result in financial or personnel gain. So, Credit Card Fraud is an illegal or fulsome use of card or unusual transaction behavior.As shown in the figure 1 there are so many frauds detected that affect the bank, merchants as well as customers. Some of them are listed below: a)Inception of mails of newly issued cards. b)Copying or replicating of card information through cloned websites. c) Phishing in which credit card number and password is hacked like through emails etc. d)Triangulation In this type of fraud, fraudster make an authentic looking website and advertises to sell goods at highly lower prices. Unaware users attract to those sites and make online transactions. They submit their card information to buy those goods. And then fraudsters use these card information to make genuine transactions.

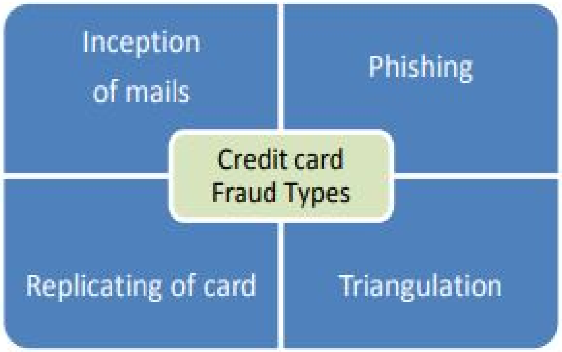


Fig: 1 General Types of Credit Card Fraud

In this paper we address some solutions to detect credit card fraud as early as possible. The following section introduce some approaches on the basis of supervised and Unsupervised learning.

II. LITERATURE SURVEY IN FRAUD DETECTION

Fraud act as the wrongful 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 a aim to attain unauthorized financial benefit.Lokesh Sharma ans Raghavendra Patidar works on emerging technology Neural Network,that can be used in banking or financial areas to detect fraud. They have been successfully applied to detect legitimate or fraudulent transactions.Association Rules can be applied to detect fraud.Linda Delamaire and Pointon et.al use the association rules to extract knowledge so that normal behavior pattern may be obtained in unlawful transactions.This proposed Methodology has been applied on data about credit card fraud of the most important retail companies in Chile. In the area of fraud detection, neural network like feed forward neural network with back propagation have found immense application.Usually such applications need to know previous data and on the behalf of this previous data they detect the fraud. Another statistically approach is feed forward network in which there is certain kind of relationship is found between user data and other parameters to get. the result. By the help of this approach or by using SOM, data can be filter out to analyze customer behavior (John T.S Kuah, M.Sriganesh). Another new emerging technology of Credit card fraud detection is based on the genetic algorithm and scatter search. Ekrem DuMan,M Hamdi Ozclik published an approach that was base on genetic algorithm and scattering search. In this approach, each transaction is scored and based on these score transactions are divided into fraudulent or legitimate transactions. They focused on a solution to minimize the wrongly classified transactions. They merge the Meta heuristic approaches scatter search and genetic algorithm.

Peer group analysis made by David Weston and Whitrow is a good solution regarding credit card fraud detection. Peer group analysis is a good approach that is based on unsupervised learning and it monitors the behavior over time as well. This peer group technique can be used to find anomalous transaction and help to detect the fraud in time.

All these technologies have their pros and cons as well. As Linda Delmaire works on association rule is a simple method that initially need large data set in which it can find frequent item set. As work done by Lokeh et.al is on Neural Network that can be applied in Supervised as well as Unsupervised Approach. As Unsupervised approach is little bit more complex but give more optimized results. John T.S Quah and M.Sriganesh works on Real time Credit card Fraud detection using computational intelligence that works on Self Organising Map. Ekrem et.al combined the genetic algorithm and Scatter search approach that is really helpful to find anomalous transcations.David Weston provides a good solution to find credit card fraud detection using Peer Group analysis method. So, the main motive of our paper is to represent all important technologies that can detect the fraud as early as possible and to avoid the loss as much as possible.

III. Various Techniques to Detect Credit Card Fraud

There are many emerging technologies that are able to detect credit card fraud detection. Some of condign technologies that will work on some parameters and able to detect fraud earlier as well are listed below:

A. Local Outlier Factor

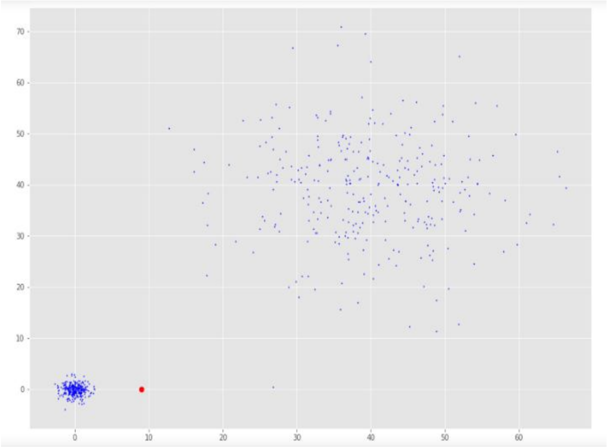
The Local Outlier Factor (LOF) algorithm is an unsupervised anomaly detection method which computes the local density deviation of a given data point with respect to its neighbors. It considers as outliers the samples that have a substantially lower density than their neighbors.

Local Outlier Factor (LOF) is a score that tells how likely a certain data point is an outlier/anomaly.

LOF $\approx 1 \rightarrow noOutlier$

LOF $\gg 1 \rightarrow outlier$

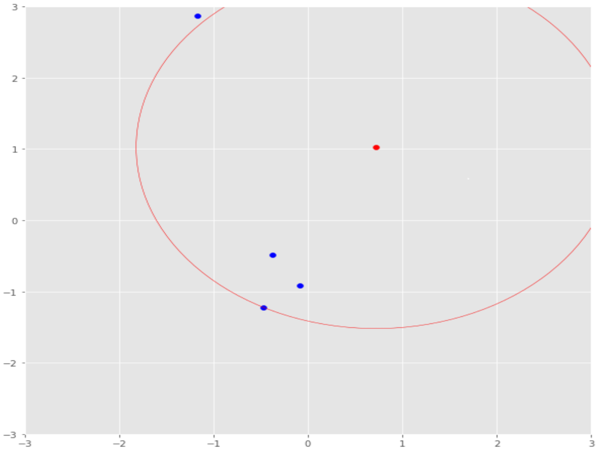
First, I introduce a parameter k which is the number of neighbors the LOF calculation is considering. The LOF is a calculation that looks at the neighbors of a certain point to find out its density and compare this to the density of other points later on. Using a right number k isnt straight forward. While a small k has a more local focus, i.e. looks only at nearby points, it is more erroneous when having much noise in the data. A large k, however, can miss local outliers.



The density of the red point to its nearest neighbors is not different from the density to the cloud in the upper right corner. However, it is probably an outlier compared to the nearest neighbors density.

K-distance

With this k defined, we can introduce the k-distance which is the distance of a point to its kth neighbor. If k was 3, the k-distance would be the distance of a point to the third closest point.



The red points k-distance is illustrated by the red line if k=3.

Reachability distance

The k-distance is now used to calculate the reachability distance. This distance measure is simply the maximum of the distance of two points and the k-distance of the second point.

$$reach\text{-}dist(a,b) = max\{k\text{-}distance(b), dist(a,b)\}$$

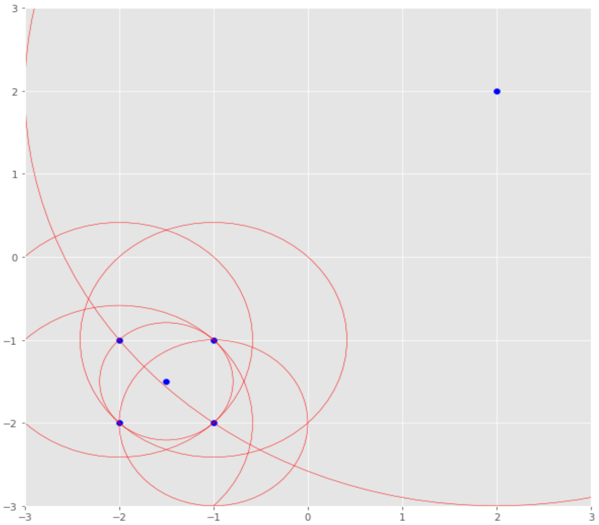
Basically, if point a is within the k neighbors of point b, the reach-dist(a,b) will be the k-distance of b. Otherwise, it will be the real distance of a and b. This is just a smoothing factor. For simplicity, consider this the usual distance between two points.

Local reachability density

The reach-dist is then used to calculate still another conceptŁŁthe local reachability density (lrd). To get the lrd for a point a, we will first calculate the reachability distance of a to all its k nearest neighbors and take the average of that number. The lrd is then simply the inverse of that average. Remember that we are talking about densities and, therefore, the longer the distance to the next neighbors, the sparser the area the respective point is located in. Hence, the less dense –the inverse.

$$lrd(a) = 1/(sum(reach\text{-}dist(a,n))/k)$$

By intuition the local reachability density tells how far we have to travel from our point to reach the next point or cluster of points. The lower it is, the less dense it is, the longer we have to travel.



The lrd of the upper right point is the average reachability distance to its nearest neighbors which are points (-1, -1), (-1.5, -1.5) and (-1, -2). These neighbors, however, have other lrds as their nearest neighbors dont include the upper right point.

LOF

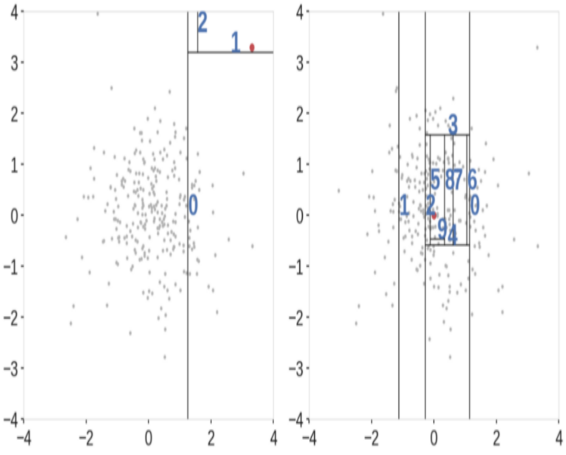
The lrd of each point will then be compared to the lrd of their k neighbors. More specifically, k ratios of the lrd of each point to its neighboring points will be calculated and averaged. The LOF is basically the average ratio of the lrd of point a to the lrds to its neighboring points. If the ratio is greater than 1, the lrd of point a is on average greater than the lrd of its neighbors and, thus, from point a, we have to travel longer distances to get to the next point or cluster of points than from as neighbors to their next neighbors. Keep in mind, the neighbors of a point a may dont consider a a neighbor as they have points in their reach which are way closer.

In conclusion, the LOF of a point tells the density of this point compared to the density of its neighbors. If the density of a point is much smaller than the densities of its neighbors (LOF $\gg$ 1), $thepointisfarfromdenseareasand, hence, anoutlier.$

B. Isolation Forest Algorithm

Isolation Forest

Isolation Forest algorithm utilises the fact that anomalous observations are few and significantly different from normal observations. The forest is built on the basis of decision trees, each of the trees having access to a sub-sample of the training data. In order to create a branch in the tree, first a random feature is selected. Afterwards, a random split value (between min and max value) is chosen for that feature. If the given observation has lower value of this feature then the one selected it follows the left branch, otherwise the right one. This process is continued until a single point is isolated or specified maximum depth is reached.



(a) Anomaly point

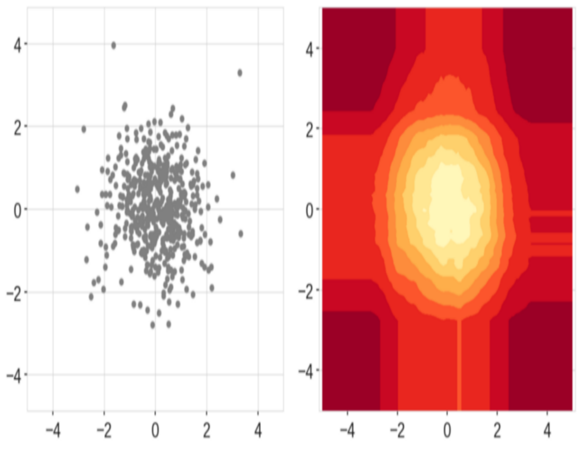
(b) Nominal point

In principle, outliers are less frequent than regular observations and are different from them in terms of values (they lie further away from the regular observations in the feature space). That is why by using such random partitioning they should be identified closer to the root of the tree (shorter average path length, i.e., the number of edges an observation must pass in the tree going from the root to the terminal node), with fewer splits necessary.

The anomaly score is created on the basis of all trees in the forest and the depth the point reaches in these trees.

Isolation Forest Problem

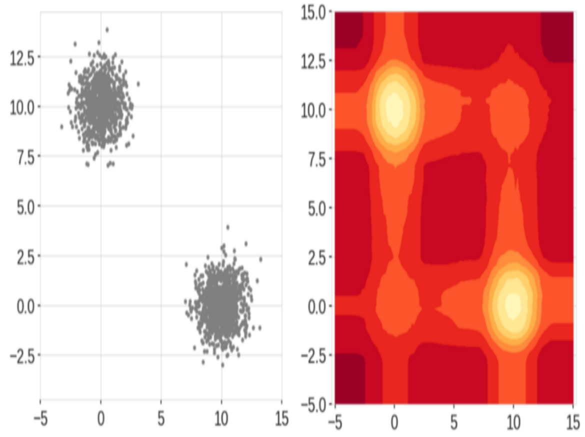
The best way to understand the issue is to see it on an example.



(a) Normally Distributed Data (b) Anomaly Score Map

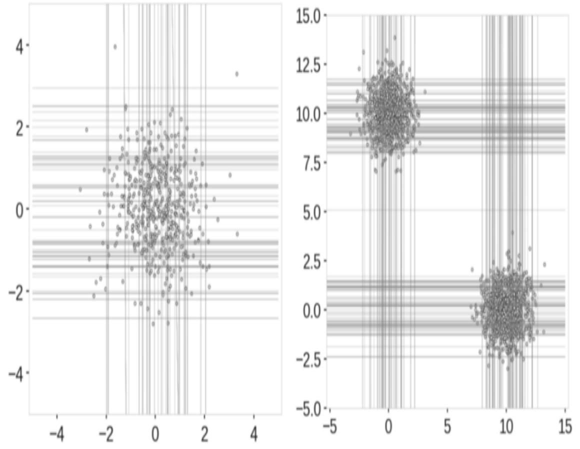
In the left picture we can see data sampled from multivariate normal distribution. Intuitively, we would assume that the anomaly score assigned to the observations would increase radially from the central point of the distribution [0, 0]. However, this is clearly not the case, as seen in the right image. What is more, there are also rectangular artifacts of lower score, such as the vertical one between point 0 and 1 on the x axis.

Lets move on to the second example. Here we see two blobs centred at points [0, 10] and [10, 0]. By inspecting the right figure we see not only the artifacts that were present before, but also two ghost clusters (approximately at [0, 0] and [10, 10]).



(a) Two normally distributed clusters (b) Anomaly Score Map

The reason for this peculiar behaviour originates from the fact that the decision boundaries of the Isolation Forest are either vertical or horizontal (random value of a random feature), as seen in the picture below, where the authors plot branch cuts generated by the IF during the training phase. We see that the branches tend to cluster where the majority of the points are located. But as the lines can only be parallel to the axes, there are regions that contain many branch cuts and only a few or single observations, which results in improper anomaly scores for some of the observations. An example might be points around [3, 0] (many branch cuts) and [3, 3] (few cuts).



(a) Single blob (b) Multiple Blobs

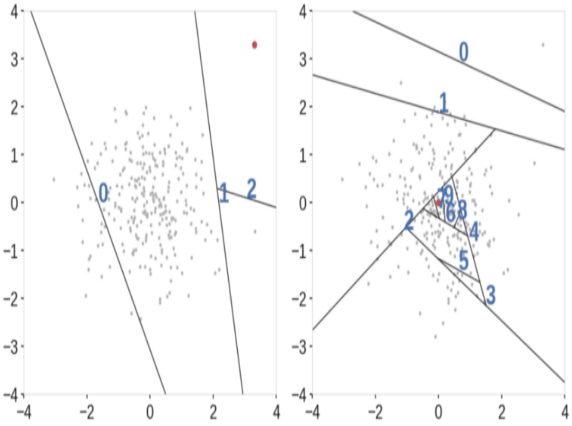
Extended Isolation Forest

Analysis of the Isolation Forests drawback led to a conclusion that the problem is caused by only horizontal/vertical branch cuts. Extended Random Forest addresses that issue by approaching the problem a bit differently. Instead of selecting a random feature and then random value within the range of data it selects:

- random slope for the branch cut
- random intercept chosen from the range of available values from the training data

These are the terms (slope/intercept) you most likely recall from the simple linear regression (y = ax + b). Lets look at a visual example! From the image below we can see the main difference from the original IF algorithm →

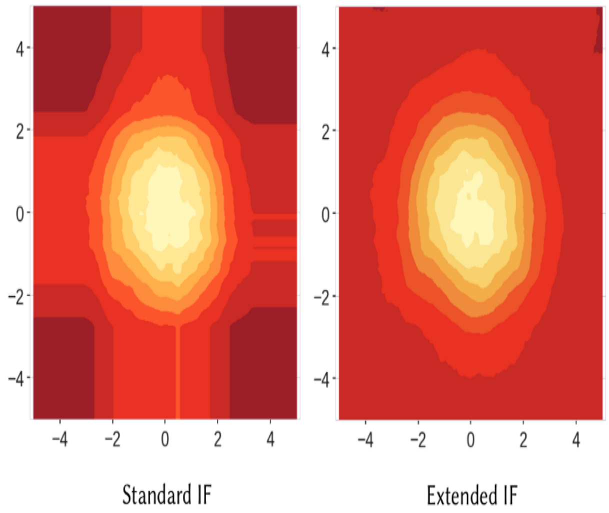
cuts that are not parallel to the axes.



(a) Anomaly (b) Nominal

Extended Random Forest generalizes well into higher dimension, where instead of straight lines we are dealing with hyperplanes.

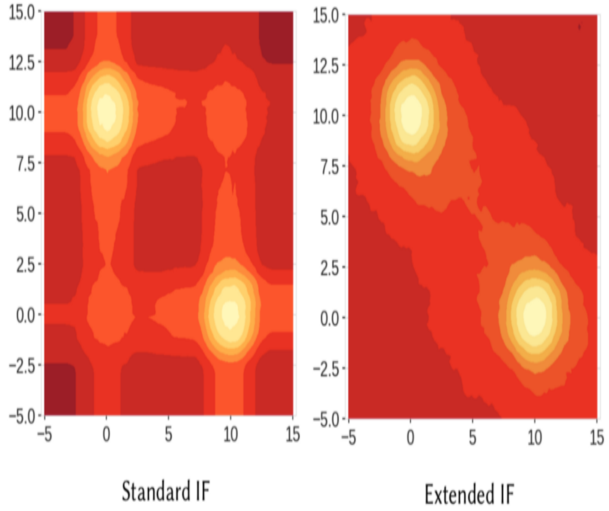
Lets have a look at the difference in the anomaly score maps generated by IF/EIF. In the images below we see that the anomaly score spreads out from the data clusters radially and there are no artifacts/ghost clusters visible.



Standard IF

Extended IF

An extra feature captured by the EIF is the higher anomaly score region directly in-between the two clusters (where they kind of link). This region can be considered as close to normal given the proximity to both clusters, but with higher score as it is far from the concentrated groups.



Standard IF

Extended IF

IV. Implementation and Comparison

Throughout the financial sector, machine learning algorithms are being developed to detect fraudulent transactions.That is exactly what we are going to be doing as well. Using a dataset of of nearly 28,500 credit card transactions and multiple unsupervised anomaly detection algorithms, we are going to identify transactions with a high probability of being credit card fraud.

We will build and deploy the following two machine learning algorithms:

- Local Outlier Factor (LOF)
- Isolation Forest Algorithm

Furthermore, using metrics suchs as precision, recall, and F1-scores, we will investigate why the classification accuracy for these algorithms can be misleading.

In addition, we will explore the use of data visualization techniques common in data science, such as parameter histograms and correlation matrices, to gain a better understanding of the underlying distribution of data in our data set.

Let’s get started!

1. Importing Necessary Libraries

```
import sys
import numpy
import pandas
import matplotlib
import seaborn
import scipy
```

Import the necessary packages

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2. The Data Set

To get started we will first build a dataset of over 280,000 credit card transactions to work on! We will import our dataset from a .csv file as a Pandas DataFrame.Furthermore, we will begin exploring the dataset to gain an understanding of the type, quantity, and distribution of data in our dataset. For this purpose, we will use Pandas’ built-in describe feature, as well as parameter histograms and a correlation matrix.

Note: One can view or download the Source code and dataset from our Github repository→ $https : //github.com/SHASHANKSKP/Credit - Card - Fraud - DEtection - .git.$

For Dataset → $https : //drive.google.com/open?id = 14DTdChjd_mVD - 4uZroi9286psSZtjNAo$

Load the dataset from the csv file using pandas

```
data = pd.read_csv(’creditcard.csv’)
```

Start exploring the dataset

```
print(data.columns)
```

Print the shape of the data

```
data = data.sample(frac=0.1, random_state = 1)
print(data.shape)
print(data.describe())
```

Plot histograms of each parameter

```
data.hist(figsize = (20, 20))
plt.show()
```

Determine number of fraud cases in dataset

```
Fraud = data[data[’Class’] == 1]
Valid = data[data[’Class’] == 0]
outlier_fraction = len(Fraud)/float(len(Valid))
print(outlier_fraction)
print(’Fraud Cases: ’.format(len(data[data[’Class’] == 1])))
print(’Valid Transactions: ’.format(len(data[data[’Class’] == 0])))
```

Correlation matrix

```
corrmat = data.corr()
fig = plt.figure(figsize = (12, 9))
sns.heatmap(corrmat, vmax = .8, square = True)
```

plt.show()

Get all the columns from the dataFrame

columns = data.columns.tolist()

Filter the columns to remove data we do not want

columns = [c for c in columns if c not in [”Class”]]

Store the variable we’ll be predicting on

target = ”Class”

X = data[columns]

Y = data[target]

Print shapes

print(X.shape)

print(Y.shape)

3. Unsupervised Outlier Detection Now that we have processed our data, we can begin deploying our machine learning algorithms. We will use the following techniques:

Local Outlier Factor (LOF)

The anomaly score of each sample is called Local Outlier Factor. It measures the local deviation of density of a given sample with respect to its neighbors. It is local in that the anomaly score depends on how isolated the object is with respect to the surrounding neighborhood.

Isolation Forest Algorithm

The IsolationForest isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.

Since recursive partitioning can be represented by a tree structure, the number of splittings required to isolate a sample is equivalent to the path length from the root node to the terminating node.

This path length, averaged over a forest of such random trees, is a measure of normality and our decision function.

Random partitioning produces noticeably shorter paths for anomalies. Hence, when a forest of random trees collectively produce shorter path lengths for particular samples, they are highly likely to be anomalies.

from sklearn.metrics import classification_report, accuracy_score

from sklearn.ensemble import IsolationForest

from sklearn.neighbors import LocalOutlierFactor

define random states

state = 1

define outlier detection tools to be compared

classifiers = ”Isolation Forest”: IsolationForest(max_samples=len(X),

contamination=outlier_fraction,

random_state=state),

”Local Outlier Factor”: LocalOutlierFactor(

n_neighbors=20,

contamination=outlier_fraction)

Fit the model

plt.figure(figsize=(9, 7))

n_outliers = len(Fraud)

for i, (clf_name, clf) in enumerate(classifiers.items()):

fit the data and tag outliers

if clf_name == ”Local Outlier Factor”:

y_pred = clf.fit_predict(X) scores_pred = clf.negative_outlier _factor

else:

clf.fit(X)

scores_pred = clf.decision_function(X)

y_pred = clf.predict(X)

Reshape the prediction values to 0 for valid, 1 for fraud.

y_pred[y_pred == 1] = 0

y_pred[y_pred == -1] = 1

n_errors = (y_pred != Y).sum()

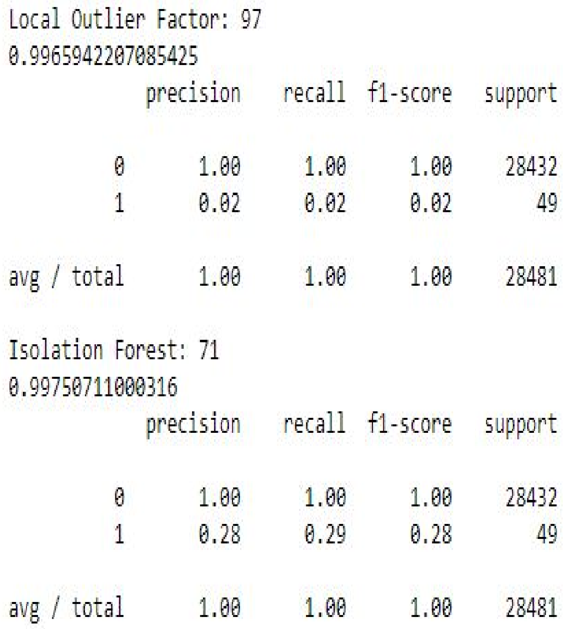
Run classification metrics

print(’: ’.format(clf_name, n_errors))

print(accuracy_score(Y, y_pred))

print(classification_report(Y, y_pred))

V. Results



```
Local Outlier Factor: 97
0.9965942207085425
             precision    recall  f1-score   support

          0       1.00      1.00      1.00     28432
          1       0.02      0.02      0.02        49

avg / total       1.00      1.00      1.00     28481

Isolation Forest: 71
0.99750711000316
             precision    recall  f1-score   support

          0       1.00      1.00      1.00     28432
          1       0.28      0.29      0.28        49

avg / total       1.00      1.00      1.00     28481
```

We can see that using Local Outlier Factor we are able to predict only 2% of Fraud cases, wheras using Isolation Forest algorithm , performance gets improved,and it can predict 28 % of Fraudulent transactions. Also the Recall and f1-score for Isolation Forest algorithm is better than Local Outlier Factor, as Isolation Forest algorithm uses multiple decision trees as classifiers.So it has improved accuracy than Local Outlier Factor. However these both are having very less precision values because of the nature of their algorithms and the sparse dataset.

VI. Conclusion

Clearly, credit card fraud is an act of criminal dishonesty.Due to Fulsome advancement in technology, the use of credit card has increased and due to this, Fraud cases are affecting it directly. One of the main motives of this study is to compare two anomaly detection algorithms Local Outlier Factor and Isolation Forest algorithm,which can be used to detect fraud effectively. If one of the above technologies is applied in bank then cases of credit card fraud will surely minimize. Here we compared these algorithms that can detect credit card fraud and save the bank from big loss.

ACKNOWLEDGMENT

This project is an outcome of the efforts put forth by many people. Firstly, we would like to thank the VIT University Management and the Dean of SCSE for giving us this opportunity to carry out this paper. Secondly, our teacher, Prof. Dr. Syed. P. Ibrahim providing his valuable inputs during our Reviews and giving us the opportunity to take-up this interesting topic. This paper is a group effort by all the team members.

REFERENCES

[1] Breunig, M. M., Kriegel, H. P., Ng, R. T., Sander, J. (2000, May). LOF: identifying density-based local outliers. In ACM sigmod record (Vol. 29, 2, pp. 93104). ACM.

[2] David J.Wetson,David J.Hand,M Adams,Whitrow and Piotr Jusczak Plastic Card Fraud Detection using Peer Group Analysis Springer, Issue 2008.

[3] Francisca Nonyelum Ogwueleka, Data Mining application In Credit Card fraud Detection Journal of Enggineering Science and Technology, Vol 6 No 3,Issue 2011.

[4] Aleskerov, E., Freisleben, B. B Rao. 1997. CARDWATCH: A Neural Network-Based Database Mining System for Credit Card Fraud Detection, Proc. of the IEEE/IAFE on Computational Intelligence for Financial Engineering, 220-226.

[5] Bolton, R. Hand, D. 2001. Unsupervised Profiling Methods for Fraud Detection, Credit Scoring and Credit Control VII.

[6] John T.S Quah,M Sriganesh Real time Credit Card Fraud Detection using Computational Intelligence ELSEVIER Science Direct,35 (2008) 1721-1732. Chepaitis, E. 1997. Information Ethics Across Information Cultures. Business Ethics: A European Review, 6: 4,

[7] V.PriyaDarshini,G.Adiline Macriga, An Efficient dta Mining For Credit Card Fraud detection Using finger Print Recognition, IJACR, Vol 2, 2012.

[8] Chan, P., Fan, W. Prodromidis, A. S Stolfo. 1999. Distributed Data Mining in Credit Card Fraud Detection. IEEE Intelligent Systems, 14; 67-74.

[9] 195-199.

[10] Philip K Chan,Wei Fan,Andias Prodromidis,J.Stolfo, Distributed Datan Mining For Credit Card Fraud Detection, IEEE Intelligent System,Special Issue On Data Mining, 1999.

[11] Kenneth Revett,Magalhaes and Hanrique Santos Data Mining a Keystroke dynamic Based Biometric Dtatabase Using Rough Set IEEE

[12] Francisca Nonyelum Ogwueleka, Data Mining application In Credit Card fraud Detection Journal of Enggineering Science and Technology, Vol 6 No 3,Issue 2011.

[13] Linda Delamaire ,Hussein Abdou and John Pointon, Credit Card Fraud and Detection technique, Bank and Bank System,Volume 4, 2009.

[14] Bolton, R. Hand, D. 2002. Statistical Fraud Detection: A Review. Statistical Science, 17; 235-249.

[15] Bolton, R. Hand, D. 2001. Unsupervised Profiling Methods for Fraud Detection, Credit Scoring and Credit Control VII.

[16] Chan, P., Fan, W. Prodromidis, A. S Stolfo. 1999. Distributed Data Mining in Credit Card Fraud Detection. IEEE Intelligent Systems, 14; 67-74.

[17] Bentley, P., Kim, J., Jung. G. J Choi. 2000. Fuzzy Darwinian Detection of Credit Card Fraud, Proc. of 14th Annual Fall Symposium of the Korean Information Processing Society.

[18] Chepaitis, E. 1997. Information Ethics Across Information Cultures. Business Ethics: A European Review, 6: 4, 195-199.

[19] European e-Business Market Watch. 2005. ICT Security, e-Invoicing and e-Payment Activities in European Enterprises, Special Report, September.

[20] Anderson, R. 2007. The Credit Scoring Toolkit: theory and practice for retail credit risk management and decision automation. New York: Oxford University Press.