كلية الحوسبة والمعلوماتية |
College of Computing and Informatics

الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Semester 2 – 2021/2022

Course Code	DS650
Course Name	Predictive Analytics for Business
Assignment type	Project
Module	All Modules

Student ID	G200007615
Student Name	Abdulaziz M. Alqumayzi
CRN	21601

Student ID	G200002614
Student Name	Enad S. Alotaibi
CRN	21601

Credit Card Fraud Detection

Enad Saud Alotaibi
G200002614@seu.edu.sa,
Saudi Electronic University,
Riyadh, Saudi Arabia

Abdulaziz Mohammed Alqumayzi
G200007615@seu.edu.sa,
Saudi Electronic University,
Riyadh, Saudi Arabia

ABSTRACT

Credit card scams are simple and easy to commit. E-commerce and other online sites have increased the number of online payment methods, raising the risk of online fraud. It is critical that credit card companies detect fraudulent credit card transactions so that customers are not charged for items they did not purchase. The global impact of credit card fraud is concerning; many businesses and individuals have lost millions of dollars. Furthermore, cyber criminals are constantly innovating sophisticated techniques; thus, it is critical to develop improved and dynamic techniques capable of adapting to rapidly evolving fraudulent patterns. With the rise in fraud rates, researchers began employing various machine learning methods to detect and analyses fraud in online transactions. This task is extremely difficult to complete, owing to the dynamic nature of fraud as well as a lack of data set for researchers. The performance of ANN, Random Forest, logistic regression, and XGBOOST on highly skewed credit card fraud data is investigated in this report. The credit card transaction dataset is sourced from European cardholders and contains 284,807 transactions. The dataset is highly unbalanced, with positive transactions accounting for 0.172 percent of all transactions. The four techniques are used on unprocessed and pre-processed data. Python is used to carry out the work. The techniques' performance is measured using accuracy, sensitivity, specificity, precision, and recall. Upon analyzing we get to know that Neural Network are spot-on predicting the normal cases with an accuracy of 99% but in the case of predicting the outliers they are not as good as anomaly detection algorithms like random forest and XGBOOST.

Keywords: E-commerce*;* XGBOOST*;* ANN*;* anomaly

1. Introduction

In credit card transactions, 'fraud' refers to the unauthorized and unwanted use of an account by someone other than the account's owner. To stop this abuse, necessary prevention measures can be implemented, and the behavior of such fraudulent practices can be studied to minimize it and protect against similar occurrences in the future. In other words, Credit Card Fraud occurs when a person uses another person's credit card for personal reasons while the owner and card issuing authorities are unaware that the card is being used.

Credit card theft is an increasing concern that costs banks and card service providers a great deal of money. Banking institutions use a variety of protective measures to try to prevent account misuse. As security solutions become more complex, fraudsters become more sophisticated, i.e., they change their strategies over time. As a result, improving fraud detection and prevention procedures Security modules aimed at preventing fraud are critical. Fraud detection has become a critical step in reducing the negative impact of fraudulent transactions on service delivery, prices, and the company's reputation. There are various methods for detecting fraud to maintain a high level of service quality. Maximum service performance while keeping the number of fatalities to a bare minimum. Fraud is costly, and early detection of fraud can save a lot of money before the information is captured. The system is highly accurate, with few false alarms.

This problem is especially difficult to solve from the standpoint of learning because it is characterized by various factors such as class imbalance. The number of valid transactions far outnumbers the number of fraudulent transactions. Furthermore, transaction patterns frequently change their statistical properties over time. However, these are not the only difficulties that must be overcome in the implementation of a real-world fraud detection system. In practice, a massive stream of payment requests is quickly scanned by automated tools that determine which transactions to authorize.

Machine learning algorithms are used to analyses all authorized transactions and report any that are suspicious. Professionals investigate these reports and contact cardholders to determine whether the transaction was genuine or fraudulent.

The investigators provide feedback to the automated system, which is then used to train and update the algorithm, resulting in improved fraud detection performance over time. Methods for detecting fraud are constantly being developed to assist criminals in adapting to their fraudulent strategies. These scams are classified as follows:

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

Financial losses due to fraud often have serious negative impacts on merchants, banks, and individual customers. Billions of dollars have been lost in recent years due to the nefarious activities of credit card fraudsters. A credit card fraud can be perpetrated using any of the following:

- Cardholder-Not-Present (CNP)
- Misplaced/ thieved card,
- Magnetic stripe obliteration,
- Hijacked account
- Card cloning and scanning.

Basic credit card fraud detection system works like this:

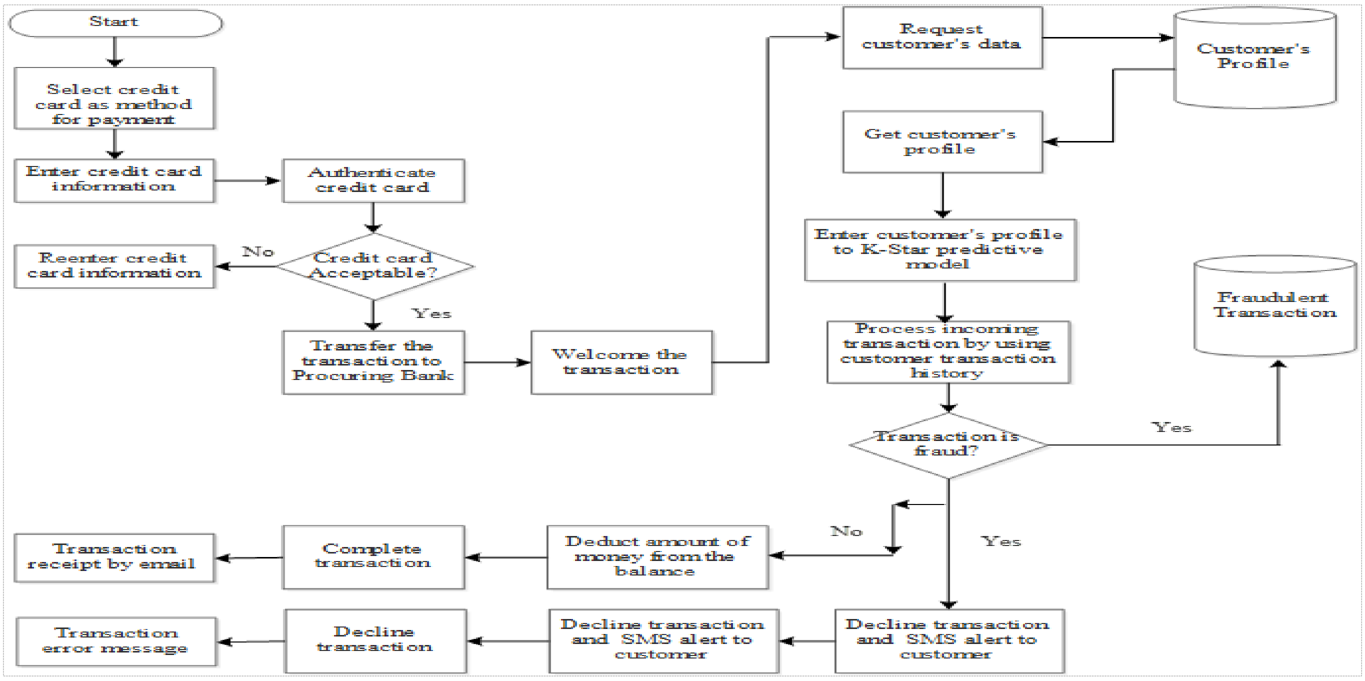

Start
Select credit card as method for payment
Enter credit card information
Authenticate credit card
Reenter credit card information
No
Credit card Acceptable?
Yes
Transfer the transaction to Procuring Bank
Welcome the transaction
Request customer's data
Customer's Profile
Get customer's profile
Enter customer's profile to K-Star predictive model
Fraudulent Transaction
Process incoming transaction by using customer transaction history
Transaction is fraud?
Yes
No
Yes
Transaction receipt by email
Complete transaction
Deduct amount of money from the balance
Transaction error message
Decline transaction
Decline transaction and SMS alert to customer
Decline transaction and SMS alert to customer

Figure 1: Credit Card fraud detection system

2. Body section

2.1 Data

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

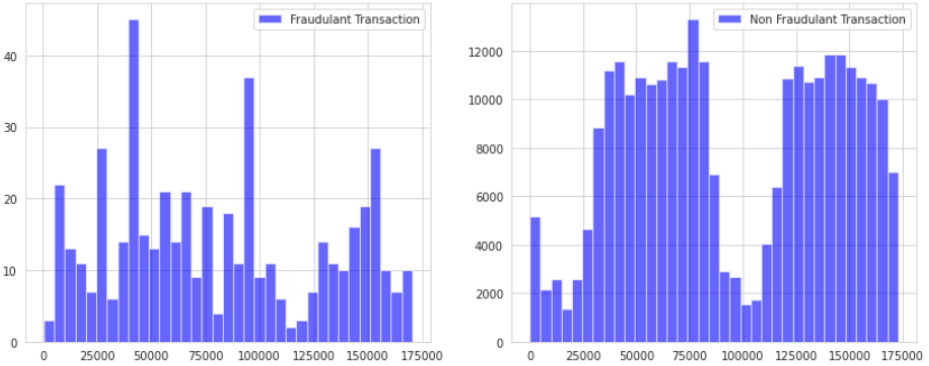


Figure 2: Fraudulent & non fraudulent transaction

2.2 Literature Review

A: Fraud Detection Process

Credit card fraud detection is the process of knowing whether a set of credit card transactions is in the category of fraudulent or legitimate instances of buying or selling something (Maes *et al.*, 2002) [13].

Some desirable characteristics of a Fraud Detection System (FDS) include efficient detection of fraud, and high effectiveness or productivity in relation to its cost in transaction checking (Quah and Sriganesh, 2008).

Increasingly, fraud investigators are depending on innovative machine learning methods to aid their investigations. The non-stationary distribution of data, the substantially one-sided classes divisions and the inaccessibility of many transactions labeled by fraud investigators have made the task of developing effective fraud detection algorithms a very demanding one. Also contributing to the difficulty is the fact that public data are barely available due to privacy concerns, thereby making it difficult to know the most efficient approach to adopt in curbing this menace.

B: Related Work

Zareapoor and Shamsolmoali (2015) in [11] applied Bagging ensemble classifier for detection of credit card fraud. They compared the performance of their proposed system with NB, SVM and KNN. The performance of their proposed technique is relatively low.

Dorronsoro et al. (2017) in [12] developed an online fraud detection system using neural classifier. The drawback of their approach is that data must be clustered by class of account making it to be time consuming.

Maes et al. (2017) in [13] used Bayesian networks classifier to detect fraud. Their approach produced excellent results. Time constraint is the major shortcoming of their method.

Randhawa *et al.*, (2017) in [14] used AdaBoost and majority voting to detect credit card fraud. The performance of the proposed system was compared with other standard models using publicly available credit card data set.

Apapan and Liu (2018) in [15] applied deep learning Auto-encoder (AE) and restricted Boltzmann machine (RBM) to create model for detecting fraud in transactions based on previous history. The results showed that their methods can accurately predict credit card detection with a large dataset. The drawback of their technique is the computational cost involved.

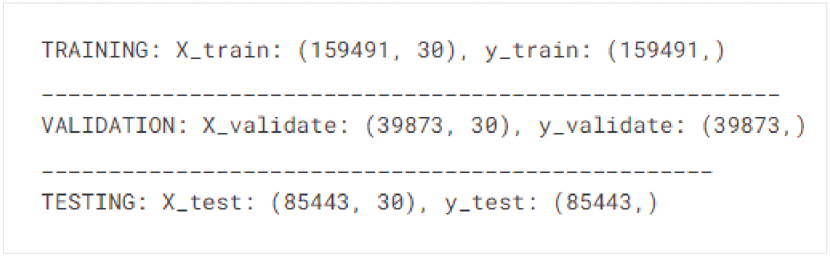
2.3 Methods

The major aspect of this project to develop a best-suited algorithm to find the outliers or frauds in the case of credit cards. Many algorithms of machine learning have been used to solve this problem including HMM (Hidden Markov Model) as described in [1], Decision Trees as explained in [2], SVM (Support Vector Machines) in [3] and Random Forests described in [4]. In terms of the overall number of frauds discovered, it also surpasses standard classifiers. [6]. The natural logarithmic function is used in logistic regression to compute probability and indicate that the results fall into a certain category. [5]. We have implemented several machine learning and deep learning algorithms and compared them and chosen the best algorithm. We have implemented following algorithms:

- Artificial Neural Network
- Random Forest
- Logistic Regression
- XGBoost

We utilized a pre-existing dataset for this. The dataset includes data on transactions done by different cardholders. The collection contains roughly 284,807 records, with just about 492 scammers among them. As a result, the dataset is severely skewed, with the positive class of scams accounting for just 0.172 percent of all transactions. Because of the PCA transformation, all the feature's columns are numeric. As a result, their value varies from -1 to 1. Because of PCA transformation, features columns V1, V2, V3, … V28 are obtained. Time and Amount columns have not been altered. The Feature Class column is a classification variable with values of 0 (Normal Case) and 1 (Exceptional Case) (Fraud).

While preprocessing the data mean is removed, and each feature/variable is scaled to unit variance using StandardScaler. This procedure is carried out in a feature-by-feature manner. Because StandardScaler includes the calculation of the empirical mean and standard deviation of each feature, it might be impacted by outliers (assuming they exist in the dataset). Then after preprocessing the unbalanced data, it is divided into 20 % test data while keeping rest of the data into training.



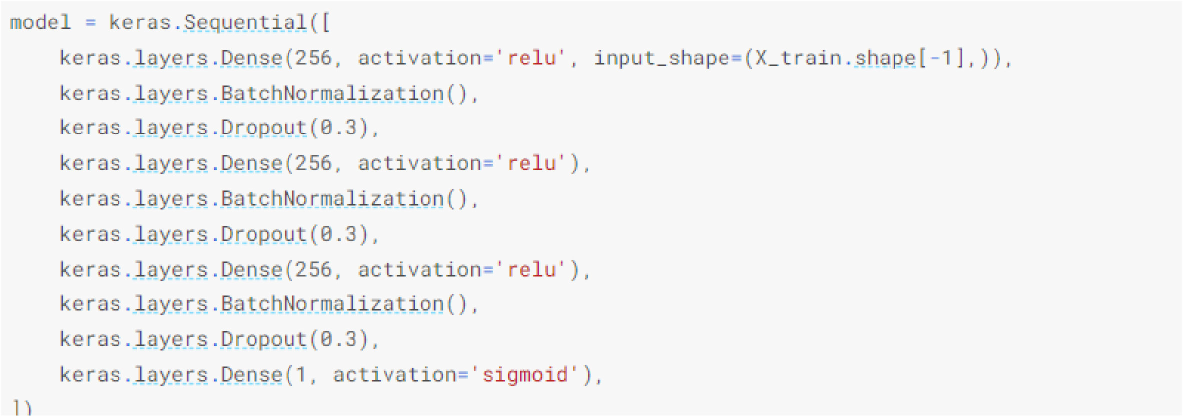
```
TRAINING: X_train: (159491, 30), y_train: (159491,)
-------------------------------------------------------
VALIDATION: X_validate: (39873, 30), y_validate: (39873,)
--------------------------------------------------
TESTING: X_test: (85443, 30), y_test: (85443,)
```

Artificial Neural Network:

ANN is a deep learning technique that is applied using Keras (in this case). Neurons make up an ANN. The input neuron is the initial layer, or input layer, and it contains each customer's transaction and amount. Weights, bias, and an activation function make up the buried layer. To fine-tune the performance, we may add as many hidden layers as we like. We're employing three levels in this example. The last layer is the output layer, which contains the categorized output. The result will be either 1 or 0, with 1 indicating a fraud case and 0 indicating a regular situation.

For data processing we have used numpy and pandas and applied PCA and Feature scaling. For visualization we have used matplotlib

Following are the model's parameters used in training the model:



```
model = keras.Sequential([
    keras.layers.Dense(256, activation='relu', input_shape=(X_train.shape[-1],)),
    keras.layers.BatchNormalization(),
    keras.layers.Dropout(0.3),
    keras.layers.Dense(256, activation='relu'),
    keras.layers.BatchNormalization(),
    keras.layers.Dropout(0.3),
    keras.layers.Dense(256, activation='relu'),
    keras.layers.BatchNormalization(),
    keras.layers.Dropout(0.3),
    keras.layers.Dense(1, activation='sigmoid'),
])
```

Figure 3 shows the loss, accuracy, precision and recall graph:

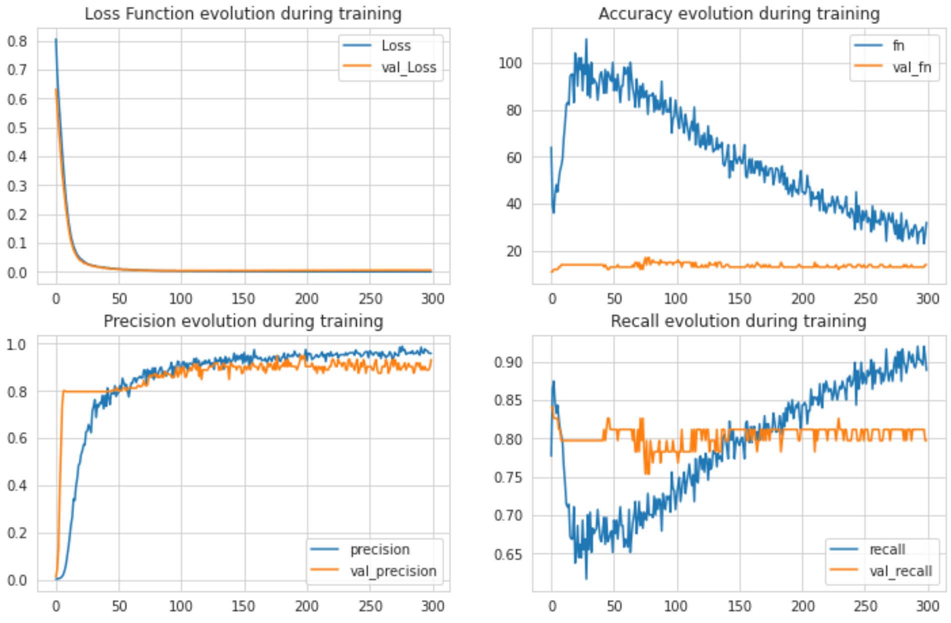


Figure 3: loss, accuracy precision and recall graph

2.4 Analysis

Only a small percentage of transactions are genuinely fraudulent (less than 1 percent). With 492 frauds in a total of 284,807 observations, the data set is substantially skewed. There were just 0.172 percent fraud cases because of this. The minimal number of fraudulent transactions justifies this biased group. The dataset is made up of numerical values from the V1 through V28 'Principal Component Analysis (PCA)' processed features. Furthermore, because there is no information about the original features, no pre-analysis or feature research can be performed. The data in the 'Time' and 'Amount' features has not been converted.

Maximum correlations come from:

- Time & V3 (-0.42)
- Amount & V2 (-0.53)
- Amount & V4 (0.4)

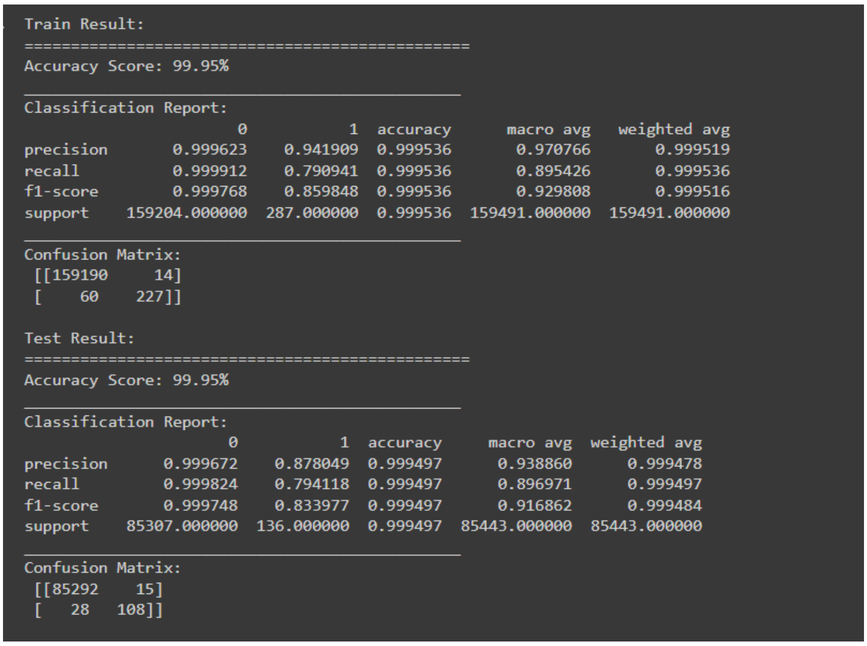
The correlation matrix also demonstrates that none of the V1 through V28 PCA components have any association with each other, although Class does have some positive and negative correlations with the V components, but none with Time or Amount. Time and Amount have been scaled as the other columns in data processing technique.

2.5 Results

Following are the libraries are used for evaluation

- Confusion Matrix
- Classification Report

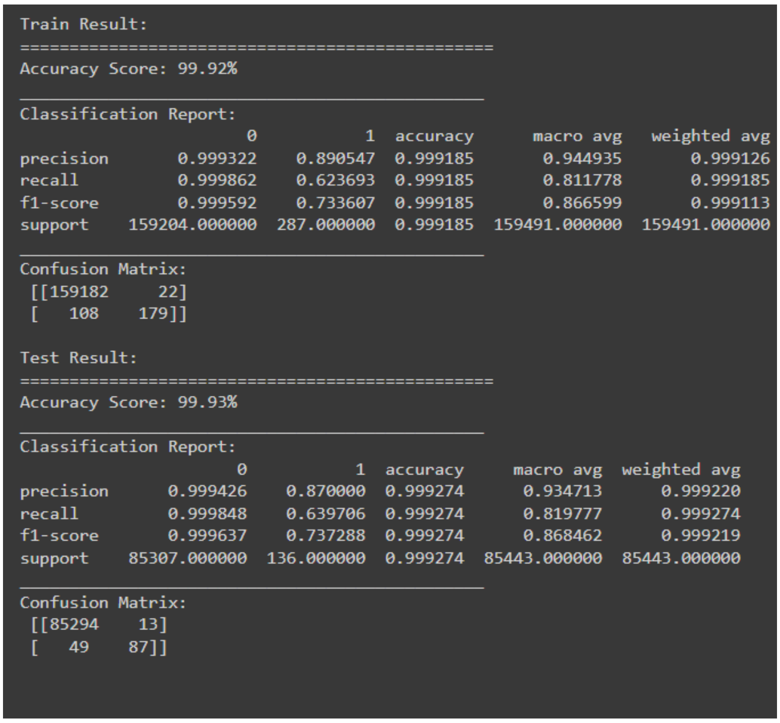
ANN Classification Report: Achieved 99.95% on training and testing data



```
Train Result:
================================================
Accuracy Score: 99.95%
_______________________________________________
Classification Report:
                       0           1  accuracy      macro avg   weighted avg
precision       0.999623    0.941909  0.999536       0.970766       0.999519
recall          0.999912    0.790941  0.999536       0.895426       0.999536
f1-score        0.999768    0.859848  0.999536       0.929808       0.999516
support    159204.000000  287.000000  0.999536  159491.000000  159491.000000
_______________________________________________
Confusion Matrix:
 [[159190     14]
 [    60    227]]

Test Result:
================================================
Accuracy Score: 99.95%
_______________________________________________
Classification Report:
                      0           1  accuracy     macro avg  weighted avg
precision      0.999672    0.878049  0.999497      0.938860      0.999478
recall         0.999824    0.794118  0.999497      0.896971      0.999497
f1-score       0.999748    0.833977  0.999497      0.916862      0.999484
support    85307.000000  136.000000  0.999497  85443.000000  85443.000000
_______________________________________________
Confusion Matrix:
 [[85292    15]
 [   28   108]]
```

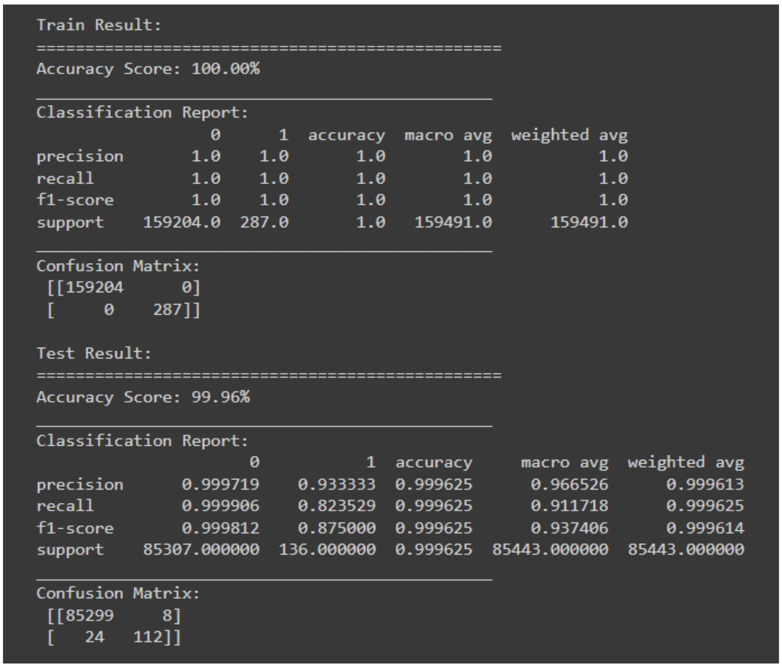
Logistic Regression Classification Report: Achieved 99.92% on training and testing data



```
Train Result:
================================================
Accuracy Score: 99.92%
_______________________________________________
Classification Report:
                       0           1  accuracy      macro avg   weighted avg
precision       0.999322    0.890547  0.999185       0.944935       0.999126
recall          0.999862    0.623693  0.999185       0.811778       0.999185
f1-score        0.999592    0.733607  0.999185       0.866599       0.999113
support    159204.000000  287.000000  0.999185  159491.000000  159491.000000
_______________________________________________
Confusion Matrix:
 [[159182     22]
 [   108    179]]

Test Result:
================================================
Accuracy Score: 99.93%
_______________________________________________
Classification Report:
                      0           1  accuracy     macro avg  weighted avg
precision      0.999426    0.870000  0.999274      0.934713      0.999220
recall         0.999848    0.639706  0.999274      0.819777      0.999274
f1-score       0.999637    0.737288  0.999274      0.868462      0.999219
support    85307.000000  136.000000  0.999274  85443.000000  85443.000000
_______________________________________________
Confusion Matrix:
 [[85294    13]
 [   49    87]]
```

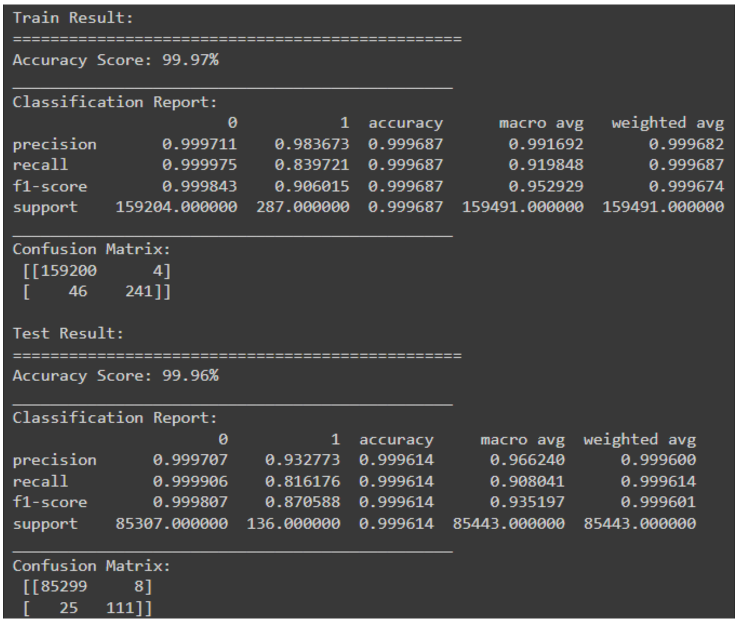
Random Forest Classification Report: Achieved 100% on training and 99.96 on testing data.



```
Train Result:
================================================
Accuracy Score: 100.00%
_______________________________________________
Classification Report:
                  0      1  accuracy  macro avg  weighted avg
precision       1.0    1.0       1.0        1.0           1.0
recall          1.0    1.0       1.0        1.0           1.0
f1-score        1.0    1.0       1.0        1.0           1.0
support    159204.0  287.0       1.0   159491.0      159491.0
_______________________________________________
Confusion Matrix:
 [[159204      0]
 [     0    287]]

Test Result:
================================================
Accuracy Score: 99.96%
_______________________________________________
Classification Report:
                      0           1  accuracy     macro avg  weighted avg
precision      0.999719    0.933333  0.999625      0.966526      0.999613
recall         0.999906    0.823529  0.999625      0.911718      0.999625
f1-score       0.999812    0.875000  0.999625      0.937406      0.999614
support    85307.000000  136.000000  0.999625  85443.000000  85443.000000
_______________________________________________
Confusion Matrix:
 [[85299     8]
 [   24   112]]
```

XGBoost Classification Report: Achieved 99.97% on training and 99.96% on testing data



```
Train Result:
================================================
Accuracy Score: 99.97%
_______________________________________________
Classification Report:
                       0           1  accuracy      macro avg   weighted avg
precision       0.999711    0.983673  0.999687       0.991692       0.999682
recall          0.999975    0.839721  0.999687       0.919848       0.999687
f1-score        0.999843    0.906015  0.999687       0.952929       0.999674
support    159204.000000  287.000000  0.999687  159491.000000  159491.000000
_______________________________________________
Confusion Matrix:
 [[159200      4]
 [    46    241]]

Test Result:
================================================
Accuracy Score: 99.96%
_______________________________________________
Classification Report:
                      0           1  accuracy     macro avg  weighted avg
precision      0.999707    0.932773  0.999614      0.966240      0.999600
recall         0.999906    0.816176  0.999614      0.908041      0.999614
f1-score       0.999807    0.870588  0.999614      0.935197      0.999601
support    85307.000000  136.000000  0.999614  85443.000000  85443.000000
_______________________________________________
Confusion Matrix:
 [[85299     8]
 [   25   111]]
```

Table 1: Comparison of different algorithms

Model	Accuracy
ANN	99.95%
Random Forest	99.96%
Logistic Regression	99.93%
XGBoost	99.96%

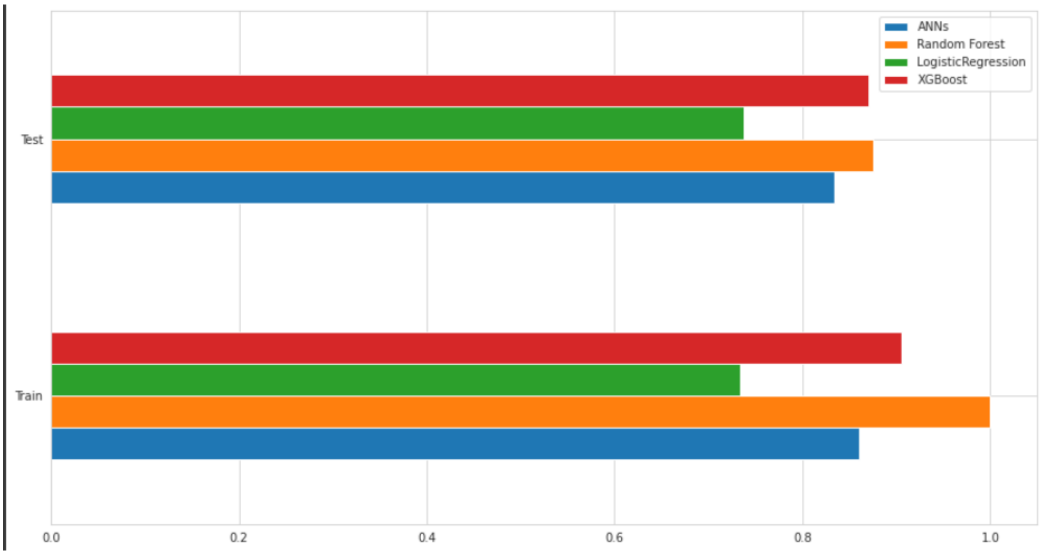


Figure 4: Model Comparison

3. Conclusion

Several algorithms have been implemented on the same data set to detect credit card frauds. All the algorithms have been analyzed and compared on basis of accuracy on basis of predicting normal cases and outliers or frauds. We implemented a different type of algorithms which include a neural network from deep learning, algorithms like Random Forest, Logistic Regression and XGBOOST. This was done to attain the best approach for the purpose. Upon analyzing we get to know that Neural Network are spot-on predicting the normal cases with an accuracy of 99% but in the case of predicting the outliers they are not as good as anomaly detection algorithms like random forest and XGBOOST.

References

[1] A. Srivastava, A. Kundu, S. Sural and A. Majumdar, “Credit card fraud detection using hidden markov model,” In:IEEE transactions on dependable and secure computing, vol. 5, no. 1, Jan.-March 2008, pp. 37-48.

[2] P. Suraj, N. Varsha and S. P. Kumar, “Predictive modelling for credit card fraud detection using data analytics,”Procedia Computer Science, 132, 385-395, 2018.

[3] S. Yusuf, E. Duman, “Detecting credit card fraud by decision trees and support vector machines,” IMECS 2011-International multiconference of Engineers and Computer Scientists 2011, 1, 442-447, 2011.

[4] T. K. Ho, “Random decision forests,” In Proceedings of 3rd international conference on document analysis and recognition, Vol. 1, pp. 278-282, IEEE, August-1995.

[5] S. Bhattacharyya, S. Jha, K. Tharakunnel, J. Westland, “Data mining for credit card fraud: A comparative study, Decision Support Systems, 50, 602-613, 2011.

[6] Y. Sahin, S. Bulkan, E. Duman, “A cost-sensitive decision tree approach for fraud detection” Expert Syst. Appl., 40, 5916-5923, 2013.

[7] S. Panigrahi, A. Kundu, S. Sural, A. Majumdar, “Credit card fraud detection: a fusion approach using dempster– shafer theory and bayesian learning,” Information Fusion, 10, 354-363, 2009.

[8] S. Maes, K. Tuyls, B. Vanschoenwinkel, B. Manderick, “Credit card fraud detection using bayesian and neural networks, 2002.

[9] M. Zareapoor, P. Shamsolmoali, “Application of credit card fraud detection: based on bagging ensemble classifier,” Procedia Computer Science, 48, 679-686, 2015.

[10] L. Zheng, et al, “A new credit card fraud detecting method based on behavior certificate,” IEEE 15th international conference on Networking, Sensing and Control (ICNSC), Zhuhai, pp. 1-6, 2018.

[11] Zareapoor, M., & Shamsolmoali, P. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier. Procedia computer science, 48(2015), 679-685.

[12] Dorronsoro, J. R., Ginel, F., Sgnchez, C., & Cruz, C. S. (1997). Neural fraud detection in credit card operations. IEEE transactions on neural networks, 8(4), 827-834.

[13] Makki, S. (2019). An efficient classification model for analyzing skewed data to detect frauds in the financial sector (Doctoral dissertation, Université de Lyon; Université libanaise).

[14] Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2018). Credit card fraud detection using AdaBoost and majority voting. IEEE access, 6, 14277-14284.

[15] Mittal, S., & Tyagi, S. (2019, January). Performance evaluation of machine learning algorithms for credit card fraud detection. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 320-324). IEEE.