iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IX **Month of publication:** September 2022

DOI: https://doi.org/10.22214/ijraset.2022.46785

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

Credit Card Fraud Detection using Various Machine Learning Algorithms

Akhil Songa[1], Sri Teja Kumar Reddy Tetali[2], Naga Sai Tanmai Raavi[3]
[1, 2, 3]*Drexel University College of Computing & Informatics*

Abstract: ***Detecting credit card fraud is probably the most typical problem in the modern day. This is due to a growth in digital shopping as well as electronic-commerce platforms. As digitalization gains popularity in this current world, people are choosing online payment methods for time and transportation ease, among other reasons.Credit card fraud has increased significantly as a result of the tremendous growth in e-commerce use. Fraudsters attempt to exploit the card and the transparency of online payments. Credit card fraud often occurs whenever the card is taken/stolen or when the details of the card are known. As a result, countering fraudsters' activities has become critical. In the modern world, people are facing loads of credit card issues. There are various techniques and Machine Learning Algorithms to overcome this problem such as using Logistic Regression, XgBoost, Naïve Bayesian, K- Nearest Neighbor (KNN), Decision Tree, Random Forest, Support Vector Machine (SVM) etc.The central goal is to safeguard electronic payments, allowing people to use e-banking safely and easily.***
Keywords: Machine Learning, Xgboost, Logistic Regression, K-nearest neighbor, Decision Tree, Random Forest.

I. INTRODUCTION

Financial fraud has become a critical issue in the corporate and banking industries. Furthermore, financial fraud has a significant impact on the company, economic instability, and people's living standards. In the paper, we are going to look into how to identify credit card frauds. Both businesses and customers are losing money due to financial fraud in credit card transactions.E-payment is made enjoyable, smooth, handy, easy, and simple to use, due to online purchases and payment services; yet, we must not overlook the capital losses that accompany e-commerce [1]. It opens the door to a new sort of deception for crooks. Organizations and banks use effective security solutions to deal with these concerns, but fraudsters vary their subtle approaches over time.As a result, improving detection and preventive strategies is critical. We have credit card transactions physically and virtually. In physical transactions, cards play a significant role in the purpose of transactions; we used to swipe the card and make transactions. On the other hand, In virtual transactions like a CNP situation, we need some essential details like cardholder name, CVV number, passwords to swipe a card for net banking [2]. While dealing with fraud we use two methods: fraud detection and fraud prevention. Fraud prevention is primarily concerned with the prevention of fraud cases, although it also monitors transactions and prohibits legal activities [3]. Whereas in fraud detection,The primary purpose is to discern between real and fake transactions. Using past data, the user's habits and behavior were examined and verified to determine if the transaction/payment was fraudulent or not. When a system fails to prevent fraudulent conduct, fraud detection becomes the responsibility of the individual.

Credit card scams can be in a variety of forms. These forms are shown below in fig:-1

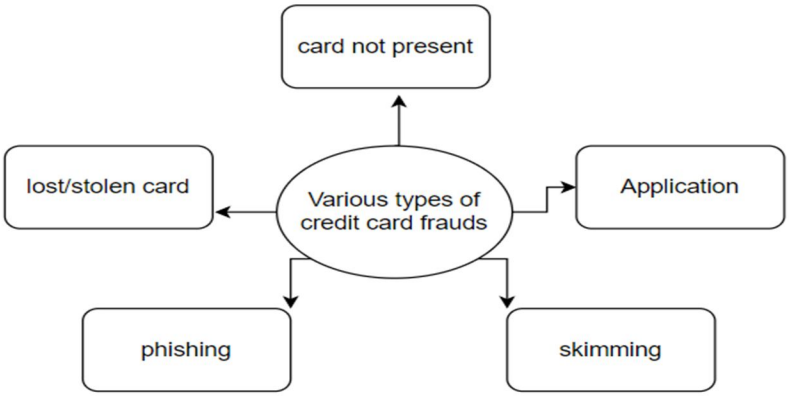


Fig 1:- different forms of credit card scams

CNP fraud, also known as card not present fraud, occurs when a trickster attempts to fool the network via impersonating another individual. Mailing and the internet are important conduits both legitimate mail order and web retailers are affected. In the skimming technique, they obtain personal information on someone else's credit card, which was used in a normal transaction. A skimmer is a tiny device that is used to grab/collect and save most of the victim's information. In the phishing technique, the fraudster/trickster may employ various tactics to fool customers into providing their credit card information by impersonating a bank or payment system on a website. Whenever a card is stolen or misplaced, there is a probability that the thief may conduct an illicit transaction before the cardholder blocks the card [4].

II. LITERATURE SURVEY

Rimpal R. Popat et al. [5] performed a survey on the dangers associated with credit card use and the methods used to commit various forms of financial fraud, most notably credit card fraud. They examined approximately seven popular techniques for detecting credit card fraud on the basis of the learning paradigm, methodologies employed, and difficulties encountered.

K. Vengatesan et al. [6] detected credit card theft using a variety of data analysis approaches. To train and test the machine learning models, we employed Logistic Regression and K Nearest Neighbors (K-NN) algorithms. The models are then validated for correctness. They concluded that K-NN outperformed Logistic Regression.

Navanshu Khare et al. [7] identified credit card fraud using a variety of machine learning models. To train and test the machine learning models, we employed Logistic Regression, Decision Tree, Random Forest, and SVM algorithms. The models are then validated for correctness. They determined that among the trained models, the Random Forest model performed the best.

S P Maniraj et al. [8] presented a few new approaches for detecting 100% fraud in credit transactions worked on data analysis and preparation, as well as implementing several anomaly detection approaches such as the Local Outlier Factor and Isolation Forest algorithm They concluded that when the entire dataset is provided to the algorithm with high precision, the precision increases.

John O. Awoyemi et al. [9] used certain algorithms to detect/identify the scam. On data sets, methods such as K-nearest neighbor, naïve bayes, and logistic regression classifiers are used for modelling, training, and testing. Finally, it was determined that K-nearest neighbor outperforms naïve bayes and logistic regression models. Ruttala Sailusha et al. [10] worked on the Random Forest Algorithm and the Adaboost algorithm to detect credit card fraud. Both methods perform well when all criteria such as accuracy, recall, and F1-score are considered. However, the Random Forest method outperforms the Adaboost algorithm in terms of value. As a consequence of these findings, they concluded that the Random Forest algorithm is the best.

A. Proposed Work

The central objective of this study is to recognize the transactions in a dataset which contains fraud and non-fraudulent transactions by using Machine Learning algorithms such as Random Forest, Decision Tree, Logistic Regression, K-nearest neighbor, XgBoost algorithm. These algorithms are then evaluated to determine which performs best in identifying fraud transactions. In the below diagram the fig 2 shows how the fraud detection system works [8].

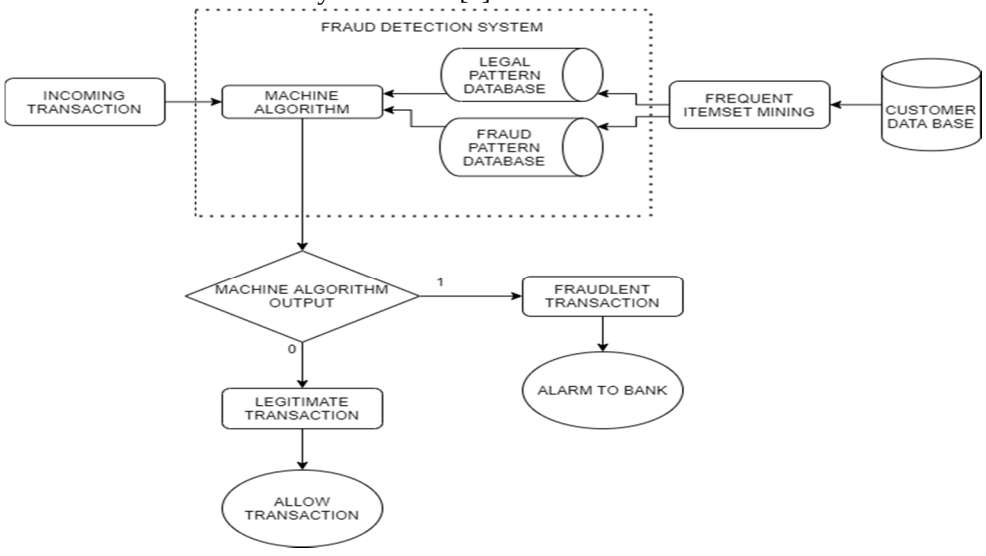


Fig 2:- Working of fraud detection system

B. Decision Tree

It is a supervised algorithm that can perform classification and regression. Decision tree uses tree structure to perform the task. Decision tree contains decision nodes, leaf nodes and branches. The CART algorithm, which is an acronym for "Classification and Regression Tree" algorithm, is used to form a decision tree. In this, the decisions or predictions are based on the characteristics of the provided dataset and the structure of the tree[11].

Entropy: It is nothing more than the degree of skepticism in our dataset or measurement of disorder.

$E(S) = -P(+)\log2\ (p+)\ - p(-)\log2(p-)$

$p(+)$ = probability of positive outcomes

$p(-)$ = probability of negative outcomes

Information gain: It serves as a decisive factor in determining which attribute should be chosen as a decision node/root node.

Information gain = $E(Y)-E(Y/X)$

The Decision Tree Algorithm diagram is plotted below in fig 3.

Algorithm

Step 1:-Choose the best attribute from the dataset as the root node.

Step 2:- Using metrics like entropy and information gain and pick the best feature from the dataset.

Step 3:- By using the best feature it creates the decision tree node.

Step 4:- Repeat the same processes recursively till you get to the leaf nodes of the tree.

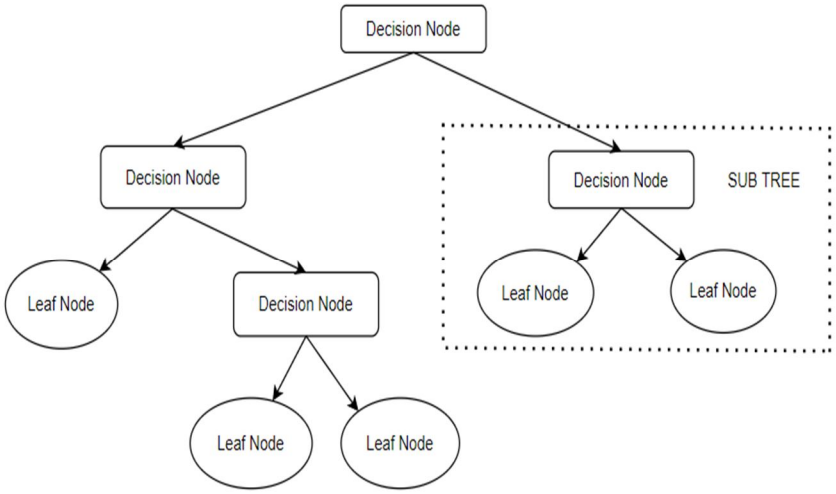


Fig 3:- Decision Tree Algorithm

C. Random Forest

The Random Forest algorithm is an ensemble approach as well as a supervised learning technique. This can be used for regression as well as classification. However, this algorithm is mostly used to solve classification problems. The Random Forest algorithm is simply nothing but generates decision trees based on the data taken and obtains the prediction from every sample data, and predicts the output by averaging the results, which reduces overfitting and is far better than a single Decision Tree [10].

The Random Forest Algorithm diagram is plotted below in fig 4.

Algorithm

Step 1:- N random data samples are collected from the dataset.

Step 2:- From each small dataset, different decision trees are built.

Step 3:- Each individual decision tree will provide different results

Step 4:- The output will be based on the voting or averaging the results

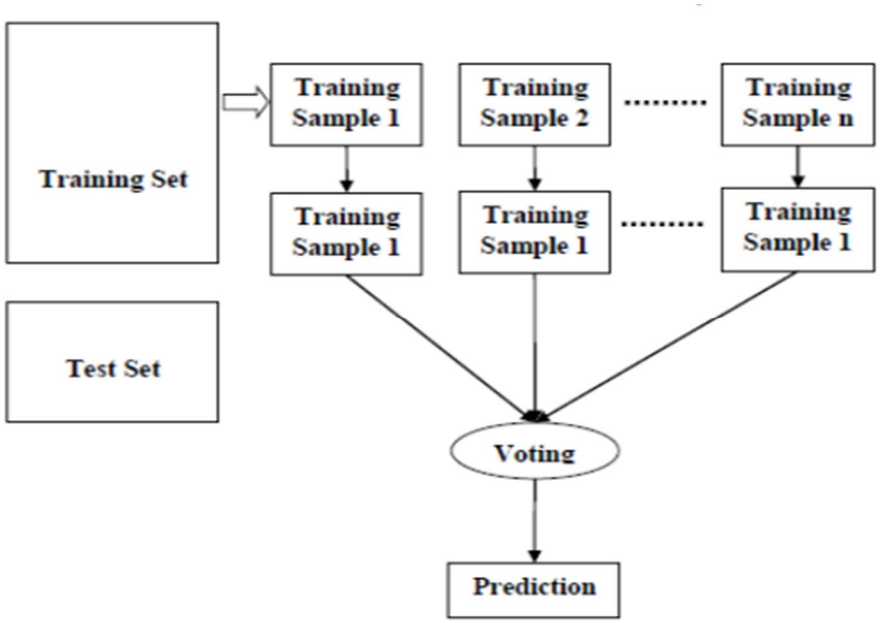


Fig 4:- Random Forest Algorithm

D. XgBoost

The full form of XgBoost is "e**X**treme **G**radient **B**oosting". It is gaining popularity for its speed and performance. The XgBoost algorithm was first implemented by " Tianqi Chen" recently because of its popularity and many developers are contributing to it. XGBoost is an ensemble technique based on decision trees that use a gradient boosting framework [12].
The XgBoost Algorithm diagram is plotted below in fig 6.

Algorithm

Step 1:- Generate the base probability and then calculate residual values for all rows using the XGBoost algorithm.
Step 2:-Now, build the decision tree by splitting the data such that the estimated gain is as high as feasible, where gain = left similarity score + right ss – root ss.
Step 3:-Now examine pruning; if the gain is less than the cover value (pr(1-pr)), the branch should be cut.
Step 4:-When fresh data arrives, use the formula log(odds)=log(p/1-p) to obtain the output of the base model first.
Step 5:-Then, with the new data, proceed through the splits and check the similarity scores of the last split before applying the sigmoid activation function, which is sigmoid (base output + learning rate (similarity score)).
Step 6:-Furthermore, sigmoid = 1/1-(e-value). For each record, a new probability will be generated for each sigmoid value .

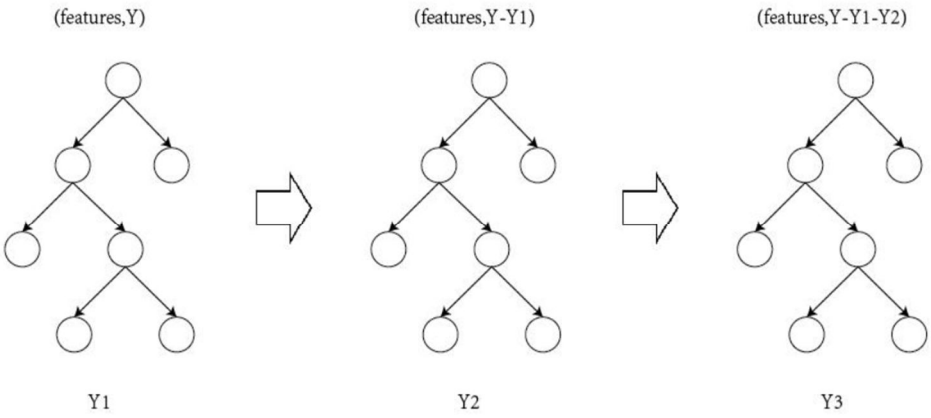


Fig 6:- XgBoost Algorithm

E. Logistic Regression

The logistic regression technique is designed for classification jobs, but it may also be used for regression. Logistic regression is a minor tweak to the linear regression technique. Linear regression techniques cannot conduct classification, however logistic regression may by employing functions such as the sigmoid function. In contrast to linear regression, which uses a straight line and logistic regression uses an S-shaped curve to accomplish two class classification jobs/tasks. The Logistic Regression Algorithm diagram is plotted below in fig 7.

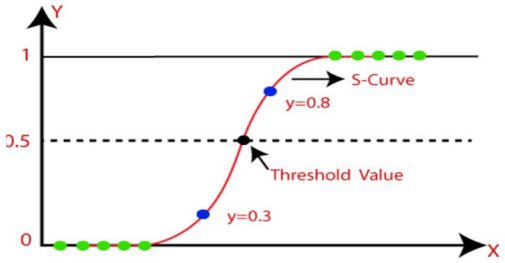


Fig 7:- Logistic Regression Algorithm

The linear equation may be expressed numerically using the following equation.

$z = \beta 0 + \beta 1x1 + \beta 2x2 + \ldots\ldots + \beta nxn$

This anticipated response number (z) is then converted into a probability ranging from 0 to 1. The sigmoid function is used to convert anticipated values to probability values. Using this sigmoid function, every real integer is turned into a probability value between 0 and 1 [13].

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

Algorithm

Step 1:- Data cleaning and preprocessing
Step 2:- Train and fit the logistics regression
Step 3:- Testing the hypothesis
Step 4:- Predicting the results based on the given input.

F. K-nearest neighbor (K-nn)

K-nearest neighbor in short known as KNN is used to perform the clustering. It is an unsupervised learning algorithm in which the data set will contain only X (features) but it doesn't contain any output Y. KNN is a lazy algorithm that doesn't form a hypothesis based on the training data; it just stores the entire data set and performs the task when a new data is entered [14]. The K-nn ALgorithm diagram is plotted below in fig 8

By using these two formulas we can calculate the distance between the new data point and the available data points so that it can categorize to which category it belongs.

1.Euclidean = squareroot((x2-x1)^2 – (y2-y1)^2)
2.Manhattan = |(x2-x1)|+|(y2-y1)|

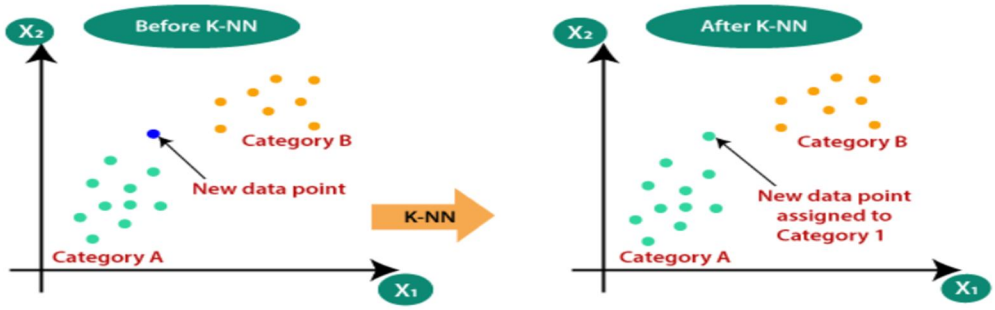


Fig 8:- K-nearest neighbor(K-nn) Algorithm

Algorithm

Step 1:- Fix the value of "k".

Step 2:- Calculate the distance between all the points in the dataset by using distance formula like (Euclidean, manhattan)

Step 3:- Find the "k" closest points based on the distance

Step 4:- Find the distance between the new point and all the points in the dataset, the new point behavior is similar to the closest point to it.

III. EVALUATION AND RESULT ANALYSIS

First and foremost, we collected our dataset from Kaggle which is a data analysis service that offers datasets. It has 31 columns, and 28 are named as v1-v28 to safeguard delicate data. The leftovers labeled/denoted as Time, Amount, and Class. Class 0 illustrates a valid transaction, whereas class 1 depicts a malicious/fraudulent transaction.

Created several graphs to visually understand the dataset and check for anomalies in it.

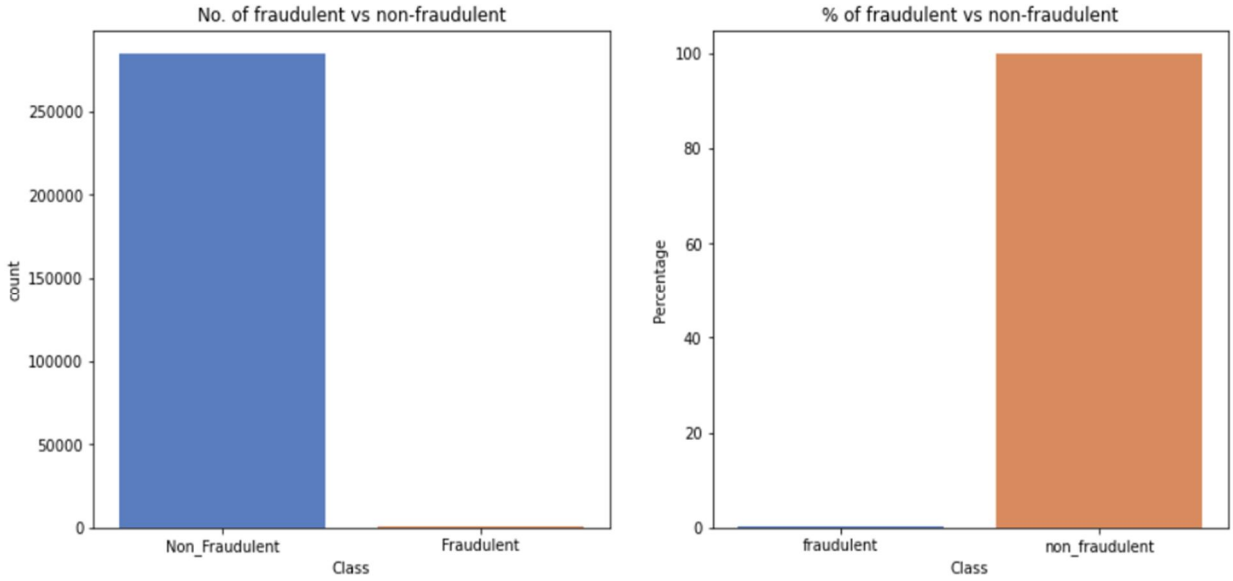


Fig 9 :- number of fraudulent vs non_fraudulent transactions

Based on the graph above in fig 9, we can only conclude that the number of fraudulent/false transactions is far fewer than the number of non-fraudulent/false transactions. As a result, we may state that the dataset is severely skewed..

Fig 10 :- distribution of classes with time

We can't anticipate anything from the preceding graph i.e in fig 10, which shows the distribution of classes over time, because fraud can occur at any moment.

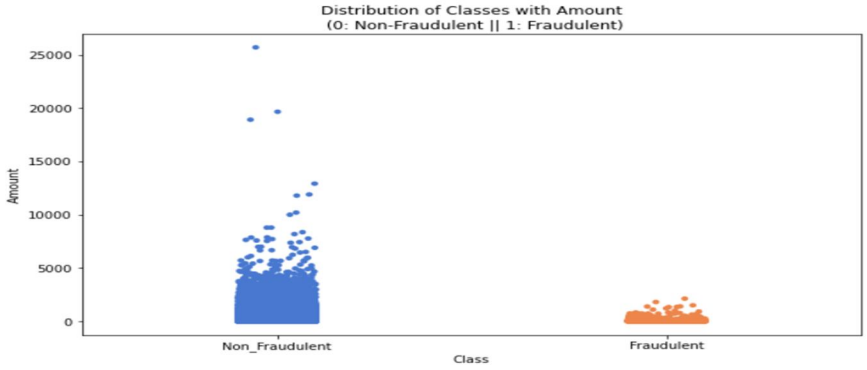


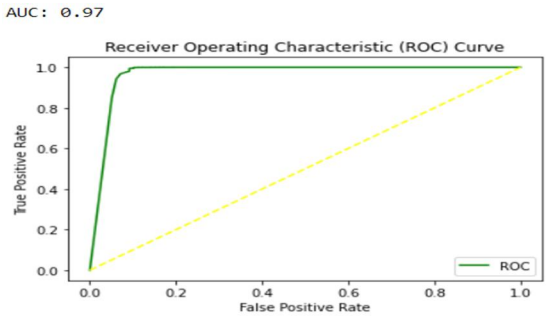
Fig 11:- distribution of classes with amount

Based on this graph i.e in fig 11, we may conclude that fraudulent transactions can occur in the range of 0 to 2500 $.
As we can see from the above visualization graphs, the data is imbalanced. So the data was then balanced using SMOTE and ADASYN methods. Models were tested on both SMOTE and ADASYN data in order to determine which produced the best results. SMOTE is a synthetic minority oversampling technique which seeks to balance class distribution by recreating minority class cases at random. ADASYN means Adaptive Synthetic that produces synthetic data, with the primary benefits of avoiding duplicating/replicating the same data and producing supplementary data for "harder to learn" cases.To compare algorithms, we must include parameters such as accuracy, precision, recall, and F1-score.
Accuracy:- The percentage of correct predictions made by our model is referred to as its accuracy.
Precision:- Precision is the proportion of relevant examples found among the retrieved instances. Recall :- Recall is the proportion of relevant instances found.
F1-score :- By considering their harmonic mean, a classifier's precision and recall are combined into a single statistic.
As the performance matrix for the models, we will apply the ROC curve and calculate the AUC Score.The ROC curve measures the model's performance at various thresholds, which will assist us in determining the best threshold for the model.

A. Result Analysis

For each algorithm, the ROC curve is depicted. When the dataset is applied to various algorithms and by using SMOTE and ADASYN techniques, the model produces varying results.
First, we run the dataset through the random forest model, and the results are shown below.

1) *Random Forest with SMOTE:* The accuracy for Random Forest occurred is 0.9995611109160493. In the fig 12, precision, recall, f1-score are same for non_fraudulent transactions and differ from that of fraud cases. Along with that ROC curve is plotted and AUC score is 0.97.



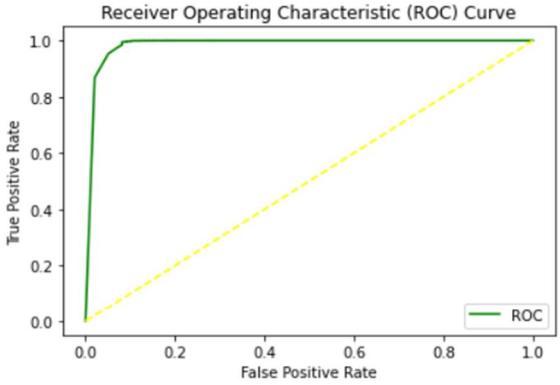
	precision	recall	f1-score	support
Fraudulent	0.94	0.83	0.88	98
Non_Fraudulent	1.00	1.00	1.00	56864
accuracy			1.00	56962
macro avg	0.97	0.91	0.94	56962
weighted avg	1.00	1.00	1.00	56962

0.9685229993798308

Fig 12:- Output for Random Forest with SMOTE

2) *Random Forest with ADASYN:* The accuracy for Random Forest occurred is 0.9995435553526912. In the fig 13, precision, recall, f1-score are same for non_fraudulent transactions and differ from that of fraud cases. Along with that ROC curve is plotted and AUC score is 0.97.

```
AUC: 0.98
```



	precision	recall	f1-score	support
Fraudulent	0.92	0.82	0.86	98
Non_Fraudulent	1.00	1.00	1.00	56864
accuracy			1.00	56962
macro avg	0.96	0.91	0.93	56962
weighted avg	1.00	1.00	1.00	56962

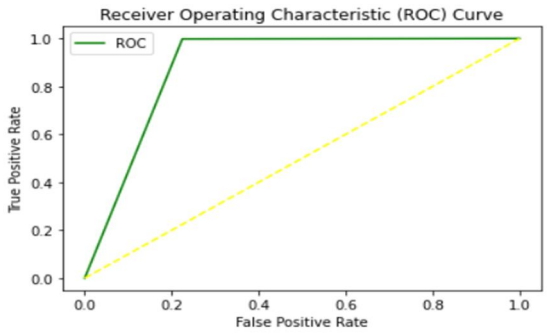
```
0.9846768300736163
```

Fig 13 :- Output for Random Forest with ADASYN

Secondly, the dataset is run through a Decision Tree model, and the results are shown below.

3) *Decision Tree with SMOTE:* The accuracy for Decision Tree occurred is 0.99750711000316. In the fig 14, precision, recall, f1-score are same for non_fraudulent transactions and differ from that of fraud cases. Along with that ROC curve is plotted and AUC score is 0.89.

```
AUC: 0.89
```



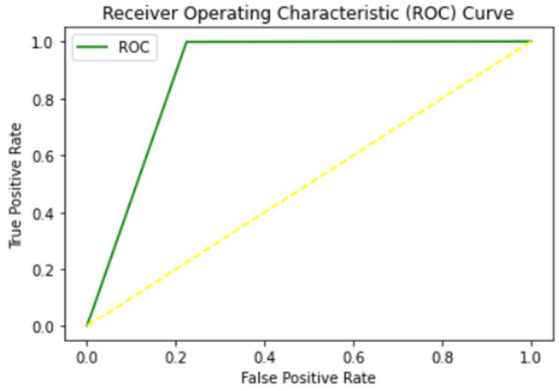
	precision	recall	f1-score	support
Fraudulent	0.40	0.78	0.53	98
Non_Fraudulent	1.00	1.00	1.00	56864
accuracy			1.00	56962
macro avg	0.70	0.89	0.77	56962
weighted avg	1.00	1.00	1.00	56962

```
0.8867702961882558
```

Fig 14:- Output for Decision Tree with SMOTE

4) *Decision Tree with ADASYN:* The accuracy for Decision Tree occurred is 0.9977528878901724. In the fig 14, precision, recall, f1-score are same for non_fraudulent transactions and differ from that of fraud cases. Along with that ROC curve is plotted and AUC score is 0.89.

```
AUC: 0.89
```



	precision	recall	f1-score	support
Fraudulent	0.46	0.78	0.58	98
Non_Fraudulent	1.00	1.00	1.00	56864
accuracy			1.00	56962
macro avg	0.73	0.89	0.79	56962
weighted avg	1.00	1.00	1.00	56962

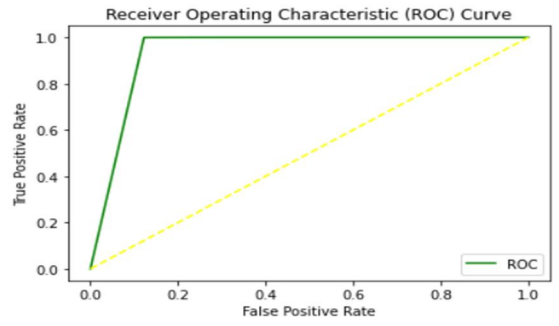
```
0.8869813260138044
```

Fig 15 :-Output for Decision Tree with ADASYN

Thirdly, the dataset is run through a K-nearest neighbor model, and the results are shown below.

5) *K-nearest Neighbor with SMOTE:* The accuracy for K-nearest neighbor occurred is 0.9980864435939749. In the fig 16, precision, recall, f1-score are same for non_fraudulent transactions and differ from that of fraud cases. Along with that ROC curve is plotted and AUC score is 0.94.

```
AUC: 0.94
```



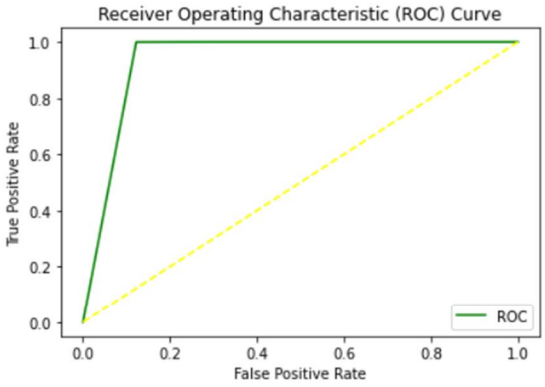
	precision	recall	f1-score	support
Fraudulent	0.96	0.77	0.85	98
Non_Fraudulent	1.00	1.00	1.00	56864
accuracy			1.00	56962
macro avg	0.98	0.88	0.93	56962
weighted avg	1.00	1.00	1.00	56962

```
0.938721047999954
```

Fig 16:- Output for K-nearest neighbor with SMOTE

6) *K-nearest neighbor with ADASYN:* The accuracy for K-nearest neighbor occurred is 0.9980688880306169. In the fig 17, precision, recall, f1-score are same for non_fraudulent transactions and differ from that of fraud cases. Along with that ROC curve is plotted and AUC score is 0.94.

AUC: 0.94



	precision	recall	f1-score	support
Fraudulent	0.96	0.77	0.85	98
Non_Fraudulent	1.00	1.00	1.00	56864
accuracy			1.00	56962
macro avg	0.98	0.88	0.93	56962
weighted avg	1.00	1.00	1.00	56962

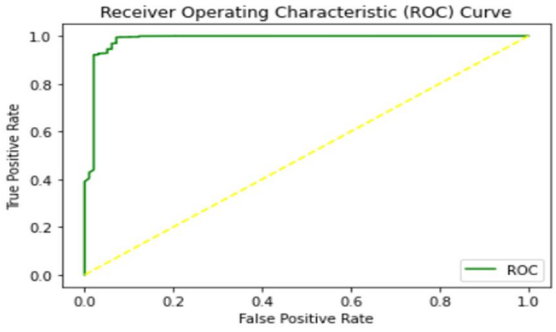
0.938721047999954

Fig 17 :- Output for K-nearest neighbor with ADASYN

Fourthly, the dataset is run through a XgBoost model, and the results are shown below.

7) *XgBoost with SMOTE:* The accuracy for Xg Boost occurred is 0.9991397773954567. In the fig 18, precision, recall, f1-score are same for non_fraudulent transactions and differ from that of fraud cases. Along with that ROC curve is plotted and AUC score is 0.98.

AUC: 0.98



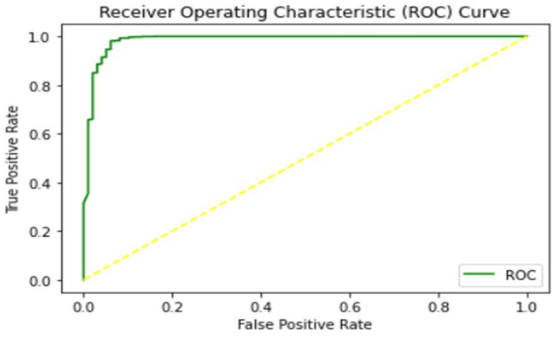
	precision	recall	f1-score	support
Fraudulent	0.69	0.87	0.77	98
Non_Fraudulent	1.00	1.00	1.00	56864
accuracy			1.00	56962
macro avg	0.84	0.93	0.88	56962
weighted avg	1.00	1.00	1.00	56962

0.9846460548907239

Fig 18 :- Output for XgBoost with SMOTE

8) *XgBoost with ADASYN:* The accuracy for Xg Boost occurred is 0.9989817773252344. In the fig 19, precision, recall, f1-score are same for non_fraudulent transactions and differ from that of fraud cases. Along with that ROC curve is plotted and AUC score is 0.99.

```
AUC: 0.99
```



	precision	recall	f1-score	support
Fraudulent	0.64	0.87	0.74	98
Non_Fraudulent	1.00	1.00	1.00	56864
accuracy			1.00	56962
macro avg	0.82	0.93	0.87	56962
weighted avg	1.00	1.00	1.00	56962

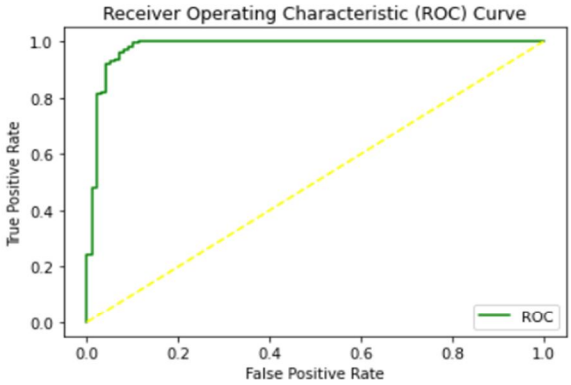
```
0.9850280978317043
```

Fig 19:- Output for XgBoost with ADASYN

Finally, the dataset is run through a Logistic Regression model, and the results are shown below.

9) *Logistic Regression with SMOTE:* The accuracy for Logistic Regression occurred is 0.98039759622730054. In the fig 20, precision, recall, f1-score are same for non_fraudulent transactions and differ from that of fraud cases. Along with that ROC curve is plotted and AUC score is 0.98.

```
AUC: 0.98
```



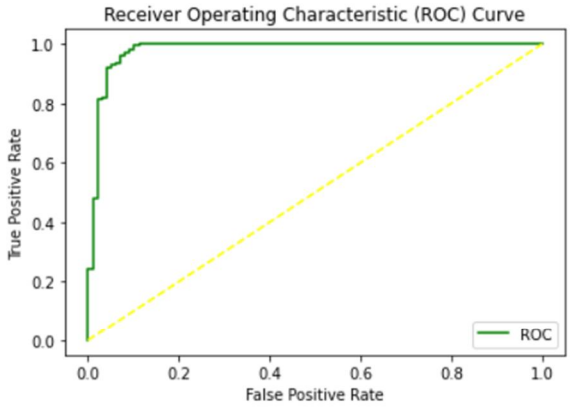
	precision	recall	f1-score	support
Fraudulent	0.87	0.54	0.67	98
Non_Fraudulent	1.00	1.00	1.00	56864
accuracy			1.00	56962
macro avg	0.93	0.77	0.83	56962
weighted avg	1.00	1.00	1.00	56962

```
0.9799370930139079
```

Fig 20 :- Output for Logistic Regression with SMOTE

10) *Logistic Regression with ADASYN:* The accuracy for Xg Boost occurred is 0.9803975962273005. In the fig 20, precision, recall, f1-score are same for non_fraudulent transactions and differ from that of fraud cases. Along with that ROC curve is plotted and AUC score is 0.98.

```
AUC: 0.98
```



	precision	recall	f1-score	support
Fraudulent	0.87	0.54	0.67	98
Non_Fraudulent	1.00	1.00	1.00	56864
accuracy			1.00	56962
macro avg	0.93	0.77	0.83	56962
weighted avg	1.00	1.00	1.00	56962

```
0.9799370930139079
```

IV. CONCLUSION

Despite the fact that there are multiple techniques to detecting fraud, we cannot claim that any one algorithm or model entirely identifies fraud. Based on the data, it can be concluded that the Random Forest Algorithm and the K-Nearest Neighbor Algorithm produce considerably superior results than other Algorithms. When comparing the results of the algorithms used, the following metrics are used: Random Forest Algorithm with ADASYN gives Accuracy:- 0.99, f1-score:-0.86, Recall:- 0.82, precision:-0.92, AUC score:- 0.98, and K-Nearest Neighbor Algorithm with SMOTE and ADASYN gives same results as the Accuracy:-0.99, f1-score:- 0.85, Recall:- 0.77, precision:- 0.96, AUC score:- 0.94. When these two outcomes are compared, the Random Forest Algorithm delivers slightly better results than the K-nearest neighbor algorithm. Finally, we can conclude that the Random Forest Algorithm is the most suitable for this dataset and appropriate for this model .

REFERENCES

[1] N. Mahmoudi, E. Duman, “Detecting credit card fraud by Modified Fisher Discriminant Analysis”, Elsevier Expert System with Application, 2015, pp. 2510-2516.
[2] M. Zareapoor, K. Seeja, M. Alam, “Analysis of credit cardfraud detection techniques: based on certain design criteria”,. International Journal Compututer Application, 2012, pp. 35–42.
[3] Y. Sachin, E. Duman, “Detecting Credit Card Fraud by Decision Tree and Support Vector Machine”, In Proceedings of the international multi Conference of Engineers and Computer Scientists, Hong Kong, 2011, pp. 1-6.
[4] J. Quah, M. Shriganesh, “Real-time credit card fraud detection using Computational Intelligence”, Expert System Application, 2008, pp. 1721-1732.
[5] Khare, N. and Sait, S.Y., 2018. Credit card fraud detection using machine learning models and collating machine learning models. International Journal of Pure and Applied Mathematics, 118(20), pp.825-838.
[6] Popat, R.R. and Chaudhary, J., 2018, May. A survey on credit card fraud detection using machine learning. In 2018 2nd international conference on trends in electronics and informatics (ICOEI) (pp. 1120-1125). IEEE.
[7] Vengatesan, K., Kumar, A., Yuvraj, S., KUMAR, V. and Sabnis, S., 2020. Credit card fraud detection using data analytic techniques. Advances in Mathematics: Scientific Journal, 9(3), pp.1185-1196.
[8] Maniraj, S.P., Saini, A., Ahmed, S. and Sarkar, S., 2019. Credit card fraud detection using machine learning and data science. International Journal of Engineering Research and, 8(09).
[9] Awoyemi, J.O., Adetunmbi, A.O. and Oluwadare, S.A., 2017, October. Credit card fraud detection using machine learning techniques: A comparative analysis.

In 2017 international conference on computing networking and informatics (ICCNI) (pp. 1-9). IEEE.

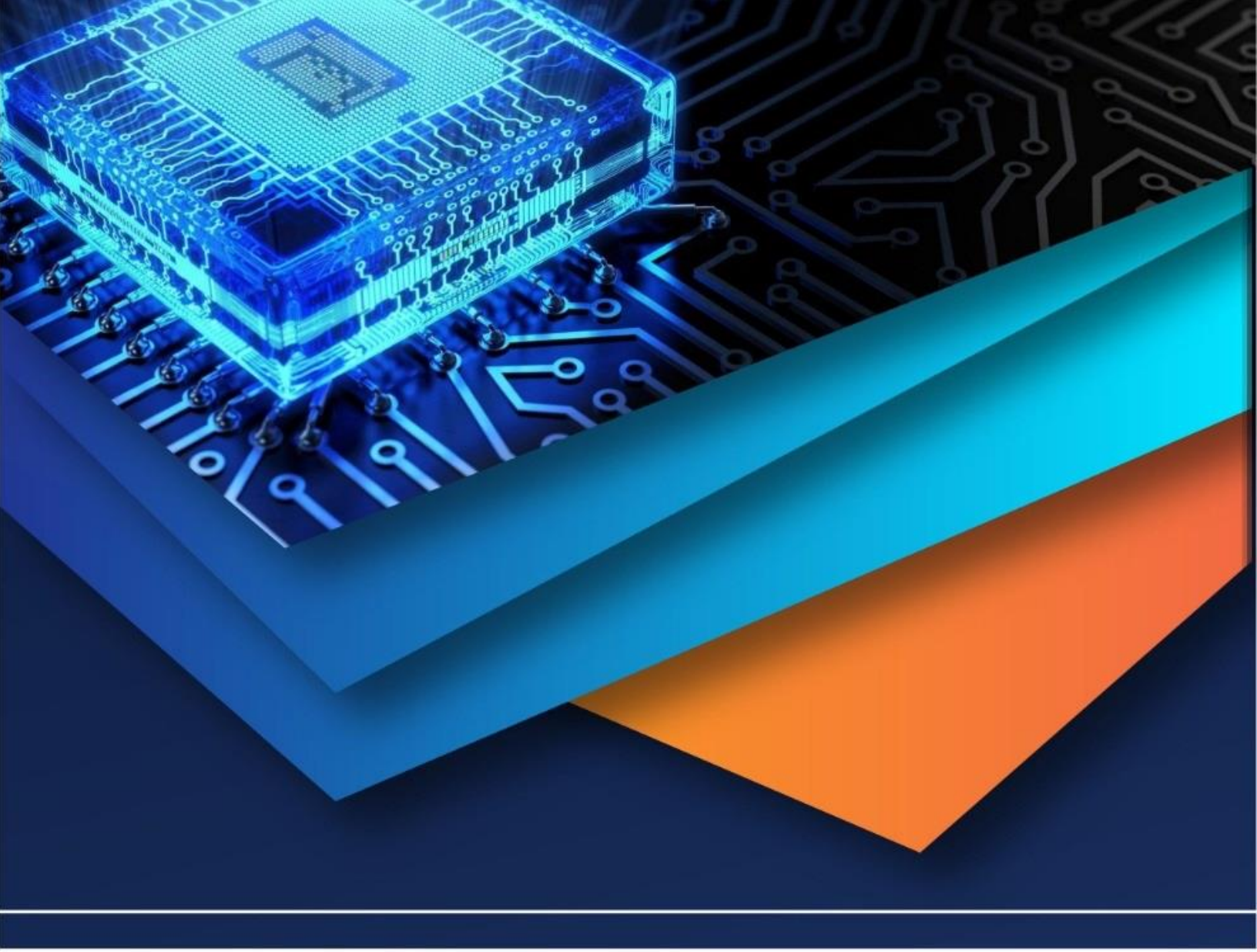
[10] Sailusha, R., Gnaneswar, V., Ramesh, R. and Rao, G.R., 2020, May. Credit card fraud detection using machine learning. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1264-1270). IEEE.

[11] E. Kirkos, C. Spathis, Y. Manolopoulos, “Data mining techniques forthe detection of fraudulent financial statements”, ExpertSystems with Applications, 2007, pp. 995–1003.

[12] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H. and Chen, K., 2015. Xgboost: extreme gradient boosting. R package version 0.4-2, 1(4), pp.1-4.

[13] Hosmer, D.W., Hosmer, T., Le Cessie, S. and Lemeshow, S., 1997. A comparison of goodness-of-fit tests for the logistic regression model. Statistics in medicine, 16(9), pp.965-980.

[14] Guo, G., Wang, H., Bell, D., Bi, Y. and Greer, K., 2003, November. KNN model-based approach in classification. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 986-996). Springer, Berlin, Heidelberg.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY