Credit Card Fraud Detection using Decision Tree and Random Forest

Dhwanir Shah[1*] and *Lokesh* Kumar Sharma[2]

[1]Department of Computer Science, St. Xavier's College (Autonomous), Ahmedabad
[2]Germany, National Institute of Occupational Health, Ahmedabad

Abstract. It is the time of technology advancement. Due to internet everything is available at the touch of a finger. There is a benefit of online shopping: first it saves lots of time and second it does not demand to go to market to buy anything. There exists various mode of payments and credit card payment is one of them. Today, there exists a good number of credit card users in the world. Every day so many credit cards transactions are taken place. Some of these transactions are fraudulent. Due to such fraudulent transactions banks and customers need to suffer. In order to prevent financial losses due to credit card fraud, a secure credit card fraud detection system is essential. Various machine learning algorithms like Naïve Bayes, Logistic regression, SVM, Decision trees, Random Forest, Genetic algorithm, J48 and AdaBoost, etc. are used for credit card fraud detection. The motive of this paper is to provide some insight about the credit card fraud along with the analysis of the dataset and also Decision tree and random forest algorithms are going to be discussed.

1 Introduction

In the time of a pandemic, online shopping has proven beneficial to the public, as they can purchase anything they want from the comfort of their homes. Online payment is comfortable, convenient, and easy to use. Now a days, at the time of shopping, many people use a credit card for payment purposes. A credit card can be described as a thin, rectangular piece of plastic or metal issued to a number of users that can be used as one of the modes of payment. Generally, credit cards offer certain credit limits, which can be used to make purchases, transfer balances, or make cash advances, and it is essential that the user pay back the loan amount in the future.

A credit card user needs to pay a minimal remittance every month by the due date on the balance. It is a fact that there are a good number of advantages to credit card usage, but we cannot ignore the financial losses that normally result from online payments done through a credit card. Nowadays, criminals use a credit card to commit fraud. We can describe a fraud as using money, goods, or services in an illegal manner. Credit card fraud can be described as when a person uses another person's credit card for personal reasons while the card owner and the card issuer company are both unaware of it.

[*]Corresponding author: dhwanir@gmail.com

The person who is using another person's credit card does not have any connection with the cardholder. The figure for the number of credit card users has increased in several countries, but due to a lack of trust in the payment system, many users don't use credit cards for payment or have abandoned the use of cards. Therefore, there is a need for a reliable fraud detection system so that credit card users can use their cards safely. Fraud detection can be described as a classification problem. A fraud can be detected after examining a large number of transactions, identifying them, and then categorising them into fraudulent and genuine transactions. Different types of credit card fraud exist; a few of them are: application fraud, duplication fraud, identity fraud, skimming, CNP, lost and stolen card fraud, mail non-receipt card fraud, account takeover, triangulation, merchant collusion, and site cloning.

2 Literature Review

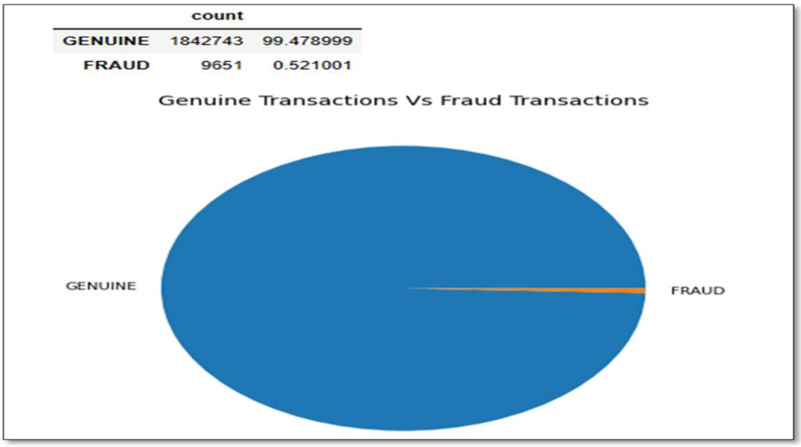
In this section, various researchers' work is going to be discussed.

K. RamaKalyani and Prof. Dr. D. Uma Devi have used genetic algorithms in their research to show how fraud is detected and how false alerts are reduced by using the customer's behaviour. According to them, if this algorithm is applied to bank credit card fraud detection systems, the probability of fraudulent transactions can be predicted soon after credit card transactions occur [1]. Rimpal R. Popat and Mr. Jayesh Chaudhary have discussed the basic information about different types of credit card fraud and also explained the usefulness of the data mining approach in fraud detection in brief. After a comparison of various machine learning algorithms, they have reached the conclusion that machine learning is preferred because of its high accuracy and detection rate [2]. S P Maniraj, Aditya Saini, Swarna Deep Sarkar, and Shadab Ahmed have used local outlier factors with isolation forests algorithms. The algorithm is achieving 99.6% accuracy and 28% precision for a tenth of the dataset taken into consideration [3]. Aman Gulati, Prakash Dubey, MdFuzailC, Jasmine Norman, and Mangayarkarasi R have taken customer behaviour and his location information into consideration for making the decision on whether the transaction is fraudulent or not. If a customer's behaviour pattern or location has changed or is different, then the transaction will be considered doubtful, and the bank will be reported to take further action. They have used the NN algorithm, which gives 80% accuracy with transaction data. The system also has the drawback that it cannot identify a fraud transaction if the fraudster is a new user in the bank [4]. Andhavarapu Bhanusri, K. Ratna Sree Valli, P. Jyothi, G. Varun Sai, and R. Rohit Sai Subash have compared Naive Bayes, Logistic Regression, Random Forest, and AdaBoost algorithms, and the performance of these algorithms has been calculated on the basis of various criteria. They have reached the conclusion that random forest with boosting technique is better in comparison to the other two [5]. John O. Awoyemi, Mr. Adebayo O. Adetunmbi, and Mr. Samuel A. Oluwadare have used Nave Bayes and K-Nearest Neighbourhood and logistic regression algorithms were developed, and the implementation of these algorithms has been done in Python. In order to solve the problem of data unbalancing, they have used oversampling and undersampling techniques, so the imbalance dataset will be converted into two datasets. The algorithms performances have been evaluated on the basis of various metrics [6]. Heta Naik has tried to compare algorithms like KNN, Random Tree, AdaBoost, and Logistic Regression and concluded that Logistic Regression is far better than the other three algorithms. It has been observed that these algorithms are not applicable for fraud detection while the transaction is in progress [7]. Varun Kumar K S, Mr. Vijaya Kumar V G, Mr. Vijay Shankar A, and Ms. Pratibha K have used time and amount features to detect and decide whether the transaction is fraudulent or not. Different algorithms have been compared on the basis of matrices like accuracy, precision, and recall, and it has been concluded that ANN gives the best accuracy [8]. Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and Andras Anderla have compared various algorithms like Logistic Regression, Random Forest, Naive Bayes, and Multilayer Perceptron on the basis of accuracy, recall, and precision and concluded that Random Forest is a more suitable algorithm for credit card fraud [9]. Navanshu Khare and Saad Yunus Sait have presented a paper in which the decision tree, support vector machines, logistic regression, and random forest algorithms have been compared on the basis of various metrics and reached the conclusion that random forest is more accurate in comparison with the other three [10].

3 Experimental Methodology

3.1 Data Analysis

The dataset and reference for analysis have been taken from the Kaggle site [12]. In this paper, we have tried to show various graphs that will provide more insight into the information in more user-friendly manner than the reference material. The dataset is a simulated credit card transaction dataset containing legitimate and fraudulent transactions from the duration January 1, 2019 to December 31, 2020. It can be seen from the following figure that the dataset is highly imbalanced.

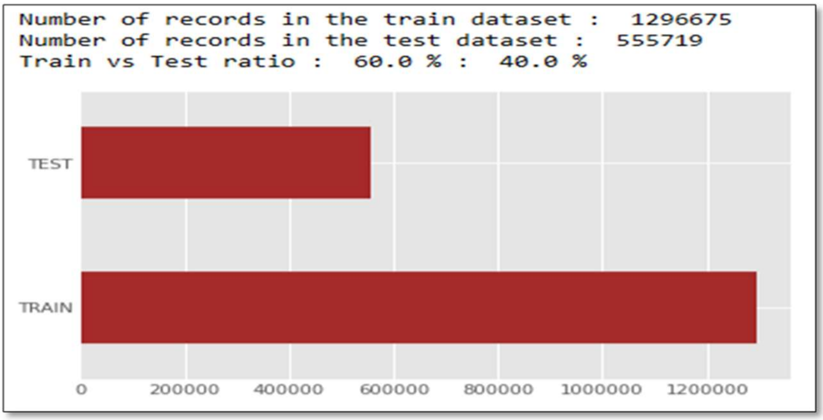


	count	
GENUINE	1842743	99.478999
FRAUD	9651	0.521001

Fig. 1. Dataset Imbalance

As part of the data cleaning, it has been taken care to check for null values as well as duplicate records in the dataset. Type casing hasbeen applied wherever; it was required. As part of the data preprocessing, OneHotEncoding method is used for category and gender features. The Target Guided Mean encoding method is used for state and trans_dayofweek features.

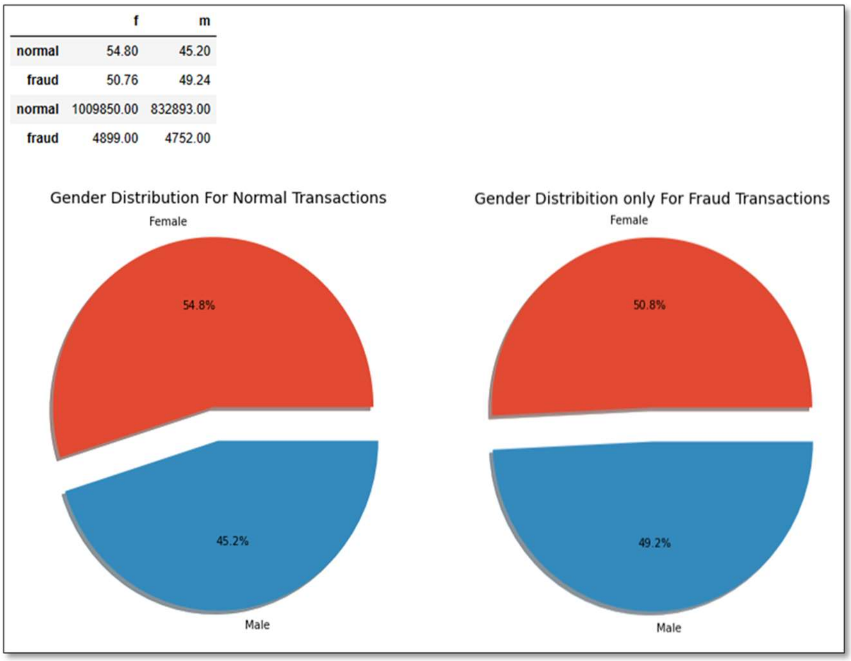
The number of records in the training dataset is 129667. The number of records in the test dataset is 555719. It can be seen that the ratio between the training and test datasets is 60.0%:40.0%.



```
Number of records in the train dataset :   1296675
Number of records in the test dataset :  555719
Train vs Test ratio :   60.0 % :   40.0 %
```

Fig. 2. Training Vs Test Dataset

The following figure shows the gender distribution among male and female card holders for fraud as well as normal transactions. It canalso be observed that the fraud transactions are equally distributed among male and female card owners.



	f	m
normal	54.80	45.20
fraud	50.76	49.24
normal	1009850.00	832893.00
fraud	4899.00	4752.00

Gender Distribution For Normal Transactions
Female
54.8%
45.2%
Male
Gender Distribition only For Fraud Transactions
Female
50.8%
49.2%
Male

Fig. 3. Gender wise Fraud Non-Fraud Transactions distribution

From the following figure, it can be observed that when genuine card owners are sleeping between the 21st and to 04 hour at that time majority of fraud transactions are taken place.

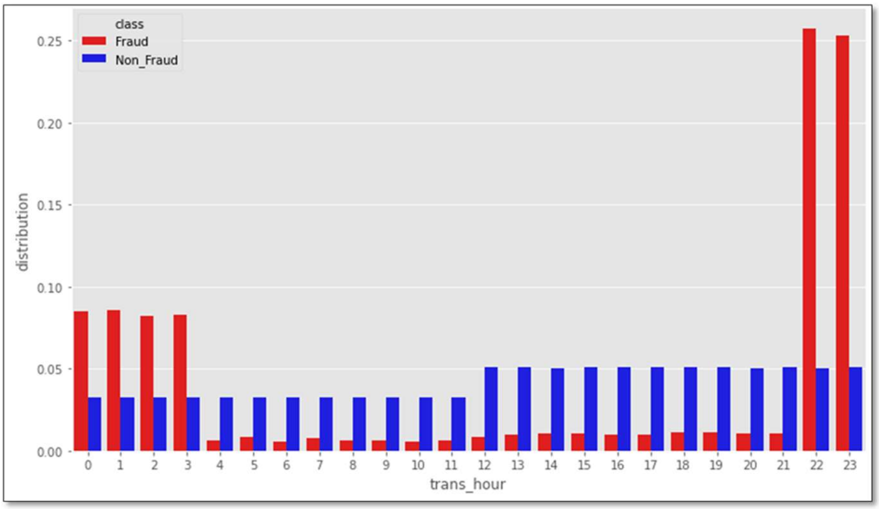


Fig. 4. Hour wise Transactions distribution

From the following figure, it can be seen that fraud transactions are occurring more in categories like grocery_pos, shopping_net, and misc_net. The attribute "is_fraud" with a value 1 indicates that the transaction is fraudulent, and 0 indicates that the transaction is not-fraudulent.

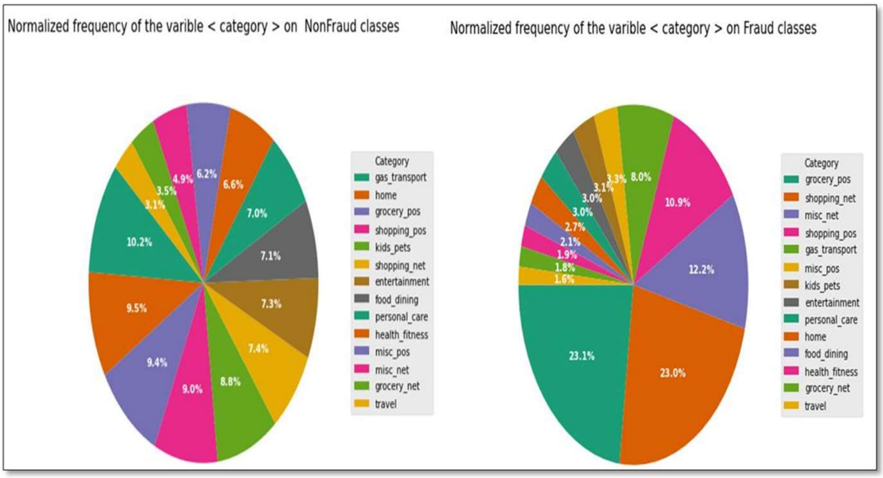


Fig. 5. Category wise Transactions distribution

3.2 Design And Implementation of Algorithms

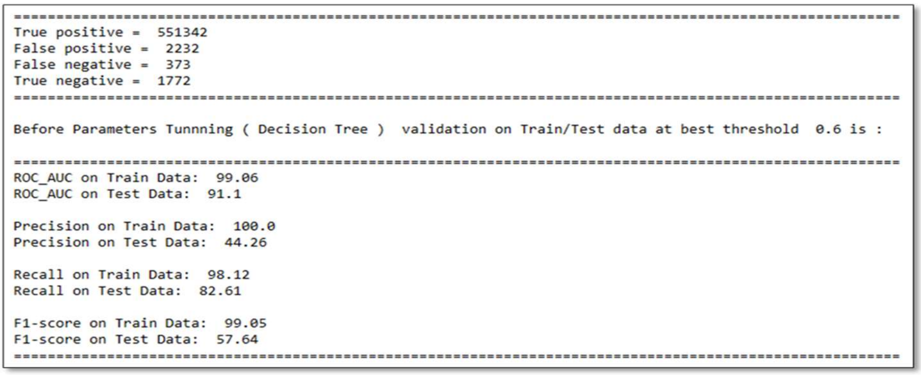
As we have seen in the previous section, that dataset is imbalanced. It has been observed that ML algorithms find difficulty in learning when classification category data are not equally distributed. Because of the large volume of data, parameter tuning may take longer. Therefore, we have taken samples from both train and test datasets to work with the different models. To balance the imbalanced dataset, we have taken the help of SMOTE and RandomUnderSampler techniques. For parameter Tuning, we have used HalvingRandomSearchCV because it is faster than GridSearchCV and RandomizedSearchCV.

Here, the data has been tested by using various ML algorithms. For any data science work, a few packages are vital to use. During the implementation, NumPy (numeric Python) is used for numeric calculation, Pandas is used for reading data and storing it in specific variables, Matplotlib is used for visualizing the data, and Seaborn is used for customization like colour setting. Anaconda navigator is used to implement machine learning algorithms. Jupyter Notebook is used to process the written code. For implementation purposes, decision trees and random forest algorithms are used.

A decision tree is one of the supervised algorithms. It is used to solve classification and regression problems. Decision trees always begin with a root node, which can be considered a starting point situated at the top. Tree is followed by splits that produce branches. A leaf node does not produce any new branches, and it results in a terminal node. Decision trees use the concept of entropy. Entropy indicates the measure of variance in the data among separate classes.

The random forest classifier is applicable for multiple conditions. It is an improved version of a decision tree classifier. A decision tree classifier is applicable to one condition only, while a random forest is applicable to multiple conditions.

Due to the nature of the dataset, any classifier will have 100% accuracy. So, accuracy is not the proper metrics and therefore, other metrices are used for evaluating the performance of the model. In the paper [11], various classifiers' results have been compared with an analysis of them, and using that reference, we have decided to use decision trees and random forest algorithms for our dataset. The following figure shows the result of the performance evaluation of the decision tree classifier before parameter tuning.

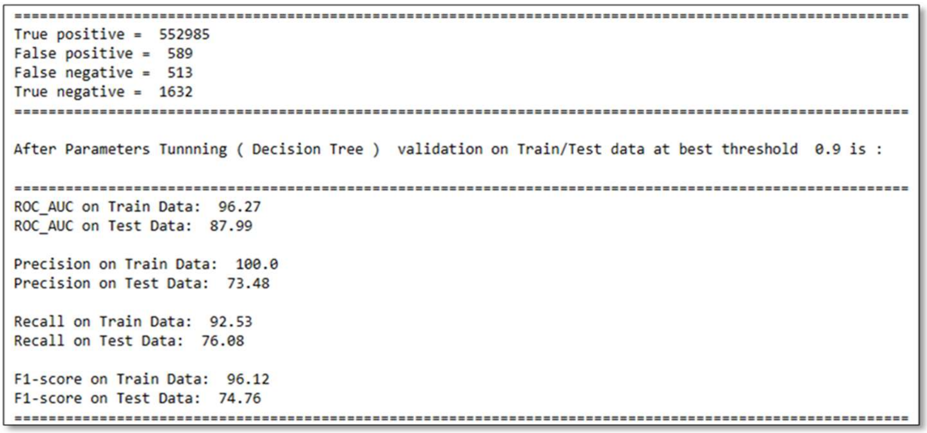


```
============================================================================================================
True positive =  551342
False positive =  2232
False negative =  373
True negative =  1772
============================================================================================================

Before Parameters Tunnning ( Decision Tree )  validation on Train/Test data at best threshold  0.6 is :

============================================================================================================
ROC_AUC on Train Data:  99.06
ROC_AUC on Test Data:  91.1

Precision on Train Data:  100.0
Precision on Test Data:  44.26

Recall on Train Data:  98.12
Recall on Test Data:  82.61

F1-score on Train Data:  99.05
F1-score on Test Data:  57.64
============================================================================================================
```

Fig. 6. Performance Evaluation of Decision Tree Before Parameter Tuning

As mentioned, hyper parameter Tuning has been done with HalvingRandomSearchCV and after parameter tuning, we have identified min_samples_split, min_samples_leaf, and 'max_depth' as the best parameters. Figure 7 shows the decision tree classifier performance and confusion matrix after parameter tuning. Figure 10 shows the confusion matrix of the decision tree classifier after the parameter tuning.

Fig. 7. Performance Evaluation of Decision Tree after Parameter Tuning



```
====================================================================================================
True positive =  552985
False positive =  589
False negative =  513
True negative =  1632
====================================================================================================

After Parameters Tunnning ( Decision Tree )  validation on Train/Test data at best threshold  0.9 is :

====================================================================================================
ROC_AUC on Train Data:  96.27
ROC_AUC on Test Data:  87.99

Precision on Train Data:  100.0
Precision on Test Data:  73.48

Recall on Train Data:  92.53
Recall on Test Data:  76.08

F1-score on Train Data:  96.12
F1-score on Test Data:  74.76
====================================================================================================
```

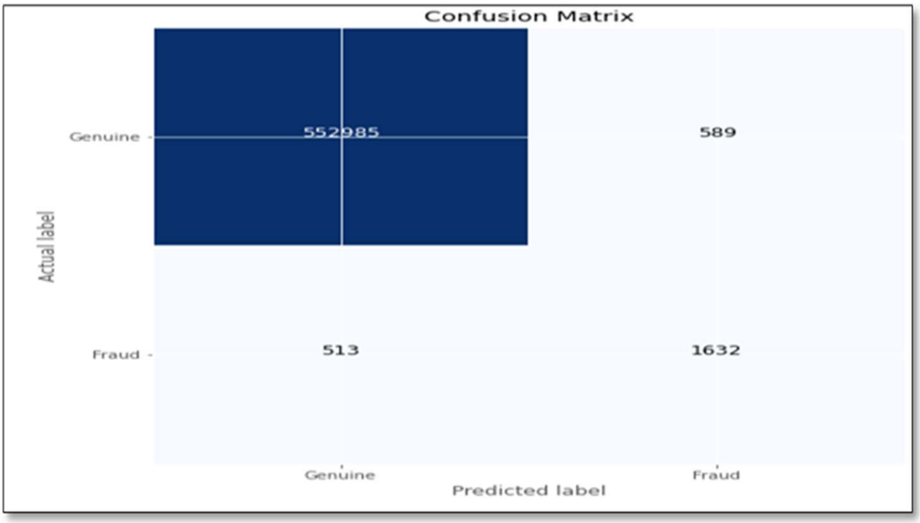
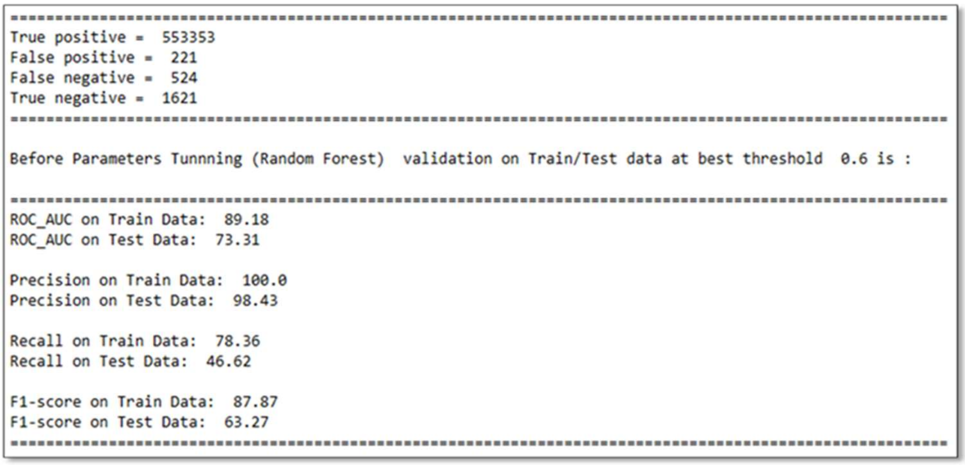


Fig. 8. Confusion Matrix for Decision Tree after Parameter Tuning

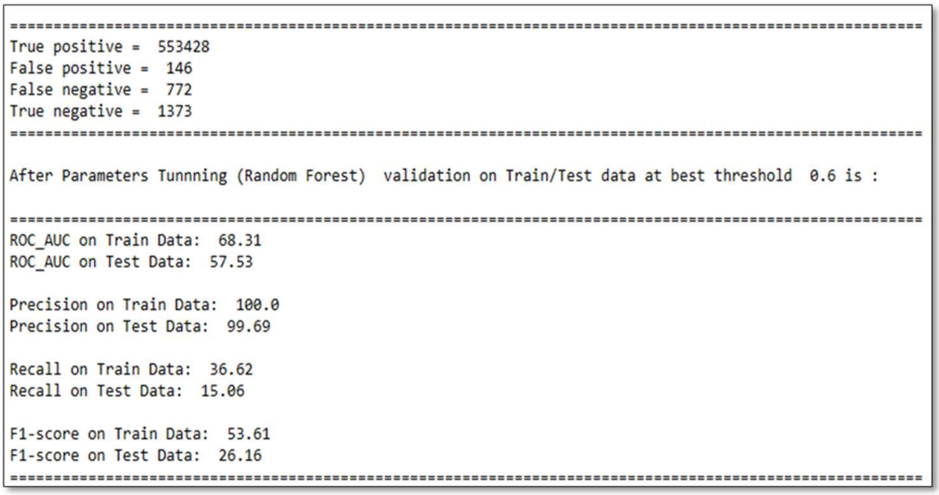
As can be seen from the figures 6 and 7, it can be observed that the classifier is suffering with overfitting problem, as it gives good performance on train data but not on test data. The following figure shows the result of performance evaluation of random forest classifier before parameters Tuning.



```
====================================================================================================
True positive =  553353
False positive =  221
False negative =  524
True negative =  1621
====================================================================================================

Before Parameters Tunnning (Random Forest)  validation on Train/Test data at best threshold  0.6 is :

====================================================================================================
ROC_AUC on Train Data:  89.18
ROC_AUC on Test Data:  73.31

Precision on Train Data:  100.0
Precision on Test Data:  98.43

Recall on Train Data:  78.36
Recall on Test Data:  46.62

F1-score on Train Data:  87.87
F1-score on Test Data:  63.27
====================================================================================================
```

Fig. 9. Performance Evaluation of Random Forest Before Parameter Tuning

Again, after hyper parameter tuning, we have the best parameters like 'n_estimators', 'min_samples_split',min_samples_leaf','max_features' and 'max_depth'. The model has been fitted again with these best parameters, and we have got the following result.The following figures show the random forest classifier's performance and confusion matrix after parameter tuning.



```
====================================================================================================
True positive =  553428
False positive =  146
False negative =  772
True negative =  1373
====================================================================================================

After Parameters Tunnning (Random Forest)  validation on Train/Test data at best threshold  0.6 is :

====================================================================================================
ROC_AUC on Train Data:  68.31
ROC_AUC on Test Data:  57.53

Precision on Train Data:  100.0
Precision on Test Data:  99.69

Recall on Train Data:  36.62
Recall on Test Data:  15.06

F1-score on Train Data:  53.61
F1-score on Test Data:  26.16
====================================================================================================
```

Fig. 10. Performance Evaluation of Random Forest After Parameter Tuning

As from the figures 9 and 10, it can be observed that the result has been improved a bit in comparison of decision tree algorithm, but the random forest classifier still gives good performance on training data but not on test data.

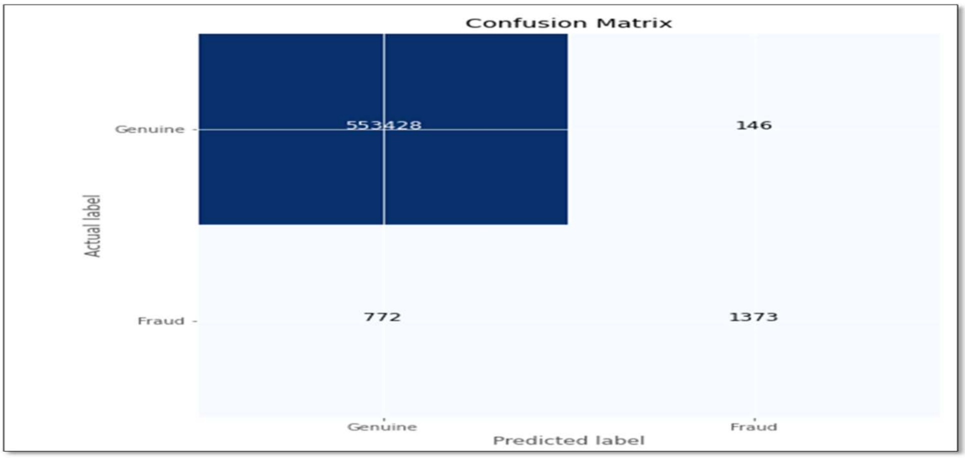


Fig. 11. Confusion Matrix for Random Forest After Parameter Tuning

4 Conclusion

This paper emphasizes the significance of technological advancements and the widespread availability of online shopping. It acknowledges the time-saving benefits and convenience of online shopping, particularly the elimination of the need to physically visit stores. Credit card payment emerges as a popular mode of transaction in this digital era, with a substantial number of credit card users worldwide. However, the increasing prevalence of fraudulent credit card transactions poses challenges for both banks and customers, resulting in financial losses. To address these issues, the paper underscores the importance of implementing a secure credit card fraud detection system. It explores the application of various machine learning algorithms, including Naïve Bayes, Logistic Regression, SVM, Decision Trees, Random Forest, Genetic Algorithm, J48, and AdaBoost, for credit card fraud detection. These algorithms play a crucial role in analyzing datasets and identifying fraudulent transactions accurately.

References

1. Ms. K. RamaKalyani and Prof. Dr. D. Uma Devi, Fraud Detection of Credit Card Payment System by Genetic Algorithm in the International Journal of Scientific & Engineering Research Volume 3, Issue 7, July-2012, ISSN 2229- 5518
2. Ms. Rimpal R. Popat and Mr. Jayesh Chaudhary, A Survey on Credit Card Fraud Detection Using Machine Learning in the Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018) IEEE Conference Record: # 42666; IEEE Xplore ISBN: 978-1-5386-3570-4
3. S P Maniraj, Aditya Saini, Swarna Deep Sarkar, Shadab Ahmed, Credit Card Fraud Detection Using Machine Learning and Data Science in the International Journal of Engineering Research and Technology (IJERT), ISSN: 2278- 0181,Vol. 8 Issue 09, September-2019
4. Aman Gulati, Prakash Dubey, MdFuzailC, Jasmine Norman and Mangayarkarasi R, Credit card fraud detectionUsing neural network and geolocation in the 14th ICSET-

2017 ,IOP Conf. Series: Materials Science and Engineering 263 (2017) 042039 doi:10.1088/1757-899X/263/4/042039

5. Andhavarapu Bhanusri, K.Ratna Sree Valli , P.Jyothi , G.Varun Sai , R.Rohith Sai Subash, Credit card fraud detection Using Machine learning algorithms in the Quest Journals, Journal of Research in Humanities and Social Science, Volume 8 ~ Issue 2 (2020)pp.: 04-11, ISSN(Online):2321-9467
6. Mr. John O. Awoyemi, Mr. Adebayo O. Adetunmbi and Mr. Samuel A. Oluwadare, Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis,978-1-5090-4642-3/17/$31.00 ©2017 IEEE
7. Ms. Heta Naik,Credit Card Fraud Detection for Online Banking Transactions in the International Journal for Research in Applied Science and Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887 Volume 6 Issue IV, April 2018
8. Mr. Varun Kumar K S, Mr. Vijaya Kumar V G, Mr. Vijay Shankar A and Ms. Pratibha K, Credit Card Fraud Detection using Machine Learning Algorithms in the International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org , ISSN: 2278-0181 , Vol. 9 Issue 07, July-2020
9. Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Credit Card Fraud Detection - Machine Learning methods in the Conference Paper, March 2019 DOI: 10.1109/INFOTEH.2019.8717766 978-1-5386-7073-6/19/$31.00 ©2019 IEEE
10. Navanshu Khare and Saad Yunus Sait, Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models in the International Journal of Pure and Applied Mathematics Volume 118 No. 20 2018, 825- 838 ISSN: 1314-3395 (on-line version)
11. Mr. Dhwanir Shah and Dr. Lokesh Kumar Sharma, A Survey on Credit Card Fraud Detection Using Machine Learning in the National Conference on Contemporary Practices in Management & Information Technology KSCON2021 (Virtual mode) (November 2021), published in the E-Book with ISBN No. 978-93-92008-00-9
12. Kaggle.com. Credit Card Fraud Detection. [online] Available at: https://www.kaggle.com/code/rahulrajml/fraud-detection-systematic-approach/data