JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

Credit Card Fraud Detection

Mohit Tiwari[1], Vipul Sharma[1], Devashish Bala[1], Devansh[1], Dishant Kaushal[1]

[1]Department of Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi

Received yyyy Month dd; Revised yyyy month dd; Accepted yyyy month dd (do not edit).

Abstract
The increase in the popularity of the Internet has brought about a rise in the usage of credit cards. People have shifted to e-banking due to the Covid pandemic. The rise is good as we move towards a digital India, but it has brought with it credit card fraud. There is, therefore, a requirement for a system that can detect these frauds. In this paper, we have analysed the recent work that has been done to identify credit card fraud. We have then developed Logistic Regression and XGBoost models to detect these frauds. To make the model more efficient and robust, we have used RandomizedSearchCV to find its optimal hyper-parameters and Synthetic Minority Oversampling Technique (SMOTE) to handle the imbalanced dataset. The models were then tested on the Kaggle dataset containing over 284,000 transactions. Both achieved a very high accuracy with a ROC-AUC score of 0.99.

Keywords: Credit card, fraud detection, Logistic Regression, Machine Learning, SMOTE, XGBoost, RandomizedSearchCV.

1. Introduction

The e-commerce industry in India is ever-growing. The 'Digital India' program has seen India's Internet user base grow to 78.2 crores. With the world suffering from Covid, more and more people are shifting to e-banking. Cash transactions have lessened, and the role of e-banking has increased significantly. Whereas the shift is a positive one, it has its downsides. Financial fraud is a serious issue, and according to a report by CNBC credit card fraud has increased significantly in recent times. Credit card fraud occurs when your credit card is used to make purchases without your consent. This can happen when a scammer has access to your credit card number and PIN. From the Nilson Report Data in 2019, it is gathered that card fraud has reached $28.65 billion worldwide. In India itself, a total of 1.17 lakh cases of banking fraud have been registered from April 2009 to Sept 2019.

Most of the online banking frauds are credit card frauds as it is the most prevailing mode of payment. Therefore, there is a need to develop a mechanism to identify fraudulent transactions. In recent times, this problem has been addressed using Machine learning. In this paper, we have proposed Logistic Regression and XGBoost models to identify fraudulent transactions. To find the optimal hyperparameters of the models, we have used RandomizedSearchCV. A complication that arises is the dataset imbalance. The dataset has a lesser percentage of fraudulent transactions and, therefore, a very high percentage of genuine transactions. To counter this problem, we have used SMOTE so that we can have a balanced dataset. Accuracy alone is not sufficient to judge the model in these kinds of problems, and so we have calculated the F1-score and ROC-AUC of the model.

The rest of the paper is as follows. In section 2 of the research, we have analysed the recent work done in this field. In Section 3, we discuss a method to detect these frauds by building logistic regression and XGBoost models using RandomizedSearchCV. The next section discusses the performance evaluation metrics and the results obtained. We have also compared the models in this section. Section 5 concludes the paper.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

2. Literature Review

The authors in [1] created a voting ensemble classifier to accurately detect credit card fraud. The authors initially collected data on users' behaviour over a banking website. They then used the Web Markov Skeleton Process (WMSP) to differentiate between normal and abnormal behaviour. Next, Random Forest was used to classify the operational features of users. The operational features include Ip address, operation time, operation device, etc. Subsequently, Support Vector Machine (SVM) was used to classify features such as transaction type, amount, account no. etc. Finally, the majority voting-based ensemble uses these results to detect fraud transactions. On testing the model, it was found that the accuracy increased from 94% to 98.5% with highly labelled data. Also, the recall, F1-score, and precision increased significantly with highly labelled data. Noticeably, the accuracy dipped as the percentage of frauds increased.

Authors in [2] used two methods to detect credit card fraud: Poisson Process Intensity Model and various machine learning models. The researchers took a highly imbalanced dataset consisting of 95662 transactions wherein fraudulent transactions were less than 0.2%. The Poisson process model requires to have clients with at least two transactions. To satisfy this condition, 812 clients were removed. The ROC-AUC scores of the 3 Poisson process models i.e., HomoModel, LinearModel, and QaudraticModel, were 0.748, 0.786, and 0.769, respectively. The authors also implemented three machine learning models: LGBM, XGBoost, and CatBoost which had ROC-AUC scores of 0.9990,0.9997, and 0.9996, respectively. There is scope to work more on the Poisson process Model using other mathematical techniques to improve the accuracy.

Research done in [3] used Artificial Neural Network and Hierarchical Temporal Memory to detect credit card frauds. The Hierarchical Temporal Memory was based on Cortical Learning Algorithms (HTM-CLA). The ANN model was trained using the Simulated Annealing technique (SA-ANN). Comparisons were also made with Long Short-Term Memory ANN (LSTM-ANN). The models were tested on the German credit card fraud benchmark dataset, and the Australian credit card fraud benchmark. The result found that the SA-ANN worked the best. HTM-CAA model was also competitive and proved to be better than LSTM-ANN.

The authors in [4] used Artificial Neural Network (ANN), Support Vector Machines (SVM), and k-nearest neighbour (KNN) to detect credit card fraud. ANN had a higher accuracy compared to SVM and KNN.

In [5], the authors worked on building fraud detection models using KNN, Naïve Bayes, Random Forest, Logistic Regression, and Support Vector Machine achieving an accuracy of 0.79, 0.76, 0.8, 0.81, and 0.82, respectively. In the research, the authors could have also found other scores such as F1-score and recall to make the study more comprehensive.

Authors in [6] proposed a Credit Card Risk Identification (CCRI) method using Random Forest Classifier and Support Vector Machine to detect fraud risks. The model selected the relevant features using Random Forest and then used Support Vector Machine to identify the fraudulent transactions. The model had an accuracy of 95.12, and it outperformed other models like Decision Tree on sensitivity and the AUC-ROC score.

The authors in [7] worked on building a hybrid approach to detect credit card fraud. They used Recursive Feature Elimination (RFE) to select relevant features, GridSearchCV to optimize hyper-parameters, and SMOTE to handle imbalanced dataset. The model was tested on three different datasets and achieved great accuracy with good recall, precision, and F1-scores.

In [8], research was done to detect fraudulent credit card transactions. The researchers initially divided the transactions into three categories i.e., high, medium, and low, using range partitioning. They then used the Sliding-window method to detect behavioural patterns of the cardholder. Subsequently they performed Synthetic Minority Over-Sampling Technique on the dataset. The Matthews Correlation Coefficient (MCC), values were calculated, and it was found that Random Forest, Decision Tree, and Logistic Regression worked the best.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

The authors in [9] worked on creating SAS with the Hadoop framework. They then build Logistic Regression, Decision Tree, and Random Forest Decision Tree models on top of it to detect frauds. The Logistic Regression and the Decision Tree had an accuracy of 70% and 72%, respectively. Random Forest outperformed both of them with an accuracy of 76% and better precision and recall.

Research done in [10] used a Random-tree-based random forest and a CART-based random forest. Dataset used to test the models was of a Chinese e-commerce company, consisting of transactions from November 2016 to January 2017. CART-based random forest achieved better accuracy, recall, and F-Measure. CART-based random was then applied to another dataset and had an accuracy of 98.67%.

Research done in [11] aimed at detecting credit card frauds using KNN and Outlier Detection. It was concluded that the KNN method is fit for detecting frauds with the limitation of memory, whereas outlier detection uses less memory and works well on larger datasets.

The authors in [12] used AdaBoost and Majority Voting to detect credit card fraud. The models were evaluated using Matthews Correlation Coefficient (MCC). AdaBoost and the majority voting method helped to improve fraud detection rates considerably. The authors then added noise to the data samples to evaluate the models. The MCC values deteriorated on the addition of noise, and the majority voting method worked best in the presence of noise.

In [13], the authors used cost-sensitive decision trees to identify fraudulent transactions. The credit card data used in the study consists of 22 million transactions done in 12 months. To counter dataset imbalance, cost-sensitive modelling is used where misclassifying a fraudulent transaction cost more than misclassifying a genuine transaction. The cost-sensitive decision tree models outperformed SVM and ANN.

The authors in [14] did a comparative study using naïve Bayes, KNN, and logistic regression models. The dataset used contains 284,807 transactions of European cardholders. The authors used a hybrid technique of under-sampling and oversampling to handle the imbalanced dataset. K-Nearest neighbour had the highest accuracy of 97.92%, which was slightly better than Naïve Bayes, which had an accuracy of 97.69%. K-Nearest Neighbour also had a better Matthews Correlation Coefficient than Naïve Bayes and Logistic Regression.

Convolutional Neural Networks were used in [15] to detect credit card frauds. Cost-based sampling method was used to counter the imbalanced dataset. To apply the CNN model features were transformed. The evaluation of the model was done using transaction data of a bank consisting of over 260 million transactions. The CNN based model outperformed state-of-art methods.

Paper [16] demonstrates an approach using the Rough set and Decision Tree Technique. To do pre-processing, the authors make use of Rough Set. This helps to select the features which have a higher dependency. Classification is done using the J48 classifier. The model is then applied to the dataset, which is taken from the UCI website, and the work is executed using MATLAB and WEKA.

The authors in [17] worked on detecting credit card frauds using various models. They made use of fraud detection techniques like Artificial Neural Networks (ANN), Hidden Markov Model, Decision Trees, K-Nearest Neighbour (KNN), Bayesian Network, Support Vector Machine (SVM), and Fuzzy Logic Based System. To compare the various techniques, they used parameters like accuracy, specificity, precision, sensitivity, false alarm rate, and the cost of the model. The dataset used was KDD from the standard KDD CUP 99. Artificial Neural network outperformed other models on all parameters, but it is also expensive to train compared to logistic regression.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

In [18], used Lightgbm Model to detect credit card frauds. The dataset used was IEEE-CIS consisting of over a million samples. In feature engineering features with over 0.95 correlation were removed to prevent overfitting. GridSearchCV was used to tune the parameters of the models. To compare the different models, the authors used the ROC-AUC score and accuracy of the models, concluding that Lightgbm performed the best with a ROC-AUC score of 0.955 and an accuracy of 0.982.

The authors in [19] used two datasets. The first was of dummy data created based on features that have a significant impact on credit card fraud detection, and the second was transformed using data reduction, normalization, and principal component analysis. The models performed better in the second experiment and achieved an accuracy of over 95%.

In paper [20], authors used Decision tree, Random Forest, and Logistic Regression to detect credit card fraud. Kaggle dataset was used, and dataset imbalance was handled using oversampling. The model was evaluated using accuracy, sensitivity, and specificity and it was found that the Random Forest performed better. The authors could have used other testing parameters such as the F1-score to evaluate the model.

3. Proposed Approach

3.1 Methodology

The first part of the methodology is to visualize the data to get a better understanding of it by using NumPy, Pandas, and Matplotlib. In this part, we check the shape of the dataset, the distribution of fraudulent and non-fraudulent transactions, check for missing values in the dataset, etc.

After this, we handle the missing values and do outlier treatment on the dataset. This is followed by preparing the dataset for modelling by analysing the relation between classes and various columns. The next part of the methodology is to mitigate skewness. Skewness is mitigated by using the PowerTransformer of sklearn. Subsequently, splitting of dataset into training set and testing set is done.

Following the splitting of the dataset, feature scaling of the required columns is done using the StandardScaler library of sklearn. To handle dataset imbalance, we have used SMOTE (Synthetic Minority Oversampling Technique), and to find the optimal values of the hyperparameters, we have used RandomizedSearchCV.

The next part is to evaluate the models. As the dataset was highly imbalanced, accuracy alone is not sufficient to judge the model. Therefore, the following parameters have been used to evaluate the model:

1. **Accuracy**
2. **Precision**
3. **Recall**
4. **Sensitivity**
5. **Specificity**
6. **F1-Score**
7. **ROC-AUC curve**

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

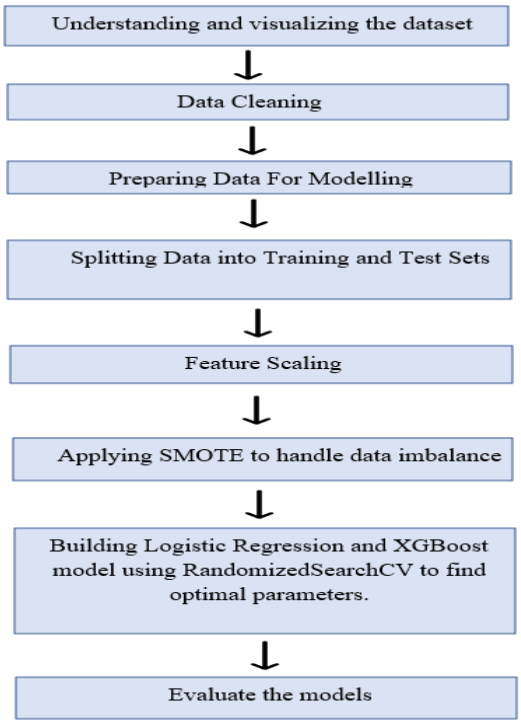

Figure 1. Methodology

3.2 Implementation

3.2.1 Data Understanding and Visualisation

The dataset we have used is of the transactions made by European cardholders in September 2013, consisting of 284,807 transactions. From Fig. 2, it is observed that the dataset has a high imbalance, there are over 284000 genuine transactions and only 492 i.e., 0.172% fraudulent transactions.

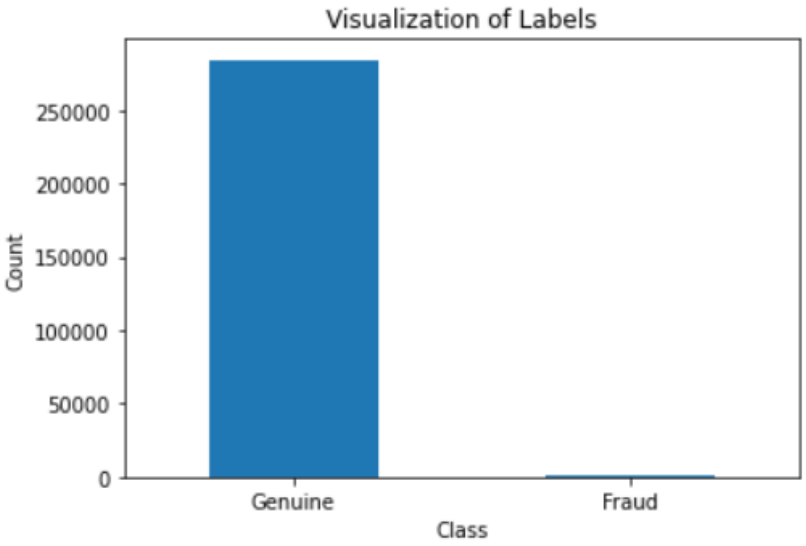

Figure 2. Distribution of Genuine and Fraud Transactions

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

```
df.shape
```

```
(284807, 31)
```

Figure 3. Shape of Dataset

Fig. 3. Shows the shape of the dataset. It contains 31 columns where column 1 stands for the time when the transaction took place, there are then 28 columns labelled from V1 to V28, followed by the amount of the transaction, and the Class column which shows if the transaction is fraudulent or not.
Principal component analysis (PCA) of columns V1 to V28 has been done to maintain the privacy of the data.

3.2.2 Missing values and outlier treatment

The next part is to detect the presence of missing values in the dataset.

```
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

Figure 4. Missing Value Detection

From Fig. 4. It is clear that the dataset has no missing values, and therefore, there is no need to handle missing data. Also, there is no need to do outlier treatment as the dataset is already PCA transformed.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

3.2.3 Preparing data for modelling and splitting it into train-test set.

In this part, the relation between the classes and time is analysed. It is seen that there is no relation between time and the classes. The skewness of the columns has been mitigated using PowerTransformer. There is both positive and negative skewness, which was removed by the yeo-johnson method of the PowerTransformer. After this, the dataset was split into training and testing set. Following the splitting of the dataset Feature Scaling of the Amount column was done using StandardScaler of sklearn library.

3.2.4 Building the Logistic Regression and XGBoost model

There are two parts to building the model. The first part is to handle the imbalanced dataset, and then the second part is to find the optimal hyperparameters of the models. To handle the data imbalance, we have used SMOTE (Synthetic Minority Oversampling Technique). SMOTE is an oversampling technique where synthetic samples are generated for fraudulent transactions. Before applying SMOTE, the no. of fraudulent and non-fraudulent transactions were 492 and 284315 respectively (can be seen in Fig. 1). After applying SMOTE, the no. of fraudulent and non-fraudulent transactions are as shown in Fig. 5.

```
from collections import Counter
counter=Counter(y_train_smote)
print('After Applying SMOTE', counter)
```

```
After Applying SMOTE Counter({0: 227449, 1: 227449})
```

Figure 5. Effect of SMOTE

To find the optimal parameters for the models, we used RandomizedSearchCV. There is no way to know the values of hyperparameters in advance, and to do this manually is very time-consuming. RandomizedSearchCV is a sklearn library similar to GridSearchCV. GridSearchCV is computationally expensive while dealing with multiple parameters, therefore, RandomizedSearchCV is used which allows us to specify the no. of parameters settings that are to be tried by specifying the value of n_iter.

4. Performance Evaluation Metrics and Results

Due to the dataset being highly imbalanced, accuracy alone would not be sufficient to judge the model. Therefore, we have to calculate the precision, recall, sensitivity, specificity, F1-score, and ROC-AUC of the model. To calculate these factors, we need to find the True positive, True Negative, False Positive, and False Negative of the model. The following are the meanings of these terms:

- **True Positive (TP):** Fraudulent transactions identified as fraudulent by the model.
- **True Negative (TN):** Genuine transactions identified as genuine by the model.
- **False Positive (FP):** Genuine transactions wrongly identified as fraudulent by the model.
- **False Negative (FN):** Fraudulent transactions identified as genuine by the model.

Using the above parameters we can calculate the accuracy, specificity, sensitivity, precision, and F1-score of the model.

1. **Accuracy:** Ratio of correctly identified transactions to total transactions.
   Accuracy = (TN + TP) / (TP + FP + FN + TN)

2. **Precision:** Ratio of True Positive to True Positive and False Positive, which is the proportion of positive identifications that were calculated correctly.
   Precision = TP / TP + FP

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

3. **Sensitivity:** Ability of the model to correctly identify the fraudulent transactions.
   Sensitivity = TP / TP + FN

4. **Specificity:** Ability of the model to correctly identify the genuine transactions.
   Specificity = TN / TN + FP

5. **F1-Score:** Calculates the weighted Harmonic mean of the sensitivity (recall) and the precision.
   F1- score = 2 × (*Precision* × *Recall*) / (*Precision* + *Recall*)

6. **ROC-AUC:** It plots the TPR (true positive rates and FPR (false positive rates) for different decision thresholds. The AUC (Area Under Curve) helps measure the ability of the model to distinguish between the fraudulent class and the genuine class of transactions.

4.1 Result Analysis

In this section we will go through the results obtained.

4.1.1 Logistic Regression Results

From Fig. 4.1, the model has an accuracy of 0.972 with a sensitivity of 0.946 and a specificity of 0.972. The precision, recall, and F1-score were 1.00, 0.97, and 0.98, as shown in Fig. 4.2. Fig. 4.3 shows the ROC-AUC curve, the ROC-AUC score of the model is 0.99. Overall, the model worked very well with most scores in the range of 0.95-1.

```
Accuracy:- 0.9721568765141674
Sensitivity:- 0.9468085106382979
Specificity:- 0.972198776113104
```

Figure 6. Accuracy, Sensitivity, and Specificity of Logistic Regression model

	precision	recall	f1-score	support
0	1.00	0.97	0.99	56868
1	0.05	0.95	0.10	94
accuracy			0.97	56962
macro avg	0.53	0.96	0.54	56962
weighted avg	1.00	0.97	0.98	56962

Figure 7. Precision, Recall, and F1-Score of Logistic Regression model

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

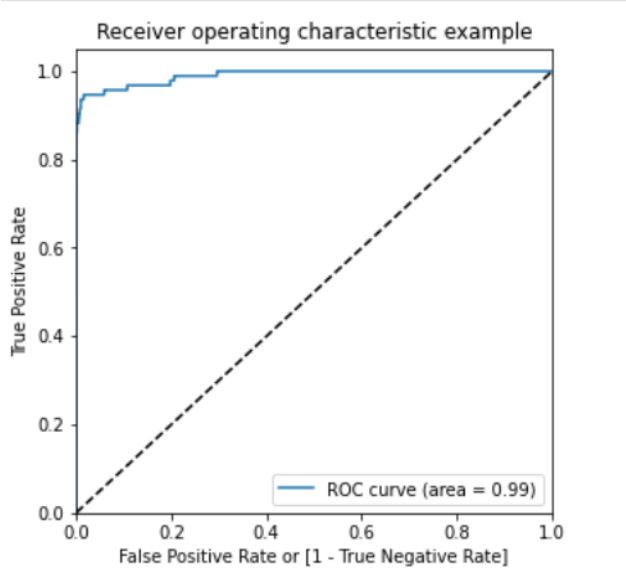

Figure 8. ROC-AUC of Logistic Regression Model

4.1.2 XGBoost Results

From Fig. 4.4, the model has an accuracy of 0.988 with a sensitivity of 0.914 and a specificity of 0.988. The precision, recall, and F1-score were 1.00, 0.99, and 0.99, as shown in Fig. 4.5. Fig. 4.6 shows the ROC-AUC curve, the ROC-AUC score of the model is 0.99. Overall, the model worked very well with most scores in the range of 0.95-1.

```
Accuracy:- 0.988149994733331
Sensitivity:- 0.9148936170212766
Specificity:- 0.9882710839136245
```

Figure 9. Accuracy, Sensitivity, and Specificity of XGBoost model

	precision	recall	f1-score	support
0	1.00	0.99	0.99	56868
1	0.11	0.91	0.20	94
accuracy			0.99	56962
macro avg	0.56	0.95	0.60	56962
weighted avg	1.00	0.99	0.99	56962

Figure 10. Precision, Recall, and F1-Score of XGBoost model

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

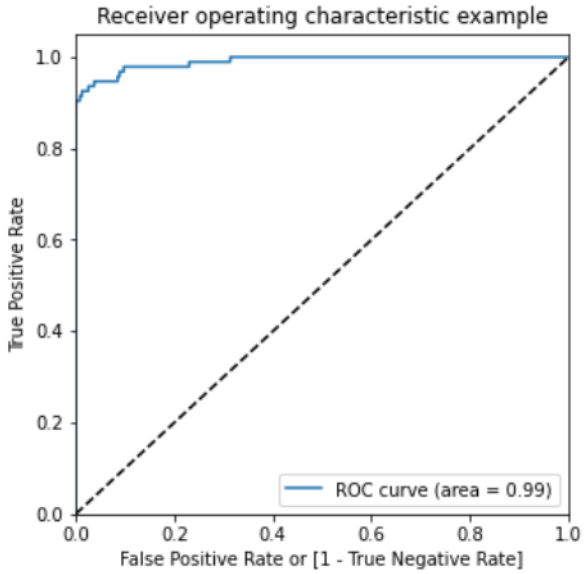

Figure 11. ROC-AUC of XGBoost Model

4.1.3 Comparison of Logistic Regression and XGBoost Model

The Logistic Regression and XGBoost models performed exceedingly well. From the comparison table, the XGBoost model had a slightly higher accuracy, specificity, recall, and F1-Score, whereas Logistic Regression had a better sensitivity. The main difference is the time taken to build the models in which Logistic Regression took only 32 minutes, whereas the XGBoost model took 2 hours and 7 minutes.

Table 1. Comparison of Logistic Regression and XGBoost Model

S. No.	***Parameter***	***Logistic Regression***	***XGBoost***
1	Accuracy	0.972	0.988
2	Sensitivity	0.946	0.914
3	Specificity	0.972	0.988
4	Precision	1.00	1.00
5	Recall	0.97	0.99
6	F1-Score	0.98	0.99
7	ROC-AUC	0.99	0.99
8	Time Taken	32 mins	2 hrs and 7 mins

5. Conclusion and Future Work

In the paper, credit card frauds were detected effectively. The literature survey analysed the recent work done. Subsequently, we studied the Kaggle dataset and prepared it for modelling. Logistic Regression and XGBoost models, were built to detect frauds. SMOTE (Synthetic Minority Oversampling Technique) was used to counter the dataset imbalance, and RandomizedSearchCV was used to find the optimal parameters of the models. To assess the model, accuracy, sensitivity, specificity, precision, recall, F1-score, and ROC-AUC of the models were calculated. The models achieved a ROC-AUC score of 0.99. XGBoost had a higher F1-score of 0.99 compared to 0.98 of Logistic Regression, but it took more time to build. Overall, the models performed exceedingly well, achieving all the scores greater than 0.9. In the future, a system to detect credit card fraud in real time can be build.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1778-1789
https://publishoa.com
ISSN: 1309-3452

References

[1] C. Sudha and D. Akila, "Majority vote ensemble classifier for accurate detection of credit card frauds," Materials Today: Proceedings, 2021, doi: https://doi.org/10.1016/j.matpr.2021.01.616.

[2] A. Izotova and A. Valiullin, "Comparison of Poisson process and machine learning algorithms approach for credit card fraud detection," Procedia Computer Science, vol. 186, pp. 721–726, 2021, doi: https://doi.org/10.1016/j.procs.2021.04.214.

[3] E.N. Osegi and E.F. Jumbo, "Comparative analysis of credit card fraud detection in Simulated Annealing trained Artificial Neural Network and Hierarchical Temporal Memory," Machine Learning with Applications, vol. 6, Jun. 2021, doi: https://doi.org/10.1016/j.mlwa.2021.100080.

[4] A. Rb and S. K. Kr, "Credit card fraud detection using artificial neural network," Global Transitions Proceedings, vol. 2, no. 1, pp. 35–41, 2021, doi: https://doi.org/10.1016/j.gltp.2021.01.006.

[5] S. Arora, S. Bindra, S. Singh, and V. K. Nassa, "Prediction of credit card defaults through data analysis and machine learning techniques," Materials Today: Proceedings, 2021, doi: https://doi.org/10.1016/j.matpr.2021.04.588.

[6] N. Rtayli and N. Enneya, "Selection Features and Support Vector Machine for Credit Card Risk Identification," Procedia Manufacturing, vol. 46, pp. 941–948, 2020, doi: https://doi.org/10.1016/j.promfg.2020.05.012.

[7] N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization," Journal of Information Security and Applications, vol. 55, p. 102596, 2020, doi: https://doi.org/10.1016/j.jisa.2020.102596.

[8] V. N. Dornadula and S. Geetha, "Credit Card Fraud Detection using Machine Learning Algorithms," Procedia Computer Science, vol. 165, pp. 631–641, 2019, doi: https://doi.org/10.1016/j.procs.2020.01.057.

[9] S. Patil, V. Nemade, and P. K. Soni, "Predictive Modelling For Credit Card Fraud Detection Using Data Analytics," Procedia Computer Science, vol. 132, pp. 385–395, 2018, doi: https://doi.org/10.1016/j.procs.2018.05.199.

[10] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random Forest for credit card fraud detection," 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), 2018, doi: https://doi.org/10.1109/ICNSC.2018.8361343.

[11] N. Malini and M. Pushpa, "Analysis on credit card fraud identification techniques based on KNN and outlier detection," 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2017, doi: https://doi.org/10.1109/AEEICB.2017.7972424.

[12] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit Card Fraud Detection Using AdaBoost and Majority Voting," IEEE Access, vol. 6, pp. 14277–14284, 2018, doi: https://doi.org/10.1109/ACCESS.2018.2806420.

[13] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," Expert Systems with Applications, vol. 40, no. 15, pp. 5916–5923, 2013, doi: https://doi.org/10.1016/j.eswa.2013.05.021.

[14] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," 2017 International Conference on Computing Networking and Informatics (ICCNI), 2017, doi: https://doi.org/10.1109/ICCNI.2017.8123782.

[15] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit Card Fraud Detection Using Convolutional Neural Networks," Neural Information Processing Lecture Notes in Computer Science, pp. 483–490, 2016, doi: 10.1007/978-3-319-46675-0_53.

[16] R. Jain, B. Gour, and S. Dubey, "A Hybrid Approach for Credit Card Fraud Detection using Rough Set and Decision Tree Technique," International Journal of Computer Applications, vol. 139, no. 10, pp. 1–6, 2016, doi: http://dx.doi.org/10.5120/ijca2016909325.

[17] Y. Jain, N. Tiwari, S. Dubey and S. Jain, "A Comparative Analysis of Various Credit Card Fraud Detection Techniques," International Journal of Recent Technology and Engineering (IJRTE), vol.7, 2019. [Online]. Available: https://www.researchgate.net/publication/332264296_A_comparative_analysis_of_various_credit_card_fraud_detection_techniques.

[18] D. Ge, J. Gu, S. Chang, and J. Cai, "Credit Card Fraud Detection Using Lightgbm Model," 2020 International Conference on E-Commerce and Internet Technology (ECIT), 2020, doi: https://doi.org/10.1109/ECIT50008.2020.00060.

[19] O. S. Yee, S. Sagadevan and N. H. A. H. Malim, "Credit Card Fraud Detection Using Machine Learning As Data Mining Technique," Journal of Tele-communications Electronic and Computer Engineering, 2018. [Online]. Available: https://www.researchgate.net/publication/326986162_Credit_Card_Fraud_Detection_Using_Machine_Learning_As_Data_Mining_Technique.

[20] S. Lakshmi and S. D. Kavila, "Machine Learning For Credit Card Fraud Detection System," International Journal of Applied Engineering Research, vol. 13, no. 24, pp. 16819-16824, 2018.