IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.13441

Enhancing Fraud Detection in Credit Card Transactions using Diverse Machine Learning Techniques

Dr. P. Sreedevi M. Tech, Ph.D.[1], Sharuk N[2], Rushmitha Sreeja K[3], Jyothsna Priya N[4], Lakshmi Teja J[5]

Computer Science Department, Rajeev Gandhi Memorial College of Engineering and Technology Nandyal, 518501[1-5]

Abstract: Regular online card exchanges have expanded as a result of innovative headways in ranges like e-commerce and monetary innovation (FinTech) applications. Credit card extortion has expanded as a result, having an affect on card backers, retailers, and as well as banks. In this manner, making frameworks to ensure the astuteness and security of credit card exchanges is pivotal. In this ponder, we utilize skewed real-world datasets from European credit cardholders to build a machine learning (ML) based system for credit card extortion discovery. We re-sampled the dataset utilizing the Synthetic Minority over- sampling method (SMOTE) in arrange to address the issue of lesson lopsidedness. We evaluated this system with the taking after machine learning methods: Extreme Gradient Boosting (XGBoost).

Keywords: SMOTE, credit card, data resampling, fraud detection, XGBoost, machine learning.

I. INTRODUCTION

Financial fraud has increased recently as a result of new paradigms and technologies emerging in industries like e-commerce and financial technology (FinTech) [1]. Credit card transactions have increased as a result of these technologies' development. As a result, the number of credit card-related financial fraud cases has sharply increased. When a criminal uses a credit card in an unauthorized or unwanted way, it is considered credit card fraud.

This occurs when credit card authentication credentials are acquired through various illegal methods, like e-commerce transaction interception or card cloning [2]. Moreover, organizations including card issuers, retailers, and small companies are impacted by credit card theft. The estimated global loss resulting from credit card theft in 2015 was $21.84 billion [3]. Credit card losses reached $28.65 billion in 2019 [4]. This is a $6.81 billion growth over the previous four years. In order to ensure the integrity and security of all systems involved in processing credit card transactions, it is imperative to employ credit card fraud detection systems.

In this work, we apply machine learning (ML) techniques for the identification of credit card fraud, and we assess these techniques on an actual dataset gathered in September 2013 from European cardholders. The dataset is incredibly unbalanced. In order to address the problem of class imbalance in the European card dataset, this study looked into the application of the Synthetic Minority Over-sampling Technique (SMOTE) [5]. Additionally, the following machine learning techniques were taken into consideration for this study: decision trees (DT), random forests (RF), support vector machines (SVM), and extreme gradient boosting (Boost).

Each of these machine learning techniques was assessed separately for both efficacy and classification quality. To further improve each method's robustness, the Adaptive Boosting (AdaBoost) algorithm was combined with it. This paper's primary contribution is a comparison of various machine learning techniques on a publicly accessible dataset of actual word card transactions.

Additionally, this study looksinto AdaBoost to improve classification quality on a heavily skewed dataset related to credit card fraud. This research work's primary contribution can be summed up as follows:

- We provide a scalable system for detecting credit card fraud. We suggest a scalable approach for detecting credit card fraud.
- We apply the SMOTE approach to address the class imbalance seen in datasets related to credit card fraud.

- To improve performance on the suggested framework, we combine the AdaBoost technique with a number of machine learning techniques. Additionally, we perform a comparative study utilizing the metrics of area under the curve (AUC), Matthew's correlation coefficient(MCC), accuracy, recall, and precision.
- To verify the efficacy of the suggested credit card fraud detection framework, we apply it to a synthetic dataset that is significantly unbalanced.

The remainder of the paper unfolds as follows: Section 2 furnishes a comprehensive review of prior studies employing ML for credit card fraud detection. Section 3 offers an elucidation of the ML methodologies utilized in this study. In Section 4, experimental analyses are carried out. Section 5 delineates the implementation of the proposed framework on a synthetic credit card fraud dataset. Lastly, Section 6 encapsulates the key findings of the research.

II. RELATED WORK

A review of prior studies that employed machine learning (ML) approaches to detect credit card fraud is given in this section.

Khatri et al. [9] executed a few ML calculations for credit card extortion location. In this investigate, the creators implemented the taking after strategies: Decision Tree (DT), k- Nearest Neighbor (kNN), Logistic Regression (LR), Random Forest (RF). To assess the ML-based credit card extortion discovery models, the analysts utilized a dataset that was produced from European cardholders in 2013 [25]. Moreover, the creators considered the affectability and the precision as the primary execution measurements. The comes about showed that the kNN calculation accomplished the most ideal comes about with an exactness of 91.11% and a affectability of 81.19%.

Rajora et al. [10] conducted a comparative inquire about of ML strategies for credit card extortion location utilizing the European cardholders' dataset. A few of the strategies that were explored incorporate the RF and the kNN methods. The creators considered the exactness and the Area Under the Curve (AUC) as the fundamental execution measurements. The results demonstrated that RF calculation accomplished a precision of 94.9% and a AUC of 0.94. In differentiate, the kNN obtained an exactness of 93.2% and an AUC of 0.93. In spite of the fact that these results are promising, this inquire about did not examine the lass awkwardness issue that exists in the dataset that was used.

Trivedi et al. [11] proposed an effective credit card fraud detection motor utilizing ML strategy. In this research, the creators considered numerous administered ML techniques including Gradient Boosting (GB) and Random Forest (RF). The creators assessed these strategies utilizing the European cardholders' dataset. The execution metrics used to survey the viability of the proposed approaches include the exactness and the exactness. The result of the tests appeared that the GB gotten an accuracy of 94.01% and an accuracy of 93.99%. On the other hand, the RF accomplished a precision of 94.00% and a precision of 95.98%.

Tanouz et al. [12] displayed a credit card extortion detection framework utilizing ML calculations. In this inquire about, the authors utilized the European cardholder's dataset to assess the execution of the proposed strategies. In addition, the authors actualized an under-sampling strategy to solve the issue of course lopsidedness that exist in the dataset that was utilized. The ML strategies considered in this work include the RF and LR. The analysts utilized the precision as the primary execution metric. The comes about demonstrated that the RF approach accomplished a extortion discovery accuracy of 91.24%. In differentiate, the LR strategy gotten an accuracy 95.16%. Moreover, the creators computed the confusion lattice to declare whether these proposed methods performed ideally for the positive and negative classes. The comes about appeared that the course lopsidedness issue that exist in the European credit card holder dataset requires further investigation.

Riffi et al. [13] executed a credit card extortion detection engine utilizing the Extraordinary Learning Machine (ELM) and Multilayer Perceptron (MLP) calculations. Both the ELM and MLP are manufactured neural systems (ANNs); however, they contrast in terms of inner design. In this research, the creators utilized the European cardholder's dataset that was generated in 2013. The creators utilized the extortion detection accuracy as the primary execution metric. The results demonstrated that the MLP strategy accomplished an accuracy of 97.84%. In differentiate, the ELM accomplished credit card fraud detection precision of 95.46%. This work concluded that the MLP beated the ELM; be that as it may, the ELM is less complex in comparison to the MLP. Randhawa et al. [14] The creators proposed a credit card fraud discovery motor utilizing Adaptive Boosting (AdaBoost) and Majority Voting (MV) strategies. In this investigate, the authors utilized the European cardholder's dataset. Moreover, the creators considered the AdaBoost strategy in conjunctions with ML strategies such as the Support Vector Machine (SVM). In the tests, the precision and the Matthews Correlation Coefficient (MCC) were considered as the main performance measurements. The comes about illustrated that the AdaBoost-SVM accomplished an exactness of 99.959% and a MCC of 0.044

DOI: 10.17148/IJARCCE.2024.13441

III. BACKGROUND ON MACHINE LEARNING ALGORITHMS.

A. AdaBoost

Boosting is an approach to ML that points at making (generating) profoundly exact models by the combination of several simple or wrong models [15], [16].

This is about implements the AdaBoost calculation in conjunction with other ML strategies to move forward their classification performance. The yield of the AdaBoost strategy is a weighted sum. This is done by combining the yield of the individual boosted models. Below is the scientific detailing of
the AdaBoost strategy:

$$G_N(x) = \sum_{t=1}^{N} g_t(x) \qquad (1)$$

where g_t is a weak learner (simple classifier) that outputs a prediction given an input vector x. t denotes an iteration. For each training sample, the prediction of a weak learner is represented by h(xn).

Further, at each t, a weak learner is selected and multiplied by a coefficient β_t in order to compute the training error, L, as follows:

$$Lt = \sum_{n} L[Gt{-}1xn + \beta th(xn)] \qquad (2)$$

where G_{t-1} is a classifier that was boosted at iteration t − 1 and β_th(xn) is a weak classifier that is considered for the final model.

B. Additional ML Methods

The AdaBoost strategy was utilized in conjunction with the following directed ML strategies: Logistic Regression (LR) [17], Decision Tree (DT) [19], Random Forest (RF) [20], Extra Trees (ET) [19], Support Vector Machine (SVM) [21], and Extreme Gradient Boosting (XGboost).

The AdaBoost approach is utilized to move forward the execution of individual classifiers with respect to execution measurements such as the exactness, the Matthew Correlation Coefficient (MCC), and Area Under the Curve (AUC). These measurements are discussed in more detail in the Tests segment of the paper.

The LR (Logit classifier) is a supervised ML strategy that is efficient for binary classification tasks [18]. The LR method uses a direct work in the Logit work in arrange to make predictions. The SVM is another directed ML technique that is utilized for relapse and classification assignments.

This method is profoundly productive on information with a tall dimensional feature space and it is flexible in terms of utilizing different kernel capacities (decision methods) [22].

The DT calculation is a non-parametric supervised ML approach that is regularly utilized for relapse and classification. This approach employments a tree-like build to make the predictions. A few of the points of interest of utilizing DT include the truth that is basic to translate and it does not require a broad information arrangement.

DTs are the establishment of algorithms such as the RF, ET, and the XGBoost. These methods frame portion of what is labeled Gathering Tree [23] since they fit numerous DTs on a given dataset in arrange to make predictions.

DOI: 10.17148/IJARCCE.2024.13441

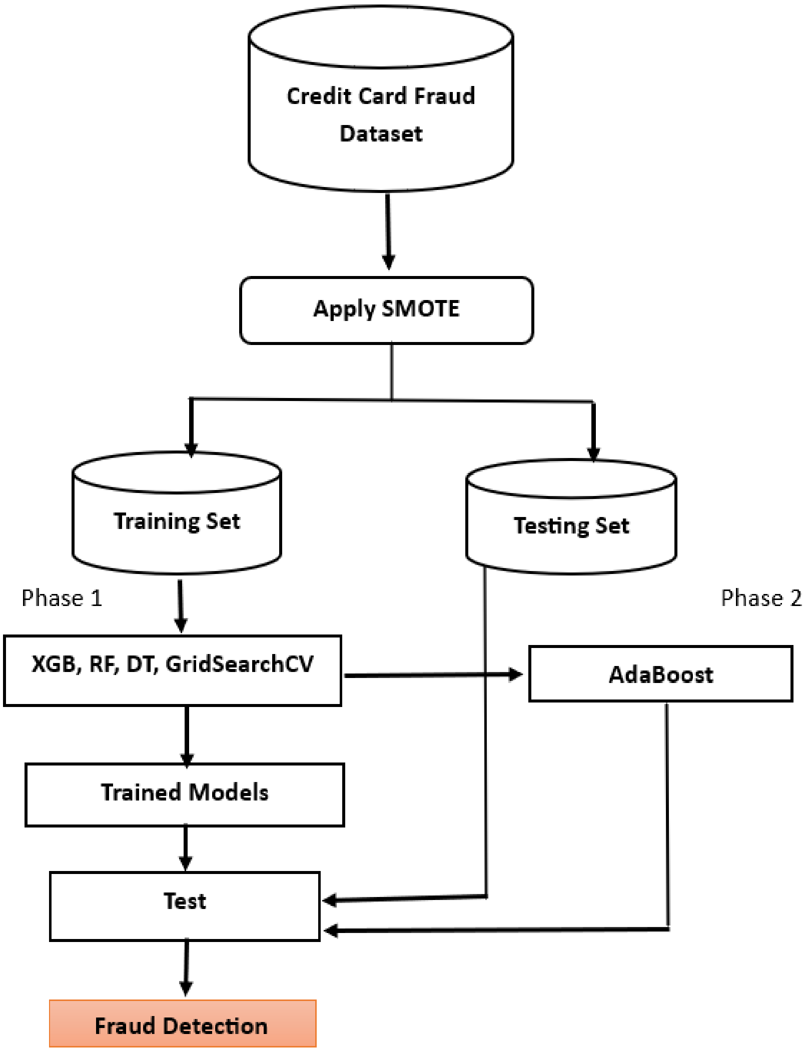


FIGURE 1. Credit card fraud detection framework

IV. RESEARCH METHODOLOGY

A. Fraud Detection Framework

The fraud detection framework used in this study is shown in Fig. 1. The credit card fraud (CCF) dataset is first loaded via the SMOTE block. The credit card fraud (CCF) dataset is split into a training and a testing set in the second stage.

XGB, ET, RF, DT, and LR models must be instantiated in the third phase. After the model is created, it is tested (with the testing set) and trained (with the training set). In addition, the training procedure makes use of the k-fold cross-validation (CV) and GridSearchCV technique to prevent overfitting and boost the accuracy of the experimental results [24]. The instantiated models pass via the AdaBoost module in the fourth stage. At the completion of the AdaBoost process, the models are trained and tested. The Fraud Detection module evaluates the performance of both the non-boosted and boosted models.

B. Dataset

The dataset used in this research was generated from European cardholders in September 2013. This dataset is highly skewed and is publicly available through Kaggle [25]. Moreover, this dataset is not synthetic; therefore, the transactions found in it occurred over a period of time Further, the dataset has 284807 card transactions in total whereby 99.828% are legitimate and 0.172% are fraudulent. Additionally, it contains 30 attributes (V1, . . ., V28), Time and Amount. All the features within the dataset are numerical. The class (label) is represented by the last column whereby the value of 0 represents a legitimate transaction and the value of 1 is a fraudulent activity. The attributes V1 to V28 do not have specific feature names due to data security and integrity reasons. The name of the features was withheld to protect the identity and types of transactions conducted by the cardholders. This dataset has been used in [9]– [14].

C. SMOTE Applied to Credit Card Fraud Dataset The Synthetic Minority over-sampling Technique SMOTE) is amongst of the most dominant techniques that are used to address the issue of class imbalance that is found in datasets such as the ones used to build credit card fraud detection ML-based models [5].

The SMOTE method generates samples of a specific class by connecting a data point with its k-nearest neighbours. The SMOTE method generates synthetic data points that are not a direct replica of the minority class instance. This is done to avoid the phenomenon of over-fitting during the training process.

```
Algorithm 1 SMOTE (T, N, k)
Input T, the total number of instances in the minority class; N, the percentage
 (amount ofSMOTE). k, the number of neighbours.

Output (N / 100) * T, the newly created syntheticdata points.
if N < 100 then
        Generate T minority class data pointsrandomly.
        T = (N/100) * T
        N = 100
    end if
    N = int(N/100)
    num_of_attrs, the number of attributes.
k, the number of nearest neighbours sample.
new_index_points, keeps tabs on the number of synthetic data points that were generated. It is initialized with 0.
synthetic_array_data, an array to keep synthetic data points.
for i range (1 to T) do
        Calculates the k nearest neighbours for i and save the indices in nn_array
        Populate (N, i, nn_array (this is a function that computes synthetic samples)
end for
Populate (N, i, nn_array)
while N ≠ 0 do
    Randomly select a number between 1 and k= rn
    for j in range (1 to num_of_attrs) do
        Calculate the difference: δ = sample[nn_array[rn][j]] – sample[i][j]

        Compute gap_btw:        gap_btw = random (0,1) – random numbers between 0 and 1
        synthetic_array[new_index] [j] = sample[i][j] + gap * δ

    end for
    increment the new index: new_index++
    N = N – 1
end while
```

DOI: 10.17148/IJARCCE.2024.13441

Algorithm 2 SMOTE Implementation – Credit Card FraudDataset

```
1: start
2: Input Credit card fraud dataset (DS) containingminority class data points
3: Output An oversampled dataset: $X_{res}$, input data and$Y_{res}$, the target
4: Import the SMOTE module from imblearn [7]5: Import pandas (pd) from pandas [8]
6: Read DS in a pd dataframe
7: Separate the dataframe into input data, X, and targetdata, Y
8: Instantiate SMOTE instance as s = SMOTE (m: r),where m is the minority class and r the ratio.
9: Fit the SMOTE instance as follows: $X_{res}$, $Y_{res}$ =sm.fit_resample (X, Y)
10: End
```

Algorithm 1 depicts the pseudo code implementation of the SMOTE technique [6] that was used in this research. Algorithm 2 describes the pseudo code implementation of the SMOTE method on the credit card dataset that is used in this research by using the Imblearn library [7].

D. Experimental Setup

The classification experiments were conducted on Google Collab [26]. The Google Compute Engine (GCE) had the following specifications: Intel(R) Xeon(R), 2 Cores, 2.30G Hz. The ML models were implemented using the Scikit-Learn ML framework [27].

E. Performance Metrics

The credit card fraud dataset used in this study includes traces of both fraudulent and legitimate transactions, denoted by 1s and 0s. As such, this machine learning problem has been formulated as a binary classification task. Performance indicators, such as accuracy (AC), recall (RC), and precision (PR), are used to assess such issues. These indicators are expressed mathematically as follows:

- False positives (FP): legitimate transactions mistakenly reported as fraudulent.
- False Negative (FN): deceptive transactions that are mistakenly categorized as legitimate.
- True positives (TP): instances of fraud that are consistently reported as fraudulent.
- True Negatives (TN): genuine transactions that are positively categorized as genuine

$$\mathrm{AC} = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} \quad (3)$$

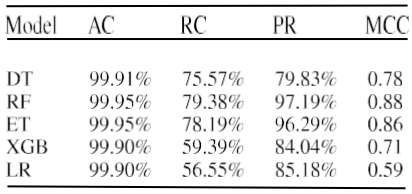
$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (4)$$

$$RC = \frac{TP}{TP + FN} \quad (5)$$

Moreover, there is a significant disparity in the European cardholder’s dataset. Consequently, evaluating the effectiveness of our suggested approach requires more than just taking into account the AC, PR, and RC indicators. We also take into account the Confusion Matrix (CM), the AUC [30], the Matthews correlation coefficient (MCC) [28], and [29] as extra performance metrics in this study. The MCC is employed in this case as a metric to assess the caliber of the categorization task.

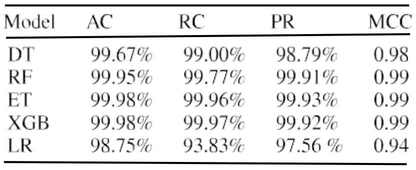
DOI: 10.17148/IJARCCE.2024.13441

TABLE 1. Results without the AdaBoost method.



Model	AC	RC	PR	MCC
DT	99.91%	75.57%	79.83%	0.78
RF	99.95%	79.38%	97.19%	0.88
ET	99.95%	78.19%	96.29%	0.86
XGB	99.90%	59.39%	84.04%	0.71
LR	99.90%	56.55%	85.18%	0.59

TABLE 2. Results withthe AdaBoost method.



Model	AC	RC	PR	MCC
DT	99.67%	99.00%	98.79%	0.98
RF	99.95%	99.77%	99.91%	0.99
ET	99.98%	99.96%	99.93%	0.99
XGB	99.98%	99.97%	99.92%	0.99
LR	98.75%	93.83%	97.56 %	0.94

The MCC metric has a value that ranges from +1 to +1. The better categorization quality, the closer the MCC is to +1. Moreover, the CM [31] is a graph that shows us the errors that a particular classifier made. Furthermore, each model's Area Under the Curve (AUC) was calculated to assess the accuracy and dependability of the categorization.

The AUC is a metric used to assess a classifier's efficacy. An ideal classifier would have an AUC value that is near to 1 because the AUC value ranges from 0 to 1 [30].

$$\text{MCC} = \frac{(\text{TN} \times \text{TP}) - (\text{FN} \times \text{FP})}{\sqrt{}(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}$$

F. Experiments, Results and Discussions

As shown in figure 1, There were two stages to the experimental procedure. We applied the ML models in the first step without using the AdaBoost technique. The findings are shown in Table 1, where the RF algorithm with an MCC of 0.88 is the one that performed best in terms of classification quality. With accuracy coefficients of 99.5%, the RF and ET classifiers outperformed the others. Each ML algorithm was partnered with AdaBoost in the second phase. The results of the trial indicated that the DT recorded a 0.20 MCC increase. The MCC spike for the XGB was 0.28. The ET and the XGB both reached an ideal AC of 99.98% for fraud detection. Additionally, Figures 2 through 4 show the confusion matrix (CM) for each model, which was calculated to determine where the algorithm had some errors. The DT algorithm performed a good job of isolating valid transactions in Fig., but it made a lot of mistakes in predicting fraudulent transactions. Nevertheless, DT- AdaBoost depicted in Figure 3 shows some enhancement in detecting fraudulent transactions. Throughout Figures 4-9, RF- AdaBoost, ET-AbaBoost, LR-AdaBoost, and XGB-AdaBoost consistently exhibit the highest proficiency in identifying fraudulent transactions.

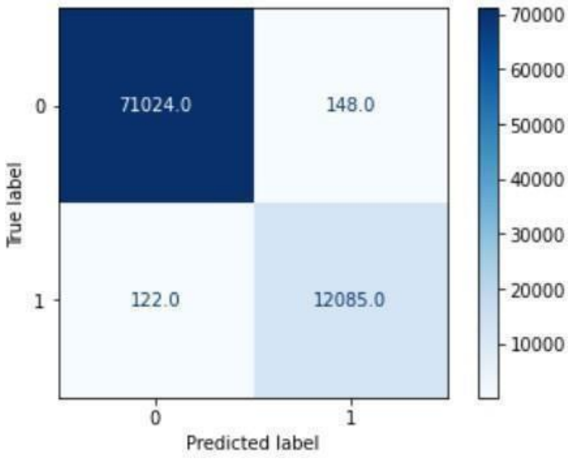


FIGURE 2. DT confusion matrix

DOI: 10.17148/IJARCCE.2024.13441

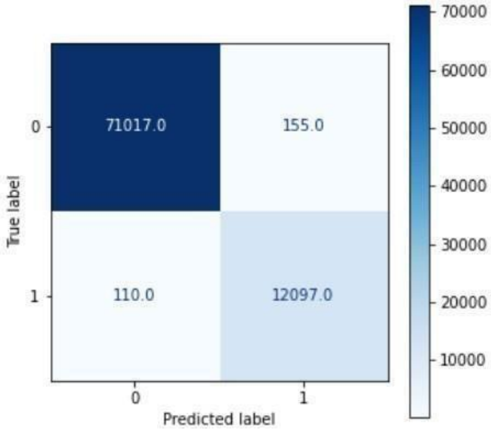


FIGURE 3. DT-AdaBoost confusion matrix

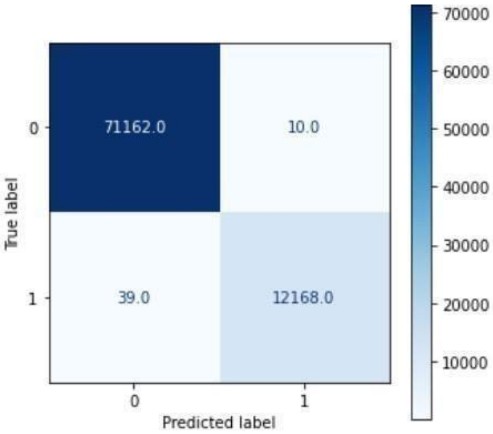


FIGURE 4. RF confusion matrix

It was seen from the results that the fraud detection ACs of the XGB-AdaBoost and the ET-AdaBoost are 5.08% higher than the RF reported in [10] and 6.78% higher than the KNN presented in [10]. The XGB-AdaBoost achieved an AC that is 8.74% greater than the work reported in [12]. Furthermore, the AC produced by the ET-AdaBoost is 4.34% higher than that of the work in [13]. Furthermore, the SMOTE-AdaBoost approaches have improved eachmodel's accuracy when precision and recall are considered.

For instance, the recall of the DT model rose to 99.00% when SMOTE-AdaBoost approaches were applied, from 75.75% while they weren't. Taking precision into consideration, the DT achieved 79.83% precision without employing the SMOTE-AdaBoost methods.

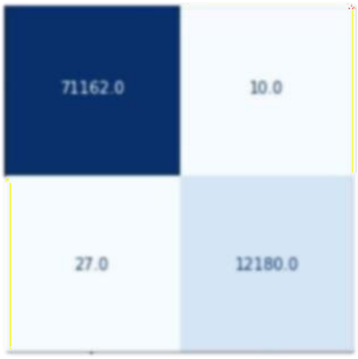


FIGURE 5. RF-AdaBoost confusion matrix

DOI: 10.17148/IJARCCE.2024.13441

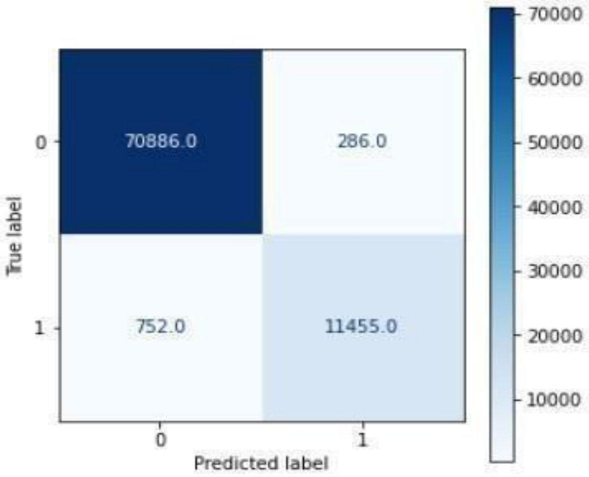


FIGURE 6. LR confusion matrix

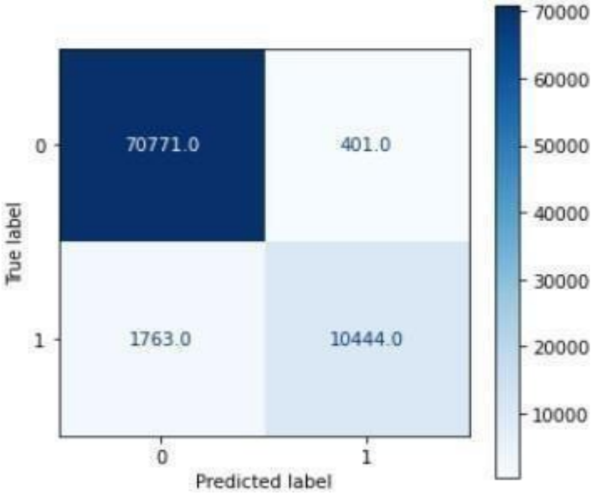


FIGURE 7. LR-AdaBoost confusion matrix

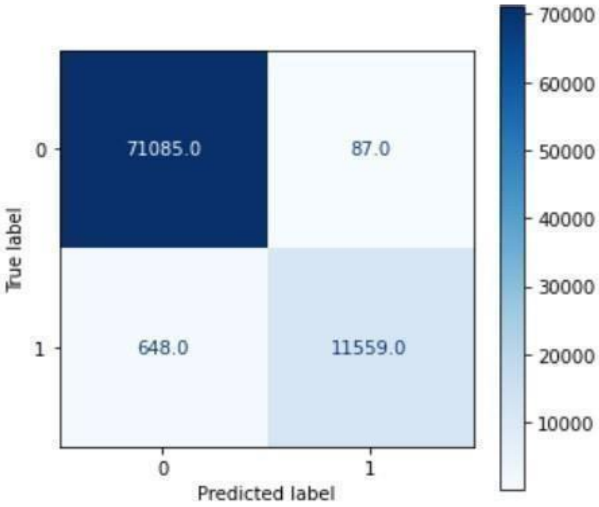


FIGURE 8. XGB confusion matrix

DOI: 10.17148/IJARCCE.2024.13441

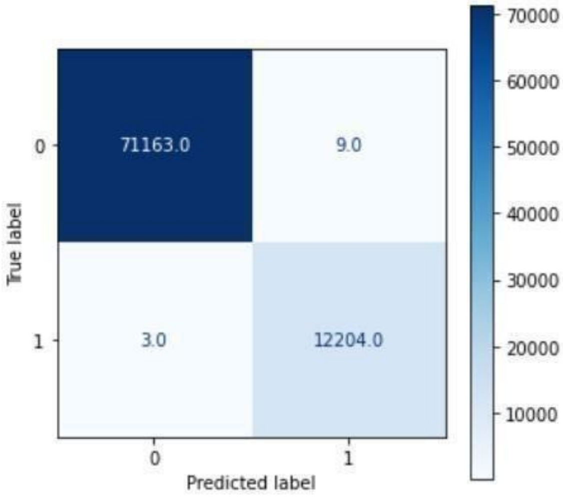


FIGURE 9. XGB-AdaBoost confusion matrix

However, when SMOTE-AdaBoost was employed, the DT model obtained a precision of 98.79%. The MCC increase from 0.78 to 0.98 as a result. This pattern is seen in every model that the study took into consideration. Comparing the RCs and PRs that all of the models produced before and after the SMOTE-AdaBoost application is shown in Figs. 8–9. Additionally, Figure 11 compares the MCCs before and after SMOTE-AdaBoost was implemented.

G. Experiments Validation

In this section uses a publicly available synthetic credit card fraud dataset for experimentation [32]. There are 29757 fraudulent credit card transactions and 24357143 legitimate ones in this dataset. Moreover, the dataset contains he following features F = {User, Card, Year, Month, Time, Day, Amount, Use Chip, Merchant Name, Merchant City, Merchant State, MCC, Zip, Errors, Is Fraud} where Is Fraud represents the class. Table 3 describes these characteristics.

TABLE 3. Synthetic dataset feature list

Feature List	Class
User, Card, Year, Month, Day, Time, Amount, Use Chip, Merchant Name, Merchant City, Merchant State, Zip, MCC, Errors,	Is Fraud

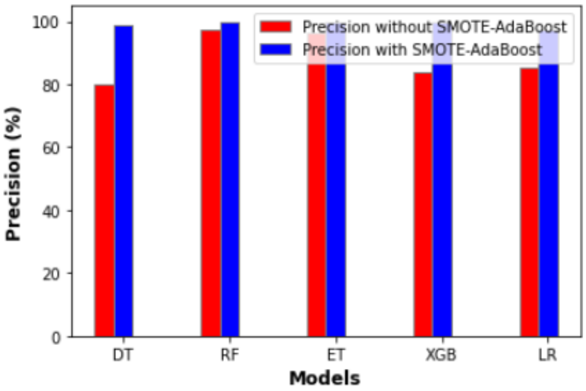


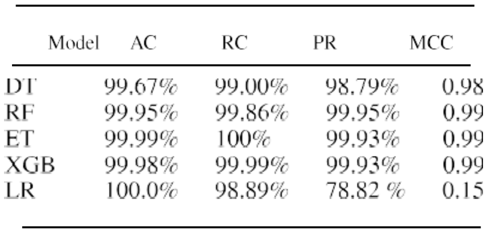
FIGURE 10. Recall, precision and MCC - Comparison.

The following models were employed in the experimental process: DT, RF, ET, XGB, and LR. AdaBoost was used to flexibly enhance each of these models. Table 4 displays the results. When compared to other models, the ET-AdaBoost model performed best, having 99.99% accuracy, 99.99% recall, 99.99% precision, and 0.99 MCC. The results

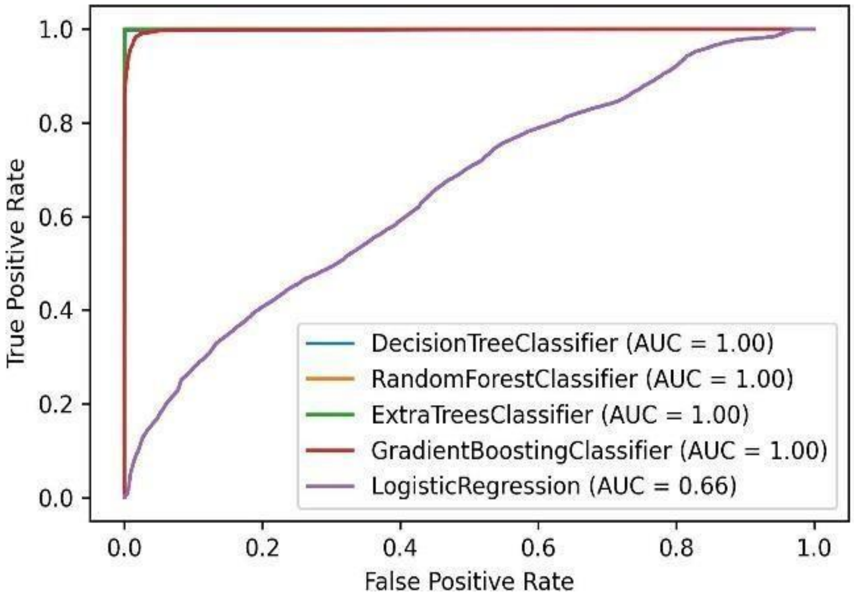
DOI: 10.17148/IJARCCE.2024.13441

generated by the DT-AdBoost, RF-AdaBoost, and RF- AdaBoost all reflect this pattern. These findings showed that a CCF detection engine's overall performance is enhanced when the SMOTE approach is used to CCF data and AdaBoost is used for the classification models. Furthermore, each of the proposed models' ROC curves is shown in Fig. 15, where results reveal that the AUC for the DT, RF, ET, and XGB was 1. By contrast, the AUC for the LR was 0.66. The MCC values presented in Table 4 are confirmed by these results.

TABLE 4. Results using AdaBoost on synthetic dataset.



Model	AC	RC	PR	MCC
DT	99.67%	99.00%	98.79%	0.98
RF	99.95%	99.86%	99.95%	0.99
ET	99.99%	100%	99.93%	0.99
XGB	99.98%	99.99%	99.93%	0.99
LR	100.0%	98.89%	78.82 %	0.15



FI GURE 11. ROC: DT, RF, XGB, LR

V. CONCLUSION

Using the September 2013-generated European credit card fraud dataset, this research constructed many machine learning methods for credit card fraud detection. This work comprised the DT, RF, ET, XGB, LR, and SVM as machine learning techniques. Furthermore, to address the problem of class imbalance in the European credit card fraud dataset and to improve classification quality, each of the suggested algorithms was combined with the AdaBoost methodology. Additionally, a comparative analysis was carried out between the approaches described in this paper and the frameworks currently in use for credit card fraud detection.

As an illustration, the accuracy of the DT-AdaBoost, RF-AdaBoost, ET-AdaBoost, and XGB-AdaBoost was 99.67%, 99.95%, 99.98%, and 99.98%, respectively. The ET-AdaBoost and the XGB-AdaBoost both received MCCs of 0.99 and 0.99, respectively, for classification quality. These results showed that the suggested ML techniques benefit from the use of the AdaBoost algorithm.

Furthermore, a highly skewed synthetic credit card fraud dataset was used to validate the framework suggested in this study, and the outcomes were ideal. As an example, the ET AdaBoost achieved an MCC of 0.99 and an accuracy of 99.99%. Additionally, the AUC value of 1 was reached by the XGB-AdaBoost, DT-AdaBoost, ET- AdaBoost, and RF-AdaBoost. We want to test and validate thesuggested approach using more credit card fraud datasets thatwe will obtain from financial organizations in subsequent work.

DOI: 10.17148/IJARCCE.2024.13441

REFERENCES

[1]. A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarachchi, ‘‘Real-time credit card fraud detection using machine learning,’’ in Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence), Jan. 2019, pp. 488–493.

[2]. S. P. Maniraj, A. Saini, S. Ahmed, and S. Sarkar, ‘‘Credit card fraud detection using machine learning and data science,’’ Int. J. Res. Appl. Sci. Eng. Technol., vol. 8, no. 9, pp. 3788–3792, Jul. 2021.

[3]. The Nilson Report. Accessed: Sep. 27, 2021. [Online]. Available:https://www.nilsonreport.com/upload/content_promo/The_Nilson_Report_10- 17-2016.pdf

[4]. The Nilson Report. Accessed: Sep. 27, 2021. [Online]. Available:https://nilsonreport.com/content_promo.php?id_promo=16

[5]. D. Elreedy and A. F. Atiya, ‘‘A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance,’’ Inf. Sci., vol. 505, pp. 32– 64, Dec. 2019

[6]. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ‘‘SMOTE: Synthetic minority over- sampling technique,’’ J. Artif. Intell. Res., vol. 16, no. 1, pp. 321–357, 2002.

[7]. Imalanced Learn. Accessed: Sep. 27, 2021. [Online].Available: https:// imbalanced-learn.org/stable/

[8]. Pandas. Accessed: Sep. 27, 2021. [Online]. Available: https://pandas. pydata.org/

[9]. S. Khatri, A. Arora, and A. P. Agrawal, ‘‘Supervised machine learning algorithms for credit card fraud detection: A comparison,’’ in Proc. 10th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence), Jan. 2020,pp. 680–683.

[10]. S. Rajora, D. L. Li, C. Jha, N. Bharill, O. P. Patel, S. Joshi, a. Puthal, and M. Prasad, ‘‘A comparative study of machine learning techniques for credit card fraud detection based on time variance,’’ in Proc. IEEE Symp. Comput. Intell. (SSCI), Nov. 2018, pp. 1958–1963.

[11]. N. K. Trivedi, S. Simaiya, U. K. Lilhore, and S. K. Sharma, ‘‘An efficient credit card fraud detection model based on machine learning methods,’’ Int. J. Adv. Sci. Technol., vol. 29, no. 5, pp. 3414–3424, 2020

[12]. R. Sailusha, V. Gnaneswar, R. Ramesh, and G. R. Rao, ‘‘Credit card fraud detection using machine learning,’’ in Proc. 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS), May 2020, pp. 967–972.

[13]. F. Z. El Hlouli, J. Riffi, M. A. Mahraz, A. El Yahyaouy, and
H. Tairi, ‘‘Credit card fraud detection based on multilayer perceptron and extreme learning machine architectures,’’ in Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV), Jun. 2020, pp. 1–5.

[14]. 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

[15]. R. E. Schapire, ‘‘Explaining adaboost,’’ in Empirical Inference. Berlin, Germany: Springer, 2013, pp. 37–52.

[16]. X. Li, L. Wang, and E. Sung, ‘‘AdaBoost with SVM- based component classifiers,’’ Eng. Appl. Artif. Intell., vol. 21, no. 5, pp. 785–795, Aug. 2008.

[17]. K. Kirasich, T. Smith, and B. Sadler, ‘‘Random forest vs logistic regression: Binary classification for heterogeneous datasets,’’ SMU Data Sci. Rev., vol. 1, no. 3, p. 9, 2018.

[18]. J. Feng, H. Xu, S. Mannor, and S. Yan, ‘‘Robust logistic regression and classification,’’ in Proc. Adv. Neural Inf. Process. Syst., vol. 27, 2014,pp. 253–261.

[19]. C.-C. Chern, Y.-J. Chen, and B. Hsiao, ‘‘Decision tree–based classifier in providing telehealth service,’’ BMC Med. Informat. Decis. Making, vol. 19, no. 1, pp. 1–15, Dec. 2019.

[20]. T. Hengl, M. Nussbaum, M. N. Wright, G. B. M. Heuvelink, and B. Gräler, ‘‘Random forest as a generic framework for predictive modeling of spatial and spatio- temporal variables,’’ PeerJ, vol. 6, p. e5518, Aug. 2018.

[21]. D. A. Pisner and D. M. Schnyer, ‘‘Support vector machine,’’ in Machine Learning. New York, NY, USA: Academic, 2020, pp. 101–121.

[22]. A. Tharwat, ‘‘Parameter investigation of support vector machine classifier with kernel functions,’’ Knowl. Inf. Syst., vol. 61, no. 3, pp. 1269–1302, Dec. 2019.

[23]. Ensemble Trees. Accessed: Sep. 27, 2021. [Online]. Available:https://scikitlearn.org/stable/modules/classes.html# module- sklearn.ensemble

[24]. T. T. Wong and P. Y. Yeh, ‘‘Reliable accuracy estimates from k-fold cross validation,’’ IEEE Trans. Knowl. Data Eng., vol. 32, no. 8, pp. 1586–1594, Apr. 2019.

[25]. Credit Card Fraud Detection. Accessed: Sep. 27, 2021. [Online].
Available: https://www.kaggle.com/mlgulb/creditcardfraud

[26]. Google Colab. Accessed: Sep. 27, 2021. [Online].
Available: https://colab.research.google.com/

[27]. Scikit-learn: Machine Learning in Python. Accessed: Sep. 27, 2021. [Online]. Available: https://scikit-learn.org/stable/

[28]. D. Chicco and G. Jurman, ‘‘The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,’’ BMC Genomics, vol. 21, no. 1, pp. 1–13, 2020.

DOI: 10.17148/IJARCCE.2024.13441

[29]. S. Boughorbel, F. Jarray, and M. El-Anbari, ‘‘Optimal classifier for imbalanced data using Matthews correlation coefficient metric,’’ PLoS ONE, vol. 12, no. 6, Jun. 2017, Art. no. e0177678.

[30]. M. Norton and S. Uryasev, ‘‘Maximization of AUC and buffered AUC in binary classification,’’ Math. Program., vol. 174, no. 1, pp. 575–612, 2019.

[31]. A. Luque, A. Carrasco, A. Martín, and A. de las Heras, ‘‘The impact of class imbalance in classification performance metrics based on the binary confusion matrix,’’ Pattern Recognit., vol. 91, pp. 216– 231, Oct. 2019.

[32]. E.R.Altman, ‘‘Synthesizing credit card transactions,’’ 2019,arXiv:1910.03033.