Received November 11, 2021, accepted December 6, 2021, date of publication December 15, 2021,
date of current version December 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3134330

Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost

EMMANUEL ILEBERI[1], YANXIA SUN[1], (Senior Member, IEEE), AND ZENGHUI WANG[2], (Member, IEEE)
[1]Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2094, South Africa
[2]Department of Electrical Engineering, University of South Africa, Florida 1709, South Africa

Corresponding author: Zenghui Wang (wangzengh@gmail.com)

This work was supported in part by the South African National Research Foundation under Grant 114911, Grant 137951, and Grant 132797, and in part by the Eskom Tertiary Education Support Programme Grants.

ABSTRACT The advance in technologies such as e-commerce and financial technology (FinTech) applications have sparked an increase in the number of online card transactions that occur on a daily basis. As a result, there has been a spike in credit card fraud that affects card issuing companies, merchants, and banks. It is therefore essential to develop mechanisms that ensure the security and integrity of credit card transactions. In this research, we implement a machine learning (ML) based framework for credit card fraud detection using a real world imbalanced datasets that were generated from European credit cardholders. To solve the issue of class imbalance, we re-sampled the dataset using the Synthetic Minority over-sampling TEchnique (SMOTE). This framework was evaluated using the following ML methods: Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Decision Tree (DT), and Extra Tree (ET). These ML algorithms were coupled with the Adaptive Boosting (AdaBoost) technique to increase their quality of classification. The models were evaluated using the accuracy, the recall, the precision, the Matthews Correlation Coefficient (MCC), and the Area Under the Curve (AUC). Moreover, the proposed framework was implemented on a highly skewed synthetic credit card fraud dataset to further validate the results that were obtained in this research. The experimental outcomes demonstrated that using the AdaBoost has a positive impact on the performance of the proposed methods. Further, the results obtained by the boosted models were superior to existing methods.

INDEX TERMS Credit card fraud, machine learning, predictive modeling.

I. INTRODUCTION

In recent years there has been an increase in financial fraud due to the growth of technologies and paradigms such as the e-commerce and the financial technology (FinTech) sectors [1]. The evolution of these technologies has sparked an increase in the number of credit card transactions. As a result, there has been a rapid spike in the number financial fraud cases that involved credit cards. Credit card *Fraud* occurs when an unauthorized or undesirable use of a credit card is made by a criminal. This happens when the credit card authentication details are stolen using different types of fraudulent techniques such as intercepting an e-commerce transaction or cloning an existing card [2]. Moreover, the impact of credit card fraud affects institutions such as card issuers, merchants, and small businesses. In 2015, the global loss due to credit card fraud was estimated at \$21.84 Billion [3]. In 2019, credit card losses reached \$28.65 Billion [4]. This represents an increase of \$6.81 Billion in 4 years. Therefore, it is crucial to implement credit card fraud detection systems that can guarantee the integrity and security of all systems that are involved in fulfilling credit card transactions.

In this paper, we implement machine learning (ML) algorithms for credit card fraud detection that are evaluated on a real world dataset which was generated from European cardholders in September 2013. This dataset is highly imbalanced. To alleviate the issue of class imbalance that is found in the European card dataset, this research investigated the use of the Synthetic Minority Over-sampling TEchnique (SMOTE) [5]. Moreover, the ML methods that were considered in this research include: Support Vector Machine (SVM), Random Forest (RF), Extra Tree (ET), Extreme Gradient Boosting (XGBoost), Logistic Regression (LR), and Decision

The associate editor coordinating the review of this manuscript and approving it for publication was Khin Wee Lai.

Tree (DT). These ML methods were evaluated individually in terms of their effectiveness and classification quality. Additionally, the Adaptive Boosting (AdaBoost) algorithm was paired with each methods to increase their robustness. The main contribution of this paper is a comparative analysis of several ML methods on a publicly available dataset that contains real word cards transactions. Moreover, this research investigate the AdaBoost to increase the quality of classification on a highly skewed credit card fraud dataset. The major contribution of this research work can be summarized as follows:

- We propose a credit card fraud detection framework that is scalable.
- We implement the SMOTE technique in order to solve the issue of class imbalance that is found in credit card fraud datasets.
- We pair the AdaBoost method with several ML methods to increase the performance on the proposed framework. Moreover, we conduct a comparison analysis using the following metrics: accuracy, recall, precision, Matthews Correlation Coefficient (MCC), and Area Under the Curve (AUC).
- We implement the proposed credit card fraud detection framework on a highly imbalanced synthetic dataset to validate its effectiveness.

The rest of the paper is organized as follows. Section 2 provides a literature review of previous work that used ML for credit card fraud detection. Section 3 provides a background of the ML methods that were used in this paper. In Section 4, we conduct the experiments. Section 5 presents the implementation of the proposed framework on a synthetic credit card fraud dataset. Section 6 concludes the research.

II. RELATED WORK

This section provides a literature review of previous researches that used ML techniques for credit card fraud detection.

Khatri *et al.* [9] implemented several ML algorithms for credit card fraud detection. In this research, the authors implemented the following methods: Decision Tree(DT), k-Nearest Neighbor (kNN), Logistic Regression (LR), Random Forest (RF), and Naive Bayes (NB). To evaluate the ML-based credit card fraud detection models, the researchers used a dataset that was generated from European cardholders in 2013 [25]. Moreover, the authors considered the sensitivity and the precision as the main performance metrics. The results showed that the kNN algorithm achieved the most optimal results with a precision of 91.11% and a sensitivity of 81.19%.

Rajora *et al.* [10] conducted a comparative research of ML methods for credit card fraud detection using the European cardholders dataset. Some of the methods that were investigated include the RF and the kNN methods. The authors considered the accuracy and the area under the curve (AUC) as the main performance metrics. The results demonstrated that RF algorithm achieved an accuracy of 94.9% and a AUC of 0.94. In contrast, the kNN obtained an accuracy of 93.2% and a AUC of 0.93. Although these results are promising, this research did not investigate the class imbalance issue that exists in the dataset that was used.

Trivedi *et al.* [11] proposed an efficient credit card fraud detection engine using ML methods. In this research, the authors considered many supervised ML techniques including Gradient Boosting (GB) and Random Forest (RF). The authors evaluated these methods using the European cardholders dataset. The performance metrics used to assess the effectiveness of the proposed approaches include the accuracy and the precision. The outcome of the experiments showed that the GB obtained an accuracy of 94.01% and a precision of 93.99%. On the other hand, the RF achieved an accuracy of 94.00% and a precision of 95.98%.

Tanouz *et al.* [12] presented a credit card fraud detection framework using ML algorithms. In this research, the authors used the European cardholders dataset to assess the performance of the proposed methods. Moreover, the authors implemented an under-sampling technique to solve the issue of class imbalance that exist in the dataset that was used. The ML methods considered in this work include the RF and LR. The researchers used the accuracy as the main performance metric. The results demonstrated that the RF approach achieved a fraud detection accuracy of 91.24%. In contrast, the LR method obtained an accuracy 95.16%. Furthermore, the authors computed the confusion matrix to assert whether these proposed methods performed optimally for the positive and negative classes. The results showed that the class imbalance issue that exist in the European credit card holder dataset requires further investigation.

Riffi *et al.* [13] implemented a credit card fraud detection engine using the Extreme Learning Machine (ELM) and Multilayer Perceptron (MLP) algorithms. Both the ELM and MLP are artificial neural networks (ANNs); however, they differ in terms of internal architecture. In this research, the authors used the European cardholders dataset that was generated in 2013. The authors used the fraud detection accuracy as the main performance metric. The results demonstrated that the MLP method achieved an accuracy of 97.84%. In contrast, the ELM attained credit card fraud detection accuracy of 95.46%. This work concluded that the MLP outperformed the ELM; however, the ELM is less complex in comparison to the MLP.

Randhawa *et al.* [14] The authors proposed a credit card fraud detection engine using Adaptive Boosting (AdaBoost) and Majority Voting (MV) methods. In this research, the authors used the European cardholders dataset. Moreover, the authors considered the AdaBoost method in conjunctions with ML methods such as the Support Vector Machine (SVM). In the experiments, the accuracy and the Matthews Correlation Coefficient (MCC) were considered as the main performance metrics. The results demonstrated that the AdaBoost-SVM achieved an accuracy of 99.959% and a MCC of 0.044.

III. BACKGROUND ON MACHINE LEARNING ALGORITHMS

A. MACHINE LEARNING ALGORITHMS

B. AdaBoost

Boosting is an approach to ML that aims at creating (generating) highly accurate models by the combination of several simple or inaccurate models [15], [16]. This research implements the AdaBoost algorithm in conjunction with other ML methods to improve their classification performance. The output of the AdaBoost method is a weighted sum. This is done by combining the output of the individual boosted models. Below is the mathematical formulation of the AdaBoost method:

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

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 $h(x_n)$. 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:

$$L_t = \sum_{n} L[G_{t-1}x_n + \beta_t h(x_n)] \tag{2}$$

where G_{t-1} is a classifier that was boosted at iteration $t-1$ and $\beta_t h(x_n)$ is a weak classifier that is considered for the final model.

C. ADDITIONAL ML METHODS

The AbaBoost method was used in conjunction with the following supervised ML methods: 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 used to improve the performance of individual classifiers with regard to performance metrics such as the accuracy, the Matthew Correlation Coefficient (MCC), and Area Under the Curve (AUC). These metrics are discussed in more detail in the Experiments section of the paper.

The LR (Logit classifier) is a supervised ML method that is efficient for binary classification tasks [18]. The LR method uses a linear function in the Logit function in order to make predictions. The SVM is another supervised ML technique that is used for regression and classification tasks. This method is highly efficient on data with a high dimensional feature space and it is versatile in terms of using different kernel functions (decision methods) [22].

The DT algorithm is a non-parametric supervised ML approach that is often used for regression and classification. This approach uses a tree-like construct to make the predictions. Some of the advantages of using DT include the fact that is simple to interpret and it does not require an extensive data preparation. DTs are the foundation of algorithms such as the RF, ET, and the XGBoost. These methods form part of what is labeled *Ensemble Tree* [23] because they fit many DTs on a given dataset in order to make predictions.

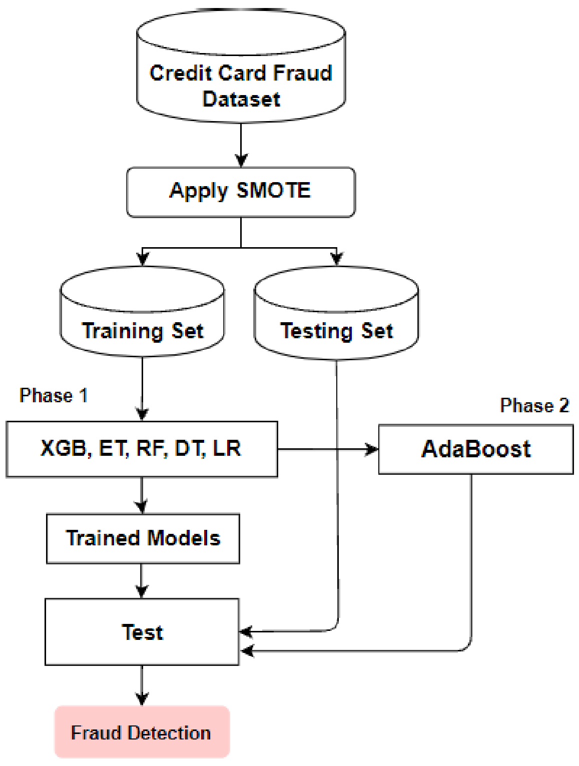


FIGURE 1. Credit card fraud detection framework.

IV. RESEARCH METHODOLOGY

A. FRAUD DETECTION FRAMEWORK

Fig.1 depicts the fraud detection framework that was implemented in this research. In the first step, the credit card fraud (CCF) dataset is loaded through the SMOTE block. In the second step, the CCF dataset is divided into a training and a test set. The third step involves the instantiation of the models (XGB, ET, RF, DT, and LR). Once the model are instantiated; they are trained (using the training set) and tested (using the testing set). Moreover, the k-fold cross-validation (CV) technique is used during the training process to avoid overfitting and to increase the reliability of the experimental results [24]. In the fourth step, the instantiated models go through the AdaBoost module. 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 were 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** 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].

Algorithm 1 SMOTE (T, N, k)

```
1:  Input $T$, the total number of instances in the minority class; $N$, the percentage (amount of SMOTE). $k$, the number of neighbours.
2:  Ouput $\frac{N}{100} * T$, the newly created synthetic data points
3:  if $N < 100$ then
4:      Generate $T$ minority class data points randomly.
5:      $T = (N/100) * T$
6:      $N = 100$
7:  end if
8:  $N = int(\frac{N}{100})$
9:  $num\_attrs$, the number of attributes
10: $k$, the number of nearest neighbours
11: $sample$,
12: $new\_index$, keeps tabs on the number of synthetic data points that were generated. It is initialized with 0.
13: $synthetic\_array$, an array to keep synthetic data points
14: for $t$ range(1 to $T$) do
15:     Calculates the $k$ nearest neighbours for $t$ and save the indices in $nn\_array$
16:     Populate($N$, $t$, $nn\_array$ (this is a function that computes synthetic samples)
17: end for
18: Populate($N, t, nn\_array$
19: while $N \neq 0$ do
20:     Randomly select an number between 1 and $k = rn$
21:     for $at$ in range (1 to $num\_attrs$) do
22:         Calculate the difference: $\delta = sample[nn\_array[rn][at]]$ - $sample[i][at]$
23:         Compute the gap: $gap$ = random(0,1) - random numbers between 0 and 1.
24:         $synthetic\_array[new\_index][at] = sample[i][at]] + gap * \delta$
25:     end for
26:     increment the new index: $new\_index$++
27:     $N = N$ - 1
28: end while
```

Algorithm 2 SMOTE Implementation - Credit Card Fraud Dataset

```
1:  Start
2:  Input Credit card fraud dataset ($DF$) containing minority 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 $DF$ in a $pd$ dataframe
7:  Separate the dataframe into input data, $X$, and target data, $y$
8:  Instantiate SMOTE instance as $sm$ = 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
```

D. EXPERIMENTAL SETUP

The classification experiments were conducted on Google Colab [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 that is used in this research contains traces of legitimate and fraudulent transactions that are labeled as 1s and 0s. Therefore, we have framed this ML task as a binary classification task. Such problems are evaluated using performance metrics that includes: the accuracy (AC), the recall (RC), and the precision (PR). The mathematical formulation of these indicators is as follows:

- False positive (FP): correct transactions that are incorrectly labeled as fraudulent.
- False Negative (FN): fraudulent transactions that are incorrectly classified as legitimate transactions.
- True positive (TP): fraudulent activities that are accurately flagged fraudulent.
- True Negative (TN): genuine transactions that are positively classified as genuine.

$$AC = \frac{TN + TP}{TP + TN + FN + Fp} \tag{3}$$

$$PR = \frac{TP}{TP + FP} \tag{4}$$

$$RC = \frac{TP}{TP + FN} \tag{5}$$

Furthermore, the European cardholders dataset is highly imbalanced. Therefore, considering the AC, PR, and the RC metrics is not enough to assess the performance of our

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 with the 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

proposed method. In this research, we consider the Matthews correlation coefficient (MCC) [28], [29], the AUC [30], and the Confusion Matrix (CM) as additional performance indicators. In this instance, the MCC is used a measure of the quality of the classification task. The value of the MCC measure varies between -1 and $+1$. The closer the MCC is to $+1$, the higher the quality of classification. Furthermore, the CM [31] is a graph that allows us to see the mistakes that were made by a specific classifier. Additionally, the Area Under the Curve (AUC) of each model was computed to evaluate the classification reliability and quality. The AUC is a measure that determines the effectiveness of a classifier. The value of the AUC varies between 0 and 1 whereby an optimal classifier would have an AUC value close to 1 [30].

$$MCC = \frac{(TN \times TP) - (FN \times FP)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{6}$$

F. EXPERIMENTS, RESULTS, AND DISCUSSIONS

As depicted in Figure 1, the experimental process was carried out in two phases. In the initial phase, we implemented the ML models without the use of the AdaBoost method. The results are depicted in Table 1 whereby the algorithm that performed optimally in terms the quality of classification is the RF with an MCC of 0.88. In terms of accuracy, the classifier that performed the best are the RF and the ET with ACs of 99.5%. In the second phase, the AdaBoost was paired with each ML algorithm. The experimental outcome showed that the DT registered an MCC increase of 0.20. The XGB registered an MCC spike of 0.28. In terms of fraud detection AC, both the XGB and the ET achieved an optimal AC of 99.98%. Furthermore, in Figure 2 - 4, the confusion matrix (CM) of each model were computed to assert where the algorithm made some mistakes. In Fig., the DT algorithm successfully isolated legitimate transactions; however, it made quite a few errors in predicting fraudulent transactions. In contrast, the DT-AdaBoost, depicted in Fig.3, shows some improvement in terms of detecting fraudulent transactions. This pattern can also be seen in Figure 4

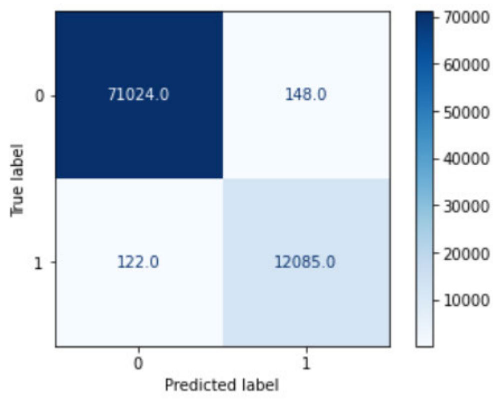


FIGURE 2. DT confusion matrix.

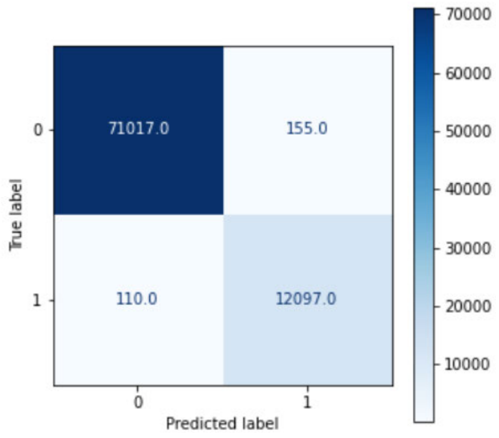


FIGURE 3. DT-AdaBoost confusion matrix.

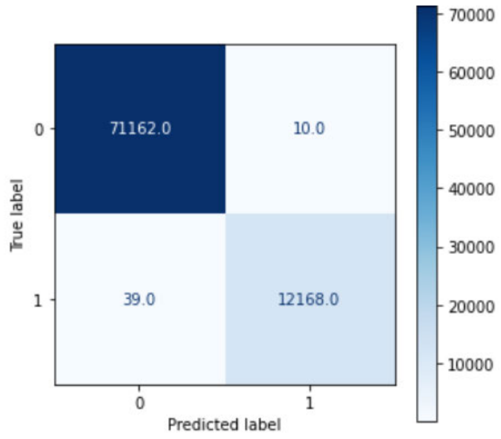


FIGURE 4. RF confusion matrix.

- Figure 9, where by the RF-AdaBoost, ET-AbaBoost, LR-AdaBoost, XGB-AdaBoost performed optimally with regards to isolating fraudulent transactions.

In Table 3, a comparison analysis was conducted between the algorithms proposed in this research and existing ML-based credit card fraud detection frameworks. The results showed that the XGB-AdaBoost and the ET-AdaBoost achieved fraud detection ACs that are 5.08% higher than the RF presented in [10] and 6.78% higher than the KNN presented in [10]. In comparison to the work presented

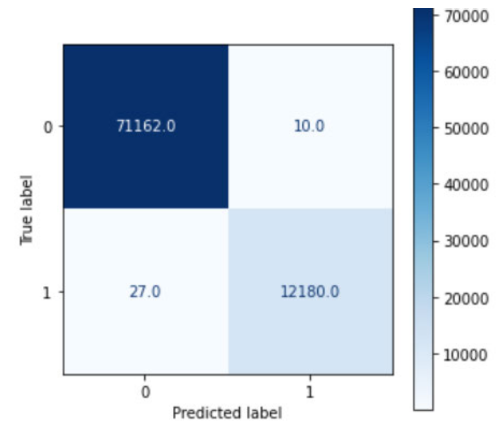


FIGURE 5. RF-AdaBoost confusion matrix.

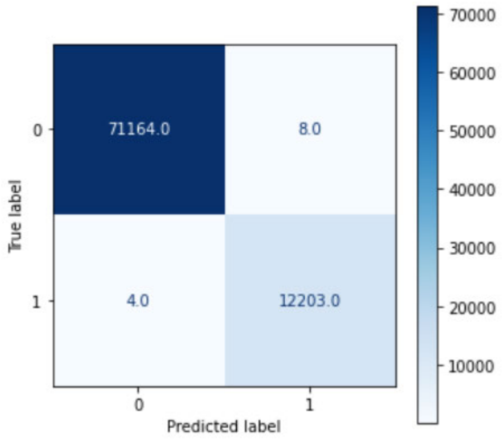


FIGURE 6. ET confusion matrix.

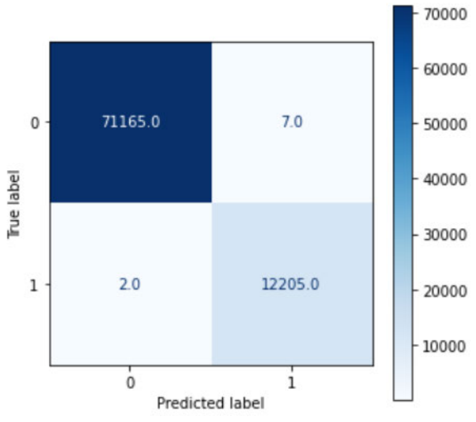


FIGURE 7. ET-AdaBoost confusion matrix.

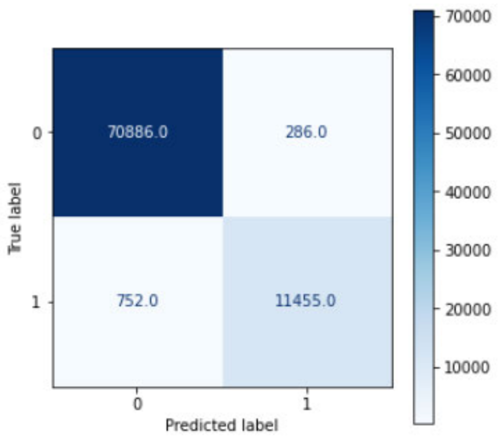


FIGURE 8. LR confusion matrix.

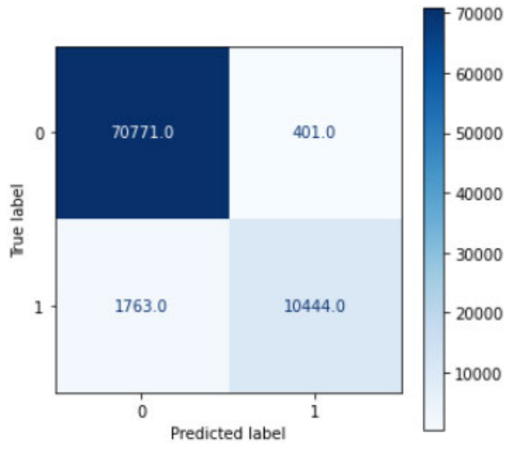


FIGURE 9. LR-AdaBoost confusion matrix.

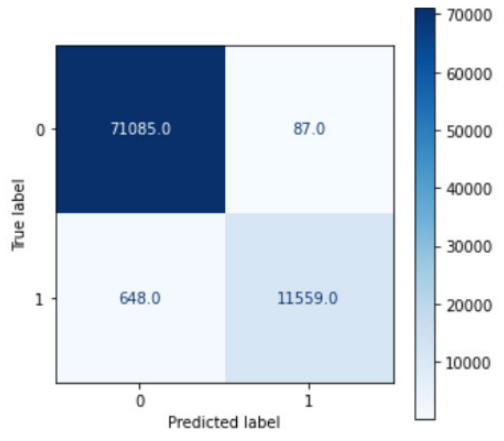


FIGURE 10. XGB confusion matrix.

in [12], the XGB-AdaBoost obtained an AC that is 8.74% higher. Further, in contrast with the work in [13], the ET-AdaBoost achieved an AC that is 4.34% higher.

Furthermore, the use of the SMOTE-AdaBoost methodologies have improved the performance of all the models when considering the precision and the recall. For instance, the DT model, without SMOTE-AdaBoost, achieved a recall of 75.75% in contrast to 99.00% when the SMOTE-AdaBoost algorithms were applied. When considering the precision, the DT achieved an precision of 79.83% without the SMOTE-AdaBoost methods. However, when the SMOTE-AdaBoost was applied, the DT model obtained a precision of 98.79%. As a result the MCC also improved from 0.78 to 0.98. This pattern is observed in all the models that were considered in this research. Fig. 12 -13 depict a comparison

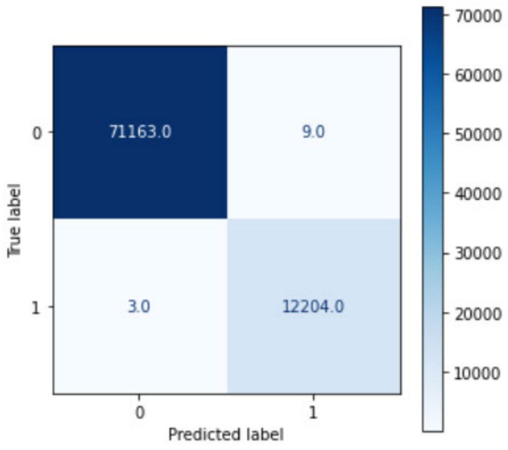


FIGURE 11. XGB-AdaBoost confusion matrix.

TABLE 3. Comparison with existing methods.

Author	Model	AC
Rajora et al. [10]	RF	94.90 %
Rajora et al. [10]	KNN	93.20 %
Trivedi et al. [11]	RF	94.00 %
Tanouz et al. [12]	RF	91.24 %
Tanouz et al. [12]	LR	95.16 %
Riffi et al. [13]	MLP	97.84 %
Riffi et al. [13]	ELM	95.46 %
Proposed method	RF-AdaBoost	99.95 %
Proposed method	DT-AdaBoost	99.67 %
Proposed method	ET-AdaBoost	99.98 %
Proposed method	XGB-AdaBoost	99.98 %

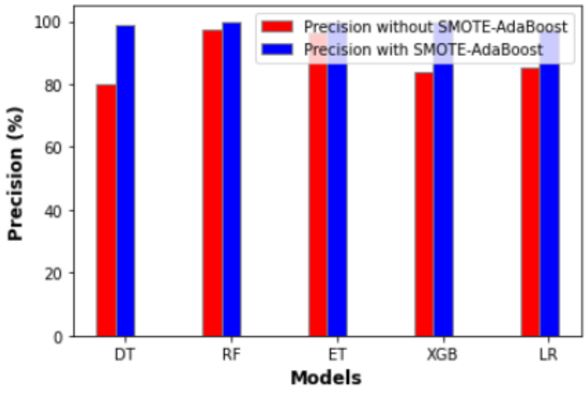


FIGURE 12. Precision comparison.

between the RCs and PRs that were obtained by all the models before and after the application of the SMOTE-AdaBoost. Additionally, Figure 14 shows the comparison of the MCCs before and after the implementation the SMOTE-AdaBoost.

G. EXPERIMENTS VALIDATION

In this section, experiments are conducted on a synthetic credit card fraud dataset which is publicly available [32]. This dataset includes 24357143 genuine credit card transactions and 29757 fraudulent ones. Moreover, the dataset contains the 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. These attributes are listed in Table 4. The experiments process were conducted using the following models: DT, RF, ET, XGB, LR. All these models were adaptively boosted (using AdaBoost). The results are listed in Table 5. The model that performed optimally in comparison to other models is the ET-AdaBoost with an accuracy of 99.99%, a recall of 99.99%, a precision of 99.99% and

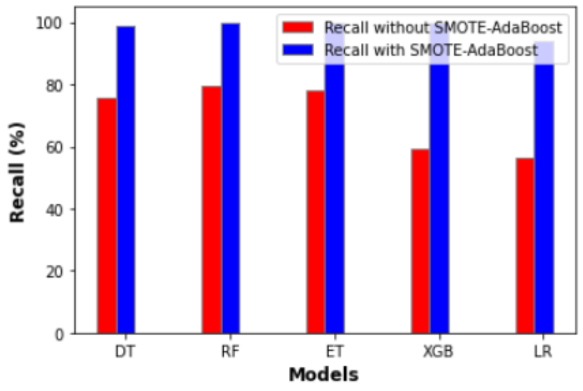


FIGURE 13. Recall comparison.

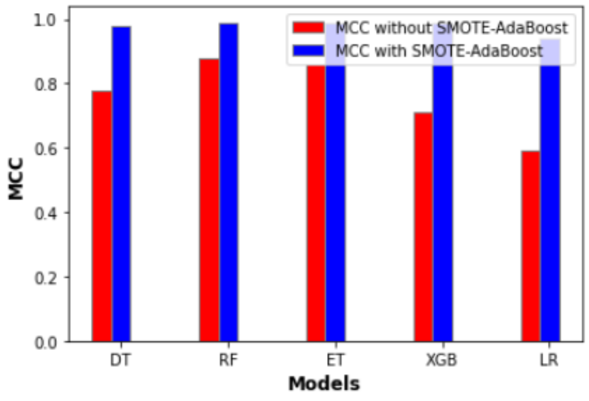


FIGURE 14. Recall, precision and MCC - Comparison.

TABLE 4. 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

TABLE 5. 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

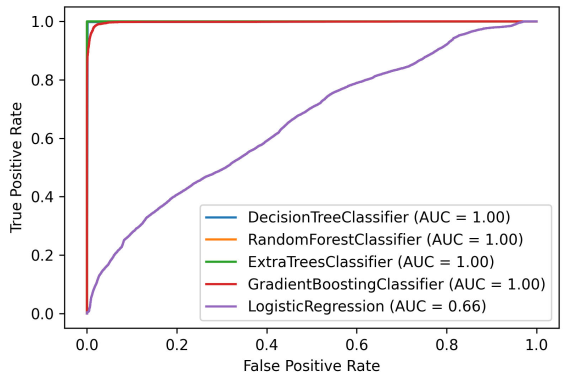


FIGURE 15. **ROC: DT, RF, ET, XGB, LR.**

MCC of 0.99. This pattern can also be observed in the outcomes obtained by the DT-AdBoost, RF-AdaBoost and RF-AdaBoost. These results demonstrated that using the SMOTE method on CCF data in conjunction with AdaBoost on the classification models has a positive impact on the overall performance of a CCF detection engine. Additionally, Fig. 15 depicts the ROC curves of each proposed model and the results show that the DT, RF, ET, and XGB obtained an AUC of 1. In contrast, the LR achieved an AUC of 0.66. These results validate the MCC values listed in Table 5.

V. CONCLUSION

This paper implemented several ML algorithms for credit card fraud detection using the European credit card fraud dataset that was generated in September 2013. The ML methods proposed in this work included the DT, RF, ET, XGB, LR and SVM. Additionally, each of the proposed algorithms was paired with the AdaBoost technique to increase the quality of classification and to deal with the issue of class imbalance that is present in the European credit card fraud dataset. Further, a comparison analysis was conducted between the methods presented in this work and existing credit card fraud detection frameworks. For instance, the DT-AdaBoost, RF-AdaBoost, ET-AdaBoost, and XGB-AdaBoost achieved accuracies of 99.67%, 99.95%, 99.98%, and 99.98%, respectively. In terms of the quality of classification, the ET-AdaBoost obtained an MCC of 0.99 and the XGB-AdaBoost achieved an MCC of 0.99. These outcomes demonstrated that using the AdaBoost algorithm has a positive impact on the proposed ML methods. Moreover, the framework proposed in this research was validated using a highly skewed synthetic credit card fraud dataset and the results were optimal. For instance, the ET-AdaBoost obtained an accuracy of 99.99% and a MCC of 0.99. Moreover, the XGB-AdaBoost, DT-AdaBoost, ET-AdaBoost, and RF-AdaBoost attained an AUC value of 1. In future works, we intend to test and validate the proposed framework on additional credit card fraud datasets that will be sourced from financial institutions.

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, D. 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.

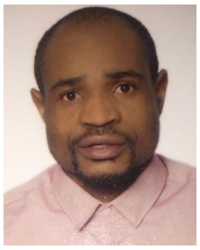
[21] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning*. New York, NY, USA: Academic, 2020, pp. 101–121.

[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://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble

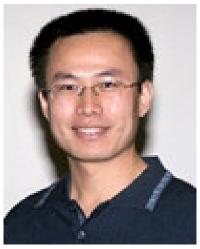
[24] T. T. Wong and P. Y. Yeh, "Reliable accuracy estimates from k-fold cross validation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1586–1594, Apr. 2019.

[25] *Credit Card Fraud Detection*. Accessed: Sep. 27, 2021. [Online]. Available: https://www.kaggle.com/mlg-ulb/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.
[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*.



EMMANUEL ILEBERI received the M.Sc. degree in telecommunications systems and computer network engineering from the Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus, in 2018. He is currently pursuing the Ph.D. degree in electrical and electronic engineering with the University of Johannesburg, South Africa. His research interests include machine learning and credit card fraud detection.

YANXIA SUN (Senior Member, IEEE) received the D.Tech. degree in electrical engineering from the Tshwane University of Technology, South Africa, and the Ph.D. degree in computer science from University Paris-EST, France, in 2012. She is currently working as a Professor with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, South Africa. Her research interests include renewable energy, evolutionary optimization, neural networks, nonlinear dynamics, and control systems.



ZENGHUI WANG (Member, IEEE) received the B.Eng. degree in automation from the Naval Aviation Engineering Academy, China, in 2002, and the Ph.D. degree in control theory and control engineering from Nankai University, China, in 2007.

He is currently a Professor with the Department of Electrical and Mining Engineering, University of South Africa (UNISA), South Africa. His research interests include industry 4.0, control theory and control engineering, engineering optimization, image/video processing, artificial intelligence, and chaos.