IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 9, Issue 2 - V9I2-1155)
Available online at: https://www.ijariit.com

Credit card fraud detection: An evaluation of Machine Learning methods performance using SMOTE and AdaBoost

Kethe Meghana
meghanakethe@gmail.com
Annamacharya Institute of Technology and Sciences, Boyanapalli, Andhra Pradesh

Nidimamidi Thahseen
nthahseen2@gmail.com
Annamacharya Institute of Technology and Sciences, Boyanapalli, Andhra Pradesh

Duragadda Dhana Lakshmi
dhanaduragadda@gmail.com
Annamacharya Institute of Technology and Sciences, Boyanapalli, Andhra Pradesh

Vamsharajula Seenu
sreenusmiley4808@gmail.com
Annamacharya Institute of Technology and Sciences, Boyanapalli, Andhra Pradesh

Vattam Veda Prakash
vedaprakash756@gmail.com
Annamacharya Institute of Technology and Sciences, Boyanapalli, Andhra Pradesh

Pola Nikhila
nikilareddy57@gmail.com
Annamacharya Institute of Technology and Sciences, Boyanapalli, Andhra Pradesh

ABSTRACT

Online card transactions have increased daily as a result of the development of technologies like e-commerce and financial technology (FinTech) apps. As a result, there has been an increase in credit card fraud that impacts banks, merchants, and card issuers. Thus, it is critical to create systems that guarantee the confidentiality and accuracy of credit card transactions. In this study, we use imbalanced real-world datasets produced from European credit cardholders to create a machine learning (ML) based framework for detecting credit card fraud. In order to address the class imbalance problem, we resampled the dataset using the Synthetic Minority over-sampling Technique (SMOTE). The following machine learning (ML) techniques were used to assess this framework: Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Decision Tree (DT), and Extra Tree (ET). For better categorization, these ML algorithms were combined with the Adaptive Boosting (AdaBoost) method. Accuracy, recall, precision, Matthews Correlation Coefficient (MCC), and Area Under the Curve were used to assess the models (AUC). Additionally, the suggested framework was used to a highly skewed synthetic credit card fraud dataset in order to further confirm the findings of this study. The results of the experiments showed that utilizing AdaBoost improves the effectiveness of the suggested strategies. The outcomes produced by the boosted models were also better than those of earlier techniques.

Keywords: *Credit Card Fraud, Machine Learning, Predictive Modelling*

1. INTRODUCTION

Due to the development of new technologies and paradigms, including those in the e-commerce and financial technology (FinTech) sectors, there has been an increase in financial fraud in recent years [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 rapidly increased. When a criminal uses a credit card in an unauthorised or undesired way, that behaviour is known as credit card fraud.

This occurs when the authentication information for a credit card is obtained through a variety of dishonest methods, such as intercepting an online transaction or copying an already existing card [2]. Furthermore, institutions impacted by credit card theft include merchants, small enterprises, and card issuers. The estimated global loss from credit card theft in 2015 was $21.84 billion [3]. Losses from credit cards totalled $28.65 billion in 2019 [4].

This is a rise of $6.81 billion in the last 4 years. Consequently, it is essential to put in place credit card fraud detection systems that can ensure the security and integrity of all systems involved in processing credit card transactions. In this study, we construct machine learning (ML) algorithms for credit card fraud detection and test them using a real-world dataset collected in September 2013 from cardholders in Europe. This dataset has a severe imbalance. In this study, the Synthetic Minority Over-sampling

Technique (SMOTE) was used to address the problem of class imbalance that is present in the European Card dataset [5].

Additionally, Support Vector Machine (SVM), Random Forest (RF), Extra Tree (ET), and Extreme Gradient credit card fraud detection models were taken into account in this study. The researchers employed a dataset that was created from European cardholders in 2013 [25]. The authors also considered sensitivity and precision to be the key performance indicators. The findings demonstrated that the KNN algorithm produced the best outcomes, with a precision and sensitivity of 91.11% and 81.19%, respectively. European Boosting (XGBoost), Logistic Regression (LR), and Decision Tree were used in a comparative study by Rajora et al. [10] to identify credit card fraud (DT).

The efficiency and classification quality of each of these ML techniques were assessed separately. To strengthen the resilience of each method, the Adaptive Boosting (AdaBoost) algorithm was combined with it. This paper's main contribution is a comparison of various machine learning techniques using a publicly available dataset of actual word card transactions. Additionally, this study investigates the AdaBoost to improve classification accuracy on a credit card fraud dataset that is extremely skewed.

2. RELATED WORK

This section gives an overview of prior studies that employed ML methods for detecting credit card fraud.
For the purpose of detecting credit card fraud, Khatri et al. [9] constructed different ML algorithms.

The author used the following techniques in this study: Decision Tree (DT), k-Nearest Neighbour (KNN), Logistic Regression (LR), Random Forest (RF), and Naive Bayes (NB). To assess the dataset of cardholders using ML. The RF and KNN approaches were just two of the techniques looked into. The accuracy and area under the curve (AUC) were thought to be the key performance indicators by the authors. The outcomes showed that the RF algorithm managed to attain an accuracy of 94.9% and an AUC of 0.94.

The KNN, in contrast, achieved an accuracy of 93.2% and an AUC of 0.93. Although the results are encouraging, the class imbalance problem that occurs in the dataset used was not addressed in this study. An effective credit card fraud detection engine utilising ML technique was proposed by Trivedi et al. [11]. The authors of this study took into account a variety of supervised machine learning approaches, such as Gradient Boosting (GB) and Random Forest (RF).

The European Cardholders dataset was used by the authors to test different techniques. Accuracy and precision are two performance indicators used to gauge how well the suggested procedures work. The results of the tests revealed that the GB achieved an accuracy and precision of 94.01% and 93.99%, respectively. The RF, on the other hand, managed an accuracy and precision of 94.00% and 95.98%, respectively.

The Extreme Learning Machine (ELM) and Multilayer Perceptron (MLP) algorithms were used by Riffi et al. [13] to create a credit card fraud detection engine. Although both the ELM and MLP are artificial neural networks (ANNs), their core architectures vary. The European Cardholders dataset, which was created in 2013, was used by the authors of this study. The accuracy of fraud detection was the primary performance metric employed by the authors. The outcomes showed that the MLP approach has a 97.84% accuracy rate. The ELM, in comparison, achieved a 95.46% accuracy rate for detecting credit card fraud. This study found that the MLP performed better than the ELM, yet the ELM is less sophisticated than the MLP.[14]

Randhawa et al. Utilizing Adaptive Boosting (AdaBoost) and Majority Voting (MV) techniques, the authors suggested a credit card fraud detection engine. The European Cardholders dataset was used in this study by the authors. Additionally, the AdaBoost technique was taken into account by the authors in conjunction with ML techniques such the Support Vector Machine (SVM). The Matthews Correlation Coefficient (MCC) and accuracy were the key performance indicators in the experiments. The outcomes showed that the AdaBoost-SVM attained a 99.959% accuracy rate and a

3. BACKGROUND ON MACHINE LEARNINGALGORITHMS

Machine Learning Algorithms

AdaBoost

Boosting is a method of machine learning that seeks to produce highly accurate models by combining a number of imperfect or simple models [15, 16]. The AdaBoost algorithm is used in this study to enhance the classification performance of other ML techniques. The weighted sum is the result of the AdaBoost technique. By merging the results of the many boosted models, this is accomplished. The AdaBoost method's mathematical formulation is provided below:

$$GN(x) = X_{t=1}^{N} gt(x)$$

where t stands for an iteration and gt is a weak learner (basic classifier) that produces a prediction given an input vector x. The prediction of a weak learner is represented by h for each training sample (xn). The training error, L, is then calculated by selecting a weak learner and multiplying it by a coefficient at each t, as follows:

$$Lt = \quad L[Gt-1xn + \beta th(xn)]$$

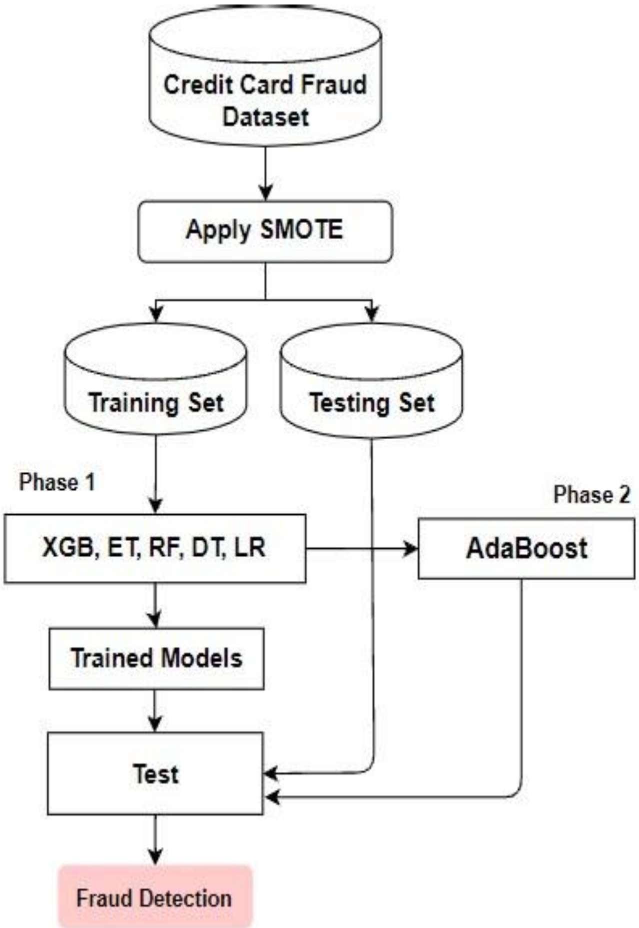
where Gt−1 is a classifier that was boosted at iteration t − 1 and βth(xn) is a weak classifier that is considered for the final model.

Additional ML Methods

The supervised ML techniques 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 were combined with the Adaboost approach (XGBoost). Individual classifiers' performance is enhanced using the AdaBoost method in terms of performance indicators including accuracy, the Matthew Correlation Coefficient (MCC), and Area Under the Curve (AUC). In the paper's Experiments section, these metrics are covered in more detail.

A supervised ML technique that is effective for binary classification applications is the LR (Logit classifier) [18]. To create predictions, the LR technique applies a linear function to the Logit function. Another supervised machine learning method used for regression and classification tasks is the SVM. This approach is flexible in its use of various kernel functions (decision procedures) and extremely effective on data with a high dimensional feature space [22].

Regression and classification are frequent uses of the DT algorithm, a non-parametric supervised ML method. The predictions are made in this method using a structure resembling a tree. Utilizing DT has some benefits, such as the ease of interpretation and lack of extensive data preparation requirements. The DT is the cornerstone of algorithms like the RF, ET, and XGBoost. Because they fit numerous DTs to a particular dataset in order to produce predictions, these techniques fall under the umbrella term "Ensemble Tree" [23].



4. RESEARCH METHODOLOGY

The architecture for detecting fraud used in this study is shown in Fig. 1. The SMOTE block is used to load the credit card fraud (CCF) dataset in the first stage. A training set and a test set are created from the CCF dataset in the second stage. The models are instantiated in the third stage (XGB, ET, RF, DT, and LR). Using the training data set, the model is trained after it is created (using the testing set). Furthermore, the k-fold cross-validation (CV) method is utilized during the training phase to prevent overfitting and boost the dependability of the experimental results [24]. The AdaBoost module is the fourth stage, which the instantiated models undergo. The models undergo training and testing at the end of the AdaBoost procedure. Both the non-boosted and boosted models' performance are examined by the Fraud Detection module.

Data Set

In September 2013, European cardholders supplied the dataset that was used for this study. This hugely biased dataset is accessible to the general public on Kaggle [25]. The transactions found in this dataset unfolded throughout time because it is not a fake dataset.

Likewise, there were 284807 total card transactions in the sample, with 99.828% of them being valid and 0.172% being fraudulent. Along with that, it has 30 properties (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].

Smote Applied to Credit Card Fraud Dataset

In order to fix the issue of class imbalance that is present in datasets like those used to design credit card fraud detection ML, amongst the most popular strategies is the Synthetic Minority over-sampling Technique (SMOTE). One of the most popular approaches to combat the issue of class imbalance that is present in datasets like those used to create credit card fraud detection ML based models is the Synthetic Minority over-sampling Technique (SMOTE)[5]. By linking a data point with its closest neighbours, the SMOTE approach creates samples of a certain class. SMOTE produces artificial data points that are not an exact duplicate of the minority class instance. By doing this, the phenomenon of over-fitting is prevented during the training process. The SMOTE technique [6] that was applied in this study is shown in pseudo code in Algorithm 1. In Algorithm 2, the Imblearn library is used to implement the SMOTE method in pseudo code on the credit card dataset used in this study [7].

Algorithm 1 SMOTE (T, N, k)
Input T, the total number of instances in the minority class; N, the percentage (amount of SMOTE). k, the number of neighbours.
Output N 100 ∗ T, the newly created synthetic data points

if N < 100 then
Generate T minority class data points randomly
T = (N/100) * T
N = 100
end if
N = int (N 100)
num_attrs, the number of attributes
k, the number of nearest neighbours
sample,
new_index, keeps tabs on the number of synthetic data points that were generated. It is initialized with 0.
synthetic_array, an array to keep synthetic data points
for t range (1 to T) do
Calculates the k nearest neighbours for t and save the indices in nn_array
Populate (N, t, nn_array (this is a function that computes synthetic samples)
end for
Populate (N, t, nn_array
while N 6= 0 do
Randomly select a number between 1 and k = rn
for at in range (1 to num_attrs) do
Calculate the difference: δ = sample[nn_array[rn][at]] - sample[i][at]
Compute the gap: gap = random (0, 1) - random numbers between 0 and 1.
synthetic_array[new_index] [at] = sample[i][at]] + gap * δ
end for
increment the new index: new_index++
N = N – 1
end while

Algorithm 2 SMOTE Implementation - Credit Card Fraud Dataset

Start
Input Credit card fraud dataset (DF) containing minority class data points
Output an oversampled dataset: Xres, input data and yres, the target
Import the SMOTE module from imblearn [7]
Import pandas (pd) from pandas [8]
Read DF in a pd data frame
Separate the data frame into input data, X, and target data, y
Instantiate SMOTE instance as sm = SMOTE (m : r), where m is the minority class and r the ratio.
Fit the SMOTE instance as follows: Xres, yres = sm.fit_resample(X, y)
End

5. PERFORMANCE METRICS
The credit card fraud dataset that was employed in this study includes traces of both honest and dishonest transactions, which are denoted by 1s and 0s. As a result, we have defined this ML problem as a binary classification task. Performance measurements like as accuracy (AC), recall (RC), and precision are used to assess these issues (PR). The

following is how these indicators are mathematically formulated:

• False positives (FP): Legitimate transactions that are mistakenly classified as fraudulent.
• False Negative (FN): fraudulent transactions that are misclassified as legal transactions.
• True positive (TP): fraudulent acts that are correctly detected as fraudulent.
• True Negative (TN): Sincere transactions that have been adjudged to be sincere.

AC = TN + TP TP + TN + FN + Fp (3)
PR = TP TP + FP (4)
RC = TP TP + FN (5)

The dataset of European cardholders is also very unbalanced. As a result, analysing the efficacy 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

Suggested strategy solely on the basis of the AC, PR, and RC measures is insufficient. As additional performance metrics in this study, we also take into account the Matthews correlation coefficient (MCC) [28, [29], the AUC [30], and the Confusion Matrix (CM). The MCC is employed in this case as a gauge for the effectiveness of the categorization task. The MCC measure's value ranges from -1 to +1. The categorization quality increases as the MCC passes +1. Secondly, the CM [31] is a graph that permits us to view the errors committed by a specific classifier. Moreover, the Area under the Curve (AUC) of each model was calculated to assess the accuracy and reliability of the diagnosis. The AUC is a metric used to assess a classifier's performance. An ideal classifier had an AUC value close to 1 [30]. The value of the AUC ranges from 0 to 1.

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

6. EXPERIMENTS, RESULTS, AND DISCUSSIONS

The experimental setup was run in two steps, as shown in Figure 1. We did not employ the AdaBoost approach when we first developed the ML models. The findings are shown in Table 1, where the RF method, with an MCC of 0.88, performed best in terms of classification quality. The RF and ET classifiers, with ACs of 99.5%, performed best in terms of accuracy. The AdaBoost was matched with each ML algorithm in the second phase. In accordance with trial results, the DT recorded an MCC uptick of 0.20.
The MCC spike in the XGB was 0.28. Both the XGB and the ET obtained an ideal AC of 99.98% for fraud detection. Additionally, the confusion matrix (CM) of each model was computed in Figures 2 through 4 to show where the algorithm made mistakes. The DT algorithm identified valid transactions in Fig. effectively, however it made plenty of mistakes in forecasting fraudulent transactions. The DT-AdaBoost, shown in Fig. 3, on the other hand, exhibits modest improvement in terms of identifying fraudulent transactions. This tendency is also obvious in Figure 4 and Figure 9, where the RF-AdaBoost, ET-Aba Boost, LR-AdaBoost, and XGB-AdaBoost all outperformed one another in terms of identifying fraudulent transactions. Table 3 contains a comparison analysis between the methods suggested in this study and current ML-based schemes for detecting credit card fraud. The findings demonstrated that the XGB-AdaBoost and the ET-AdaBoost produced fraud detection ACs that are correspondingly 5.08% and 6.78% higher than the RF and KNN reported in [10]. The Transformations that occur obtained an AC that is 8.74% higher than the work shown in [12]. Besides that, the ET- AdaBoost achieved an AC that is 4.34% higher than the work in [13], in comparison. Furthermore, all of the models now perform better when precision and recall are factored in because to the adoption of SMOTE-AdaBoost methodology. For instance, the DT model achieved a recall of 75.75% without SMOTE-AdaBoost, compared to 99.00% when the SMOTE-AdaBoost algorithms were used. Without using SMOTE-AdaBoost techniques, the DT managed to attain a precision of 79.83%. But when SMOTE- AdaBoost was used, the DT model was able to achieve a precision of 98.79%. Consequently, the MCC increased from 0.78 to 0.98. All of the models taken into account in this study exhibit this tendency.

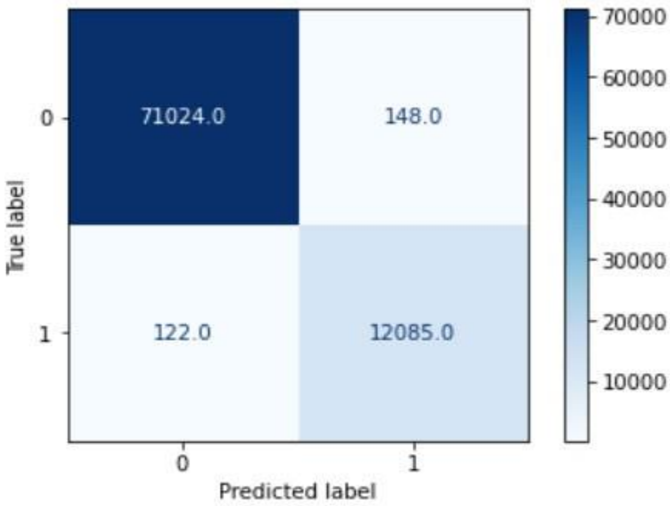


Figure 2. DT-AdaBoost confusion matrix

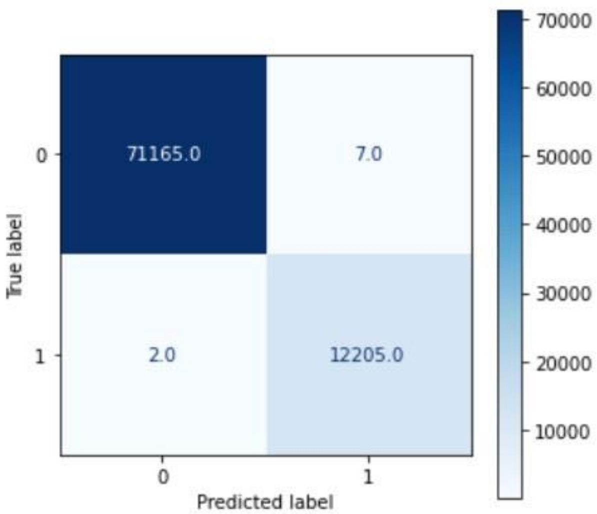


Figure 1. ET-AdaBoost confusion matrix.

Experiments Validation

The studies in this section make use of a publicly accessible synthetic credit card fraud dataset [32]. 24357143 legitimate credit card transactions and 29757 fraudulent ones are included in this dataset. The dataset also includes the following features. Use Chip, Merchant Name, Merchant City, Merchant State, MCC, Zip, Errors, User, Card, Year, Month, Time, Day, Amount Is Fraud, with

Is Fraud standing in for the class. Table 4 includes a list of these qualities. The DT, RF, ET, XGB, and LR models were used in the experimentation process. Each of these models received an adaptive boost (using AdaBoost).

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 %

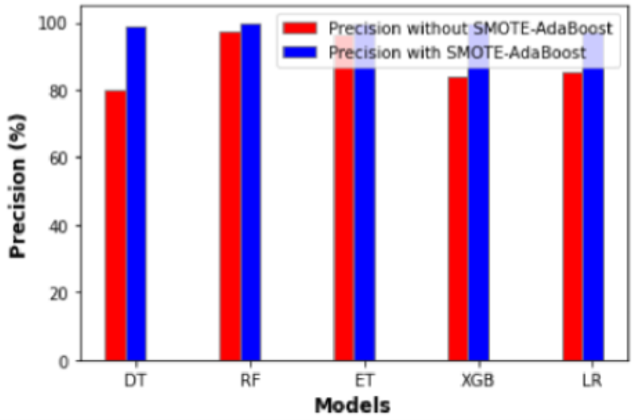


Figure 3. Precision Comparison

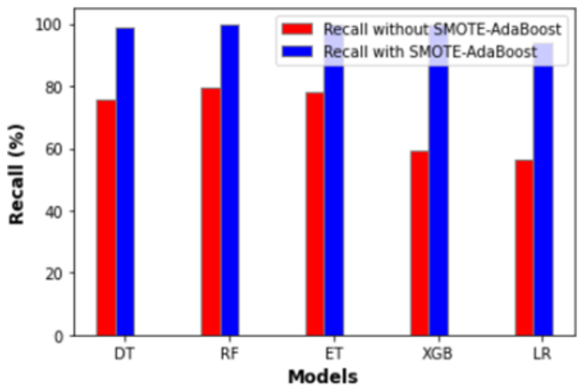


Figure 4. Recall comparison

7. CONCLUSION

Using the European credit card fraud dataset created in September 2013, this paper constructed different ML algorithms for credit card fraud detection. The DT, RF, ET, XGB, LR, and SVM were among the proposed ML techniques in this paper. In order to improve classification accuracy and address the problem of class imbalance that is present in the European credit card fraud dataset, each proposed algorithm was also combined with the AdaBoost method. Additionally, a comparison analysis was done between the techniques in this work and the frameworks already in place for detecting credit card fraud. For instance, the accuracy of the DT-AdaBoost, RF-AdaBoost, ET-AdaBoost, and XGB-AdaBoost, respectively, was 99.67%, 99.95%, 99.98%, and 99.98%. The XGB-AdaBoost and ET-AdaBoost both achieved an MCC of 0.99 in terms of classification quality. These results showed that the AdaBoost algorithm has a beneficial effect on the suggested ML techniques. Additionally, the framework suggested in this study was evaluated using a dataset on synthetic credit card fraud that was extremely skewed, and the outcomes were excellent.

8. 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] TheNilsonReport.Accessed:Sep.27,2021.[Online].Available:https://nilsonreport.com/content_promo.php?id_promo=16
[5] D. El reedy 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. Kegel Meyer, ''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.

