Utilizing GANs for Credit Card Fraud Detection: A Comparison of Supervised Learning Algorithms

Bandar Alshawi
Department of Information Systems, College of Computer and Information Systems, Umm Al-Qura University, Saudi Arabia
bmhshawi@uqu.edu.sa (corresponding author)

Received: 24 September 2023 | Revised: 15 October 2023 | Accepted: 28 October 2023

 DOI: https://doi.org/10.48084/etasr.6434

ABSTRACT

The evolution and improvements in electronic commerce and communications around the world have stimulated credit card use. With the support of smartphone wallets, electronic payments have become the most popular payment method for personal and business use; however, the past few years have also seen a major increase in fraudulent transactions. Corporations and individuals experience very negative impacts from such fraud. Therefore, fraud detection systems have received a lot of attention recently from major financial institutions. This paper proposes a fraud detection approach that deals with small and imbalanced datasets using Generative Adversarial Networks (GANs) for sample generation. Six machine-learning algorithms were applied to real-world data. The accuracy of all six algorithms was above 85% and the precision was above 95%. Five of the six algorithms had a recall score greater than 90%. Furthermore, the Receiver Operating Characteristics (ROC), which measure performance at different thresholds, demonstrated scores greater than 0.90, except Naïve Bayes, which scored 0.81. The proposed approach outperformed the same algorithms in other studies.

Keywords-fraud detection; credit card fraud; generative adversarial network; supervised learning; naive bayes; decision tree; imbalance datasets

I. INTRODUCTION

During the recent years, there has been a substantial increase in financial frauds, primarily due to the technological changes and cybersecurity breaches. In 2022, such frauds impacted personal assets by 10% and organizations by 39% [1]. Financial institutions had $28.58 billion in losses in credit card fraud worldwide in 2020, and this figure is expected to reach $49.43 billion by 2030 [2]. The employment of Artificial Intelligence (AI) techniques to monitor and detect fraudulent transactions can prevent an enormous amount of losses in the banking sector and increase the degree of trust customers have in financial institutions. Several studies have used Machine Learning (ML) to detect fraudulent transactions [3]. According to [4], 26% of credit card transactions were either attempted or actual fraud transactions. However, detecting fraudulent transactions is challenging for ML [5]. In [6], a comparison between ensemble learning and supervised algorithms was made in network traffic. In [7], a credit card fraud detection technique was presented, based on the Decision Tree (DT) algorithm.

The current study addressed one of the main issues in ML, namely the lack of sufficient data and unbalanced datasets. The use of Generative Adversarial Networks (GANs) was used to generate synthetic data from a real-world dataset. Following the generation of the dataset, different ML techniques were applied: Logistic Regression (LR), DT, Random Forest (RF), Naïve Bayes (NB), extreme gradient boosting (XGBoost), and Adaptive Boosting (Ada-Boost). This study aimed to investigate the fineness of employing various AI techniques to detect credit card anomalies and to examine the ability of GANs to balance unbalanced datasets. The main contributions of this study are:

- Proposing a solution for an imbalanced dataset in tabular data.
- Implementing GANs to generate synthetic data.
- Implementing 6 ML algorithms and evaluate their results.

II. RELATED WORKS

A method was proposed in [8] for the detection and assessement of the risk of credit card fraud. This method was based on an algorithm that can measure the relationships between variables and any related information to increase accuracy and decrease dimensionality. In [9], supervised and unsupervised credit card fraud detection methods were compared, showing huge performance restrictions when using supervised learning, with unsupervised learning performing better. In [10], a Financial Fraud Detection (FFD) model was introduced, based on ontology alert. The FFD model consisted of 40 classes and subclasses and was capable of identifying transaction anomalies by initiating different alerts based on their severity levels. In [11], an LR algorithm was applied to detect credit card anomalies that occur during the purchase of movie tickets. In [12], supervised and unsupervised techniques were used to find suspicious behaviors in bookkeeping, using a real ledger dataset and applying data vectorization to solve sub-ledger account size irregularities and to improve performance. Support Vector Machines (SVMs) and DT techniques were used in [13] to detect credit card frauds, using a highly imbalanced real dataset, and the results showed that DT clearly outperformed SVMs. In [14], several ML techniques were evaluated to detect fraudulent transactions, including outlier detection and ensemble methods. The influence of feature selection on performance was measured using feature engineering and analysis. In [15], recent advances in ML and Deep Reinforming Learning (DRL) were investigated to create a credit card fraud detection system, using a highly imbalanced dataset and resampling methods. In [16], nonlinear models were investigated, proposing credit card fraud detection methods that can be interpreted and those that are not tied in a particular way. These methods can be used simultaneously along any ML technique and provide the required tracing info linking inputs and outputs, thus avoiding reaching the black-box model. In [17], a Multiple Classifier System (MCS) was used on credit card fraud datasets, achieving accurate fraud identification by using the sequential decision combination technique. This study used two learning algorithms: NB and C4.5, showing great classification results on majority and minority class samples for C4.5 and NB, respectively. In [18], the efficiency of using multiple neural networks in conjunction with hybrid data resampling was demonstrated to detect credit card abnormalities, using an AdaBoost technique with long short-term memory as its base. The proposed approach outperformed other algorithms, such as SVM and DT, on a real credit card dataset. In [19], a credit card fraud detection system was tested, combining CatBoot and Deep Neural Networks (DNNs) as a base learner, on an IEEE-CIS dataset consisting of 590,540 instances. CatBoost was used to test whether users' overlapping prediction rates could improve. In [20], an experimental study was conducted on an imbalanced dataset, using classification algorithms such as C5.0, SVM, ANN, and NB. In this study, the problem of the imbalanced dataset was solved by balancing the classes using Random Oversampling (RO), which replicates the observations as long as the balance between classes is not reached. In [21], an intelligent mechanism for credit card fraud detection was used, which employed an optimized light gradient boosting machine (OLightGBM), which was tuned by integrating a Bayesian-based hyperparameter optimization. The experiment was carried out on two real-world datasets, reaching 98.40% accuracy. In [22], a DNN was used as a representation feature model, and fraud transactions were identified by mapping the actual features of transactions to deep representations. This study also presented an improved loss function that recognizes angles among features along with distances and, consequently, can fully supervise deep feature learning. This method was evaluated in two large credit card datasets [22]. Table I presents a summary of recent works, indicating inconsistencies and areas for improvement regarding the listed algorithms.

TABLE I. COMPARISON OF RELATED WORKS

Classifier	Accuracy (%)	Recall (%)	Year of publication	Reference
LR	94.51	97.36	2021	[15]
	36	71	2017	[23]
	99	69	2022	[3]
DT	88	91	2022	[24]
	99	74	2022	[3]
NB	97	82	2017	[23]
	99	14	2022	[3]
RF	99.99	99.98	2021	[15]
	92	86	2022	[24]
XGBoost	99.97	99.94	2021	[15]
	99	95	2022	[3]
	92	91	2022	[24]
Ada-boost	99	75	2021	[25]
	97	96	2021	[15]

III. METHOD

A. *Generative Adversarial Network (GAN)*

A GAN is an unsupervised deep learning model that was invented in 2014 and has received great attention since [26-27]. The word "generative" indicates that it is a model that can learn any distribution of data, mimicking the original distribution. It is called "adversarial" since 2 neural networks are trained in a combative setting. The final word, "network", indicates the use of a deep neural network as an AI training algorithm. A very common and major problem during modeling is having imbalanced datasets. GANs can be used to generate synthetic data that can provide a solution to this complexity, as in [28]. A GAN consists of combined neural networks that conflict with each other. The first model is called the generator, and the second is called the discriminator. The generator network collects and generates samples, while after training, the probability of the discriminator making mistakes increases. Figure 1 presents the GAN architecture. A GAN equation can be:

$$(G, D) = \mathbb{E}_{x\sim\mu_{ref}}[\ln D(x)] + \mathbb{E}_{z\sim\mu_z}\left[ln\left(1 - D\big(G(Z)\big)\right)\right] \quad (1)$$

where G is the generative model to be trained on the training data x, and D is the discriminator that discriminates among the various classes of data. The discriminator uses a binary classification with sigmoid activation to determine whether the data are generated or received from an actual sample, given an output that ranges between 0 and 1. From (1), z adds a slight noise sample as an input to the generator, $\sim\mu_{ref}$ represents the distribution from the actual data, and $\sim\mu_z$ is the distribution from the generator.

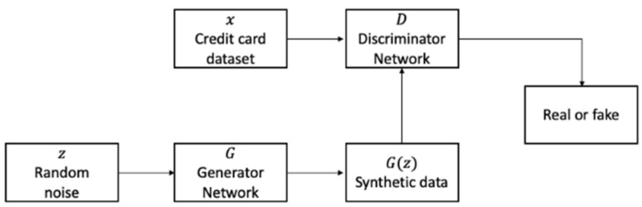

Fig. 1. Generative adversarial network.

B. GANs for Tabular Data

Although GANs are effective in generating images and textual data, the use of GANs on tabular data raises numerous challenges:

- Data types can be numerical or categorical.
- GANs distribute image data over space. Tabular data can be non-Gaussian, and then the network will not be able to propagate gradient details.
- GAN generators do not recognize imbalanced categorical columns as a result of using samples generated from a standard multivariate distribution.

This study used the CTGAN model [29], which is based on the GAN method and solves the non-Gaussian challenge by applying mode-specific normalization. CTGAN also uses all existing features of the dataset.

C. Logistic Regression (LR)

LR is often used to solve classification and predictive problems. The most common LR model is the binary one, which predicts binary classes by calculating the link between independent and dependent variables using statistical estimation. Considering the result is a probability, the dependent variable is restrained between 1 and 0. LR is widely used in classification problems and is well known for its performance, efficiency, and simplicity. It has also been used in similar experiments [3, 12, 30-31]. The LR formula is:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \cdots + \beta_k X_{ki} + \varepsilon_i \qquad (2)$$

where Y_i is the dependent variable, X_{1i} and X_{ki} are the independent variables, β_0 is the intercept, β_1 and β_k are the population coefficients, and ε_i is a random error.

D. Naïve Bayes (NB)

NB is an ML classifier based on Bayes' theorem. NB is a probabilistic algorithm that is widely adopted in different classification problems. NB assumes that all the features are independent from one another. The posterior probability $P(x|c)$ is given by:

$$\boldsymbol{P(x|c)} = \frac{\boldsymbol{P(x|c)P(c)}}{\boldsymbol{P(x)}} \qquad (3)$$

and indicates how frequently x occurs when c has already occurred.

E. Random Forest (RF)

RF is an ML classifier technique that is used to deal with classification and regression problems. Following forest initiation, RF is trained by the bagging method, which is an ensemble algorithm used to fit numerous models on multiple subsets of a training dataset and then merges all predictions. The equation for RF is given by:

$$Gini = 1 - \sum_{i=1}^{c} (p_i)^2 \qquad (4)$$

where p represents the corresponding frequency of the observed class in the dataset, and c is the number of classes.

F. Decision Tree (DT)

DT can be used for both regression and classification problems. A DT consists of nodes and the first node is the root located on top of the tree. At each level of the tree, there is a condition that leads to a binary split of the existing node into subnodes. A decision will be made when a subnode, which is also called a leaf node, does not split anymore. Leaf nodes resemble a class label, while attributes are expressed on the nonleaf node of the tree. DT offers numerous advantages, including straightforward implementation, numerical and categorical data support, and providing instinctive knowledge expression. DT decisions are restricted to a binary output, affecting the performance the tree can handle.

G. Extreme Gradient Boosting (XGBoost)

XGBoost is an open-source implementation gradient boost algorithm. The distinction of XGBoost is its ability to control overfitting behavior while gaining superior performance. XGBoost is recognized for its speed, dealing with one of the well-known gradient-boosted tree inefficiencies, achieved by taking care of probable loss for all the splits to produce a new branch. The objective function in XGBoost consists of the loss function and a regularized term:

$$\mathcal{L}(\emptyset) = \sum_i l(\hat{\mathcal{Y}}_i, \mathcal{Y}_i) + \Omega(\emptyset) \qquad (5)$$

where $\emptyset$ represents the learned parameter set, l is used to calculate the distinction between $\hat{\mathcal{Y}}_i$ and $\mathcal{Y}_i$, and Ω represents the regularization term.

H. Adaptive Boosting (Ada-boost)

Ada-boost is one of the first boosting methods. It operates by creating the model and having all data points equal with their weights, spreading greater weight to data points that are misclassified. This step gives greater importance to data points with higher weights. The Ada-boost equation is given by:

$$F_T(x) = \sum_{t=1}^{T} f_t(x) \qquad (6)$$

I. Evaluation Metrics

The performance of the classification methods was evaluated and tested using a confusion matrix. The first metric used was accuracy, where True Positive (TP) indicates all legitimate transactions that are predicted accurately, True Negative (TN) is the total fraudulent transactions predicted accurately as fraudulent, False Positive (FP) is the total legitimate transactions predicted mistakenly as fraudulent, and False Negative (FN) is the total fraudulent transactions predicted incorrectly as legitimate. The accuracy metric was used to calculate the ratio of legitimate transactions to the total number of transactions [31]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (7)$$

The second metric was precision, which measures the ratio of accurately classified positives from all the positive predictions:

$$Precision = \frac{TP}{(TP+FP)} \qquad (8)$$

Equations (9) and (10) display the Recall and F-measure metric calculations.

$$Recall = \frac{TP}{(TP+FN)} \quad (9)$$

$$F - measure = \frac{2*Precision*Recall}{Precision+Recall} \quad (10)$$

Matthews' Correlation Coefficient (MCC) was also used to measure the quality of the classifiers:

$$MCC = \frac{TP*TN - FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (11)$$

IV. RESULTS AND DISCUSSION

A. *Original Data*

The dataset contains credit card transactions made by European cardholders during two days of October 2018 [25]. This dataset is imbalanced, with 284,809 legitimate and 492 fraudulent transactions. The data do not provide the original attributes for privacy concerns and were transformed using Principal Component Analysis (PCA). The result of that transformation is indicated by its attributes starting from V1 to V28. The data also include the amount of the transaction, the time separating each transaction, and a class label associating 0 to a fraudulent transaction and 1 to a legitimate one.

B. *Synthetic Data*

Several studies tried to fix the complexity of imbalanced datasets using Minority Oversampling (MOTE) [12, 22, 31-34]. The proposed approach used synthetic data generated using GAN. After generating the data, 6 ML algorithms were trained and applied.

C. *Performance Evaluation and Confusion Metrics*

Table II shows the metric evaluation of the six ML algorithms. Another way to evaluate the results is through a confusion matrix, which can be used for both binary and multiclass classifications. In the current state, which is binary classification, the confusion matrix generates a 2×2 table, consisting of TP, FP, TN, and FN.

TABLE II. PERFORMANCE EVALUATION

Algorithm	Accuracy	Precision	Recall
LR	96%	95%	91%
DT	96%	96%	92%
NB	85%	98%	64%
RF	97%	98%	92%
XGBoost	98%	98%	96%
Ada-boost	97%	97%	93%

Five models had more than 95% accuracy. The NB algorithm had the lowest accuracy (85%), a precision of 98%, and only 64% recall. Figure 2 indicates that NB classified 567 legitimate transactions as fraudulent. The LR model had 96% accuracy. Of the transactions detected by the LR model, 95% were truly relevant according to the precision score, and Figure 3 shows its confusion matrix. Figure 4 shows the confusion matrix of DT, which also had a high accuracy of 96%. Its precision score indicates that it relented only 4% of relevant instances, while recall shows that it predicted 8% of transactions as false negatives.

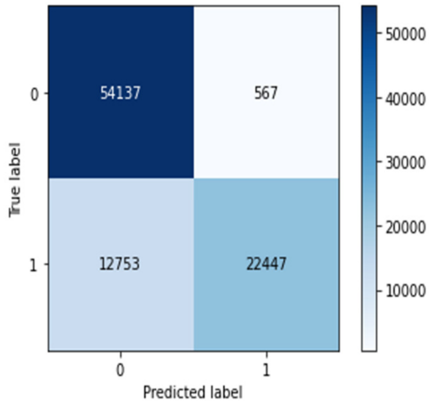

Fig. 2. Confusion matrix of NB.

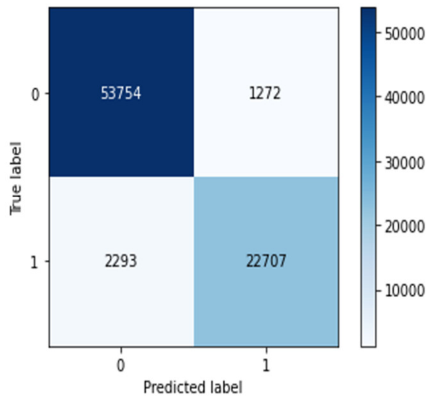

Fig. 3. Confusion matrix of LR.

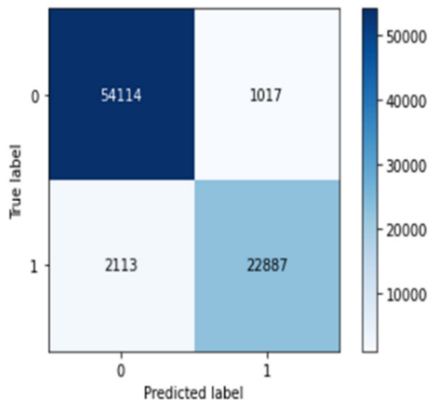

Fig. 4. Confusion matrix of DT.

Figure 5 shows that RF classified 421 fraudulent transactions as legitimate and miscategorized 1944 legitimate transactions while having 97% accuracy. Based on recall, RF classified 8% of legitimate transactions as fraudulent. The XGBoost algorithm exhibited the highest percentage values of the six algorithms, predicting 98% of anomalies accurately. XGBoost scored 98% precision, revealing that the model classified only 2% of fraud transactions as legitimate. From the XGBoost confusion matrix shown in Figure 6, 1,015 legitimate transactions were classified as fraud, which is considered trivial compared to the NB model. When comparing the existing results with Table II, there are major improvements, especially in NB. The explanation of the variations in the performance of NB regarding the other algorithms is its limitation in dealing with complicated tasks and its ineffectiveness on numerical data. Ada-Boost was second in scores. The accuracy score showed that the model misclassified 3% of all transactions. Figure 7 shows the confusion matrix of Ada-Boost.

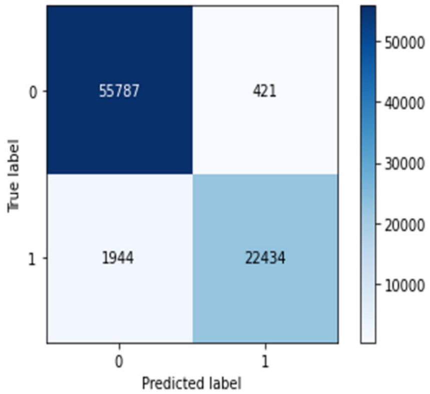

Fig. 5. Confusion matrix of RF.

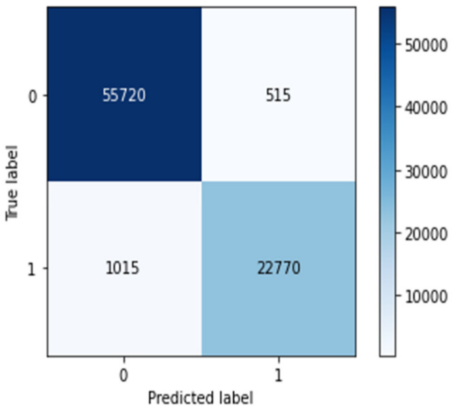

Fig. 6. Confusion matrix of XGBoost.

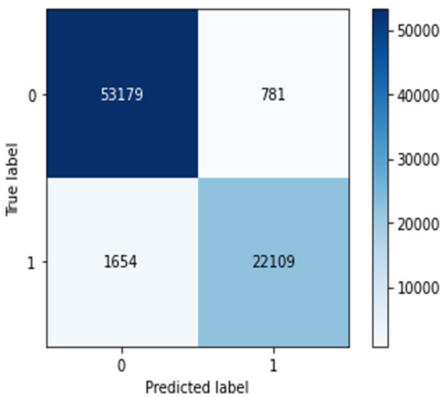

Fig. 7. Confusion matrix of Ada-Boost.

Table III shows further evaluation metrics applied to evaluate the models.

TABLE III. PERFORMANCE EVALUATION: ROC, MCC, AND F1-SCORE

Algorithm	ROC	MCC	F1-score
LR	0.94	0.90	93%
DT	0.95	0.91	94%
NB	0.81	0.70	77%
RF	0.96	0.93	95%
XGBoost	0.98	0.97	97%
Ada-Boost	0.96	0.93	95%

Table III shows that the MCC score of the LR model was 0.90, implying that it was able to predict 90% of legitimate and fraudulent transactions. The NB MCC score shows that 81% of legitimate and fraudulent transactions were predicted. XGBoost had the highest MCC score of 97%. One more method to verify the accuracy of the models is the F1-score, which combines precision and recall. This metric was expected to obtain great results using XGBoost and outperform NB, as XGBoost is designed to deal with large datasets. The 5 models received scores of over 90%. The comparative results shown in Figure 9 illustrate a comparison of the existing results with numerous recent studies. Several metric methods should be used to assess the performance of each classifier. Relying on accuracy alone or two metrics will not reflect an accurate result. For example, in the case of NB in [3], immense accuracy and moderate precision were achieved, while the recall and F1-score were extremely low suggesting class-imbalance concerns.

The Receiver Operating Characteristic (ROC) curve is an effective technique to measure the performance of a model at various thresholds. Figure 8 shows the ROC of the six models, with all of them achieving almost perfect results except for NB.

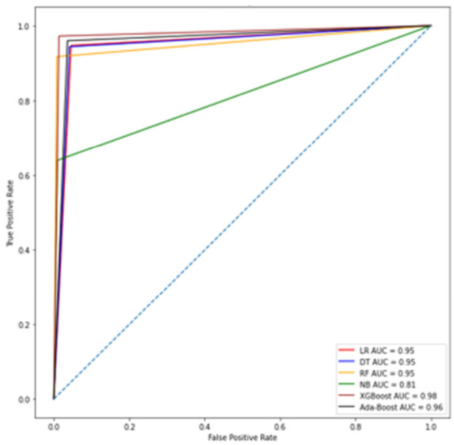

Fig. 8. ROC of six models.

V. CONCLUSION

Credit card fraud detection systems are crucial for financial institutions. This study compared 6 ML algorithms to detect credit card frauds in real-world data, using CTGAN to overcome the deficiencies of small and imbalanced datasets. Based on the results, all the tested algorithms achieved an accuracy of over 85% and a precision score of over 95%. Five algorithms achieved a recall greater than 90%. Lastly, all algorithms achieved a ROC curve greater than 0.90, except Naïve Bayes, which scored 0.81.

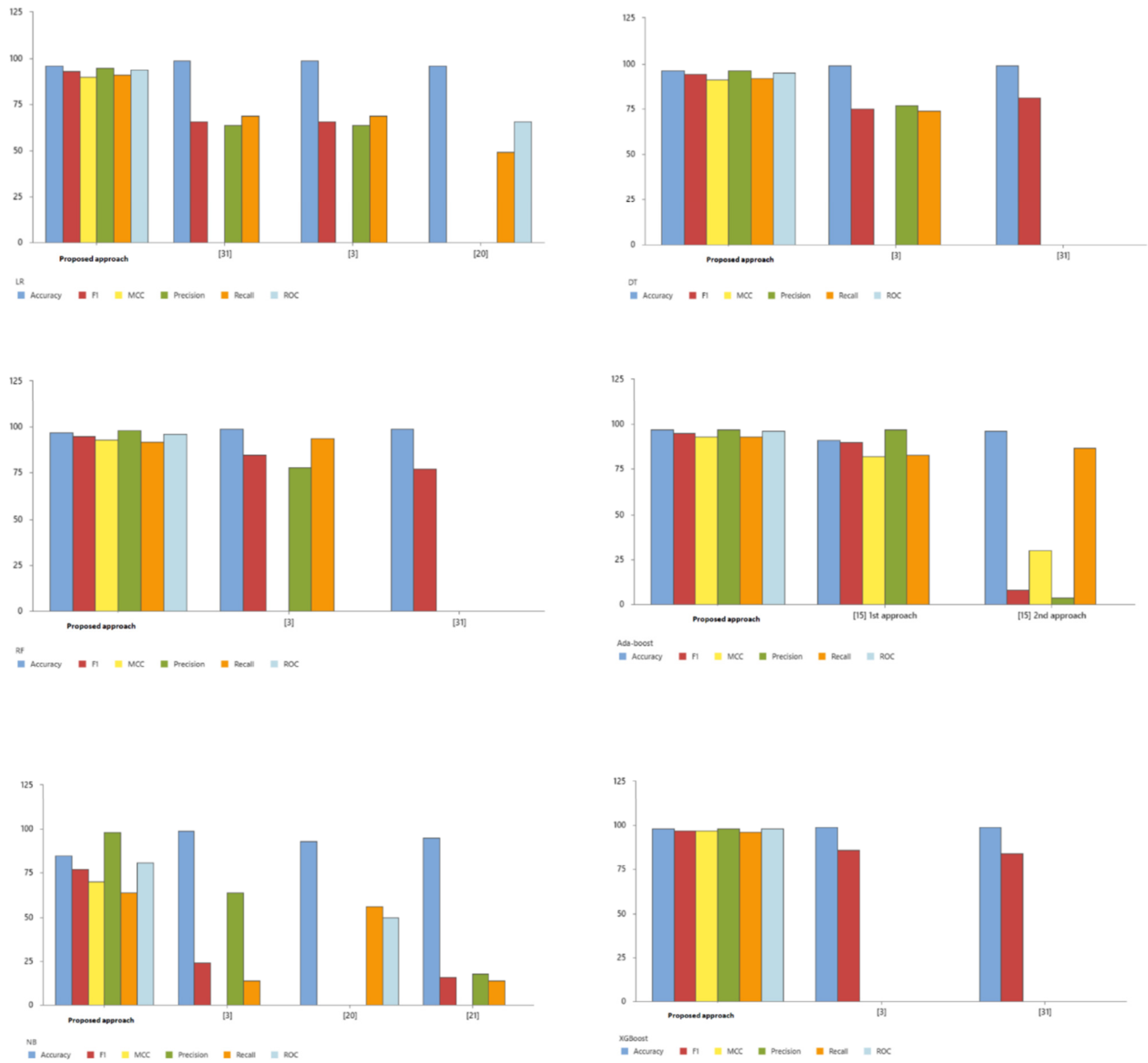

Fig. 9. Comparative analysis.

Furthermore, the results of the 6 ML models were compared with previous studies that used imbalanced data, showing a significant improvement in all metrics.

Future work will involve the testing of additional ML algorithms and workarounds, the improvement of NB, the exploration of more GANs and their effectiveness in tabular data, and the use of other datasets.

REFERENCES

[1] "Global Cybersecurity Outlook 2023," World Economic Forum, Jan. 2023. [Online]. Available: https://www3.weforum.org/docs/WEF_Global_Security_Outlook_Report_2023.pdf.

[2] "Card Fraud Worldwide," *Nilson Report 1209*. https://nilsonreport.com/newsletters/1209/.

[3] N. S. Alfaiz and S. M. Fati, "Enhanced Credit Card Fraud Detection Model Using Machine Learning," *Electronics*, vol. 11, no. 4, Jan. 2022, Art. no. 662, https://doi.org/10.3390/electronics11040662.

[4] "Payments Fraud and Control Report," Association for Financial Professionals, Apr. 2022. [Online]. Available: https://www.jpmorgan.com/content/dam/jpm/commercial-banking/insights/cybersecurity/highlights-afp-2022-payments-fraud-and-control-report.pdf.

[5] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3797, 2018, https://doi.org/10.1109/TNNLS.2017.2736643.

[6] M. Machoke, J. Mbelwa, J. Agbinya, and A. E. Sam, "Performance Comparison of Ensemble Learning and Supervised Algorithms in Classifying Multi-label Network Traffic Flow," *Engineering, Technology & Applied Science Research*, vol. 12, no. 3, pp. 8667–8674, Jun. 2022, https://doi.org/10.48084/etasr.4852.

[7] A. S. Alraddadi, "A Survey and a Credit Card Fraud Detection and Prevention Model using the Decision Tree Algorithm," *Engineering, Technology & Applied Science Research*, vol. 13, no. 4, pp. 11505–11510, Aug. 2023, https://doi.org/10.48084/etasr.6128.

[8] J. Chaquet-Ulldemolins, F. J. Gimeno-Blanes, S. Moral-Rubio, S. Muñoz-Romero, and J. L. Rojo-Álvarez, "On the Black-Box Challenge for Fraud Detection Using Machine Learning (I): Linear Models and Informative Feature Selection," *Applied Sciences*, vol. 12, no. 7, Jan. 2022, Art. no. 3328, https://doi.org/10.3390/app12073328.

[9] X. Niu, L. Wang, and X. Yang, "A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised." arXiv, Apr. 23, 2019, https://doi.org/10.48550/arXiv.1904.10604.

[10] M. Ahmed, K. Ansar, C. B. Muckley, A. Khan, A. Anjum, and M. Talha, "A semantic rule based digital fraud detection," *PeerJ Computer Science*, vol. 7, Art. no. e649, Aug. 2021, https://doi.org/10.7717/peerj-cs.649.

[11] A. Alshutayri, "Fraud Prediction in Movie Theater Credit Card Transactions using Machine Learning," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10941–10945, Jun. 2023, https://doi.org/10.48084/etasr.5950.

[12] A. Bakumenko and A. Elragal, "Detecting Anomalies in Financial Data Using Machine Learning Algorithms," *Systems*, vol. 10, no. 5, Oct. 2022, Art. no. 130, https://doi.org/10.3390/systems10050130.

[13] Y. G. Şahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2011*, Mar. 2011, vol. 1, pp. 442–447.

[14] B. Stojanović *et al.*, "Follow the Trail: Machine Learning for Fraud Detection in Fintech Applications," *Sensors*, vol. 21, no. 5, Jan. 2021, Art. no. 1594, https://doi.org/10.3390/s21051594.

[15] T. K. Dang, T. C. Tran, L. M. Tuan, and M. V. Tiep, "Machine Learning Based on Resampling Approaches and Deep Reinforcement Learning for Credit Card Fraud Detection Systems," *Applied Sciences*, vol. 11, no. 21, Jan. 2021, Art. no. 10004, https://doi.org/10.3390/app112110004.

[16] J. Chaquet-Ulldemolins, F. J. Gimeno-Blanes, S. Moral-Rubio, S. Muñoz-Romero, and J. L. Rojo-Álvarez, "On the Black-Box Challenge for Fraud Detection Using Machine Learning (II): Nonlinear Analysis through Interpretable Autoencoders," *Applied Sciences*, vol. 12, no. 8, Jan. 2022, Art. no. 3856, https://doi.org/10.3390/app12083856.

[17] S. N. Kalid, K.-H. Ng, G.-K. Tong, and K.-C. Khor, "A Multiple Classifiers System for Anomaly Detection in Credit Card Data With Unbalanced and Overlapped Classes," *IEEE Access*, vol. 8, pp. 28210–28221, 2020, https://doi.org/10.1109/ACCESS.2020.2972009.

[18] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, "A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection," *IEEE Access*, vol. 10, pp. 16400–16407, 2022, https://doi.org/10.1109/ACCESS.2022.3148298.

[19] N. Nguyen *et al.*, "A Proposed Model for Card Fraud Detection Based on CatBoost and Deep Neural Network," *IEEE Access*, vol. 10, pp. 96852–96861, 2022, https://doi.org/10.1109/ACCESS.2022.3205416.

[20] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. S. Hacid, and H. Zeineddine, "An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection," *IEEE Access*, vol. 7, pp. 93010–93022, 2019, https://doi.org/10.1109/ACCESS.2019.2927266.

[21] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," *IEEE Access*, vol. 8, pp. 25579–25587, 2020, https://doi.org/10.1109/ACCESS.2020.2971354.

[22] Z. Li, G. Liu, and C. Jiang, "Deep Representation Learning With Full Center Loss for Credit Card Fraud Detection," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 569–579, Apr. 2020, https://doi.org/10.1109/TCSS.2020.2970805.

[23] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," *IEEE Access*, vol. 10, pp. 39700–39715, 2022, https://doi.org/10.1109/ACCESS.2022.3166891.

[24] R. Soleymanzadeh, M. Aljasim, M. W. Qadeer, and R. Kashef, "Cyberattack and Fraud Detection Using Ensemble Stacking," *AI*, vol. 3, no. 1, pp. 22–36, Mar. 2022, https://doi.org/10.3390/ai3010002.

[25] "Credit Card Fraud Detection." https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

[26] E. Brophy, Z. Wang, Q. She, and T. Ward, "Generative Adversarial Networks in Time Series: A Systematic Literature Review," *ACM Computing Surveys*, vol. 55, no. 10, Oct. 2023, Art. no. 199, https://doi.org/10.1145/3559540.

[27] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, Montreal, QC, Canada, 2014, vol. 27.

[28] I. Ashrapov, "Tabular GANs for uneven distribution." arXiv, Oct. 01, 2020, https://doi.org/10.48550/arXiv.2010.00638.

[29] C. Charitou, S. Dragicevic, and A. d'Avila Garcez, "Synthetic Data Generation for Fraud Detection using GANs." arXiv, Sep. 26, 2021, https://doi.org/10.48550/arXiv.2109.12546.

[30] S. K. Shirgave, C. J. Awati, R. More, and S. S. Patil, "A Review on Credit Card Fraud Detection Using Machine Learning," *International Journal of Scientific & Technology Research*, vol. 8, no. 10, Oct. 2019.

[31] H. M and S. M.N, "A Review on Evaluation Metrics for Data Classification Evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, Mar. 2015, https://doi.org/10.5121/ijdkp.2015.5201.

[32] E. Ileberi, Y. Sun, and Z. Wang, "Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost," *IEEE Access*, vol. 9, pp. 165286–165294, 2021, https://doi.org/10.1109/ACCESS.2021.3134330.

[33] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *2017 International Conference on Computing Networking and Informatics (ICCNI)*, Lagos, Nigeria, Jul. 2017, pp. 1–9, https://doi.org/10.1109/ICCNI.2017.8123782.

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