IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 February 2024, accepted 19 March 2024, date of publication 22 March 2024, date of current version 23 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3380823

RESEARCH ARTICLE

Identifying Fraudulent Credit Card Transactions Using Ensemble Learning

JABER JEMAI[1], ANIS ZARRAD[2], AND ALI DAUD[3]
[1]CIS Division, Higher Colleges of Technology, Abu Dhabi, United Arab Emirates
[2]University of Birmingham Dubai, Dubai, United Arab Emirates
[3]Faculty of Resilience, Rabdan Academy, Abu Dhabi, United Arab Emirates

Corresponding author: Ali Daud (alimsdb@gmail.com)

ABSTRACT Recognizing fraudulent credit card transactions is one of the main issues facing banking institutions. Since each transaction that completes the authentication procedure must be authorized by financial institutions, a hacker might pose as the actual cardholder and execute a fraudulent transaction. In this paper, we investigated the capacity of ensemble learning methods to identify credit card frauds on two distinct data sets: the Sparkov synthetic dataset and the real dataset of consumers in the European Union. XGBoost models, random forests, and naive Bayes classifiers are applied and assessed on both datasets. Accuracy, precision, recall, and F1 score are used to measure performance. According to the results, most ensemble classifiers perform exceptionally well on the real-world dataset, but significantly poorly on the simulated dataset. This study showed that, unlike in simulated environments, credit card transaction management scripts are quickly learned in deterministic settings. It is discussed that a larger danger of card information leakage results from strict determinism and lack of randomness.

INDEX TERMS Fintech, credit card fraud detection, ensemble learning, machine learning, simulated data set, real-world data set.

I. INTRODUCTION

Credit Card Fraud (CCF) detection problem consists of spotting credit card transaction anomalies. In 2020, Visa and MasterCard had more than 2.183 million cardholders [1]. Approximately 1.4 million identity theft reports were reported in 2020, including 393.207 cases of CCF [2]. The loss due to fraudulent usage of credit cards reached $28.6 billion in 2020, where it was $23.97 billion. In 2017 with an increase of 19.3 %, it is expected to reach $408 billion in the next decade according to Nilson Report of 2022 [3]. This amount shows the big loss and how it is continuously increasing despite the efforts of banks to limit the effects of credit card misuse. It inevitable to detect CCF in this era of digital payments.

To address credit card misuse, several approaches have been developed belonging mainly to two classes [2]: statistical approaches and machine learning-based approaches. From a statistical point of view, detecting a fraudulent transaction is equivalent to detecting an outlier from a data set. Consequently, a variety of statistical techniques have been developed and used namely: box and whisker plots, normal distribution-based, cluster-based, etc. The second class of credit card fraud detection techniques is composed of machine learning classifiers. A classifier is designed, developed, fitted, and tuned on a training data set to separate authentic and fraudulent transactions. Classifiers aim to learn to detect correctly the type of transaction beforehand.

Machine learning techniques applied to CCF problems include decision trees, support vector machines, neural networks, regression methods, etc. [1], [4], [5]. The performance of such methods varies depending on the used data set. To improve the quality of the classification step, ensemble learning was proposed. The basic idea of ensemble learning is to build an enhanced classifier from a set of naive/basic classifiers. The objective is then to create a strong meta-learner (ensemble model) from a list of basic learners (naive classifiers). Many ensemble models have been proposed in the literature depending on the technique used in combining basic classifiers. For instance, bagging methods create many

The associate editor coordinating the review of this manuscript and approving it for publication was Dominik Strzalka.

samples with replacements from the training set and use them to fit in parallel basic classifiers. An aggregation technique is also used to build the final strong classifier like voting. The second type of ensemble learning is comprised of boosting techniques. Boosting consists of creating sequentially basic classifiers, where the prediction error is propagated from one model to the subsequent. This process will help in boosting the performance of the last classifiers. Many variants of boosting models have been developed including Adaboost, Gradient boosting, and XGBoost among others [6]. Stacking the third type of ensemble method (known also as stacked generalization) combines classifiers to find an improved model [7]. In this paper, a set of ensemble-based classifiers is developed for the detection and prevention of abnormal transactions on credit cards. Three techniques are proposed, a naive Bayes classifier and one from each class of ensemble methods: bagging and boosting. Based, on a data set of credit card transactions of European Union consumers available in the Kaggle repository [8] and the synthetic Sparkov dataset [9]. An attempt to find the best ensemble-based model for solving the credit card detection problem is made in this work. The main objective of this paper is to compare the performance of ensemble models in learning real and synthetic datasets and not equate ensemble methods in general to other machine learning techniques.

In this paper, the motivations are presented in section II, and the research methodology in section III. In section IV, we will review the credit card fraud detection and prevention literature by formally defining the CCF problem, and analyzing the performance of the developed strategies on the used data sets. Section V will report the details of the implementation of the machine learning approach; by describing the used data sets in subsection V-A. The details of the classification methods will be given in subsection V-B and the obtained results in 5.3. The findings of the research project will be reported in section VI. The conclusions and perspectives will be presented in section VII.

II. MOTIVATIONS

The usage of online payment technologies is continuously increasing via credit cards, digital wallets, cryptocurrencies, etc. Such a fact will increase consequently the number of fraudulent transactions and misappropriated funds. A fraudulent transaction is an instance of a failure in the security system of the service provider. The aims/contributions of the research paper are to:

1) *Study the learnability level of ensemble methods from real-life and synthetic data sets.*
2) *Identify the source of the failure of the credit card fraud prevention systems.*
3) *Assess the vulnerability of the credit card transactions requests processing scripts.*
4) *Proposal of remedial actions to improve the process of approving/declining a transaction on credit cards.*

III. RESEARCH METHODOLOGY

The main objective of this paper is to study the performance of ensemble methods on real and simulated CCF data sets. That's to analyze the outputs of business scripts implemented to process credit card transactions and how their determinism nature constitutes a vulnerability and a source of security systems failure. As depicted in Figure 1, two standard data sets (European consumers *eu* and Sparkov simulated *sp*) are used. The *eu* dataset [8] is already processed and has 28 features selected using the Principal Component Analysis (PCA) technique. The Sparkov data set is preprocessed by removing irrelevant features and keeping 22 variables only, encoding categorical variables, and standardizing all numeric features (refer to subsection V-A).

Since both data sets are heavily unbalanced toward authentic transactions (label 1), three strategies for handling unbalanced data sets namely oversampling, undersampling, and the Synthetic Minority Oversampling Technique (SMOTE) are implemented. At this point, there are 6 training sets to fit three classification models: the naive Bayes (*nb*), random forest (*rf*), and the XGBoost (*xgb*). The performance indicators of the 18 fitted models are collected. F1 score, accuracy, recall, and precision performance metrics are used to evaluate the performance of methods.

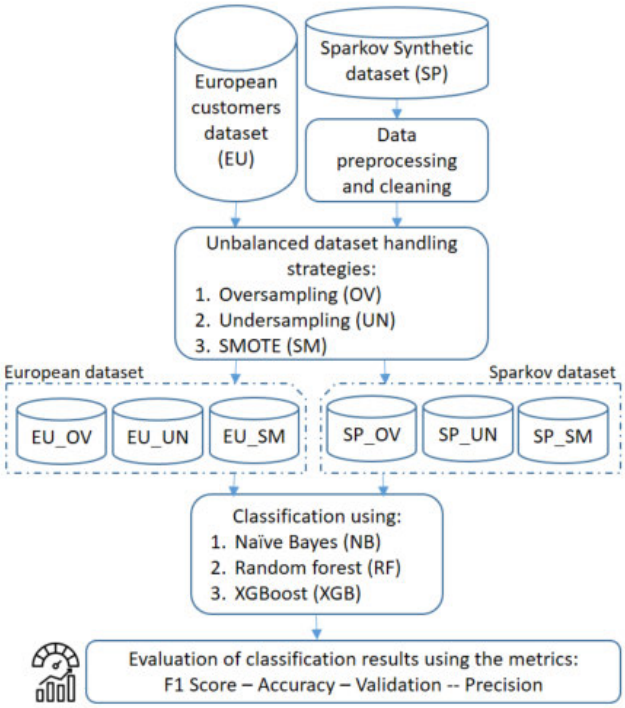


FIGURE 1. The research methodology.

IV. RELATED WORK

Nowadays, electronic payment (e-payment) constitutes one of the main trending Fintech solutions. E-payment is defined simply as the transfer of funds via electronic channels like digital wallets, credit or debit cards, or mobile banking. Such

technology offers several advantages, like reduced costs in terms of time and resources, efficiency, a cashless economy, transparency of transactions, etc. However, e-payments are not always secure and the loss of credit card information will lead surely to fund loss [10]. The number of payments completed using credit cards is continuously increasing. Consequently, the risk of fraud on credit cards will grow and the lost funds will follow the same pattern. CCF detection and prevention is then a priority for financial services providers like banks, insurance, and card payment networks (Visa, MasterCard).

The problem is defined as the detection of doubtful transactions being completed by fake cardholders [10], [11]. Typically, the fraudulent transaction should be spotted before its completion to avoid fund loss. If a hacker gets access to the card information, he can easily steal the card funds. The first remedy to CCF is then to secure more card information [1]. Even though the measures to avoid the leakage of credit card details are continuously improved and implemented by e-payment providers, still, the number and amounts of fraudulent transactions are still increasing as stated in [3]. Such a fact led to the design of innovative techniques to prevent CCF using statistical and machine learning approaches. That's by studying the data collected from previous transactions.

Statistical inference approaches for CCF look for outliers detection in the transactions data sets [11]. An outlier is a data point lying out of the normal range of the distribution. In CCF, an outlier is a transaction completed by a hacker that led to stolen funds. Data visualization via some charts like box plots, and probability distribution helps to the pattern of fraudulent transactions [12]. Hybrid approaches employing fuzzy logic and neural networks have been designed also to handle CCF detection [13]. [14], [15] stated that graph analysis and unsupervised clustering analyses are widely used to detect fraudulent transactions. In a wider context, graph, and network analytics can be used also to spot criminal relationships in money laundering transactions [14]. Other statistical-based approaches have been also proposed to solve the CCF detection, like the Dempster-Shafer theory and Bayesian learning, BLAST-SSAHA hybridization, Hidden Markov Model (HMM), Fuzzy Darwinian logic [11].

Machine learning constitutes the second class of approaches developed to handle the CCF. One can see that almost all machine learning techniques have been used. Depending on the problem's representation as a clustering or classification, machine learning models have been designed accordingly. Therefore, in Table 1, the techniques proposed under each class are summarised.

The before-mentioned classification or clustering methods for solving the CCF are based on the use of CCF data sets. The European customers' data set [8] is the most studied data set in the CCF literature. The *eu* data set is used by supervised learning algorithms where the label (type of the transaction is given). Most of the papers dealing with the CCF have employed the *eu* data set to fit various classification techniques. Our paper uses also the same data set along with the Sparkov simulated data set but for a different purpose than other papers. The Sparkov data set is a simulated CCF data set [9]. Some other data sets have been used also like the Brazilian data set and the Commercial Banks in China data sets [2]. It is worth noting that all CCF data sets are imbalanced toward authentic transactions. Therefore, strategies for handling data sets are usually employed like SMOTE, oversampling, and undersampling [4], [10], [16], [17]. The evaluation of the performance of the designed approaches for classification CCF is mainly based on the F1 score, ROC curve, accuracy, precision, and recall. For the clustering CCF, the metrics like homogeneity, completeness, and v-measure. In classification models, ensemble methods usually outperform basic methods like decision trees and Naive Bays [1], [5].

TABLE 1. Machine learning models proposed for the CCF.

CCF representation	ML Model	References
Classification	Logistic regression	[4], [10], [16]
	Random forest	[4], [10], [17]
	Naive Bayes	[4], [10], [16]
	Decision trees	[4]
	Gradient Boosting	[4]
	Neural networks (NN)	[2], [17]–[19]
	Support Vector Machines	[2]
	Deep convolutional NN	[20], [21]
	Deep belief NN	[22]
Clustering	K-Nearest Neighbors	[16], [23], [24]
	DBSCAN	[25]

The majority of the literature on the CCF, as described in this section, focuses on the creation and implementation of methods for identifying fraudulent credit card transactions. Such methods may be based on statistical inference methods, machine learning methods, etc. Without examining the reasons for these security system failures, the research concentrates on how the CCF might be prevented. In this essay, we try to illustrate one of the flaws in the business process that allows fraudulent transactions to be approved. To emphasize the predictability of credit card transaction approval/denial scripts, we shall use ensemble approaches.

V. EMPIRICAL STUDY

A. DATASETS AND DATA PREPROCESSING

This study has used two data sets on credit card transactions:

1) The European Union cardholders data set [8]: It contains 284,807 transactions made during two days in 2013 by European cardholders. The data set is highly imbalanced with 492 fraudulent transactions only. Each transaction is described by the time, amount, and 28 variables obtained by applying the PCA to the original features that have been hidden for confidentiality reasons. The type of the transaction is given in the binary variable 'Class'.

2) Simulated Credit Card Transactions generated using Sparkov simulator [9]: The data set [9] contains 1048574 legitimate transactions and 555718 fraud transactions simulated from the duration January 2019 and December 2020. The transactions are described by 22 variables including the date, amount, and the binary label 'is_fraud' (0: not fraud, 1: fraud). The data set *eu* was made ready for classification and does not need further processing. The preprocessing of the Sparkov simulated data set *sp* is completed as follows:
   - Initially, we removed the following columns as they are irrelevant for the classification of the type of the transaction: transaction time (*unix_time*), credit card number (*cc_num*), merchant name (*merchant*), customer first name (*first*), customer last name (*last*), customer gender (*gender*), customer street (*street*), customer city (*city*), customer state (*state*), customer position latitude (*lat*), customer position longitude (*long*), transaction number (*trans_num*), merchant position latitude (*merch_lat*), and merchant position longitude(*merch_long*).
   - The second step was to convert the date and time variables: the transaction date and time (*trans_date_trans_time*) and the customer date of birth (*dob*) into integers.
   - The encoding of categorical variables category and job type using the label encoder.
   - Data standardization to the normal scale by dividing each variable by its standard deviation after subtracting its mean.

   The data preprocessing phase of the data set *sp* resulted in a data set composed of 1852394 rows and 8 columns including the label *'is_fraud'*.

The objective of the preprocessing process depicted above is to ensure the robustness of the selected algorithms against varied types of noisy data. Mainly, the standardization step is to remove outliers and to ensure that no null values exist in the datasets.

B. CLASSIFICATION

The classification step was conducted using the three classifiers: the naive Bayes, the random forest, and the XGBoost algorithm. Below, we describe the three algorithms and their respective parameter settings:

1) Naive Bayes classifiers [26] are examples of network models. They are Bayesian probabilistic classifiers that make substantial assumptions about the independence of data set attributes. Naive Bayes models use the Maximum-likelihood training algorithm to fit their parameters to the training set. The computational complexity of the Naive Bayes classifier is $\theta(Nd)$ where N is the size of the training set and d is the number of features.
2) Random forests [27] are ensemble learning methods that can be used for classification problems. A random forest classifier is built by combining several basic decision trees. Random forests belong to the class of voting-based ensemble methods where the final output is the one selected by most of the trees. It is worth noting that random forests can avoid overfitting to their training set. Random forests are based on decision tree classifiers; their complexity is $\theta(tdnlog(n))$ where t is the number of trees used in the *rf* model.
3) XGBoost [28] is an open-source software library that implements optimized distributed gradient boosting machine learning algorithms under the Gradient Boosting framework. XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine-learning library for regression, classification, and ranking problems. The computation complexity of *xgb* is O(tdxlogn), where x is the number of non-missing entries in the training set.

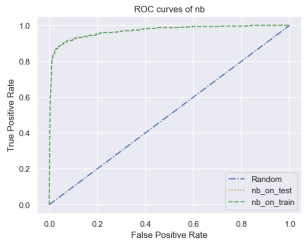
In this paper, three classifiers are implemented using the machine learning library Scikit-learn (Sklearn) of Python. Sklearn is an open-source machine-learning library widely used in academia and industry. For the naive Bayes model, the Gaussian Naive Bayes classifier (gaussianNB) is used with the default parameters of the model without additional settings. The random forest model was fitted from the same library using the following parameters:

$$\cdot\ n\_estimators = 5$$
$$\cdot\ max\_samples = 0.2$$
$$\cdot\ max\_features = 0.3$$
$$\cdot\ max\_depth = 3$$

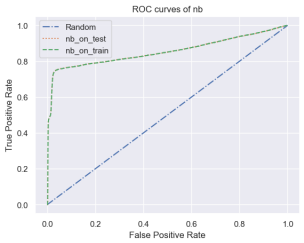
For the XGBoost model, default parameters of the Sklearn library with 5 estimators and a maximum depth of trees of 3, are used ($n\_estimators = 5$ and $max\_depth = 3$). It's important to highlight that all models have been fitted using the same parameters as mentioned above for all training sets.

C. COMPUTATIONAL RESULTS

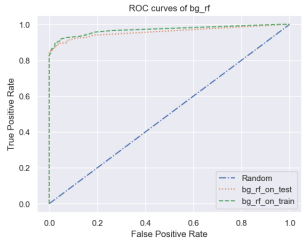
The pre-processing and the handling of the imbalance of the data sets yield 6 sets to be used to fit the 3 models. To properly train and evaluate the classifiers, we divided each data set into 3 subsets: the training set (70%), the test set (15%), and the validation set(15%). The training set is used to fit each model. The test set is used to evaluate the performance of the fitted model during the training phase. The validation set is used to assess the quality of the final model. Additionally, the validation set, separated from the original data set, will help to avoid the problem of data leakage. To handle overfitting and data leakage issues, we implemented all models using the KFold cross-validation technique. We used the KFold cross-validator from the *model_selection* package of the Sklearn library with $n\_splits = 5$.



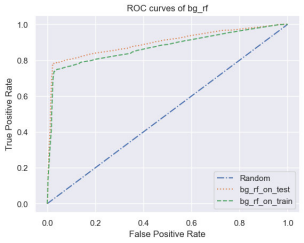
(a) ROC of nb on eu dataset using oversampling



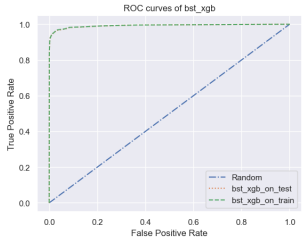
(b) ROC of nb on sp dataset using oversampling



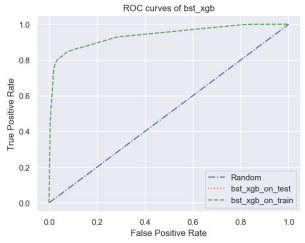
(c) ROC of rf on eu dataset using undersampling



(d) ROC rf on sp dataset using undersampling



(e) ROC of xgb on eu dataset using SMOTE



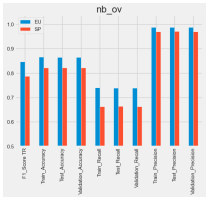
(f) ROC of xgb on sp dataset using SMOTE

FIGURE 2. A sample of ROC curves under different scenarios.

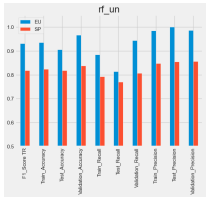
The results of the classification phase were obtained after fitting the *nb*, *rf*, and *xgb* models on the data sets *eu* and *sp* under three strategies (*un*, *ov*, and *sm*). The experiment offered then 18 fitted models each corresponding to a classifier trained on a data set using a certain strategy. For example, *eu_ov_nb* is the naive Bayes classifier fitted on the European data set using the oversampling strategy. Similarly, *sp_sm_xgb* is the XGBoost model fitted to the Sparkov data set using the SMOTE strategy. To evaluate the performance of the models during the training, test, and validation phases, we use the F1 score, accuracy, precision, and recall metrics.

The ROC curves under different scenarios show that the performance of all classifiers is better on the real dataset *eu* than on the synthetic dataset *sp* as shown in the sample plotted in Figure 2. It is worth noting that the *xgb* classifiers were able to outperform other classifiers under all schemes. Comparing the sampling techniques, we can confirm that SMOTE method helped to balance the training sets and allowed the classifier to better discriminate between fraudulent and genuine transactions.

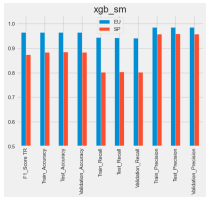
The performance of the *nb* classifier is lowest for all metrics and on all data sets. Its highest training accuracy of 0.86 for the model *eu_sm_nb*. The same *nb* model has the same accuracy of 0.86 on the test and the validation



(a) nb classifier performance summary



(b) rf classifier performance summary



(c) xgb classifier performance summary

FIGURE 3. Classifiers performance by data set.

sets. It is important to mention that the *nb* performs more on the European data set than the Sparkov data set for all metrics. The performance of *rf* and *xgb* are quite similar for the European and Spakov data sets. However, they perform much better on the real data set *eu* than the Sparkov data set *sp*. That can be seen in Figure 3. In Figure 3, we plotted the evaluation metrics for some selected models. It is worth noting that the precisions are lower than the accuracy and recall for all models.

Figure 4 [1] details the performance of the fitted model on the European data set *eu* under each strategy. We can observe that the models' performance improves considerably using the SMOTE strategy on the *eu* data set. The worst performance is shown using the undersampling strategy. The same finding can be seen in Figure 5, where the models' performance is better using SMOTE on the *sp* data set. We can see also in the same two figures that *rf* and *xgb* outperform the *nb* classifier under all scenarios.

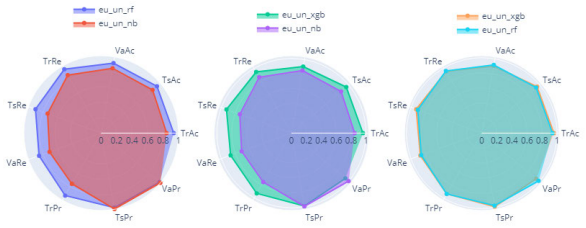
F1 score [2] is the harmonic mean of the precision and recall and it represents both precision and recall in one metric. Figure 6 represents the probability distribution of the F1 score by the model (Figure 6b) and by the data set (Figure 6a). The F1 score by the model (Figure 6b) is lower for the *nb* classifier and higher for the *rf* and *xgb* models. This joins the finding stated above. However, it is more variable for both ensemble models than the *nb* classifier. Comparing the F1 score distribution by data set shows in 6a that it is higher and more consistent for the Sparkov data set whereas it is low and more variable for the European real data set. Moreover, we can see clearly that the average F1 score obtained for all models is much higher on the *eu* data set compared to the synthetic data set *sp* as shown in 7a. The detailed gaps in the performance measure of the fitted models on both data sets are depicted in Figure 7b.

VI. RESEARCH FINDINGS

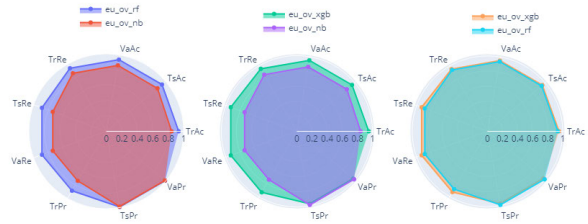
Currently, the process used to approve the transactions on credit cards is based on passing successfully the

[1] Legend for Figures 4 and 5:
TrAc: Train Accuracy *TsAc*: Test Accuracy *VaAc*: Validation Accuracy
TrPr: Train Precision *TsPr*: Test Precision *VaPr*: Validation Precision
TrRe: Train Recall *TsRe*: Test Recall *VaRe*: Validation Recall

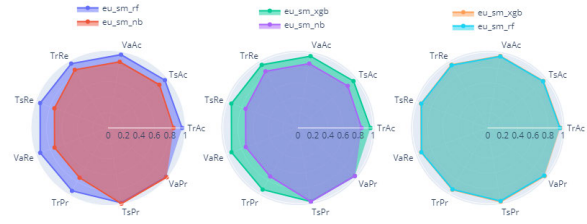
[2] $F_1 = 2 * (precsion * recall)/(precision + recall)$



(a) EU data set with undersampling strategy.

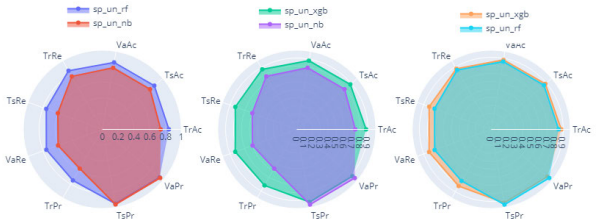


(b) EU data set with oversampling strategy.

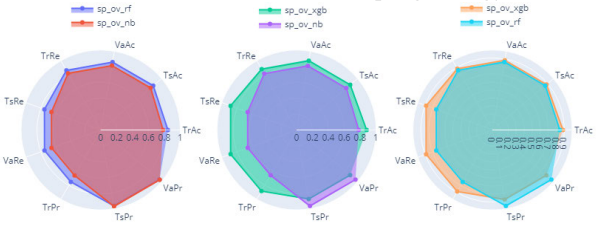


(c) EU data set with SMOTE sampling strategy.

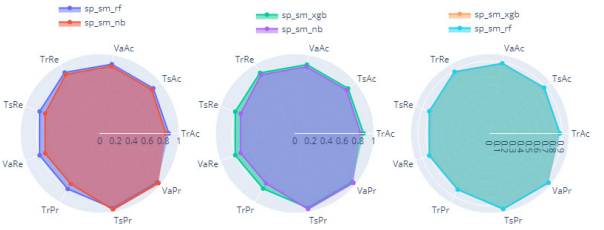
FIGURE 4. Comparison of the performance of the classifier on the European data set.



(a) SP data set with undersampling strategy.

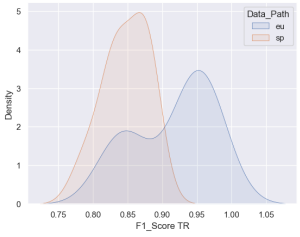


(b) SP data set with oversampling strategy.

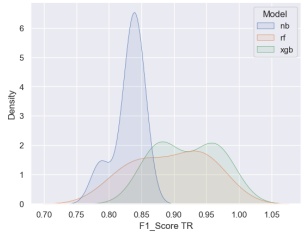


(c) SP data set with SMOTE sampling strategy.

FIGURE 5. Comparison of the performance of the classifier on the Sparkov data set.



(a) Density probability distributions of F1 score by data set

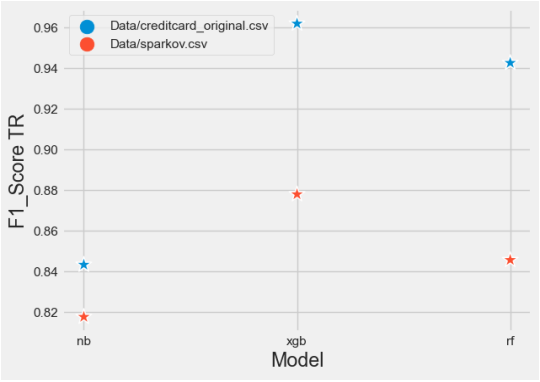


(b) Density probability distributions of F1 score by model

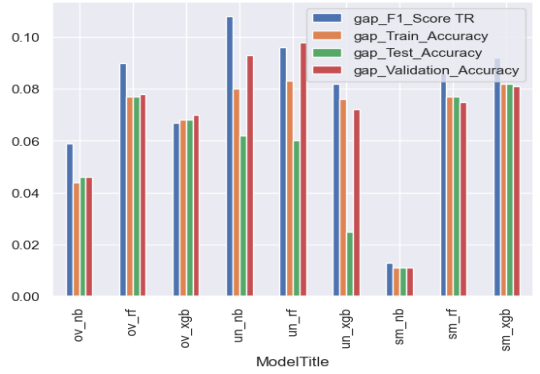
FIGURE 6. Density probability distributions of F1 score.

authentication step. If a user (true card owner or hacker) knows the details of the card (number, expiration date, Card Code Verification CCV, PIN code), he can get all his requests on this credit card approved. However, such card details can be easily leaked. Different ways might be used for that like lost or stolen cards, counterfeit or doctored or faked cards, phone scams, and phishing scams among others. The authentication process is then clearly deterministic and vulnerable. This fact explains fundamentally, the performance of ensemble methods in correctly separating authentic from fraudulent transactions on the European consumers' real data set and they fail to perform well on the synthetic data set of Sparkov. In the context of our study and experiment, both data sets have been used to fit the same type of classification models (*nb*, *rf* , and *xgb*) under the same settings. The results of the experiment show clear gaps (see Figure 7b) in all performance evaluation metrics. Such a gap is mainly explained by the nondeterministic nature of the Sparkov data set where the bias is much higher. We can conclude that the main source of the vulnerability of credit card request processing scripts is their determinism and absence of any form of uncertainty or bias. To overcome such limitations and mitigate this vulnerability, we recommend considering the following remedial actions:

- Focus on more descriptors of the operation like:
  - The location from where the transaction is being completed and if it is a previously known location used by the true cardholder.
  - The type of the transaction: online payment, cash withdrawal, fund transfer, etc.
  - The amount of the transaction; is it in the normal range of the previous operations completed by the true cardholder?
- Vary the authentication scheme like the Two Factor Authentication (2FA) by using:
  - Text message (SMS),
  - Verification emails,
  - Authenticator applications,



(a) Average F1 score of each classifier on the EU and Sparkov data sets.



(b) Gaps in the performance of classifiers on the EU and Sparkov data sets.

FIGURE 7. Performance gaps comparison between *eu* and *sp* data sets.

  - Push notifications on mobile phones,
  - Fast IDentity Online (FIDO) using biometric identification like fingerprint, face, eyes, or voice.
- Use Multi-factor Authentication (MFA) by requiring the customer to provide more verification factors from the above.

The current study revealed the importance of randomization in the datasets used to train machine learning algorithms. Such findings can be strengthened by investigating the behavior of other machine learning methods like pre-trained or hybrid deep models (like BERT, CNN, BiGRU) on other datasets. On the other hand, the tuning of the selected machine learning algorithms can lead to better performance.

VII. CONCLUSION AND FUTURE WORK

The detection of malicious transactions on credit cards helps in avoiding a big loss of money for financial institutions. That loss is continuously increasing despite the efforts deployed by financial stakeholders. In this paper, a comparative study of ensemble methods' performance is discussed in classifying credit card transactions as authentic or malicious. It is found that XGBoost and bagging methods outperform basic classification techniques like the naive Bayes. However, they lack overfitting on real data sets. On simulated data sets, the performance of all classifiers decreases since data generation did not follow an a priori script or distribution. The high performance of ensemble methods on real data can be explained by the fact that the approval of credit card transactions by bankers follows a strict script easily discovered by transfer learning. Such a finding can be seen as a credit card transaction processing script vulnerability. In future works, multiplying, varying, and randomizing the factors should be used during the authentication phase. From a technical point of view, addressing the explainability and interpretability of the algorithms and understanding their decision-making process will be an interesting perspective for future studies.

REFERENCES

[1] Z. Faraji, "A review of machine learning applications for credit card fraud detection with a case study," *SEISENSE J. Manage.*, vol. 5, no. 1, pp. 49–59, Feb. 2022.

[2] 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.

[3] Nilson Report. *Card Fraud Worldwide*. Accessed: May 2023. [Online]. Available: https://nilsonreport.com/

[4] N. S. Alfaiz and S. M. Fati, "Enhanced credit card fraud detection model using machine learning," *Electronics*, vol. 11, no. 4, p. 662, Feb. 2022.

[5] B. Arora and Sourabh, "A review of credit card fraud detection techniques," *Recent Innov. Comput.*, pp. 485–496, 2022.

[6] S. Srinidhi, K. Sowmya, and S. Karthika, "Automatic credit fraud detection using ensemble model," in *ICT Analysis and Applications*. Springer, 2022, pp. 211–224.

[7] M. Sabih and D. K. Vishwakarma, "A novel framework for detection of motion and appearance-based anomaly using ensemble learning and LSTMs," *Exp. Syst. Appl.*, vol. 192, Apr. 2022, Art. no. 116394.

[8] Kaggle. (2022). *European Cardholders Dataset*. Accessed: May 2023. [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[9] *Sparkov Data Generation on Github*, Sparkv simulator, 2020.

[10] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Credit card fraud detection–machine learning methods," in *Proc. 18th Int. Symp. INFOTEH-JAHORINA (INFOTEH)*, Mar. 2019, pp. 1–5.

[11] S B. E. Raj and A A. Portia, "Analysis on credit card fraud detection methods," in *Proc. Int. Conf. Comput., Commun. Electr. Technol. (ICCCET)*, 2011, pp. 152–156.

[12] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," 2010, *arXiv:1009.6119*.

[13] T. Razooqi, P. Khurana, K. Raahemifar, and A. Abhari, "Credit card fraud detection using fuzzy logic and neural network," in *Proc. 19th Commun. Netw. Symp.*, 2016, pp. 1–5.

[14] M. E. Lokanan, "Financial fraud detection: The use of visualization techniques in credit card fraud and money laundering domains," *J. Money Laundering Control*, vol. 26, no. 3, pp. 436–444, Apr. 2023.

[15] B. Lebichot, F. Braun, O. Caelen, and M. Saerens, "A graph-based, semi-supervised, credit card fraud detection system," in *Proc. 5th Int. Workshop Complex Netw. Appl. (COMPLEX Network)*. Springer, 2017, pp. 721–733.

[16] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *Proc. Int. Conf. Comput. Netw. Informat. (ICCNI)*, Oct. 2017, pp. 1–9.

[17] I. Sohony, R. Pratap, and U. Nambiar, "Ensemble learning for credit card fraud detection," in *Proc. ACM India Joint Int. Conf. Data Sci. Manage. Data*, Jan. 2018, pp. 289–294.

[18] R. Bin Sulaiman, V. Schetinin, and P. Sant, "Review of machine learning approach on credit card fraud detection," *Human-Centric Intell. Syst.*, vol. 2, nos. 1–2, pp. 55–68, Jun. 2022.

[19] 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.

[20] J. I.-Z. Chen and K.-L. Lai, "Deep convolution neural network model for credit-card fraud detection and alert," *J. Artif. Intell.*, vol. 3, no. 2, pp. 101–112, 2021.
[21] A. Alharbi, M. Alshammari, O. D. Okon, A. Alabrah, H. T. Rauf, H. Alyami, and T. Meraj, "A novel text2IMG mechanism of credit card fraud detection: A deep learning approach," *Electronics*, vol. 11, no. 5, p. 756, Mar. 2022.
[22] J. M. V and D. Vimal, "Population based optimized and condensed fuzzy deep belief network for credit card fraudulent detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 9, 2020.
[23] V. Vaishali, "Fraud detection in credit card by clustering approach," *Int. J. Comput. Appl.*, vol. 98, no. 3, pp. 29–32, Jul. 2014.
[24] P. Chougule, A. D. Thakare, P. Kale, M. Gole, and P. Nanekar, "Genetic K-means algorithm for credit card fraud detection," *Int. J. Comput. Sci. Inf. Technologies (IJCSIT)*, vol. 6, no. 2, pp. 1724–1727, 2015.
[25] L. Han, "Advanced DBSCAN: A clustering algorithm for personal credit reference system," in *Proc. Intell. Syst. Conf. (IntelliSys)*. Cham, Switzerland: Springer, 2020, pp. 370–381.
[26] K. M. Leung, "Naive Bayesian classifier," *Polytech. Univ. Dept. Comput. Sci./Finance Risk Eng.*, vol. 2007, pp. 123–156, Nov. 2007.
[27] A. Prinzie and D. Van den Poel, "Random forests for multiclass classification: Random MultiNomial logit," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1721–1732, Apr. 2008.
[28] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

JABER JEMAI received the master's and Ph.D. degrees in computing and information systems from the University of Tunis and the master's Diploma degree in data science and business analytics from The University of Texas at Austin.

He is currently an Associate Professor in computer information systems with the Higher Colleges of Technology, United Arab Emirates. His research interests include the application of artificial intelligence and optimization techniques in various business problems like green transportation and logistics, healthcare operations, and fin-tech. He maintains a strong record of publications in refereed international journals and conferences.

ANIS ZARRAD received the M.Sc. degree from Concordia University, Canada, and the Ph.D. degree from the University of Ottawa, Canada.

He has been an Associate Professor with the School of Computer Science, University of Birmingham Dubai, Dubai, since 2018. From 2020 to 2022, he was the Digital Lead of the Computer and Engineering Programs. His current research interests include search-based software engineering (SBSE), encompassing requirements, project planning, and maintenance to reengineering. He actively collaborates with researchers from various universities.

Dr. Zarrad has contributed to over ten conference and workshop program committees. Additionally, he serves as a reviewer for esteemed journals published by Elsevier, Springer, and IEEE.

ALI DAUD received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in July 2010. Currently, he is a Full Professor with the Faculty of Resilience, Rabdan Academy, Abu Dhabi, United Arab Emirates. He has 13 years' post-Ph.D. experience of teaching, supervision, and research at B.S., M.S., and Ph.D. level. He has published more than 100 research papers in reputed international impact factor journals and conferences. He has taken part in many research projects as well and have written and acquired many research fundings. He has proven and extensive experience in data mining, artificial intelligence (machine learning/deep learning) applications to social networks, data science, and natural language processing.