Machine Learning Techniques for Detecting Fraud in Credit Card Transactions

Tariq Mahmood[1,*], Seyedeh Khadijeh Hashemi[2,†], Seyedeh Leili Mirtaheri[2,†] and Sergio Greco[1,†]

[1]Department of Computer Science, Modeling, Electronics and Systems Engineering, University of Calabria, Italy
[2]Department of Electrical and Computer Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran

Abstract

Credit cards became one of the most popular payment methods as technology advanced and e-commerce services expanded, resulting in an increase in the volume of banking transactions. Furthermore, the significant increase in the number of frauds necessitates high banking transaction costs. As a result, detecting fraudulent activities has become an intriguing topic, attracting a large number of researchers. We consider the use of class weight-tuning hyperparameters to control the weight of fraudulent and legitimate transactions in this study. We use Bayesian optimization, in particular, to optimize the hyperparameters while taking into account practical issues such as unbalanced data. We propose weight-tuning as a pre-process for unbalanced data, as well as using CatBoost and XGBoost to improve the performance of the LightGBM method by taking the voting mechanism into account. Finally, we use deep learning to fine-tune the hyperparameters, particularly our proposed weight-tuning one, in order to improve performance even further. We conducted some experiments to evaluate the proposed methods on real-world data. We use recall-precision metrics in addition to the common ROC-AUC to better cover unbalanced datasets. We use a 5-fold cross-validation method to test CatBoost, LightGBM, and XGBoost separately. Furthermore, the performance of the combined algorithms is evaluated using the majority voting ensemble learning method. According to the results, LightGBM and XGBoost achieve the best level criteria of ROC-AUC = 0.95, precision 0.79, recall 0.80, F1 score 0.79, and MCC 0.79. Also, by using deep learning and the Bayesian optimization method to tune the hyperparameters, we meet the following criteria: ROC-AUC = 0.94, precision = 0.80, recall = 0.82, F1 score = 0.81, and MCC = 0.81. This is a big improvement over the state-of-the-art methods we compared it to.

Keywords

Machine Learning, Deep Learning, Hyper parameter, Unbalanced Data, Bayesian Optimization

1. Introduction

In recent years, there has been a significant increase in the volume of financial transactions due to the expansion of financial institutions and the popularity of web-based e-commerce. Fraudulent transactions have become a growing problem in online banking, and fraud detection has always been challenging [2, 3].

Note: This is the short version of reference [1]

SEBD 2023: 31st Symposium on Advanced Database System, July 02–05, 2023, Galzignano Terme, Padua, Italy

[*] Corresponding author.

[†] These authors contributed equally.

mahmood.tariq@dimes.unical.it (T. Mahmood); s.greco@dimes.unical.it (S. Greco)

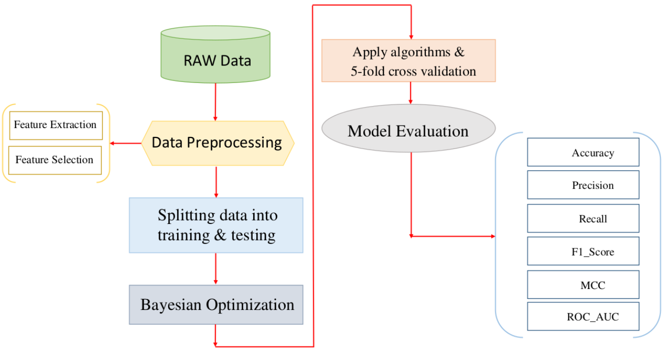

Figure 1: Proposed framework for credit card fraud detection.

Along with credit card development, the pattern of credit card fraud has always been updated. Fraudsters do their best to make it look legitimate, and credit card fraud has always been updated. Fraudsters do their best to make it look legitimate. They try to learn how fraud detection systems work and continue to stimulate these systems, making fraud detection more complicated. Therefore, researchers are constantly trying to find new ways or improve the performance of the existing methods [4].

There are two mechanisms, fraud prevention and fraud detection, that can be exploited to avoid fraud-related losses. Fraud prevention is a proactive method that stops fraud from happening in the first place. On the other hand, fraud detection is needed when a fraudster attempts a fraudulent transaction [5].

Fraud detection in banking is considered a binary classification problem in which data is classified as legitimate or fraudulent [6]. Because banking data is large in volume and with datasets containing a large amount of transaction data, manually reviewing and finding patterns for fraudulent transactions is either impossible or takes a long time. Therefore, machine learning-based algorithms play a pivotal role in fraud detection and prediction [7].

In this paper, we propose an efficient approach for detecting credit card fraud that has been evaluated on publicly available datasets and has used optimized algorithms LightGBM, XGBoost, CatBoost, and logistic regression individually, as well as majority voting combined methods, as well as deep learning and hyperparameter settings. An ideal fraud detection system should detect more fraudulent cases, and the precision of detecting fraudulent cases should be high, i.e., all results should be correctly detected, which will lead to the trust of customers in the bank, and on the other hand, the bank will not suffer losses due to incorrect detection.

The main contributions of this paper are summarized as follows:

- We adopt Bayesian optimization for fraud detection and propose to use the weight-tuning hyperparameter to solve the unbalanced data issue as a pre-processing step.
- We propose a majority-voting ensemble learning approach to combine CatBoost, XGBoost, and LightGBM and review the effect of the combined methods on the performance of fraud detection on real, unbalanced data.
- To better cover the unbalanced datasets, we use recall-precision in addition to the typically used ROC-AUC. We also evaluate the performance using F1_score and MCC metrics.

Table 1
The features of the **credit-card fraud** dataset that is used in this paper.

Variable Name	Description	Type
$V_1, V_2, ..., V_{28}$	Transaction feature after PCA transformation	Integer
Time	Seconds elapsed between each transaction with the first transaction	Integer
Amount	Transaction Value	Integer
Class	Legitimate or Fraudulent	0 or 1

According to the results, the proposed methods outperform the existing and based methods. For evaluations, we use publicly available datasets and also publish the source codes [1] with public access to be used by other researchers.

2. Related Work

A number of methods have been proposed by researchers in order to detect and prevent fraudulent credit card transactions. Halvaiee & Akbari develop a new fraud detection model called the AIS-based fraud detection model (AFDM). To improve fraud detection accuracy, they use Immune System Inspired Algorithms (AIRS). According to their paper, their proposed AFDM improves accuracy by up to 25%, reduces costs by up to 85%, and reduces system response time by up to 40% compared to basic algorithms [8].

Randhawa et al., analyze the effectiveness of machine learning algorithms for detecting credit card fraud. To evaluate the available datasets, they used Naive Bayes, stochastic forest and decision trees, neural networks, linear regression (LR), and logistic regression, as well as support vector machine standard models. By applying AdaBoost and majority voting, they propose a hybrid method. Additionally, they add noise to the data samples in order to evaluate robustness. On publicly available datasets, they demonstrate that majority voting is effective at detecting credit card fraud[9].

In [10] the authors propose a group learning framework based on partitioning and clustering of the training set. Their proposed framework has two goals: 1) to ensure the integrity of the sample features, and 2) to solve the high imbalance of the dataset. The main feature of their proposed framework is that every base estimator can be trained in parallel, which improves the effectiveness of their framework.

To detect fraud in credit card transactions, Altyeb et al., propose an intelligent approach consisting of a Bayesian-based hyperparameter optimization algorithm for tuning LightGBM parameters [11]. A publicly available dataset of credit card transactions is used for their experiments. Transactions from both legitimate and fraudulent sources are included in these datasets. Their evaluation results are reported in terms of accuracy, the area under the receiver operating characteristic curve (ROC-AUC), precision, and F1-score metrics.

Verma and Tyagi investigate machine learning algorithms in order to determine the best supervised ML-based algorithm for credit card fraud detection in the presence of an imbalanced

[1] The codes are available at https://github.com/khadijehHashemi/Fraud-Detection-in-Banking-Data-by-Machine-Learning-Techniques

Table 2
The transaction label distribution in the "credit card" dataset This unbalanced data is expected in real-life datasets.

No. of Transactions	No. of legitimate Transactions	No. of fraudulent Transactions	Legitimate (%)	fraudulent (%)
284,807	284,315	492	99.83%	0.17%

dataset. They evaluate five classification techniques and show that the supervised vector classifier and logistic regression classifier outperform other algorithms in an imbalanced dataset [12].

2.1. Proposed approach

The proposed framework for credit card fraud detection is presented in Fig.1. We first apply the pre-processing on the dataset and further split the data into two sections: training and testing, followed by performing Bayesian optimization on the training data to find the best hyper-parameters that lead to the improvement of the performance. We use the cross-validation method to obtain performance comparison in an unbalanced set and then examine the algorithms using different evaluation metrics, including accuracy, precision, recall, the Matthews correlation coefficient (MCC), the F1-score, and AUC diagrams.

we use a real dataset so that the outcome of the proposed algorithm can be used in practice. We consider a dataset named “credit card” that contains 284,807 records of two days of transactions made by credit card holders in September 2013. There are 492 fraudulent transactions, and the rest of the transactions are legitimate. The positive class (frauds) accounts for 0.172% of all transactions; hence, the dataset is highly imbalanced. the original features and background information about the data are not given due to confidentiality and privacy considerations. PCA yielded the following principal components: V_1, V_2, V_{28}. The untransformed features with PCA are "time" and "amount." The "Time" column contains the time (in seconds) elapsed between each transaction and the first transaction in the dataset. The feature "Amount" shows the transaction amount. Feature "Class" is the response variable, and it takes the value 1 in case of fraud and 0 otherwise. The summary of the variables and features is presented in Table.1

The total number of fraudulent transactions are significantly lower than the total number of legitimate transactions, indicating that the data distribution is unbalanced as shown in Table 2. This data imbalance causes performance issues in machine learning algorithms, and having a class with the majority of the samples influences the evaluation results[9]. Therefore, in many studies, under-sampling and over-sampling methods are used to solve the data imbalance problem [13]. Using under-sampling methods leads to data loss [14]. Besides, using over-sampling methods leads to the production of duplicate data that doesn’t provide information (the data and information are different, and the subject is discussed under the “Entropy”). Some researchers use synthetic minority oversampling (SMOTE) as a solution, which avoids the drawbacks of under and oversampling [15, 11, 16]. However, the SMOTE method causes an increase in the false-positive rate, which is not acceptable in banking for customer orientation. To solve this problem, in this study, we use class weight tuning hyperparameter to solve the mentioned disadvantages [15, 11, 16]. However, the SMOTE method causes an increase in the

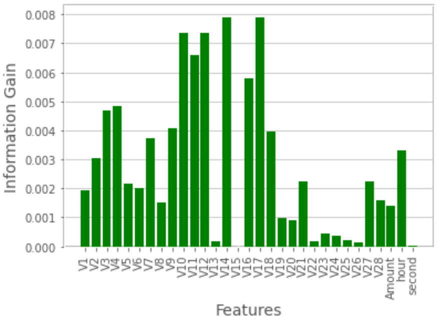

Figure 2: Feature importance diagram

false-positive rate, which is not acceptable in banking for customer orientation. To solve this problem, in this study, we use a class weight tuning hyperparameter to solve the mentioned disadvantages.

3. Features extraction and selection

The “time” feature includes the time (in seconds) elapsed between each transaction and the first transaction. The features are unknown except for “Time” and “Amount”, and we have no additional information. Feature selection tries to find a subset of features that improve the classifier’s performance on effectively detecting credit card fraud [17]. The information gain (IG) method is used to select the most important features that lead to a dimension reduction of the training data. Information gain functions by extracting similarities between credit card transactions and then awarding the greatest weight to the most significant features based on the class of legitimate and fraudulent credit card transactions [18, 11]. Fig.2 shows the diagram of the IG, and the top six features extracted by this method have been used to evaluate the proposed algorithm.

4. Algorithms

Hyperparameters have a significant effect on the performance of machine learning models. We refer to optimization as the process of finding the best set of hyperparameters that configure a machine learning algorithm during its training. In this paper, we use the Bayesian optimization algorithm to tune the hyperparameters that lead to computational time reduction and performance improvement.

4.1. Logistic Regression

This algorithm could not be used for unbalanced data. Therefore, we used hyperparameter class weight to solve the class imbalance prior to applying logistic regression. We show that the ROC-AUC curve cannot be used for the evaluation of unbalanced data and leads to false interpretations.

4.2. LightGBM

The LightGBM algorithm is built on the GBDT framework and aims to improve computational efficiency, particularly on big data prediction problems [19]. The high-performance LightGBM algorithm can quickly handle large amounts of data, and the distributed processing of data [11]. In LightGBM, the histogram-based algorithm and trees' leaf-wise growth strategy with a maximum depth limit is adopted to increase the training speed and reduce memory consumption. The tuned hyperparameters include the "num_leaves", which is the number of leaves per tree, "max_depth", which denotes the maximum depth of the tree, and "learning_rate" which is also balanced by tuning the weight of the class. With the excessive increase of the leaves, the problem fits horizontally. Therefore, we need to consider a suitable range for this algorithm to obtain good optimization results.

4.3. XGBoost

Extreme Gradient Boosting (XGBoost) has become a dominant algorithm in the field of applied machine learning. This algorithm is a hybrid technique in which new models are added to fix errors caused by existing models. XGBoost includes parallel computation to construct trees using all the CPUs during training. Instead of traditional stopping criteria (i.e., criterion first), it makes use of the "max depth" parameter and starts tree pruning from the backward direction, which significantly improves the computational performance and speed of XGBoost [19]. XGBoost employs a more regularised technique called "formalization" to control over-fitting and achieve better performance [20]. The tuned hyperparameters include learning rate, number of trees, and maximum tree depth, as well as applying weight to classes.

4.4. Majority-voting

Ensemble learning (EL), which is a type of machine learning, combines several classifiers, minimizes the error of the classifiers, and achieves more reasonable results than a single technique. A voting majority classifier is not a real classifier, but a method that is trained and evaluated in parallel in order to use the different features of each algorithm. We can train the data using different hybrid algorithms to predict the final output. The final result of the prediction is determined by a majority of votes according to two different strategies: hard voting and soft voting. If voting is hard, it uses the predicted class labels to vote for the majority law. Otherwise, if the vote is soft, it predicts the class label based on "Argmax," the sum of the predicted probabilities, which is recommended for a set of well-calibrated classifiers. In this case, the probability vector is calculated on average for each predicted class (for all classifiers). The winning class is the one with the highest value [21, 22].

4.5. Deep learning

Deep learning is shown to be a very promising solution to deal with fraud in financial transactions, making the best use of banks' big data. [23]. In this paper, we use a sequential model, which is a linear stack of layers to construct an artificial neural network model. We use the Relu activation function, and in the last layer, we use "Sigmoid", since our output is binary.

Table 3
Deep Learning Model Results

Model	Accuracy	AUC	Recall	Precision	F1-score	MCC
Keras	0.9994	0.9401	0.8222	0.8043	0.8132	0.8129

Table 4
Performance evaluation of Algorithms

Model	Accuracy	AUC	Recall	Precision	F1-score	MCC
Log_Reg	0.97477	0.9578	0.8730	0.0617	0.1143	0.2248
LGBM	0.99919	0.9472	0.7990	0.7534	0.7699	0.7727
XGB	0.99923	0.9517	0.7949	0.7862	0.7830	0.7864
CatBoost	0.99880	0.9390	0.8096	0.6431	0.7066	0.7158
Vot_Lg, Xg, Ca	0.99924	0.9501	0.8033	0.7720	0.7825	0.7847
Vot_Lg, Xg	0.99927	0.9522	0.8012	0.7901	0.7901	0.7925
Vot_g, Ca	0.99923	0.9492	0.8097	0.7681	0.7823	0.7852
Vot_Lg, Ca	0.99912	0.9459	0.8075	0.7260	0.7581	0.7620

The Sigmoid function generates values in a range of zero and one. The function of the Relu activation function is in many ways similar to the function of our biological neurons. We use kernel-initializer, which defines the method of determining the random weights of the primary Keras layers. To overcome the unbalanced data problem, we consider the ratio of 1 to 4 for the weight of the majority class to the minority class. This causes an increase in the processing speed as well as increasing the efficiency of the model. The size of the input layer is equal to the number of features plus the extracted features. We also remove the "time" feature. To build the Keras model, we optimise the number of layers and neurons, the number of epochs, and the batch size, which leads to an increase in speed. Commonly, batch size is set to 32 or 128. However, our dataset is highly unbalanced, and by choosing the common batch size, there may be no fraud cases in the batch during training. Therefore, our range is chosen so that we can see fraudulent samples in each batch. Also, by choosing a larger batch size, the processing is faster, and we also need less memory. Large epoch sizes can result in either over- or under-fitting. Therefore, selecting the appropriate range for optimization not only increases the efficiency of the algorithm but also reduces the time required to find the optimal points. By performing Bayesian optimization, the number of neurons in the first hidden layer is set to 86, the number of epochs is set to 117, and the batch size is set to 1563.

5. Experimental results and discussion

We use the stratified 5-fold cross-validation method and the boosting algorithms with the Bayesian optimization method to evaluate the performance of the proposed framework. We extract the hyperparameters and evaluate each algorithm individually before using the majority voting method. We examine the algorithms in triple and double precision. The comparison results are presented in Table.3

Most studies in the literature rely on AUC diagrams to evaluate performance. However, as

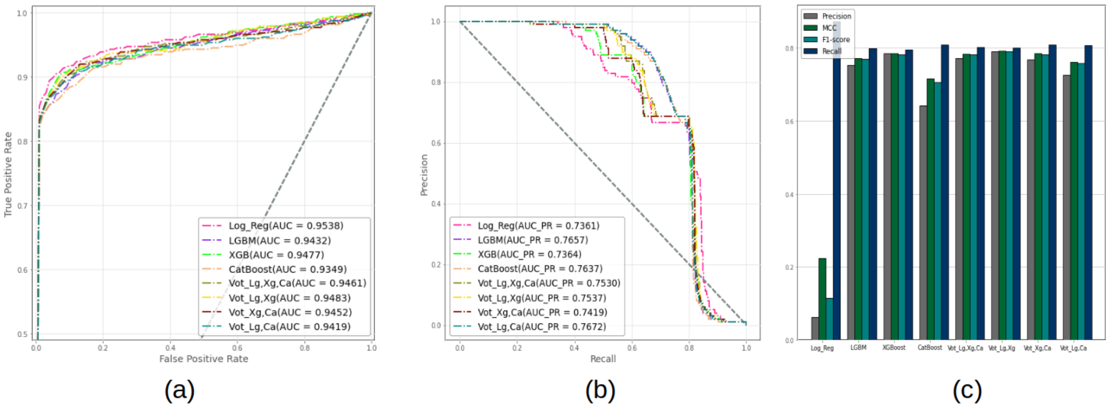

Figure 3: (a) ROC - AUC Curve (b) Precision-Recall curve (c) Performance comparing algorithms with different evaluation criteria .

Table 5
Performance comparison of the proposed approach and the method presented in [11]

Model	Accuracy	AUC	Recall	Precision	F1-score
Method presented in [11]	0.984	0.909	0.406	0.973	0.569
Proposed LightGBM	0.9992	0.947	0.799	0.753	0.769
Proposed Approach	0.9993	0.952	0.801	0.79	0.79

can be seen from the ROC-AUC curve in Fig.3 (a), the value of AUC in severely unbalanced data is not a good evaluation metric. It is influenced by the real positives and considers the negatives irrelevant. According to the ROC-AUC Fig.3, the logistic regression algorithm 0.9583 has the highest number of fraud detection, but it has the lowest value in other criteria. The precision-recall curve is illustrated in Fig.3 (b) and shows the system performance in a more precise manner compared with the ROC-AUC curve. Comparing the precision, recall, and F1-score as well as the MCC, the algorithms used are shown in Fig.3 (c). The evaluation results of the proposed approach using different pre-processing and class weight hyperparameter tuning to deal with the problem of data unbalance. In Table 5, it is shown that the proposed methods outperform the intelligence method.

6. Conclusion and future work

In this paper, we studied the credit card fraud detection problem in real unbalanced datasets. We proposed a machine-learning approach to improve the performance of fraud detection. Our experimental results showed that the proposed LightGBM method improved the fraud detection cases by 50 percent and the F1-score by 20 percent compared with the recently presented method in [17]. For future studies and work, we propose using other hybrid models as well as working specifically in the field of CatBoost by changing more hyperparameters.

References

[1] S. K. Hashemi, S. L. Mirtaheri, S. Greco, Fraud detection in banking data by machine learning techniques, IEEE Access (2022).

[2] J. Nanduri, Y.-W. Liu, K. Yang, Y. Jia, Ecommerce fraud detection through fraud islands and multi-layer machine learning model, in: Future of Information and Communication Conference, Advances in Information and Communication, Springer, San Francisco, USA, 2020, pp. 556–570.

[3] H. Feng, Ensemble learning in credit card fraud detection using boosting methods, in: 2021 2nd International Conference on Computing and Data Science (CDS), IEEE, 2021, pp. 7–11.

[4] M. Soltani Delgosha, N. Hajiheydari, S. M. Fahimi, Elucidation of big data analytics in banking: a four-stage delphi study, Journal of Enterprise Information Management 34 (2020) 1577 – 1596. doi:10.1108/JEIM-03-2019-0097.

[5] N. Kumaraswamy, M. K. Markey, T. Ekin, J. C. Barner, K. Rascati, Healthcare fraud data mining methods: A look back and look ahead, Perspectives in Health Information Management 19 (2022).

[6] E. F. Malik, K. W. Khaw, B. Belaton, W. P. Wong, X. Chew, Credit card fraud detection using a new hybrid machine learning architecture, Mathematics 10 (2022) 1480.

[7] K. Gupta, K. Singh, G. V. Singh, M. Hassan, G. Himani, U. Sharma, Machine learning based credit card fraud detection - a review, in: 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022, pp. 362–368. doi:10.1109/ICAAIC53929.2022.9792653.

[8] N. S. Halvaiee, M. K. Akbari, A novel model for credit card fraud detection using artificial immune systems, Applied soft computing 24 (2014) 40–49.

[9] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, A. K. Nandi, Credit card fraud detection using adaboost and majority voting, IEEE access 6 (2018) 14277–14284.

[10] H. Wang, P. Zhu, X. Zou, S. Qin, An ensemble learning framework for credit card fraud detection based on training set partitioning and clustering, in: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE, 2018, pp. 94–98.

[11] A. A. Taha, S. J. Malebary, An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine, IEEE Access 8 (2020) 25579–25587.

[12] P. Verma, P. Tyagi, Analysis of supervised machine learning algorithms in the context of fraud detection, ECS Transactions 107 (2022) 7189.

[13] F. Itoo, M. Meenakshi, S. Singh, Comparison and analysis of logistic regression, naive bayes and knn machine learning algorithms for credit card fraud detection, International Journal of Information Technology 13 (2021) 1503–1511.

[14] J. Zou, J. Zhang, P. Jiang, Credit card fraud detection using autoencoder neural network, arXiv preprint arXiv:1908.11553 (2019).

[15] M. Puh, L. Brkić, Detecting credit card fraud using selected machine learning algorithms, in: 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, 2019, pp. 1250–1255.

[16] D. Almhaithawi, A. Jafar, M. Aljnidi, Example-dependent cost-sensitive credit cards fraud detection using smote and bayes minimum risk, SN Applied Sciences 2 (2020) 1–12.

[17] J. Cui, C. Yan, C. Wang, Learning transaction cohesiveness for online payment fraud detection, in: The 2nd International Conference on Computing and Data Science, 2021, pp. 1–5.

[18] M. Rakhshaninejad, M. Fathian, B. Amiri, N. Yazdanjue, An ensemble-based credit card fraud detection algorithm using an efficient voting strategy, The Computer Journal (2021).

[19] W. Liang, S. Luo, G. Zhao, H. Wu, Predicting hard rock pillar stability using gbdt, xgboost, and lightgbm algorithms, Mathematics 8 (2020) 765.

[20] S. B. Jabeur, C. Gharib, S. Mefteh-Wali, W. B. Arfi, Catboost model and artificial intelligence techniques for corporate failure prediction, Technological Forecasting and Social Change 166 (2021) 120658.

[21] F. N. Khan, A. H. Khan, L. Israt, Credit card fraud prediction and classification using deep neural network and ensemble learning, in: 2020 IEEE Region 10 Symposium (TENSYMP), IEEE, 2020, pp. 114–119.

[22] A. Goyal, J. Khiari, Diversity-aware weighted majority vote classifier for imbalanced data, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.

[23] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, P. Beling, Deep learning detecting fraud in credit card transactions, in: 2018 Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2018, pp. 129–134.