IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 December 2022, accepted 19 December 2022, date of publication 26 December 2022,
date of current version 11 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3232287

RESEARCH ARTICLE

Fraud Detection in Banking Data by Machine Learning Techniques

SEYEDEH KHADIJEH HASHEMI[1], SEYEDEH LEILI MIRTAHERI[1], AND SERGIO GRECO[2]
[1]Department of Electrical and Computer Engineering, Faculty of Engineering, Kharazmi University, Tehran 15719-14911, Iran
[2]Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria, 87036 Rende, Italy

Corresponding author: Seyedeh Leili Mirtaheri (Mirtaheri@khu.ac.ir)

ABSTRACT As technology advanced and e-commerce services expanded, credit cards became one of the most popular payment methods, resulting in an increase in the volume of banking transactions. Furthermore, the significant increase in fraud requires high banking transaction costs. As a result, detecting fraudulent activities has become a fascinating topic. In this study, we consider the use of class weight-tuning hyper-parameters to control the weight of fraudulent and legitimate transactions. We use Bayesian optimization in particular to optimize the hyperparameters while preserving practical issues such as unbalanced data. We propose weight-tuning as a pre-process for unbalanced data, as well as CatBoost and XGBoost to improve the performance of the LightGBM method by accounting for the voting mechanism. Finally, in order to improve performance even further, we use deep learning to fine-tune the hyperparameters, particularly our proposed weight-tuning one. We perform some experiments on real-world data to test the proposed methods. To better cover unbalanced datasets, we use recall-precision metrics in addition to the standard ROC-AUC. CatBoost, LightGBM, and XGBoost are evaluated separately using a 5-fold cross-validation method. Furthermore, the majority voting ensemble learning method is used to assess the performance of the combined algorithms. 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, according to the results. By using deep learning and the Bayesian optimization method to tune the hyperparameters, we also meet the ROC-AUC = 0.94, precision = 0.80, recall = 0.82, F1 score = 0.81, and MCC = 0.81. This is a significant improvement over the cutting-edge methods we compared it to.

INDEX TERMS Bayesian optimization, data Mining, deep learning, ensemble learning, hyper parameter, unbalanced data, machine learning.

I. 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 [1], [2].

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 [3].

People who commit fraud usually use security, control, and monitoring weaknesses in commercial applications to achieve their goals. However, technology can be a tool to combat fraud [4]. To prevent further possible fraud, it is important to detect the fraud right away after its occurrence [5].

Fraud can be defined as wrongful or criminal deception intended to result in financial or personal gain. Credit card fraud is related to the illegal use of credit card information

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

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

for purchases in a physical or digital manner. In digital transactions, fraud can happen over the line or the web, since the cardholders usually provide the card number, expiration date, and card verification number by telephone or website [6].

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 [7].

Fraud detection in banking is considered a binary classification problem in which data is classified as legitimate or fraudulent [8]. 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 [9]. Machine learning algorithms and high processing power increase the capability of handling large datasets and fraud detection in a more efficient manner. Machine learning algorithms and deep learning also provide fast and efficient solutions to real-time problems [10].

In this paper, we propose an efficient approach for detecting credit card fraud that has been evaluated on publicly available datasets and has used optimised 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-process step. We also suggest using CatBoost and XGBoost alongside LightGBM to improve performance. We use the XGBoost algorithm due to the high speed of training in big data as well as the regularization term, which overcomes overfitting by measuring the complexity of the tree, and it does not require much time to set the hyperparameters. We also use the Catboost algorithm because there is no need to adjust hyperparameters for overfitting control, and it also obtains good results without changing hyperparameters compared to other machine learning algorithms.
- 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. We also propose to use deep learning for adjusting and fine-tuning the hyperparameters.
- To evaluate the performance of the proposed methods, we perform extensive experiments on real-world 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. 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.

The reminder of this paper is organized as follows: In Section II we review the related state-of-the-art. The proposed approach for credit card fraud detection including the dataset, pre-processing, feature extraction and feature selection, algorithms, framework, and evaluation metrics, is presented in Section III. Section IV discusses the evaluation results of the experiments performed, and finally Section V concludes the paper.

II. RELATED WORKS

In order to prevent fraudulent transactions and detect credit card fraud, several methods have been proposed by researchers. A review of state-of-the-art related works is presented in the following.

Halvaiee & Akbari study a new model called the AIS-based fraud detection model (AFDM). They use the Immune System Inspired Algorithm (AIRS) to improve fraud detection accuracy. The presented results of their paper show that 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 [11].

Bahnsen et al. developed a transaction aggregation strategy and created a new set of features based on the periodic behaviour analysis of the transaction time by using the von Mises distribution. In addition, they propose a new cost-based criterion for evaluating credit card fraud detection's models and then, using a real credit card dataset, examine how different feature sets affect results. More precisely, they extend the transaction aggregation strategy to create new offers based on an analysis of the periodic behaviour of transactions [12].

Randhawa et al. study the application of machine learning algorithms to detect fraud in credit cards. They first use Naive Bayes, stochastic forest and decision trees, neural networks, linear regression (LR), and logistic regression, as well as support vector machine standard models, to evaluate the available datasets. Further, they propose a hybrid method by applying AdaBoost and majority voting. In addition, they add noise to the data samples for robustness evaluation. They perform experiments on publicly available datasets and show that majority voting is effective in detecting credit card fraud cases [6].

Porwal and Mukund propose an approach that uses clustering methods to detect outliers in a large dataset and is resistant

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

to changing patterns [13]. The idea behind their proposed approach is based on the assumption that the good behaviour of users does not change over time and that the data points that represent good behaviour have a consistent spatial signature under different groupings. They show that fraudulent behaviours can be detected by identifying the changes in this data. They show that the area under the precision-recall curve is better than ROC as an evaluation criterion [13].

The authors in [14], 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.

Itoo et al. use three different ratios of datasets and an oversampling method to deal with the problem of data imbalance. Authors use three machine learning algorithms: logistic regression, Naive Bayes, and K-nearest neighbor. The performance of the algorithms is measured based on accuracy, sensitivity, specificity, precision, F1-score, and area under the curve. They show that the logistic regression-based model outperforms the other commonly used fraud detection algorithms in the paper [15].

The authors in [16] propose a framework that combines the potential of meta-learning ensemble techniques and a cost-sensitive learning paradigm for fraud detection. They perform some evaluations, and the results obtained from classifying unseen data show that the cost-sensitive ensemble classifier has acceptable AUC value and is efficient as compared to the performances of ordinary ensemble classifiers.

Altyeb et al. propose an intelligent approach for detecting fraud in credit card transactions [17]. Their proposed Bayesian-based hyperparameter optimization algorithm is used to tune the parameters of a LightGBM. They perform experiments on publicly available credit card transaction datasets. These datasets consist of fraudulent and legitimate transactions. Their evaluation results are reported in terms of accuracy, area under the receiver operating characteristic curve (ROC-AUC), precision, and F1-score metrics.

Xiong et al. propose a learning-based approach to tackle the fraud detection problem. They use feature engineering techniques to boost the proposed model's performance. The model is trained and evaluated on the IEEE-CIS fraud dataset. Their experiments show that the model outperforms traditional machine-learning-based methods like Bayes and SVM on the used dataset [18].

Viram et al. evaluate the performance of Naive Bayes and voting classifier algorithms. They demonstrate that in terms of evaluated metrics, particularly accuracy, the voting classifier outperforms the Naive Bayes algorithm [19].

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 dataset. They evaluate five classification techniques and show that the supervised vector classifier and logistic regression classifier outperform other algorithms in an imbalanced dataset [20]. The summary of the literature review is presented in Fig. 1.

TABLE 1. 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 Fraudlent	0 or 1

III. PROPOSED APPROACH TO DETECTING CREDIT CARD FRAUD

The proposed framework for fraud detection is presented in Fig. 2. As this figure shows, we first apply the desired pre-processing on the data and further divide the data into two sections: training and testing, followed by performing Bayesian optimization on the training data to find the best hyperparameters 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. These steps are explained in detail as follows:

A. DATASET

In this paper, we use a real dataset so that the outcome of the proposed algorithm can be used in practice. We consider a dataset named "creditcard" 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. This dataset is available and can be accessed through https://www.kaggle.com/mlg-ulb/creditcardfraud.

This dataset contains only numerical input variables resulting from a principle component analysis (PCA) transformation. Unfortunately, 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.

B. DATA PRE-PROCESSING

As illustrated in Table 2, the total number of fraudulent transactions is significantly lower than the total number of

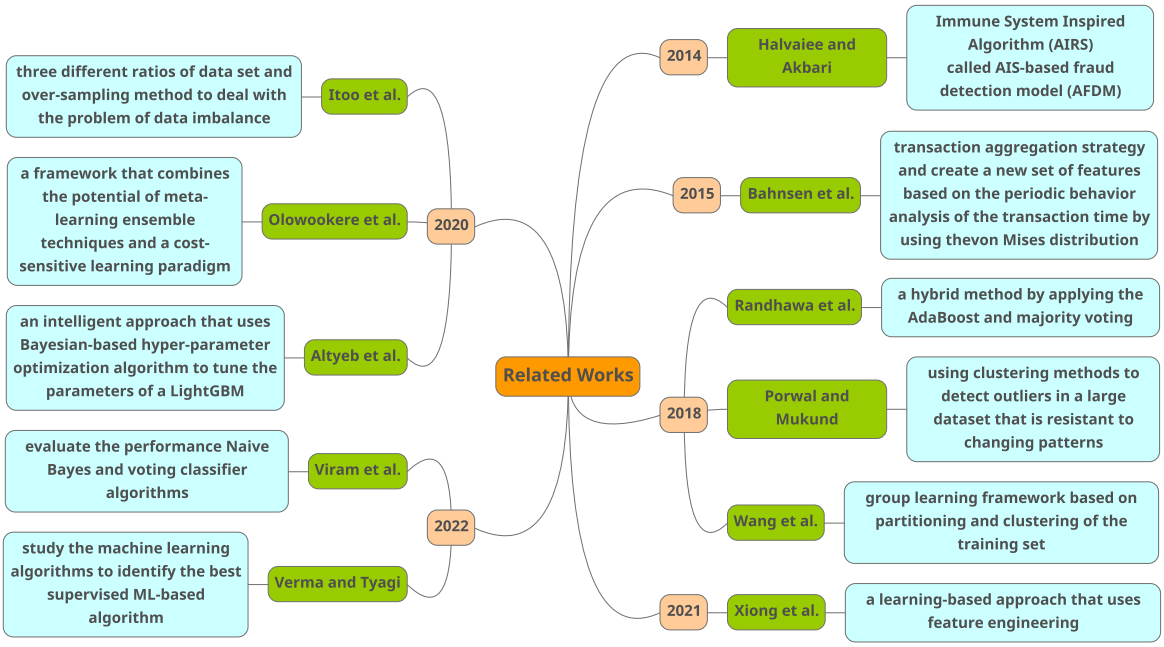

FIGURE 1. Summary of the related works on fraud detection in banking industry with machine learning techniques.

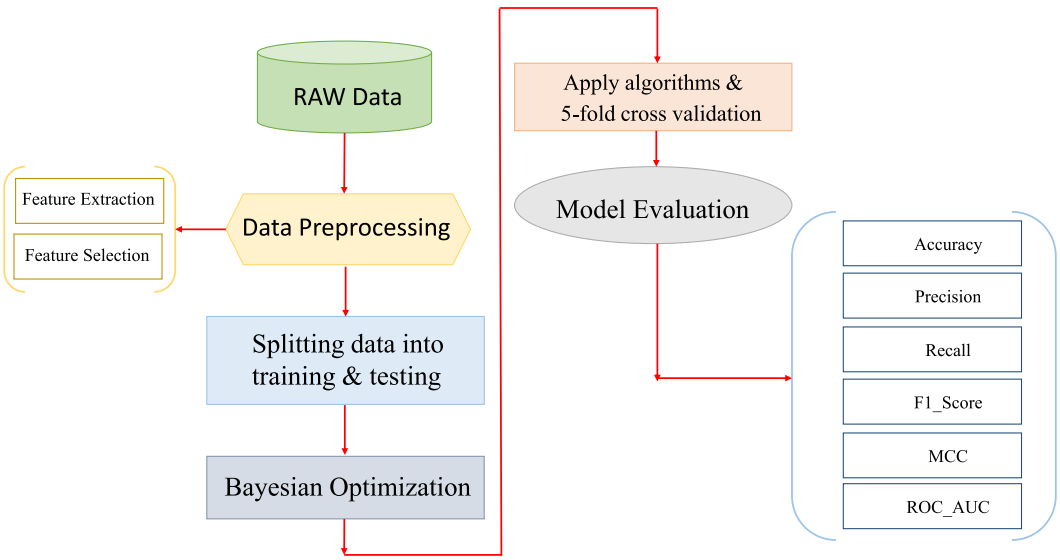

FIGURE 2. Proposed framework for credit card fraud detection.

legitimate transactions, indicating that the data distribution is unbalanced. In real datasets for credit card fraud detection, unbalanced data is expected. This data imbalance causes performance issues in machine learning algorithms, and having a class with the majority of the samples influences the evaluation results [6]. Therefore, in many studies, under-sampling and over-sampling methods are used to solve the data imbalance problem [15]. Using under-sampling methods leads to data loss [21]. 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 over sampling [5], [17], [22]. However, the SMOTE method causes an increase in the false-positive rate, which is not acceptable in banking

TABLE 2. 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%

for customer orientation. To solve this problem, in this study, we use class weight tuning hyperparameter to solve the mentioned disadvantages [5], [17], [22]. 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.

C. FEATURE EXTRACTION

The "time" feature includes the time (in seconds) elapsed between each transaction and the first transaction. To make the most of the feature, we expand it to extract the transaction hour feature, which gives us more information than the time feature itself.

D. FEATURE SELECTION

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 [23]. 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 [17], [24]. The information gain method has been proven to be computationally efficient and shows leading performance in terms of precision [17]. Therefore, we also consider the IG method for feature selection in the proposed framework. Figure 3 shows the diagram of the IG, and the top six features extracted by this method have been used to evaluate the proposed algorithm.

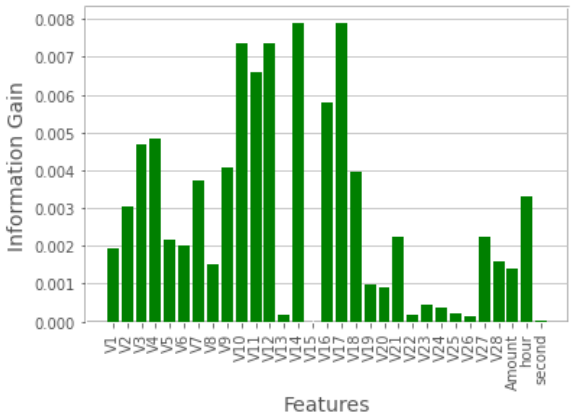

FIGURE 3. Feature importance diagram that shows the IG for the unknown features of the "creditcard" dataset. The top six features are used in evaluations.

E. 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. Recently, it was shown that the Bayesian method is capable of finding the optimised values in a much smaller number of training courses compared with evolutionary optimization methods [25], [26]. In this paper, we use the Bayesian optimization algorithm to tune the hyperparameters that lead to computational time reduction and performance improvement.

1) LOGISTIC REGRESSION

Logistic regression is a predictive analysis that finds out if two or more variables are related to each other. This method determines whether there is a relationship between one binary dependent variable and one or more ordinal, nominal, interval, or ratio-level independent variables [27].

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.

2) LightGBM

The LightGBM algorithm is built on the GBDT framework and aims to improve computational efficiency, particularly on big data prediction problems [28]. The high performance LightGBM algorithm can quickly handle large amounts of data, and the distributed processing of data [17]. In LightGBM, the histogram-based algorithm and trees' leaf-wise growth strategy with a maximum depth limit are 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.

3) XGBoost

eXtreme Gradient Boosting (XGBoost) has become a dominant algorithm in the field of applied machine learning. XGBoost is a type of decision tree algorithm with boosted gradients. It is preferred over other gradient boosting machines (GBMs) due to its fast execution speed, model performance, and memory resources [28]. 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 [28]. XGBoost employs a more regularised technique called ''formalization'' to control over-fitting and achieve better performance [29]. The tuned hyperparameters include learning rate, number of trees, and maximum tree depth, as well as applying weight to classes

4) CatBoost

Category Boosting (CatBoost) is a new gradient boosting algorithm proposed by Prokhorenkova et al. [29]. CatBoost is a competitive candidate in the realm of classifiers for highly unbalanced data. [30]. CatBoost machine learning algorithm is a particular type of Gradient boosting on the decision trees as it can handle categorical, ordered features, and the overfitting of the model is taken care of by Bayesian estimators [31]. CatBoost doesn't require extensive data training like other machine learning models and can be successfully applied to diverse types and formats of data [29], [30]. CatBoost has both CPU and GPU implementations, the GPU implementation allows for much faster training and is faster than both state-of-the-art open-source GBDT GPU implementations, XGBoost and LightGBM, on ensembles of similar sizes [32]. CatBoost uses a more efficient strategy hat reduces over-fitting and allows the use of the whole dataset for training. We perform a random permutation of the dataset, and also, for data imbalance problems, we use a class weight hyperparameter.

5) MAJORITY VOTING

Ensemble learning (EL), which is a type of machine learning, combines several classifiers, minimises 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 [27], [33].

$$\hat{y} = \text{argmax} \frac{1}{N_{\text{Classifiers}}} \sum_{Classifiers} (p_1, \ldots, p_n) \qquad (1)$$

6) DEEP LEARNING

Deep learning algorithms are a class of machine learning algorithms where multiple hidden layers are used to improve the outcome. 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. [34]. Deep learning is a generic term that refers to machine learning using a deep multi-layer artificial neural network (ANN). It is a biologically inspired model of human neurons, composed of multilevel hidden layers of nonlinear processing units, where each neuron is able to send data to a connected neuron within the hidden layers. These processing units discover intermediate representations in a hierarchical manner. The features discovered in one layer form the basis for the processing of the succeeding layer. In this way, deep learning algorithms learn intermediate concepts between raw input and target knowledge [34].

In this paper, we use a sequential model, which is a linear stack of layers to construct an artificial neural network model. Our model has a dense class, which is a very common layer and is often used. In the neural network, the activation function is used to increase the predictive power. This function divides input signals into output signals. We use the Relu activation function, and in the last layer, we use ''Sigmoid'', since our output is binary. The Sigmoid function generates values in a range of zero and one. In the ''Relu'' function, if the value x is smaller than or equal to zero, the output is zero. The function of the Relu activation function is in many ways similar to the function of our biological neurons.

Neural networks require initial weighting. 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 underfitting. 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. The details of our model are presented in Table 3.

Following Keras and with the help of the compile method and Adam's optimizer, we perform weight updates and use binary-cross entropy for the loss function that finalises the configuration of the learning and training process.

TABLE 3. Details of our deep learning model used in the paper are provided. The total parameters are set to 7593, and all are trainable.

Layer(Type)	Output Shape	Param No.
dense (Dense)	(None, 86)	2752
dense-1 (Dense)	(None, 44)	3828
dense-2 (Dense)	(None, 22)	990
dense-3 (Dense)	(None, 1)	23

F. EVALUATION METRICS

We apply a cross-validation test to evaluate the performance of the proposed model for credit card fraud detection. Similar to [6], [17], We use a stratified 5-fold validation test to obtain a reliable performance comparison in the unbalanced set. The dataset is divided randomly into five separate subsets of equal size, where the number of samples in each class is divided into equal proportions in each category. In all steps of validation, a single subset (20% of the dataset) is reserved as the validation data to test the performance of the proposed approach, while the remaining four subsets (80% of the dataset) are employed as the training data. We repeat this process five times until all subsets are used. The average performances of the five test subsets are calculated, and the final result is the performance of the proposed approach on a 5-fold cross-validation test.

To be fair in our comparisons, we use the common metrics for our evaluations, including accuracy, precision, recall, the Matthews correlation coefficient (MCC), the F1-score, and AUC diagrams. Positive numbers represent fraudulent transactions in our experiments, while negative numbers represent legitimate ones. True positive (TP) represents fraudulent transactions that have been classified as such. False positives (FP) indicate the number of legitimate transactions misclassified as fraudulent. The true negative (TN) represents legitimate transactions classified as legitimate, and the false negative (FN) indicates the misclassified fraudulent transactions as legitimate [15]. The mathematical expressions for the metrics used are given in Eq. (2) to Eq. (6).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + TN} \tag{3}$$

$$\text{Precision} = \frac{TP}{TP + TN + FP + FN} \tag{4}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{5}$$

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

Accuracy Accuracy quantifies the total performance of the classifier and is defined as the number of correct predictions made by the model. When dealing with data that isn't balanced, this criterion doesn't give good results because it also gives a high value if even one fraudulent transaction is found. **Recall** shows the efficiency of the classifier in detecting actual fraudulent transactions. **Precision** measures the reliability of the classifier and **F1-Score** is the harmonic average of recall and precision measures, that considers both false negatives and positives.

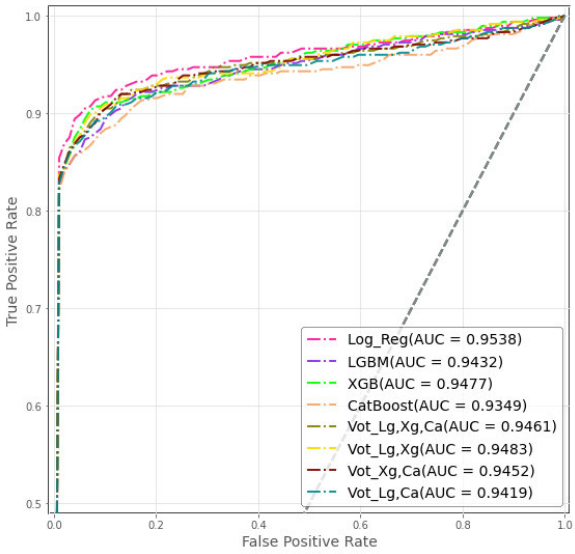

FIGURE 4. ROC_AUC curve.

ROC-AUC is a measure of separability that demonstrates the model's ability to differentiate between classes [15]. ROC-AUC is a graphical plot of the false positive rate (FPR) and the true positive rate (TPR) at different possible levels [17]. The area under the ROC curve is not a suitable criterion for evaluating fraud detection methods since it only considers positive values.

The precision and recall curves are commonly used to compare classifiers in terms of precision and recall. Usually, in this two-dimensional graph, the precision rate is plotted on the y-axis and the recall is plotted on the x-axis. There is no good way to describe the true and false positives and negatives using one indicator. One good solution is to use MCC, which measures the quality of a two-class problem, taking into account the true and false positives and negatives. It is a balanced measure, even when the classes are of different sizes [6].

IV. 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 5.

Most studies in the literature rely on AUC diagrams to evaluate performance. However, as can be seen from the ROC-AUC curve in Fig. 4, 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

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

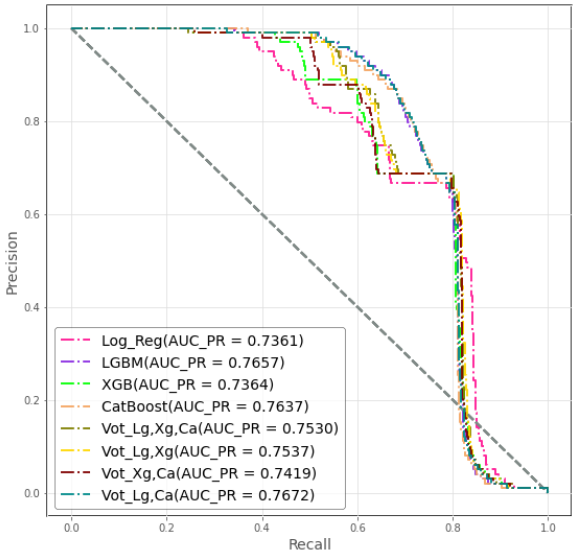

FIGURE 5. Precision_Recall curve.

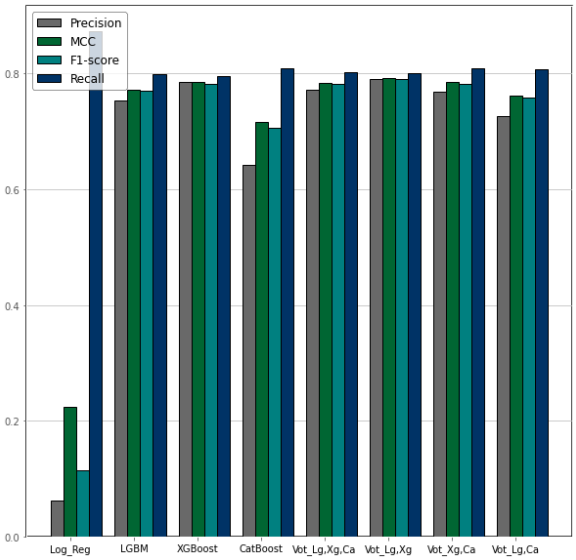

FIGURE 6. Performance comparing algorithms with different evaluation criteria.

irrelevant. According to the ROC-AUC Fig. 4, the logistic regression algorithm 0.9583 has the highest number of fraud detection, but it has the lowest value in other criteria.

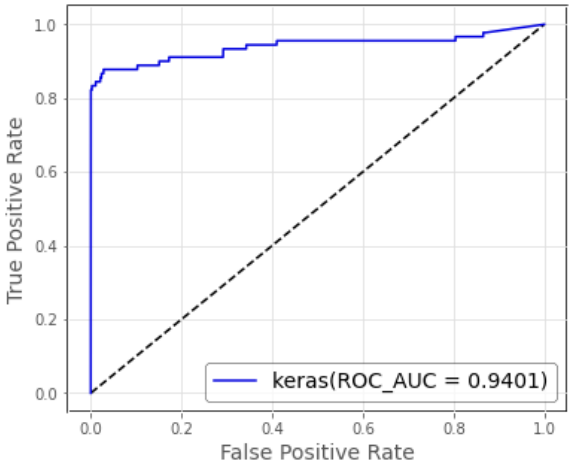

FIGURE 7. ROC curve of deep learning.

TABLE 5. 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

The precision-recall curve is illustrated in Fig. 5 and shows the system performance in a more precise manner compared with the ROC-AUC curve. However, the results cannot be cited because false negatives are far from the view of this diagram. As Fig. 5 shows, the highest value belongs to the combination of the CatBoost and LightGBM algorithms with a value of 0.7672, and the lowest value belongs to logistic regression and is 0.7361.

Comparing the precision, recall, and F1-score as well as the MCC, the algorithms used are shown in Fig. 6. The best performance is related to the combination of lightGBM and XGBoost algorithms, which have an MCC value of 0.79 and an F1-score of 0.79. In individual algorithms, XGBoost has the highest values.

According to the digits obtained in Table 5, deep learning has achieved better performance compared with individual algorithms and majority voting ensemble learning. The MCC and F1-score metrics have values of 0.8129 and 0.8132, respectively. The area under the ROC curve in the deep learning method is illustrated in Fig. 7 and shows a value of 0.9401.

TABLE 6. Performance comparison of the proposed approach and the method presented in [17].

Model	Accuracy	AUC	Recall	Precision	F1-score
Method presented in [17]	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

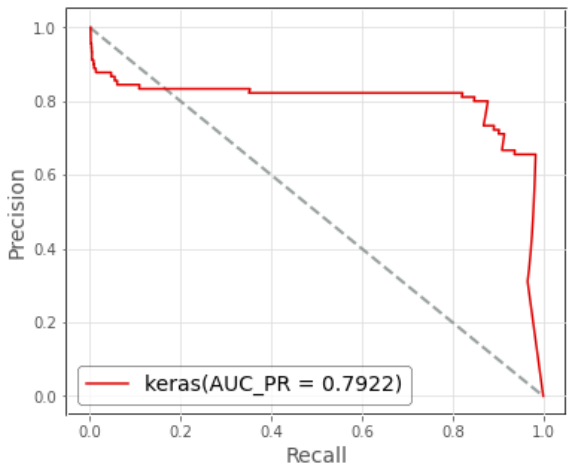

FIGURE 8. Precision- recall curve of deep learning.

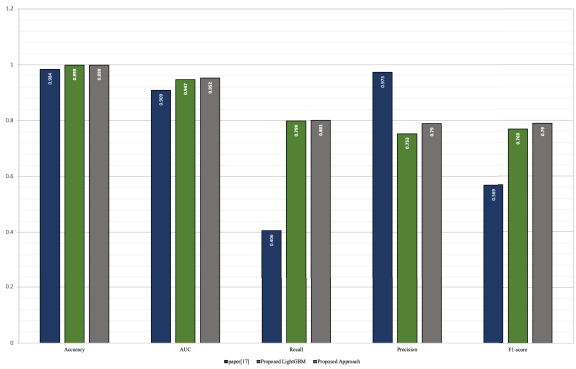

FIGURE 9. Performance comparison of the proposed approach with the paper [17] based on the different evaluation criteria.

The diagram of the Precision-Recall curve is shown in Fig. 8, and shows the value as 0.7922.

The evaluation results of the proposed approach using different pre-processing and class weight hyperparameter tuning to deal with the problem of data unbalance compared to the paper [17] are shown in Fig. 9. The results show improvement of both methods compared to the method presented in [17].

According to the Table 6, it is shown that the proposed methods outperform the intelligence method presented in [17] using common metrics and a public dataset.

V. 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. We used a publicly available ''credit card'' dataset with 28 features and 0.17 percent of the fraud data. We proposed two methods. In the proposed LightGBM, we used class weight tuning to choose the proper hyperparameters. We used the common evaluation metrics, including accuracy, precision, recall, F1-score, and AUC. Our experimental results showed that the proposed LightGBM method improved the fraud detection cases by 50% and the F1-score by 20% compared with the recently presented method in [17]. We improve the performance of the algorithm with the help of the majority voting algorithm. We also improved the criteria by using the deep learning method. The assurance of the results of MCC for unbalanced data proved that, compared to other criteria of evaluation, it's stronger. In this paper, by combining the LightGBM and XGBoost methods, we obtained 0.79 and 0.81 for the deep learning method. Using hyper parameters to address data unbalance compared to sampling methods, in addition to reducing memory and time needed to evaluate algorithms, also has better results.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, especially the hyperparameter number of trees. Also, due to hardware limitations in this study, the use of stronger and better hardware may bring better results that can ultimately be compared with the results of this study.

REFERENCES

[1] J. Nanduri, Y.-W. Liu, K. Yang, and Y. Jia, ''Ecommerce fraud detection through fraud islands and multi-layer machine learning model,'' in *Proc. Future Inf. Commun. Conf.*, in Advances in Information and Communication. San Francisco, CA, USA: Springer, 2020, pp. 556–570.
[2] I. Matloob, S. A. Khan, R. Rukaiya, M. A. K. Khattak, and A. Munir, ''A sequence mining-based novel architecture for detecting fraudulent transactions in healthcare systems,'' *IEEE Access*, vol. 10, pp. 48447–48463, 2022.
[3] H. Feng, ''Ensemble learning in credit card fraud detection using boosting methods,'' in *Proc. 2nd Int. Conf. Comput. Data Sci. (CDS)*, Jan. 2021, pp. 7–11.
[4] M. S. Delgosha, N. Hajiheydari, and S. M. Fahimi, ''Elucidation of big data analytics in banking: A four-stage delphi study,'' *J. Enterprise Inf. Manage.*, vol. 34, no. 6, pp. 1577–1596, Nov. 2021.
[5] M. Puh and L. Brkić, ''Detecting credit card fraud using selected machine learning algorithms,'' in *Proc. 42nd Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2019, pp. 1250–1255.
[6] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, ''Credit card fraud detection using AdaBoost and majority voting,'' *IEEE Access*, vol. 6, pp. 14277–14284, 2018.
[7] N. Kumaraswamy, M. K. Markey, T. Ekin, J. C. Barner, and K. Rascati, ''Healthcare fraud data mining methods: A look back and look ahead,'' *Perspectives Health Inf. Manag.*, vol. 19, no. 1, p. 1, 2022.
[8] E. F. Malik, K. W. Khaw, B. Belaton, W. P. Wong, and X. Chew, ''Credit card fraud detection using a new hybrid machine learning architecture,'' *Mathematics*, vol. 10, no. 9, p. 1480, Apr. 2022.
[9] K. Gupta, K. Singh, G. V. Singh, M. Hassan, G. Himani, and U. Sharma, ''Machine learning based credit card fraud detection—A review,'' in *Proc. Int. Conf. Appl. Artif. Intell. Comput. (ICAAIC)*, 2022, pp. 362–368.

[10] R. Almutairi, A. Godavarthi, A. R. Kotha, and E. Ceesay, "Analyzing credit card fraud detection based on machine learning models," in *Proc. IEEE Int. IoT, Electron. Mechatronics Conf. (IEMTRONICS)*, Jun. 2022, pp. 1–8.
[11] N. S. Halvaiee and M. K. Akbari, "A novel model for credit card fraud detection using artificial immune systems," *Appl. Soft Comput.*, vol. 24, pp. 40–49, Nov. 2014.
[12] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Expert Syst. Appl.*, vol. 51, pp. 134–142, Jun. 2016.
[13] U. Porwal and S. Mukund, "Credit card fraud detection in e-commerce: An outlier detection approach," 2018, *arXiv:1811.02196*.
[14] H. Wang, P. Zhu, X. Zou, and S. Qin, "An ensemble learning framework for credit card fraud detection based on training set partitioning and clustering," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Oct. 2018, pp. 94–98.
[15] F. Itoo, M. Meenakshi, and S. Singh, "Comparison and analysis of logistic regression, Naïve Bayes and knn machine learning algorithms for credit card fraud detection," *Int. J. Inf. Technol.*, vol. 13, no. 4, pp. 1503–1511, 2021.
[16] T. A. Olowookere and O. S. Adewale, "A framework for detecting credit card fraud with cost-sensitive meta-learning ensemble approach," *Sci. Afr.*, vol. 8, Jul. 2020, Art. no. e00464.
[17] 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.
[18] X. Kewei, B. Peng, Y. Jiang, and T. Lu, "A hybrid deep learning model for online fraud detection," in *Proc. IEEE Int. Conf. Consum. Electron. Comput. Eng. (ICCECE)*, Jan. 2021, pp. 431–434.
[19] T. Vairam, S. Sarathambekai, S. Bhavadharani, A. K. Dharshini, N. N. Sri, and T. Sen, "Evaluation of Naïve Bayes and voting classifier algorithm for credit card fraud detection," in *Proc. 8th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2022, pp. 602–608.
[20] P. Verma and P. Tyagi, "Analysis of supervised machine learning algorithms in the context of fraud detection," *ECS Trans.*, vol. 107, no. 1, p. 7189, 2022.
[21] J. Zou, J. Zhang, and P. Jiang, "Credit card fraud detection using autoencoder neural network," 2019, *arXiv:1908.11553*.
[22] D. Almhaithawi, A. Jafar, and M. Aljnidi, "Example-dependent cost-sensitive credit cards fraud detection using SMOTE and Bayes minimum risk," *Social Netw. Appl. Sci.*, vol. 2, no. 9, pp. 1–12, Sep. 2020.
[23] J. Cui, C. Yan, and C. Wang, "Learning transaction cohesiveness for online payment fraud detection," in *Proc. 2nd Int. Conf. Comput. Data Sci.*, Jan. 2021, pp. 1–5.
[24] M. Rakhshaninejad, M. Fathian, B. Amiri, and N. Yazdanjue, "An ensemble-based credit card fraud detection algorithm using an efficient voting strategy," *Comput. J.*, vol. 65, no. 8, pp. 1998–2015, Aug. 2022.
[25] A. H. Victoria and G. Maragatham, "Automatic tuning of hyperparameters using Bayesian optimization," *Evolving Syst.*, vol. 12, no. 1, pp. 217–223, Mar. 2021.
[26] H. Cho, Y. Kim, E. Lee, D. Choi, Y. Lee, and W. Rhee, "Basic enhancement strategies when using Bayesian optimization for hyperparameter tuning of deep neural networks," *IEEE Access*, vol. 8, pp. 52588–52608, 2020.
[27] F. N. Khan, A. H. Khan, and L. Israt, "Credit card fraud prediction and classification using deep neural network and ensemble learning," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, Jun. 2020, pp. 114–119.
[28] W. Liang, S. Luo, G. Zhao, and H. Wu, "Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms," *Mathematics*, vol. 8, no. 5, p. 765, May 2020.
[29] S. B. Jabeur, C. Gharib, S. Mefteh-Wali, and W. B. Arfi, "CatBoost model and artificial intelligence techniques for corporate failure prediction," *Technol. Forecasting Social Change*, vol. 166, May 2021, Art. no. 120658.
[30] J. Hancock and T. M. Khoshgoftaar, "Medicare fraud detection using CatBoost," in *Proc. IEEE 21st Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, Aug. 2020, pp. 97–103.
[31] B. Dhananjay and J. Sivaraman, "Analysis and classification of heart rate using CatBoost feature ranking model," *Biomed. Signal Process. Control*, vol. 68, Jul. 2021, Art. no. 102610.
[32] Y. Chen and X. Han, "CatBoost for fraud detection in financial transactions," in *Proc. IEEE Int. Conf. Consum. Electron. Comput. Eng. (ICCECE)*, Jan. 2021, pp. 176–179.
[33] A. Goyal and J. Khiari, "Diversity-aware weighted majority vote classifier for imbalanced data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.
[34] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, "Deep learning detecting fraud in credit card transactions," in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2018, pp. 129–134.

SEYEDEH KHADIJEH HASHEMI received the B.Sc. and M.Sc. degrees in computer engineering. She is currently a Former Student with the Department of Electrical and Computer Engineering, Kharazmi University. Her master's thesis has been performed on fraud detection for banking with machine learning techniques. Her research interest includes application of machine learning techniques, focusing on banking.

SEYEDEH LEILI MIRTAHERI is currently a Faculty Member with the Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran. She is researching next-generation high-performance computing systems and GPU computing. She has published more than 50 papers in credible conferences and journals. Her research interests include distributed and parallel systems, exascale computing, cluster computing, mathematics, and scientific computing. She worked on distributed systems and done several successful industrial experiments in these areas. She received an Exemplary Professor of Kharazmi University, in 2020, and also she received a Leading Young Researcher in Alborz Province, in 2020. She received the First Award of Inventions at National Science Foundation Invention Festival, in 2011, the Iran University of Science and Technology (IUST) Awards for Excellence in Researching, in 2009, the Second Level Reward of National Science Foundation in Ph.D., in 2009, the First Award for presenting "CSharifi: Kernel Level Cluster Management System Software," at the Khwarizmi Young Awards, in 2008, the Grant of Excellent Researcher of National Science Foundation, in 2008, and the Iranian Organization of Scientific and Industrial Research appreciation to cooperating and presenting "A Cluster Management System Software" at the Khwarizmi International Awards, in 2007.

SERGIO GRECO is currently a Full Professor with the Department of Informatics, Modeling, Electronics and System Engineering (DIMES), University of Calabria, Rende, Italy. He has written over 220 papers, including more than 60 journal papers in prestigious conferences and journals. His research interests include database theory, data integration and exchange, inconsistent data, incomplete data, data mining, knowledge representation, logic programming, and computational logic and argumentation theory.