Journal of Industrial and Systems Engineering
Vol. 14, No. 2, pp. 159-171
Spring (April) 2022

Improving the multilayer Perceptron neural network using teaching-learning optimization algorithm in detecting credit card fraud

Ehram Safari[1*] , Mozhdeh Peykari[2]

[1] *Faculty member of the Institute of Communication and Information Technology, Tehran, Iran*
[2]*Department of Information and Technology Engineering, Science and Research Branch of Azad University, Tehran, Iran*

e.safari@itrc.ac.ir, peykari@gmail.com

Abstract

Due to the necessity of electronic transactions with credit cards in this modern era and that fraudulent activity with credit cards are on the rise, the development of automated systems that can prevent such financial fraud is considered vital. This study presents a method for detecting credit card fraud by deploying a neural network that distinguishes between legitimate and illegitimate transactions and detects fraudulent activities with stolen physical credit cards. For this purpose, after collecting data in the preprocessing stage, cleaning and normalizing the data, the feature selection operation is performed using fisher discriminant analysis. After that, a multilayer perceptron (MLP) neural network is trained during the post-processing period using the teaching learning-based optimization algorithm (TLBO) to optimize credit card fraud detection. In this algorithm, local search (exploitation) is done using the teacher phase, and global searching(exploration) is done using the student phase. Moreover, the fisher discriminant analysis algorithm reduces within-class scattering. It increases between-class diffusion to increase classification accuracy and decrease the CPU time of the algorithm in the training phase. The latest available algorithms such as AdaBoost, Random Forest, CNN, and RNN are also compared with the proposed method. The results show that the proposed algorithm outperforms the mentioned algorithms regarding some standards criteria and CPU time.

Keywords: Classification, fraud detection, multilayer Perceptron neural network, teaching-Learning optimization algorithm

1- Introduction

One of the biggest threats to businesses today is credit card fraud. Therefore, understanding the mechanism of fraud is essential to counteract its effects. Scammers use a variety of methods to use and forge credit cards to commit such illegal acts(Kim et al., 2019). Fraud actually means accessing services, goods, and money without the permission of the owner (Singh and Jain, 2019). Different approaches are applied to improve the performance in this area. One of the most important approaches is the use of extracted data and neural networks (Lebichot et al., 2019).

*Corresponding author
ISSN: 1735-8272,

The quick growth of electronic marketing has led to a substantial increase in using credit cards for web-based shopping, and thus fraud in credit cards has blown up
(Roy et al., 2018). The discovery of financial crime is an emerging field. Criminals try to achieve their goals by committing illegal acts. In addition, it is clear that unknown aspects of smart crime detection have not yet been explored.

In this study, we intend to classify and identify fraudulent activities in credit cards using MPL neural network trained by TLBO evolutionary algorithm. To solve this problem, evolutionary algorithms are used that have operators to exit the local optimal points. In meta-heuristic algorithms, there are parameters that must be optimally determined, and the performance of the algorithm depends on the parameter setting. The TLBO algorithm performs local search operation (exploitation) using the teacher phase and the global search operation (exploration) using the student phase. There is also no parameter to be set in this algorithm. These factors have led to the selection of this algorithm in this research. Also, during training the MLP neural network by the TLBO algorithm, while determining the optimal values of weights and biases, the optimal values of activation functions are also determined in both the hidden and output layers. In the literature, it has been observed that feature selection has not been studied. In this research, the fisher discriminant analysis algorithm is used to choose the feature and reduce the input space. This algorithm uses linear mapping to transfer data to a new space and select features that have low within-class variance and high between-class variance to increase fraud detection accuracy. In the research literature, it was observed that in the preprocessing stage, data cleaning with record removal and normalization have been less common. Therefore, normalization is one of the operations performed in this research in order to unify the effect of inputs, and in order to clean the data in this research, the central statistical average is used to substitute the missing value with a value that most values are around it. However, there has not been any research contributed in this area using our proposed method. As fraudsters are finding new ways to commit a crime, it is of paramount importance to examine new approaches with the intention to find a more precise way to prevent these illegal activities, which lead to disastrous money losses in electronic marketing.

Xuan et al. (2018) took into account two types of random forests to determine the behavioral characteristics of safe and unsafe transactions. Furthermore, they made a comparison between proposed algorithms with an available basic classification. Randhava et al. (2019) proposed several machine learning algorithms to identify credit card fraud. For the first time, they used some typical models. Then, the combined methods used by AdaBoost and most voting methods are used. Positive empirical findings show that the majority voting approach has high accuracy in determining credit card fraud cases. Zhang et al. (2019) devised a new method in feature engineering to detect credit card fraud with deep learning. Examination results show that the method is a useful and practical framework for detecting credit card fraud. Dubai et al. (2020) discovered credit card fraud by deploying an artificial neural network together with post-propagation. It gives better results than comparing methods. It will be very practical in the future because the proposed model also provides real-time results and predicted results of customer transactions. Taha et al. (2020) presented a smart method for determining fraud in credit card purchases by deploying an optimal gradient boosting machine. In the suggested method, a Bayesian-based hyperparameter optimizing algorithm is adapted to adjust the arguments of the optimal gradient boosting machine. In comparison, the method works better than other methods in terms of accuracy (98.4%), accuracy (92.88%), sensitivity (97.34%), and harmonic mean (95.56%).) was obtained. Giannini et al. (2020) managed a set of credit card fraud diagnosis rules with gamification. The classification consists of two steps: In the first step, the system quickly determines whether to authorize the transaction. Next, the system performs a slower examination grounded on a bigger data field. Shakur et al. (2019) deployed some machine learning methods to predict fraudulent and non-fraudulent transactions through classification. Normalization and analysis of the main elements are used in preprocessing the data set. The classification accuracy is 95%. Yee et al. (2018) detected credit card fraud by utilizing machine learning techniques such as data mining. In this article, a classification based on monitoring with Bayesian network classifiers called k2 and simple Bayesian logistics and J48 is used. After preprocessing the data set using normalization and principal component analysis, most classifications reached 95% accuracy. Save et al. (2017) proposed a system that detects fraudulent activities in credit card transactions by utilizing a decision tree with combining Lohan and Hunt algorithms. The results illustrate the high accuracy of the proposed method in detecting credit card fraud. Fu et al. (2016) detected credit card cheating based on a convolutional neural network. The framework was proposed to capture inherent patterns of trained fraudulent behaviors from marked data. Transaction data is provided by a feature matrix, on which a convolutional neural network is used to identify a series of latent patterns per sample. Examinations with large transactions of a large commercial bank in the real world show its higher efficiency in comparison with some advanced methods. Evaluation of the available literature shows that there are no references considering the assumption of the proposed algorithm. Table 1 shows Summary of Literature review.

Table 1. Research literature summary

Method	Disadvantages	Reference
Random Forest	Use of weak classifier, non-normalization of data	Xuan et al. (2018)
AdaBoost	Computational complexity in classification, lack of feature selection	Randhava et al. (2019)
Deep Learning	Use record deletion in data wiping	Zhang et al. (2019)
Backpropagation Neural Network	Use of classical algorithms in education	Dubai et al. (2020)
Gradient Boosting	The complexity of the input space due to the lack of feature selection	Taha et al. (2020)
Gamification	Use the above rules in classification	Giannini et al. (2020)
Artificial Neural Network	Select activation functions by trial and error	Shakur et al. (2019)
Bayesian, Logistics and J48	High computational complexity in teaching different models	Yee et al. (2018)
Decision Tree	Unable to transfer data from one directory to another	Save et al. (2017)
Convolutional Neural Network	Use record deletion in data wiping	Fu et al. (2016)

2- Methodology

According to figure 1, this part expresses all the steps of the research, which include data collection, data cleaning, data normalization as well as feature selection in the preprocessing stage. After that, the identification process of credit card fraud is implemented by MLP neural network, trained by the optimization algorithm based on teaching and learning.

2-1- Data collection

The data set used in this study was provided by a Kaggle engaged in the competition. It contains nearly 284808 anonymized credit card transactions labeled as fraudulent or genuine.

2-2- Data cleaning

In many real-world data mining applications, even with large amounts of data and adequate storage space, some data may be lost in existing samples. But the problem starts with the fact that missing values cannot be ignored for small data sets. One solution is to replace and clear the missing values with fixed values. In this research, the mean value in each property is utilized for the missing values. In other words, the mean is calculated based on the available values for each property and is substituted in the samples without value.

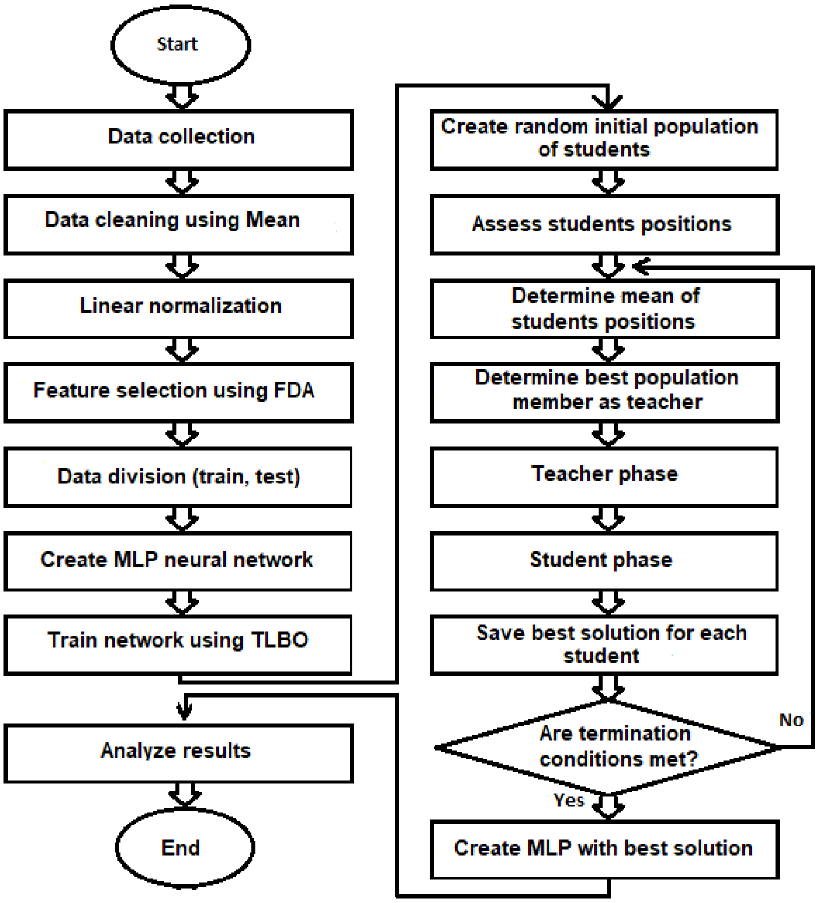

Fig 1. Proposed approach flowchart

2-3- Data normalization

Due to the fact that the range of properties changes are not the same, as well as diverse variable units, larger values impact more substantially on the methods used, which does not necessarily mean that they are more important. To solve this problem, data normalization is used. In this study, with linear normalization, the data are normalized to the range of [-1,1] (Bard, 2018)

$$X = 2 \times \frac{x - \min(x)}{\max(x) - \min(x)} - 1 \quad (1)$$

min (x) is the minimum input vector x. max (x) is the maximum input vector x. X is normalized.

2-4- Feature selection

Fisher's discriminant analysis is a statistical approach used in machine learning as well as pattern recognition to discover the linear combination of properties that best divides classes of objects. Fisher's discriminant analysis is very similar to variance analysis and regression analysis; in all three statistical methods, the dependent variable is used as a combination of the rest. Fisher's discriminant analysis is also similar to principal component analysis and factor analysis; both of these statistical methods are used to linearly combine variables in a way that best describes the data. A major application of both methods is to reduce the number of data dimensions. However, these methods have major differences: in fisher's discriminant analysis, class differences are modeled, while in principal component analysis, class differences are ignored (Chowdhury et al., 2018). Fisher discriminant analysis algorithm maps data from the input space to a new area to reduce within-class scatter and increase between-class scatter to improve accuracy in the classification stage. In this research, using Fisher's discriminant analysis algorithm, among the 31 features in the database, 18 features have been chosen for more accurate classifying due to their high dispersion.

2-5- Using MLP trained with TLBO

Neural networks emerged as an applied advancement that has been utilized successfully in a variety of fields. The most important benefit of neural networks is adaptability, independent organization, and instantaneous processing. The framework of these networks consists of three layers: input, middle, and output, each of which includes multiple processing elements linked to the neurons in the next layer in a weighted way. The quantity of elements in the first and last layers is dependent on the quantity variables. Nevertheless, the decision about the number of elements in the middle layer is made based on trial and error, which in our case, is usually the best solution (Vang-Mata, 2020). In neural networks, the neurons of every layer are linked to the whole elements of the previous layer via a directional connection. Based on the weight given to each connection, each neuron's effect is determined. An activation function calculates the neurons' output by getting these weighted values. Sigmoid, as well as linear functions, are widely utilized for MLP networks that are progressive. In this research, the hyperbolic tangent is utilized in the middle layer, and the hardlims function is utilized in the output. The linking network neurons weights are selected through the training process. Smart optimization approaches are amongst the most beneficial and advanced ones that can achieve optimized weights in the neural network (Rao and Patel, 2012).

The TLBO algorithm based on the teaching and learning of a class was first introduced by Rao et al. in 2012. The TLBO algorithm utilizes the ability of students in the classroom and teacher training to students to improve the educational level of the classroom. The main elements of TLBO are the teacher and the student, which form two important and principled parts of this algorithm. The output of the algorithm is the same as the student's grades and their level of knowledge, and the quality and ability of the teacher in this field are very effective. Therefore, the teacher in each class selects the best student in that class so that he or she can improve their grades with the guidance of other students. This process is followed in the teacher phase. Also, students try to improve their grades by learning from each other, which is followed in the student learning phase. The TLBO algorithm is a modern optimization algorithm grounded on the population of the class. The optimization procedure performed on the class population can be divided into two sections: 1. Teacher phase, 2. Student phase.
These phases are described as follows (Chowdhury et al., 2018):

- **Teacher phase**

This phase is the beginning section of the algorithm in which students attempt to develop their knowledge level and grades depending on the level of information and knowledge of the teacher. During this training process, the teacher tries to utilize all their abilities to improve the result of the average class, i.e., Meank, towards their knowledge, i.e., $Teacher^k$. As a result, this difference in the level of knowledge between the class average and the teacher is formulated in the form of relation (2).

$$Difference\_Mean^k = Teacher^k - TF^k * Mean^k \tag{2}$$

In relation (2), TF^k is the learning coefficient that controls the amount of average movement towards the teacher and its value is selected randomly 1 or 2 based on the relation round ((1 + rand (0)). Round is a function used to round numbers. According to equation (2), each student updates their position based on equation (3).

$$X_{new}^k = X_{old}^k + rand(.) \times Diference\_Mean^k \tag{3}$$

Where the new and old positions of the students are k, respectively, if the newly generated answer has a better objective function from the point of view of problem optimization, the previous result is replaced with it. If not, the same previous answer is preserved in the population. It is important to note that the population output from the starting phase, i.e., the teacher phase, is the population input for the second phase, i.e., the student phase (Rao and Patel, 2012).

• **Student phase**

This phase is the second section of the TLBO optimization process, in which students improve their knowledge level and information based on interactions and compromises between themselves. Each student randomly selects another student and changes knowledge level based on equation (4).

$$X_{new} = \begin{cases} X_i + rand(.) \times (X_i - X_j) & if\ f(X_i) < f(X_j)\ \ i \neq j \\ X_i + rand(.) \times (X_j - X_i) & else \end{cases} \tag{4}$$

If this change in knowledge level improves the student's score, this position replaces the previous position otherwise the previous position remains in the search space. It should be noted that the population output from the second phase, ie the student, is considered as the population input for the next iteration.

As their name implies, combined models combine computational intelligence solutions that try to use each other's capabilities to solve problems and improve solutions. Different solutions are usually integrated or used in this type of system. Typically, using one approach within another system causes these systems to cover each other's weaknesses, increase strengths, or have both capabilities. These systems include neuro-evolutionary systems. The remarkable combination of the power of neural systems in estimating the high speed of evolutionary algorithms is in calculating the optimal parameters in neural networks. For this purpose, training a MLP neural network by an optimization algorithm grounded on teaching and learning is described. The MLP neural network consists of 6 hidden layers with 150 neurons. The activation function used in the hidden layer is transit, while the activation function utilized in the output layer is hardlims, which are chosen by trial and error. Firstly, an MLP neural network is constructed using random weights and biases. Next, the weights and biases are passed as decision variables to the training and learning-based optimization algorithm, which consists of two stages of initial preparation and repetition stage with 100 iteration numbers. The following steps are explained in detail for MLP neural network training.

• **The initial preparation stage in the TLBO**

At this stage, a population of 500 students is created that each student consists of two parts: decision variables (weights and biases of MLP) and objective function (cost function in this case). Decision variables are considered MLP neural network weights for each student, and the mean squared error per instructional data is considered as the objective function. Figure 2 demonstrates a student position, which is MLP weights. At this stage, the position of all students is randomly assigned in the range [-1, 1] and the average neural network error squares are calculated for each member of the population.

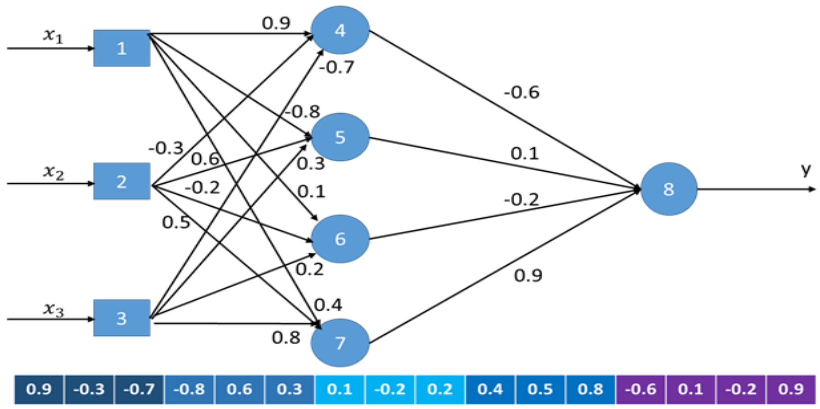

Fig 2. MLP weights as student's position

• **Iteration step in TLBO Algorithm**

Determining the teacher: In this step, the best student is selected as a teacher for the objective function value. The teacher is a member of the student population that has the lowest mean square error.

Teaching phase (teacher phase): In this step, the new position of each student is calculated grounded on the present positions of the student, teacher, and average. The teacher phase changes the values of the weights in the MLP neural network. Figure 3 shows the student position before and after the teacher phase. The new position of the student is evaluated, i.e., the training inputs are applied to it, the training outputs are calculated, and the mean square error is calculated and considered as the student's new objective function value. It is saved if the student's mean square error in the new position is better than the previous position. Otherwise, it is discarded.

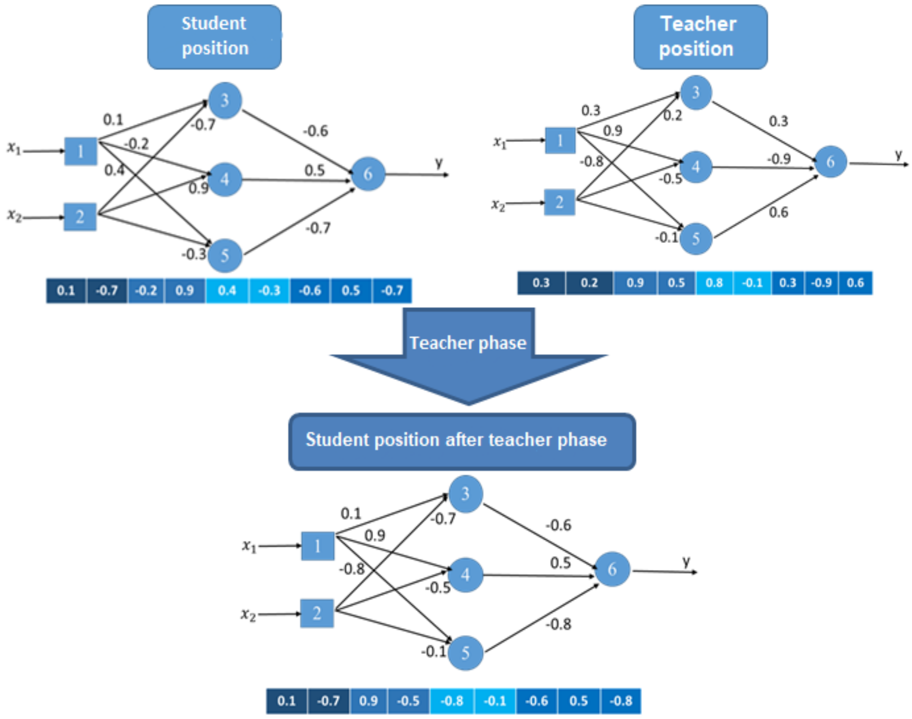

Fig 3. Student’s new position after teacing phase

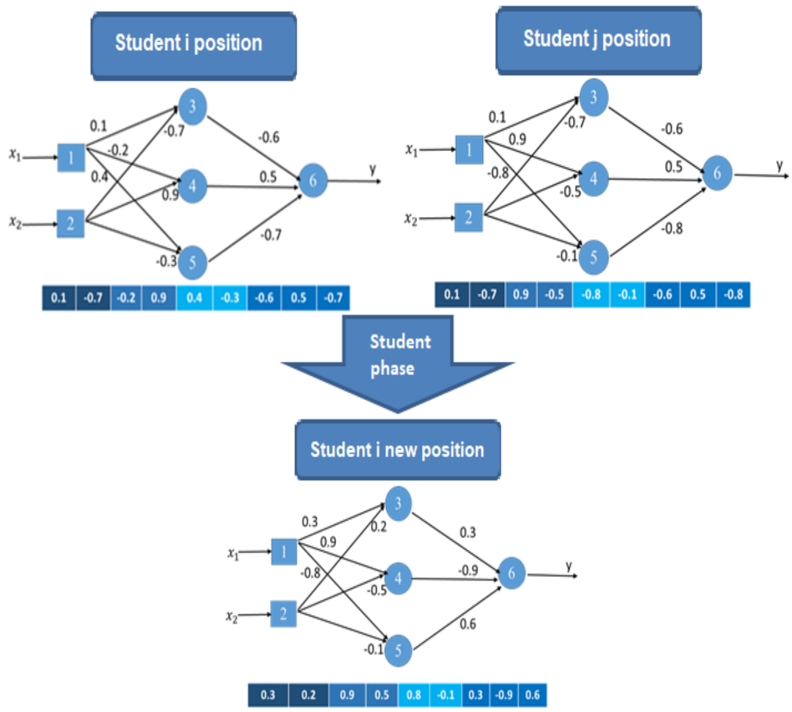

Fig 4. Student’s new position after student phase

Student phase: In this step, the new position of each student is calculated grounded on the present positions of the student and another student who is randomly selected. The student phase changes the values of the weights in the MLP neural network. Figure 4 shows the student's position before and after the learning phase. The new position of the student is evaluated, i.e. the training inputs are applied to it and the training outputs are calculated, and the mean square error is calculated and considered as the student's new objective function value. If the student's mean square error in the new position is better than the previous position, it is saved, otherwise, it is skipped. The above operation is repeated till the terminating conditions are met. The output of the optimization algorithm depends on the student's teaching and learning, which has the best values of weights in the MLP neural network so that the mean square error is the lowest possible.

3- Simulation results

This part expresses the outcome of the proposed model and other algorithms in identifying credit card fraud is reviewed, so first, we discuss the results of using a trained MLP neural network with a training-based optimization algorithm. All simulation steps have been performed in MATLAB 2020 software environment.

The data set used in this study was provided by a Kaggle engaged in the competition. It contains nearly 284808 anonymized credit card transactions labeled as fraudulent or genuine. The information provided includes 31 anonymized features. The response variable has been pre-labeled as 1 for fraudulent transactions and 0 for non-fraudulent transactions. There is a significant class imbalance problem associated with our data set. Only 0.14% of the transactions are classified as fraudulent compared with 99.86% classified as legitimate. 80% of the data is utilized in model training and 20% of the data is employed in model testing.

Since the dataset travails from class imbalance, the legitimate transactions are under-sampled at the account level. We consider there might be a temporal component in the data. Consequently, we must ensure that each account we sample includes all its transactions present in the training set. This implicated separating the dataset into fraudulent and non-fraudulent datasets and extracting unique account numbers from the fraudulent dataset. After that, we randomly sample non-fraudulent transactions from the non-fraudulent dataset for the extracted account numbers. The sampling ratio is chosen is10:1 (Non-fraudulent: fraudulent), which has been demonstrated to be excellent for credit card fraud detection. Ultimately, we utilize one-hot encoding to describe the categorical variables.

3-1- Results of using MLP trained with TLBO

In this research, to test the introduced classifications and the proposed model, the standard performance criteria used in most research are used. After classification training by the training database, test data with N samples in which N_P represents positive samples and N_N represents negative samples ($N=N_P+N_N$) constitute the classifier's input. After the classifier performs the classification operation, TP demonstrates the figure of positive transactions that the classifier also classifies as a positive transaction, FP illustrates the number of positive transactions that the classifier has mistakenly considered as a positive example, or in other words, $N_P=TP+FP$. Similarly, TN demonstrates the negative samples that the classifier classifies as negative, and FN reflects the number of negative transactions that the classifier mistakenly classifies as negative. In other words, $N_N=TN+FN$. Using the introduced symbols, critical performance criteria are introduced in table 2 (Bhatia, 2019).

Table 2. Some standard performance criteria

Name	Formula	Description
Recall,Sensitivity	$\frac{TP}{TP+FN}$	the ratio of positive transactions that are validly classified as positive samples
Specificity	$\frac{TN}{TN+FP}$	the ratio of negative transactions that are validly classified as negative samples
Precision	$\frac{TP}{TP+FP}$	the percentage of transactions that are positively classified to the total figure of positive transactions classified by the classifier
Negative Predictive Value	$\frac{TN}{TN+FN}$	the percentage of negatively classified transactions to the total figure of negatively classified transactions
Accuracy	$\frac{TP+TN}{TP+TN+FN+FP}$	the percentage of samples that are correctly classified

In this section, the results of identifying credit card fraud by the MLP neural network trained by the training-based optimization algorithm for training, test, and the whole data are discussed. The considered architecture contained six hidden layers, each with 150 nodes and the activation function in the hidden layer is hyperbolic tangent. In contrast, the activation function in the output layer is hardlims. This model is trained using a proposed TLBO algorithm. In the training-based optimization algorithm, the population size is 50, and the maximum frequency of the algorithm is 500. Table 3 shows the standard efficiency criteria for detecting credit card fraud by a trained MLP neural network with a TLBO algorithm for training, experimental, and all data.

Table 3. Standard efficiency criteria in MLP neural network improved by TLBO algorithm in credit card fraud detection

Data	Recall	Specificity	Precision	Negative Predictive Value	Accuracy
Train Data	97.5	99.8	99.7	97.6	**98.6**
Test Data	94	98.8	99.1	92.2	**96**
All Data	96.7	99.6	99.6	96.6	**98.1**

Figure 5 shows the regression diagram for training, test, and all data in a trained MLP with a training-based optimization algorithm. The regression coefficient is 0.973 for training data, 0.92 for test data, and 0.962 for total data. The closer the regression coefficient is to one, the more significant the correlation between the target and model outputs and the lower the detection error.

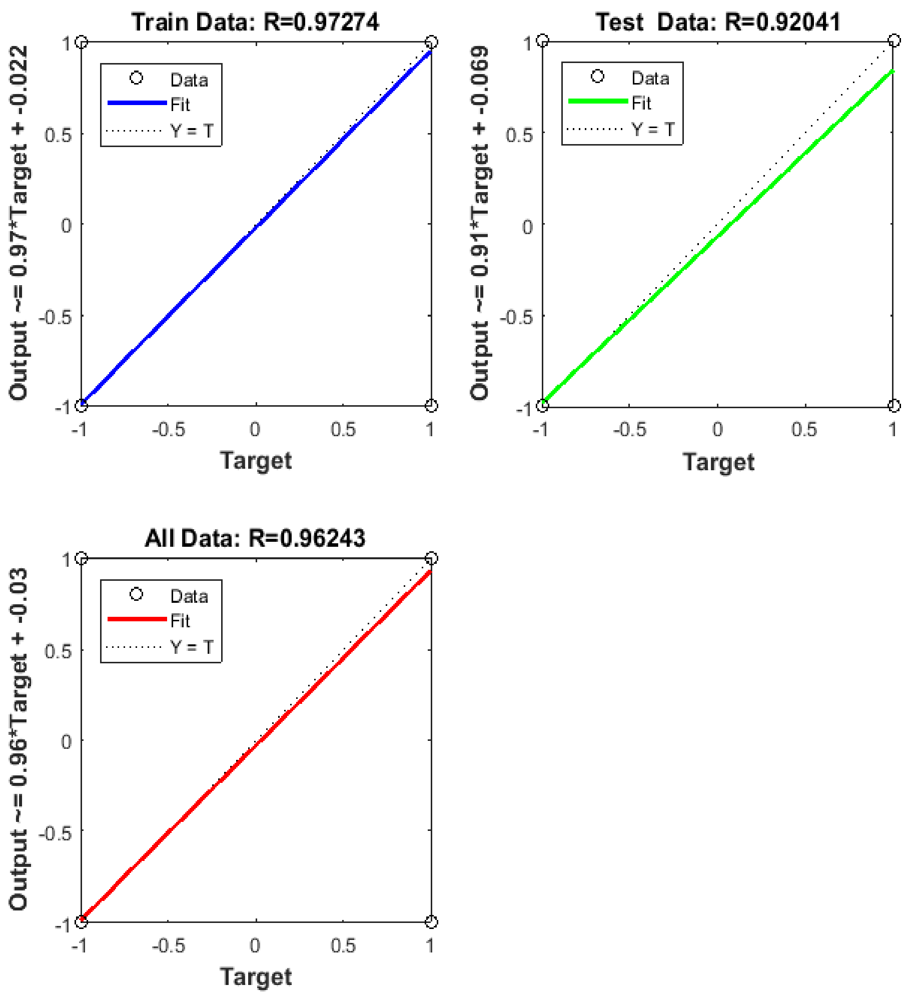

Fig 5. Regression diagram for different types of data in MLP neural network trained with TLBO algorithm

3-2- Comparing the proposed approach results

In table 4, the proposed approach is compared to Adabost, Random Forest, DP, CNN and RNN algorithms (Rout, 2021) in standard efficiency criteria such as recall, specificity, precision, negative predictive value and accuracy. Table 2 indicates that the proposed performance is superior to that of Adabost, Random Forest, DP, and RNN algorithms and is the same as that of RNN. Table 3 shows the CPU time for the proposed algorithm, AdaBoost, Random Forest, DP, CNN, and RNN algorithms in minutes. As shown in E 5, the CPU time of the training phase using the proposed algorithm is about 180 minutes, which is even better than that of Adabost, Random Forest, DP, CNN, and RNN algorithms. As a result, it can be said that the run time of the proposed algorithm is one-quarter of that of the RNN algorithm, which has a similar performance. The importance of this issue becomes more and more when the number of data is very high.

Since the proposed method escape from the local optimal solution due to the use of the TLBO algorithm, it works better than DP and RNN method, which uses gradient descent algorithms. Compared to other methods such as AdaBoost and random forest, the proposed algorithm is one of the deep learning methods and has better performance. Because the proposed method uses a metaheuristic algorithm for learning, its convergence speed is much higher than the gradient descent algorithms.

Table 4. Comparison of results

	MLP+TLBO	Adaboost	Random Forest	DP	CNN	RNN
Accuracy	98.1	97.8	97.1	98	98.1	98
Negative Predictive Value	96.6	95.3	94.7		96.7	96.4
Precision	99.6	98.8	98.5	99.4	99.6	99.5
Specificity	99.6	99.1	98.8	99.3	99.7	99.4
Recall	96.7	95.4	95.2	96.3	96.8	96.6

Table 5. CPU time of different algorithm

	MLP+TLBO	Adaboost	Random Forest	DP	CNN	RNN
CPU time	180min	250 min	240 min	800 min	850 min	950 min

4- Conclusion

This study has provided a system for detecting credit card fraud by deploying a neural network that distinguishes between legitimate and illegitimate transactions and detects fraudulent activities with stolen physical credit cards. For this purpose, after collecting data in the preprocessing stage, cleaning and normalizing the data, the feature selection operation is performed using Fisher Discriminant Analysis. After that, during the post-processing period, MLP neural network, which contains six hidden layers, each with 150 nodes and the activation function of the neurons in the hidden layer is hyperbolic tangent while the activation function in the output layer is hardlims, is trained by making use of TLBO to optimize credit card fraud detection. This research's innovation is deploying TLBO in training MLP neural network. In this algorithm, local search (exploitation) is done using the teacher phase and global search (exploration) is done using the Student Phase. Moreover, the Fisher Discriminant Analysis algorithm reduces within-class scattering and increases between-class scattering to increase classification accuracy.

The data set used in this study was provided by a Kaggle engaged in the competition. It contains details of nearly 284808 anonymized credit card transactions labeled as fraudulent or genuine. There is a significant class imbalance problem associated with our data set. Only 0.14% of the transactions were classified as fraudulent compared with 99.86% classified as legitimate. We use a sampling technique to reduce the ratio chosen to10:1 (Non-fraudulent: fraudulent), which has been demonstrated to be excellent for credit card fraud detection. After selecting18 features were selected from 31 features, using TLBO the MLP neural network was trained. The results were compared to the latest available algorithms such as Adaboost, Random Forest, CNN, and RNN. Results indicate that the proposed performance is superior to that of AdaBoost, Random Forest, DP, and RNN algorithms and is the same as that of RNN.

Furthermore, it was shown that the CPU time of the training phase using the proposed algorithm is about 180 minutes, which is even better than that of AdaBoost, Random Forest, DP, CNN, and RNN algorithms. As a result, it can be said that the run time of the proposed algorithm is one-quarter of that of the RNN algorithm, which has a similar performance in terms of some standard efficiency criteria such as recall, specificity, precision, negative predictive value, and accuracy. The importance of this issue becomes more and more when the number of data is very high.

References

Bhatia, P. (2019). Data mining and data warehousing: principles and practical techniques. Cambridge University Press.

Chowdhury, A., Yang, J., & Drineas, P. (2018). Randomized Iterative Algorithms for Fisher Discriminant Analysis. arXiv preprint arXiv:1809.03045.

Draghici, S. (2016). Statistics and data analysis for microarrays using R and bioconductor. CRC Press.

Dubey, S. C., Mundhe, K. S., & Kadam, A. A. (2020, May). Credit Card Fraud Detection using Artificial Neural Network and BackPropagation. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 268-273). IEEE

Fu, K., Cheng, D., Tu, Y., & Zhang, L. (2016, October). Credit card fraud detection using convolutional neural networks. In International conference on neural information processing (pp. 483-490). Springer, Cham.

Gianini, G., Fossi, L. G., Mio, C., Caelen, O., Brunie, L., & Damiani, E. (2020). Managing a pool of rules for credit card fraud detection by a Game Theory based approach. Future Generation Computer Systems, 102, 549-561.

Kim, E., Lee, J., Shin, H., Yang, H., Cho, S., Nam, S. K., ... & Kim, J. I. (2019). Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning. Expert Systems with Applications, 128, 214-224.

Lebichot, B., Borgne, Y. A. L., He-Guelton, L., Oblé, F., & Bontempi, G. (2019, April). Deep-learning domain adaptation techniques for credit cards fraud detection. In INNS Big Data and Deep Learning conference (pp. 78-88). Springer, Cham.

Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2018). Credit card fraud detection using AdaBoost and majority voting. IEEE access, 6, 14277-14284.

Rao, R., & Patel, V. (2012). An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. International journal of industrial engineering computations, 3(4), 535-560.

Rout, M. (2021). Analysis and comparison of credit card fraud detection using machine learning. In Advances in electronics, communication and computing (pp. 33-40). Springer, Singapore.

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

Save, P., Tiwarekar, P., Jain, K. N., & Mahyavanshi, N. (2017). A novel idea for credit card fraud detection using decision tree. International Journal of Computer Applications, 161(13).

Shukur, H. A., & Kurnaz, S. (2019). Credit card fraud detection using machine learning methodology. International Journal of Computer Science and Mobile Computing, 8(3), 257-260.

Singh, A., & Jain, A. (2019). Adaptive credit card fraud detection techniques based on feature selection method. In Advances in computer communication and computational sciences (pp. 167-178). Springer, Singapore.

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

Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., & Jiang, C. (2018, March). Random forest for credit card fraud detection. In 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC) (pp. 1-6). IEEE.

Yang, X. S. (2019). Introduction to algorithms for data mining and machine learning. Academic press.

Yee, O. S., Sagadevan, S., & Malim, N. H. A. H. (2018). Credit card fraud detection using machine learning as data mining technique. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(1-4), 23-27.

Zhang, X., Han, Y., Xu, W., & Wang, Q. (2019). HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. Information Sciences.

Vang-Mata, R. (2020). Multilayer Perceptrons: Theory and Applications. Computer Science, Technology and Applications Series, 153.

West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: a comprehensive review. Computers & Security, 57, 47-66.