

Fraud Detection of Credit Cards Using Neuro-fuzzy Approach Based on TLBO and PSO Algorithms

Maryam Ghodsi ^a, Mohammad Saniee Abadeh ^{b,*}

^a Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

^b Faculty of Computer and Electrical Engineering, Tarbiyat Modarres University, Tehran, Iran

Received 14 September 2016; revised 5 November 2016; accepted 18 January 2017; available online 16 October 2017

Abstract

The aim of this paper is to detect bank credit cards related frauds. The large amount of data and their similarity lead to a time consuming and low accurate separation of healthy and unhealthy samples behavior, by using traditional classifications. Therefore in this study, the Adaptive Neuro-Fuzzy Inference System (ANFIS) is used in order to reach a more efficient and accurate algorithm. By combining evolutionary algorithms with ANFIS, the optimal tuning of ANFIS parameters is achieved by the Teaching-Learning-Based Optimization (TLBO) and the Particle Swarm Optimization (PSO). The aim of using this approach is to improve the network performance and to reduce calculation complexities compared to gradient descent and least square methods. The proposed algorithm is implemented and evaluated on credit cards data to detect fraud. The results demonstrate superior performance of the designed scheme compared to other intelligent identification methods.

Keywords: Credit Cards Fraud Detection, Teaching-Learning-Based Optimization (TLBO), Adaptive Neuro-Fuzzy Inference System (ANFIS), Particle Swarm Optimization (PSO).

1. Introduction

Fraud is spreading all over the world, as the information technology and communication channels develop more and more, causing large financial losses. Financial institutions pay an extreme attention to rapid solutions of fraudulent activities detection. Fraud detection is a necessary tool and probably the best way to stop the financial fraud, due to its direct influence on the institutions' customer service, decrease of operational costs and remaining a reliable financial service provider. Beside, because of the prosperity and development of electronic banking and electronic payment, the fraud is spreading in credit cards; thus, banks and credit cards issuing organizations are making serious

efforts to prevent the abuse of costumers' accounts by taking security measures.

Credit cards fraud includes the illegal use of a card or its information without the owners' knowledge. Generally, defrauders gain access to internal information of the card by different ways and all along with the change of technology, criminals modify their methods. An algorithm can produce more than a single result. In the recent years, famous algorithms in the field of credit cards fraud detection have been introduces, which will be presented briefly as below.

In 2011 [1], by using the combination of Bayesian learning for each card owner and according to the sequence of its purchases, a model has been formed and in the case of

* Corresponding author. Email: saniee@modares.ac.ir

a purchase behavior variation, it could declare a probable problem in the form of unwanted data. This method presents many advantages such as high accuracy, low false alarm and suitable detection rate. But the most important drawback of this method remains in its high implementation complexity, as a model has to be kept from the behavior of each person in order to use it if necessary.

Next, an algorithm has been presented by the influence of BLAST method and its combination with SSAHA in [2]. This algorithm can perform the balancing of costumers' behavior in two steps. Consequently, all costumers' behaviors are assimilated and could exploit similar models for all. The resulting scheme has provided acceptable performance with enough accuracy, but its major drawback was that with the increase of data numbers, the detection of suspicious simultaneous behaviors encountered certain difficulties due to the sensitivity of balancing to the sequence numbers.

Another algorithm which has a high rank in criminal activities detection is based on the nearest neighborhood. The main advantage of this algorithm is that it does not require a learning phase; thus, it could be compatible with new data in the minimum time possible and perform acceptable results [3]. The most famous algorithm in this field is called the SODRNN, which can be updated in a very little amount of time. The main disadvantage of this scheme is its very high memory consumption.

In [4], Artificial Neural Networks (ANNs) and Bayesian method have been used. The main advantage of this method is that it does not require any system reprogramming and can perform the learning from the existing data. But its main drawback remains in the necessity of a very high speed system and the non-suitable response time for updating the network.

The use of fuzzy based methods is one of the best solutions when encountering uncertainties. In [5], by using ANNs, a fuzzy method has been developed which possess the capability of algorithm parallelization and can provide suitable results in the system. The problem of using these algorithms is that they are only applicable for a specific class of bank errors and does not have a high accuracy in detection.

The idea of using Support Vector Machines (SVM) has been introduced in [6] in order to achieve a decision level for separating suspicious and healthy data, and it has provided

very suitable results. Due to its insensitivity to problem dimension, this method shows a suitable complexity and its only drawback is that it is sensitive to the increase of data and cannot support large datasets.

In [7], a two steps method is proposed to identify and detect the fraud: step 1 consists of the Self-Organizing Map (SOM) technique to map the (users' accounts) data into a two dimension topological space. The second step, the classification of fraudulent and healthy data is achieved by a simple method called the threshold binary classification algorithm. In this paper, the threshold binary classification algorithm is proposed based on U-matrix network. In the mapping stage of users' accounts, the saved data in the matrix reflect the users' sequential activities. In order to perform the execution, some transactions have been considered for each person and are assumed to be in separate matrices.

In [8], the SOM method is used to analyze the initial data before the detection and to use the Growing Hierarchical SOM (GHSOM) approach for the detection of fraudulent financial pattern. A classification rule for financial fraud detection has been presented based on topological patterns. The SOM and GHSOM methods have been compared with each other and different classification algorithms have been used in the detection phase for addressing the problems of financial fraud detection.

The aim of this study is to detect bank credit cards related frauds. The large amount of data and their similarity lead to a time consuming and low accurate separation of healthy and unhealthy samples behavior, by using traditional classifications. Therefore in this paper, by using the problem mapping from a high complexity environment to a simple one, one has tried to improve the method performance and to reach desired results in a suitable time interval. Also, in order to analyze the data and detect the crime, the Adaptive Neuro-Fuzzy Inference System (ANFIS) has been used and thus, the fuzzy system will be capable of learning and the ANNs' operation will be more clarified. This is due to this fact that ANNs are low level calculation structures which operate very well on raw data, but the fuzzy logic deals with high level inference and uses linguistic information obtained from human knowledge. However in this paper, in order to design and learn the mentioned network, the Teaching-Learning-Based Optimization (TLBO) and the Particle Swarm Optimization (PSO) are used. Indeed, the

combination of evolutionary and metaheuristic algorithms with neuro-fuzzy systems leads to an intelligent optimization algorithm which is used for the optimal tuning of system parameters.

The remaining parts of this paper is organized as follows: in the second and third sections, TLBO and PSO algorithms are described briefly, respectively. In Section 4, the ANFIS structure is presented. The proposed algorithm and the training of ANFIS network by TLBO and PSO algorithms are introduced in order to improve the detection rate of data. Also in section 6, the results of the proposed algorithm implementation for the fraud detection on credit cards' data are presented and compared with previous methods. In the last section, some conclusion remarks are presented.

2. Particle Swarm Optimization Algorithm

The PSO method is a global optimization algorithm which can deal with problems with a single point solution in n-dimensional space. In such a space, assumptions are made and an initial velocity is assigned to particles. Also, the communication channels between particles are taken into consideration.

Next, these particles move in the response space and the obtained results are calculated based on an "eligibility index" after each time interval. Along with the time, particles accelerate toward the particles with higher eligibility index which are situated in the same communication group.

Each particle possesses a position which determines the dimension of coordinates of the particle in the search space. The particle position changes along the time as the particle moves. $x_i(t)$ denotes the i^{th} particle position at time t . Also, each particle needs a velocity in order to move in the space. $v_i(t)$ is the i^{th} particle velocity at time t . By adding velocity to each particle's position, one can get the new position for the particle. The particle position updating equation is expressed as follows:

$$X(t+1) = X_i(t) + V_i(t+1) \quad (1)$$

$$X_i(t) \sim U(x_{\min}, x_{\max})$$

A cost function is used to evaluate that whether a particle's position is suitable or not. The particles have the

capability to remember their best position in all their life. The best individual experience of a particle or the best position met by a particle is called y_i (in some of algorithms y_i is also noted as p_{best}). Particles can be aware of the best position met by all the group, which is called \hat{y}_i (in some of algorithms \hat{y}_i is also noted as g_{best}). The particle velocity vector in the optimization process reflects the particle's experimental knowledge and the particles society information. Each particle considers two components for moving into the search space:

Cognitive component: $y_i(t) - x_i(t)$ is the best solution that can be achieved by a single particle.

Social component: $\hat{y}_i(t) - x_i(t)$ is the best solution that can be noted by the whole group.

Two main models exist for the standard PSO algorithm which are the calculation of their velocity vector based on both of the cognitive and social components. These two models are named PSO_{lbest} and PSO_{gbest} and their difference remains in their neighborhood size which is considered for each particle. Also, the pseudo code for the PSO algorithm is depicted in Figure 1.

```

For each particle  $i \in \mathbf{1}, \dots, s$  do
    Randomly initialize  $x_i$ 
    Randomly initialize  $v_i$  (or just set  $v_i$  to zero)
    Set  $y_i = x_i$ 
end for
Repeat
For each particle  $i \in \mathbf{1}, \dots, s$  do
    Evaluate the fitness of particle  $i$ ,  $f(x_i)$ 
    Update  $y_i$  using  $y_i(t+1) = \begin{cases} y_i(t) & \text{iff } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{iff } f(x_i(t+1)) < f(y_i(t)) \end{cases}$ 
    Update  $\hat{y}$  using  $\hat{y}(t) \in \{y_0, y_1, \dots, y_s\} = \min\{f(y_0(t)), f(y_1(t)), \dots, f(y_s(t))\}$ 
For each dimension  $j \in \mathbf{1}, \dots, N_d$  do
    Apply velocity update using
    
$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + c_2r_{2j}(t)(\hat{y}_j(t) - x_{ij}(t))$$

Endloop
    Apply position update using  $x_i(t+1) = x_i(t) + v_i(t+1)$ 
Endloop
Until some convergence criteria is satisfied
  
```

Fig. 1. Pseudo code of PSO algorithm

3. The Teaching-Learning Based Optimization Algorithm

One of the most important properties of the TLBO algorithm is its independence from parameters, as this algorithm has the minimum number of possible parameters.

The principle of TLBO algorithm is based on the teaching of a teacher in a classroom [9]. The teacher has an important role in students' learning by teaching lessons in the classroom and thus, the better learning of student depends on the teaching method. Besides, the revision of materials between the students can improve the learning process. This idea is the basis of the TLBO algorithm in solving optimization problems. The operation mechanism of the TLBO algorithm includes two parts: the first part is the teacher contribution for the enhancement of the class' scientific level. The second part is the interaction and revision of materials by the student of this class.

3.1. Teaching Phase

In this phase, the best member of society is chosen as teacher and it will guide the population average toward him/her. This is similar to the real world case where a teacher does the same job. A good teacher is someone who elevates the scientific level of persons to his own level. In practice, students' level does not increase to the teachers' one, but it gets close depending on the class capabilities. This part is modelled as follows:

$$\bar{X}_{diff}^k = rand() \times (T^k - R_t \times M^k) \quad (2)$$

Where T^k is the teachers' k^{th} iteration, M^k is the class average in the k^{th} iteration, R^t is the teaching coefficient which can be 0 or 1 and is valued randomly in each iteration. Also, \bar{X}_{diff}^k denotes the scientific difference between the teacher and students.

The population is formed as below in the next iteration:

$$\bar{X}_{new}^{k+1} = \bar{X}_{old}^k + \bar{X}_{diff}^k \quad (3)$$

Where \bar{X}_{old}^{k+1} and \bar{X}_{new}^{k+1} are the population members in the previous and new iterations, respectively. A cost function is defined for the new member of population and its value is compared to the one obtained from the same member in the previous iteration. If the former is less than the latter, the old member is substituted by the new one.

3.2. Learning Phase

In this phase, the population members (which are considered to be classmates) extend their knowledge by cooperating with each other. This is also similar to the real world case.

Students enhance their scientific level by two methods: one is the class attendance and the other one is the materials' revision between the students. In order to model this part, it is assumed that each student engages randomly into conversion with another student and its mathematical model is expressed as follows:

$$\bar{X}_{new} = \bar{X}_{old} + rand() \times (\bar{X}_i - \bar{X}_j) \quad (4)$$

Where \bar{X}_i, \bar{X}_j are the i and j^{th} population members, respectively. \bar{X}_{old} and \bar{X}_{new} are the old and the new population members. After the calculation of the new population member, its cost function value is compared to the one obtained from the same member in the previous iteration. If the former is less than the latter, the old member is substituted by the new one. This procedure is repeated for a determined time and the TLBO algorithm flowchart is illustrated in Figure 2. For further details about the TLBO algorithm, refer to [9, 10].

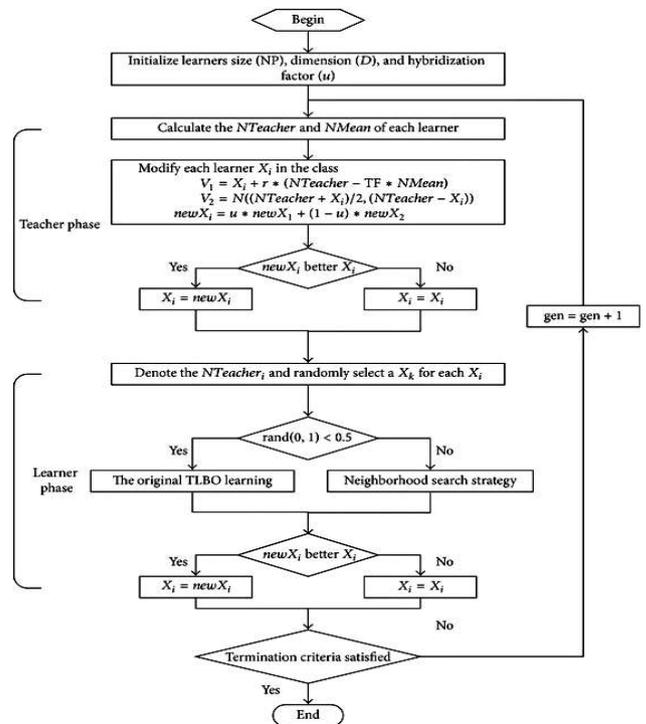


Fig. 2. TLBO algorithm flowchart

4. Adaptive Neuro-Fuzzy Inference System Structure

The ANFIS has been introduced for the first time by Jang [11] and by training a fuzzy inference system in an adaptive ANNs' framework. The ANFIS structure consists of some nodes in different layers which are connected to each other. The output of this network depends on the tunable parameters of these nodes. The network learning rules determine the parameters' updating method for minimizing the error. A fuzzy inference system is a framework based on fuzzy theory and If-Then rules. The ANFIS structure has three main elements: 1) rule base, 2) data base, and 3) reasoning mechanism.

The fuzzy rule base includes fuzzy If-Then rules. The data base along with membership functions used in fuzzy rules and also reasoning mechanism execute the output inference procedure from the input variables.

The ANFIS network is a set of If-Then rules as below:

$$\begin{aligned}
 & \text{if } x_1 \text{ is } A_1^i, x_2 \text{ is } A_2^i, \dots, x_n \text{ is } A_n^i \text{ then } y & (5) \\
 & = \sum_{i=1}^R \overline{W}_i \\
 & = (a_{i,1} x_1 + \dots + a_{i,n} x_n \\
 & + a_{i,n+1} x_{n+1})
 \end{aligned}$$

Such that $x_i (i = 1, 2, \dots, n)$ is the network input, A and y are the fuzzy sets and network output, respectively.

The ANFIS structure is shown in Figure 3. As it can be seen, the type of used functions for each node in a layer is similar. $O_{l,i}$ is the output of i^{th} node from the l^{th} layer.

The output of each layer can be expressed as:

Layer 1: the nodes of this layer are adaptive. The output of each node is:

$$O_{1,i} = \mu_{A_i}(X) \quad (6)$$

Such that x is the input value of node i and A_i is the fuzzy set related to this node. For each input, we can have one or more fuzzy sets. The Gaussian function $\mu_{A_i}(x)$ is defined as follows:

$$\mu_{A_i}(X) = \exp\{-\frac{(x-c_i)^2}{\sigma_i^2}\} \quad (7)$$

This function determines the membership of input x to the fuzzy set A_i . It is obvious that c_i, σ_i are the i^{th} node

parameters of the first layer of ANFIS and should be trained.

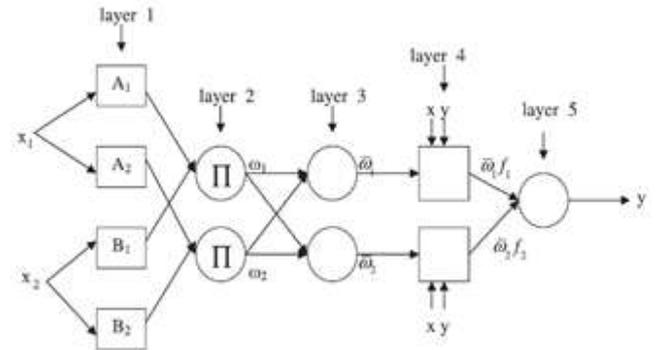


Fig. 3. ANFIS network structure

Layer 2: the node value is constant in this layer and denoted the output of i^{th} rule. The output of this layer in the product of all the outputs as below:

$$O_{2,i} = W_i = \mu_{A_i}(X) \cdot \mu_{B_i}(X), i = 1, 2, \dots \quad (8)$$

Layer 3: the node value is constant in this layer and calculates the intensity ratio of the i^{th} rule as follows:

$$O_{3,i} = \overline{W}_i = \frac{W_i}{W_1 + W_2}, i = 1, 2, \dots \quad (9)$$

Layer 4: the nodes in this layer are the executers of each rule's output as below:

$$O_{4,i} = \overline{W}_i F_i = (a_i x + a_i \text{ bias}) \quad (10)$$

Parameters $\alpha_{i \text{ bias}}, \alpha_i$ are as result part parameters and are updated in the training phase.

Layer 5: the single existing node in this layer is a constant node which denotes the final value of output parameter as a set of input signals:

$$O_{5,i} = \frac{\sum_i W_i f_i}{\sum_i W_i} \quad (11)$$

5. Training ANFIS based on Evolutionary Algorithm

The learning algorithm is used to tune all the tunable parameters (target function and conclusion parameters) and to obtain output parameters values of ANFIS which is adapted with training data.

Here we have three population categories randomly named as $P_\alpha, P_\beta, P_\gamma$. The population P_α corresponds to parameters a and the variable numbers of each particle of this population is equal to the product of network rules' number and network inputs' number plus one (this number is equal to the number of parameters in a). The population P_β corresponds to parameters c_i , i.e. the categories' centers. The population P_γ corresponds to parameters σ_i or actually the spread. Each particle of each population $P_\alpha, P_\beta, P_\gamma$ possesses N members, where N is equal to the total number of fuzzy sets.

5.1. ANFIS Training Based on PSO Algorithm

The PSO algorithm calculates the velocity vector and the new position for each particle. Then, the particle new position is applied to the evaluation function and thus based on that, the best experience of each particle and also the best experience of the group are obtained. When the operation has been repeated for all the particles, finally, each particle possesses a value called the best individual experience and the group has a value called the best group experience. In the next iteration for each particle, these values have to be calculated. This procedure continues until reaching a threshold which can be as a determined number of iterations or as a reaching time to a determined value of the error. Finally, after finishing the training, the best member of population is considered as trained ANFIS network parameters.

Figures 4 and 5 depict the pseudo code of the algorithm and the ANFIS-PSO algorithm flowchart, respectively.

5.2. ANFIS Training Based on TLBO Algorithm

According to the evaluation function (which is the network output error), the TLBO algorithm considers the best population member as teacher. Then, the teacher tries to share his/her information with the population and students/initial population will update their information as well. Therefore, the class average is improved and a new teacher will be required. Now, by applying the new population to the evaluation function, the best member is again chosen as teacher. This procedure can be repeated until reaching a threshold limit which can be equal to a determined number of iterations or error value. Finally, once the training has finished, the best population members are thus trained as the ANFIS network parameters.

```

Begin
Create ANFIS Structure
Random Initialize of ANFIS's Parameters ( $P_\alpha, P_\beta, P_\gamma$ )
Calculate the Number of Parameters
// Call PSO Algorithm
Function PSO (problem) returns a state that is Global Best
Input: Populationsize, Number of Weights  $c_1, c_2, w, V_{max}, N_d$ 
Output:  $S_{best}$ 
Fitness: Mean Square Error;
Population  $\leftarrow 0$ ;
For each particle  $i \in 1, \dots, s$  do
Randomly initialize  $x_i$ 
Randomly initialize  $v_i$  (or just set  $v_i$  to zero)
Set  $y_i = x_i$ 
Endfor
repeat
For each particle  $i \in 1, \dots, s$  do
Evaluate the fitness of particle  $i, f(x_i)$ 
Update  $y_i$  using  $y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases}$ 
Update  $\hat{y}$  using  $\hat{y}(t) \in \{y_0, y_1, \dots, y_s\} = \min\{f(y_0(t)), f(y_1(t)), \dots, f(y_s(t))\}$ 
For each dimension  $j \in 1, \dots, N_d$  do
Apply velocity update using
 $v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + c_2r_{2j}(t)(\hat{y}_j(t) - x_{ij}(t))$ 
Endloop
Apply position update using  $x_i(t+1) = x_i(t) + v_i(t+1)$ 
Endloop
Evaluate Population (Population);
 $S_{best} \leftarrow$  Get Best Solution (Population);
Return  $S_{best}$ 
// End of PSO Algorithm
Consider Total Dimensions of  $S_{best}$  as the parameters of the ANFIS
End

```

Fig. 4. TLBO algorithm flowchart

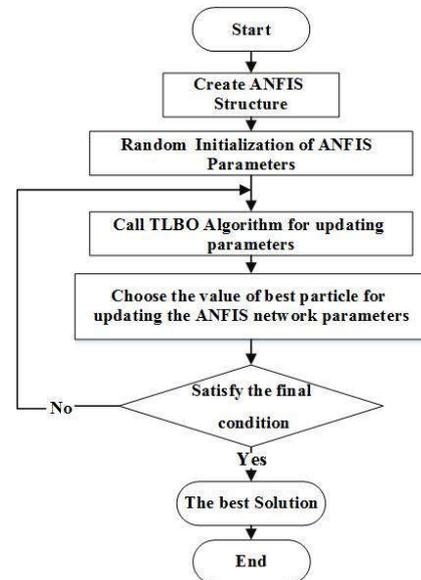


Fig. 5. Training ANFIS algorithm flowchart based on PSO algorithm

In order to further understand and summarize the aforementioned algorithm, the pseudo code and flowchart of the proposed ANFIS-TLBO algorithm are illustrated in Figures 6 and 7.

```

Begin
Create ANFIS Structure
Random Initialize of ANFIS's Parameters ( $P_a, P_k, P_c$ )
Calculate the Number of Parameters
    // Call TLBO Algorithm
Function TLBO (problem) returns a state that is Global Best
    Input: Populationsize, Number of Weights, Nd
    Output:  $S_{best}$ 
    Fitness: Mean Square Error;
    Population  $\leftarrow$  0;
For each particle  $i \in 1, \dots, s$  do
    Randomly initialize  $\bar{X}_i^k$ 
endfor

For each iteration  $k \in 1, \dots, It$  do
    Compute the average class  $\{M^k\}$ 
    For each particle  $i \in 1, \dots, s$  do
    Evaluate the fitness of student,  $i, f(\bar{X}_i^k)$ 
    Update  $T^k$  using  $T^k \in \{\bar{X}_0^k, \bar{X}_1^k, \dots, \bar{X}_s^k\} = \min\{f(\bar{X}_0^k), f(\bar{X}_1^k), \dots, f(\bar{X}_s^k)\}$ 
    // Teacher Phase
    Updated students in the teacher phase
    
$$\bar{X}_i^{k+1,1} = \bar{X}_i^k + rand() \times (T^k - R_i \times M^k)$$

    Update  $\bar{X}_i$  using  $\bar{X}_i^{k+1} = \begin{cases} \bar{X}_i^k & \text{if } f(\bar{X}_i^{k+1,1}) \geq f(\bar{X}_i^k) \\ \bar{X}_i^{k+1,1} & \text{if } f(\bar{X}_i^{k+1,1}) < f(\bar{X}_i^k) \end{cases}$ 
    // Learner Phase
    Updated students in the learner phase
    
$$j = randi(Nd - 1)$$


$$\bar{X}_i^{k+1,2} = \bar{X}_i^{k+1,1} + rand() \times (\bar{X}_i^{k+1} - \bar{X}_j^{k+1})$$

    Update  $\bar{X}_i$  using  $\bar{X}_i^{k+1} = \begin{cases} \bar{X}_i^{k+1,1} & \text{if } f(\bar{X}_i^{k+1,2}) \geq f(\bar{X}_i^{k+1,1}) \\ \bar{X}_i^{k+1,2} & \text{if } f(\bar{X}_i^{k+1,2}) < f(\bar{X}_i^{k+1,1}) \end{cases}$ 
    Endloop
Endloop
    Evaluate Population (Population);
     $S_{best} \leftarrow$  Get Best Solution (Population);
    Return  $S_{best}$ 
    // End of TLBO Algorithm
    Consider Total Dimensions of  $S_{best}$  as the parameters of the ANFIS
End
    
```

Fig. 6. Pseudo code of ANFIS-TLBO proposed algorithm

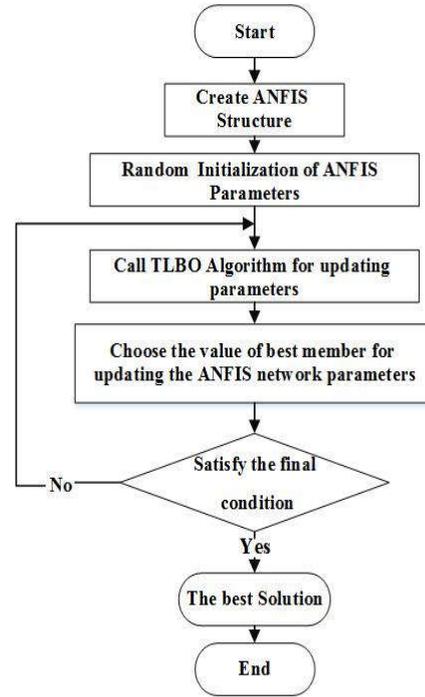


Fig. 7. Training ANFIS algorithm flowchart based on TLBO algorithm

6. Implementation Results of the Proposed Algorithm for Credit Cards' Fraud Detection

In order to implement the algorithm, at first, one should determine the network characteristics for credit cards' fraud detection. Thus, the parameter values for implementing the ANFIS-PSO and ANFIS-TLBO algorithms are shown in Tables 1 and 2, respectively.

The value of maximum velocity achieved by other particles (V_{max}) has been considered to be equal to 1.2. Also, the initial value of the performed motion inertial weight has been assumed to be 0.8 in order to emphasize the scanning ability. These two parameters control the effect of previous motion produced by other particles. During the program iterations, the latest parameter is finally reduced linearly to 0.1 in order to benefit from the property. This is due to the fact that at the end we are approaching the program response and one need to provide small variations in the displacement value.

One of the most important characteristics of TLBO algorithm is its independence from parameters; because this algorithm possesses the minimum number of possible

parameters and therefore, it has a special privilege. Thus, as it can be seen in Table 2. The only parameter required in TLBO algorithm is the learning coefficient, which chooses stochastically 1 or 2 with equal probability.

German and Australian credit cards' datasets have been used in the implementation.

As it can be seen, the particle number (N) has been assumed to be 30, because according to experiments, a higher number of particles does not lead to a better accuracy and only increase the program execution's time. The particle algorithm is executed 500 times. This value is achieved based on several experiments by trial and error. The greater value for iterations leads to an excessive learning and the lesser one causes an incomplete learning.

Table 1. ANFIS-PSO algorithm parameters

Parameters	Value
Total German data number	1000
Total Australian data number	690
German data number for network training	700
Australian data number for network training	480
Input number in German data	24
Input number in Australian data	14
Output number	1
Number of fuzzy sets for each input	3
Size of populations P_a, P_c, P_α	30
Number of PSO algorithm execution	$N(t) = 500$
Inertia weight	$w(0) = 0.8$
Individual learning coefficient	$c_1(0) = 1$
Group learning coefficient	$c_2(0) = 2$
Maximum particle velocity	$V_{\max} = 1.2$

Table 2. ANFIS-TLBO algorithm parameters

Parameters	Value
Total German data number	1000
Total Australian data number	690
German data number for network training	700
Australian data number for network training	480
Input number in German data	24
Input number in Australian data	14
Output number	1
Number of fuzzy sets for each input	3
Size of populations P_a, P_c, P_α	30
Number of TLBO algorithm execution	$N(t) = 500$
Learning coefficient	{1,2}

In order to study the performance of the proposed algorithm, one need a suitable method to evaluate the efficiency of the designed scheme, because the more accurate the evaluation method, the more correct the analysis of results will be. In this paper, the following formula is used to evaluate the proposed approaches ANFIS-PSO and ANFIS-TLBO:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

According to the above equation, in order to obtain the accuracy, the mess matrix including four components as TP, TN, FP and FN should be formed during the program execution. Each of these components denotes specific data explained as below:

- TP: number of correct data detected for category 1
- TN: number of correct data detected for category 0
- FP: number of incorrect data detected for category 0
- FN: number of incorrect data detected for category 1

One of the other aims of learning on data is the capability of extending the results to similar data. In order to declare the generalization and compare the proposed method with the previous ones, one should use the validation; therefore, data are divided into two phases of train and test data. At first, the network is trained with train data (which constitutes 70% of the total data) and finally the accuracy is evaluated by test data (the remaining 30% of data).

Like all of the intelligent evolutionary algorithms, this algorithm cannot be referred as correct with a single program run, and this is due to the use of stochastic parameters. Therefore, the program has been executed 25 times and by averaging the results, more accurate and more reliable data are achieved.

Also, the proposed approaches are compared with other methods such as ANFIS, Gaussian Multiple Model (GMM) method, Self-Organizing Mapping (AOM) with RBF kernel, the GHSOM method [12], the Fuzzy Artificial Immune Systems (FAIS), the Simple Artificial Immune Systems (SAIS) [14] and the improved version of FAIS (IFAIS) [15].

6.1. Simulation Results with German Credit Cards' Data

The German credit cards' datasets include 20 properties of 1000 transactions of credit cards in Germany which have been used in KDD99 competitions. As some of its data are descriptive, in order to use them in different categories, another version has been presented which shows the data in 24 properties in numerical form [16].

Due to the use of random parameters in PSO and TLBO algorithms (as in all of the intelligent evolutionary algorithms), one cannot approve the validation of results with a single program execution. Therefore, the program has been run 25 times and by taking average of the derived results, more accurate and more reliable information are presented. Next, the performance and efficiency of the proposed ANFIS-PSO and ANFIS-TLBO algorithms have been compared with the aforementioned methods. The results of comparison are given in Table 3.

As the data similarities are very high and their separation is very difficult, the data separation is previous papers has not been higher than 76% in the best case. As one can see in Table 3, the proposed algorithm possesses higher detection accuracy compared to other ones.

Table 3. Comparison of the fraud detection validation in German credit cards with the proposed methods and the previous ones

Evaluation Method	Test Data Accuracy
GHSOM	0.650
GMM	0.750
ANFIS	0.755
FAIS	0.72
SAIS	0.754
SMO with RBF kernel	0.70
IFAIS using simple memory	0.723
IFAIS using 3-layer memory	0.749
ANFIS-PSO	0.792
ANFIS-TLBO	0.810

Also, the error decrease is shown in Figures 8, 9 and 10 when using PSO, TLBO, and gradient descent training algorithms, respectively. The error sum of squares average in equal iteration numbers for credit cards' data is equal to 0.0039, 0.00375 and 0.0029 for gradient descent, PSO and TLBO algorithms, respectively. The superior performance of TLBO over the other approaches is thus proved.

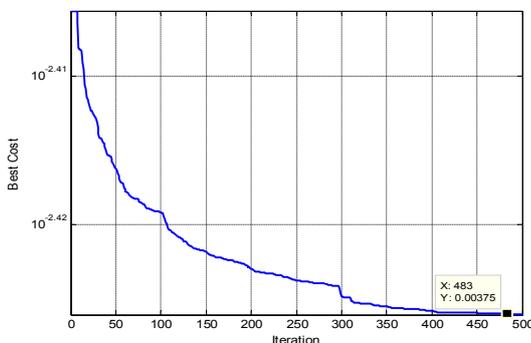


Fig. 8. Error decrease in German credit cards' fraud detection by using the PSO algorithm

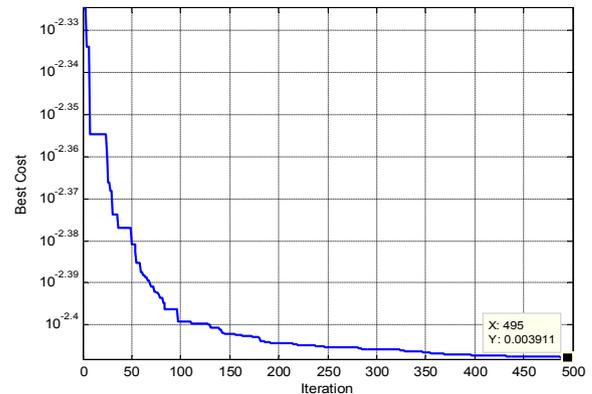


Fig. 9. Error decrease in German credit cards' fraud detection by using the gradient descent algorithm

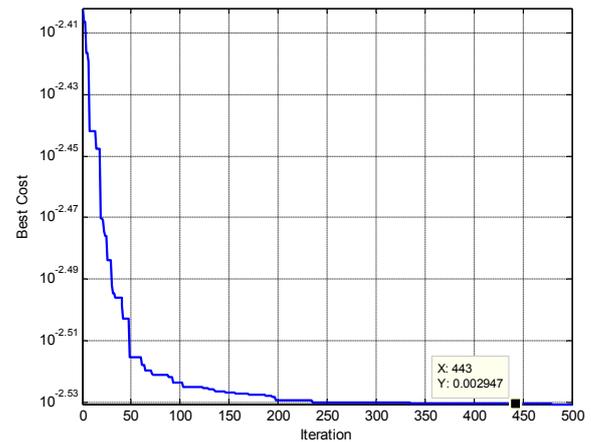


Fig. 10. Error decrease in German credit cards' fraud detection by using the TLBO algorithm

6.2. Simulation Results with Australian Credit Cards' Data

The Australian credit cards' datasets include 14 properties of 690 transactions of credit cards in Australia. The data are classified into fraudulent and normal and are shown by 0 and 1, respectively [17].

Similar to the last section, the performance and efficiency of the proposed ANFIS-PSO and ANFIS-TLBO algorithms are shown. The program has been run 25 times and the averages of the derived results are compared with previous approaches. The comparison results are shown in Table 4.

Table 4. Comparison of the fraud detection validation in Australian credit cards with the proposed methods and the previous ones

Evaluation Method	Test Data Accuracy
ANFIS	0.755
FAIS	0.855
SAIS	0.852
SMO with RBF kernel	0.855
IFAIS using simple memory	0.865
IFAIS using 3-layer memory	0.878
ANFIS-PSO	0.875
ANFIS-TLBO	0.892

It can be seen that the ANFIS-PSO and ANFIS-TLBO algorithms possess higher detection accuracy compared to other algorithms.

Also, the error decrease is shown in Figures 11 and 12 when using PSO and TLBO training algorithms, respectively. The error sum of squares average in equal iteration numbers for credit cards' data is equal to 0.0029 and 0.0021 for PSO and TLBO algorithms, respectively. From these results, one can infer the following facts: 1) The ANFIS-TLBO algorithm has superior performance over the PSO and gradient descent ones. 2) As the cost function value for Australian data is less than in the German case, the ANFIS-TLBO algorithm has shown a better performance when encountering Australian credit cards' data compared to the German ones.

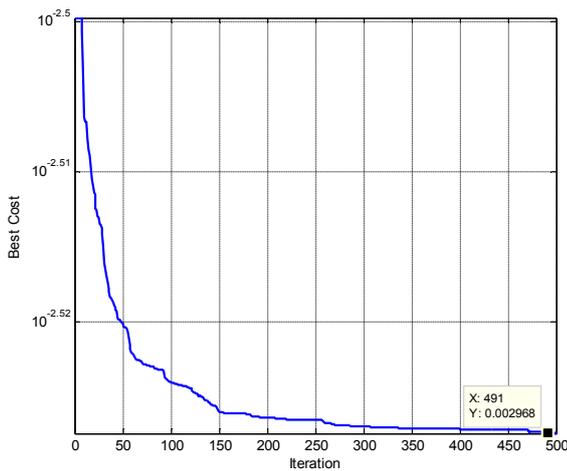


Fig. 11. Error decrease in German credit cards' fraud detection by using the PSO algorithm

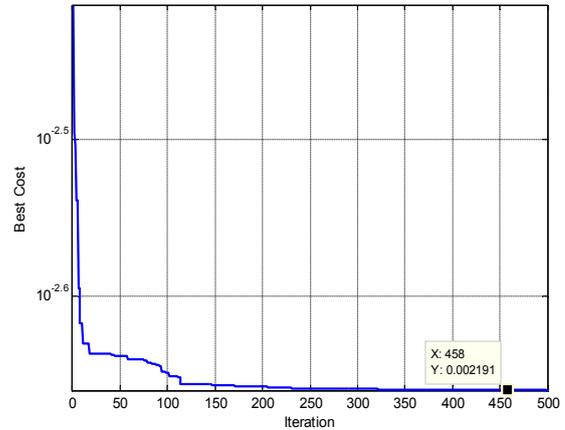


Fig. 12. Error decrease in German credit cards' fraud detection by using the TLBO algorithm

The results of applying train and test data to the ANFIS-PSO network are shown in Figures 13 and 14, respectively, and are compared with real values. One can see that the ANFIS-PSO algorithm shows the same performance compared to train and test data. This is a proof to the efficiency of the proposed algorithm generalization, because its performance has not been decreased when encountering new data.

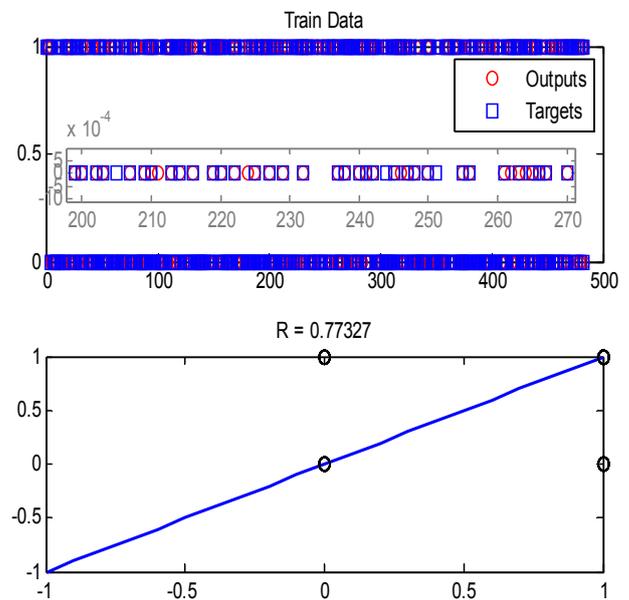


Fig. 13. Results of training the ANFIS network based on PSO

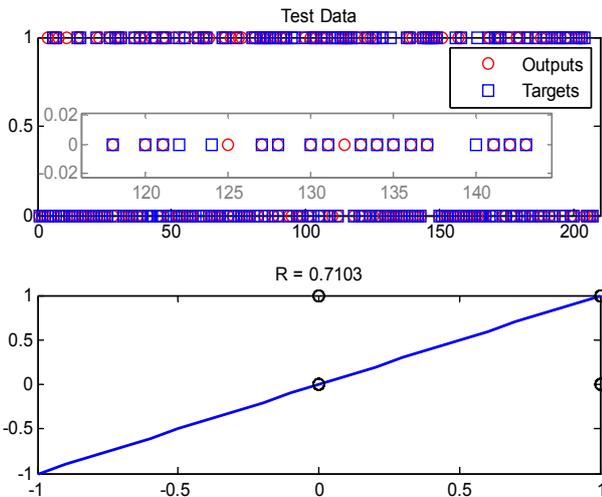


Fig. 14. Results of testing the ANFIS network based on PSO

The results of applying train and test data to the ANFIS-TLBO network are shown in Figures 15 and 16, respectively, and are compared with real values. One can see that the ANFIS-TLBO algorithm shows the same performance compared to train and test data. This is a proof to the efficiency of the proposed algorithm generalization, because its performance has not been decreased when encountering new data.

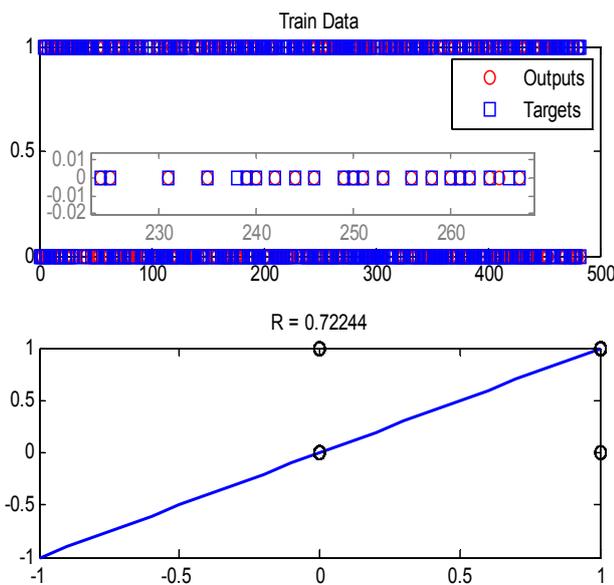


Fig. 15. Results of training the ANFIS network based on TLBO

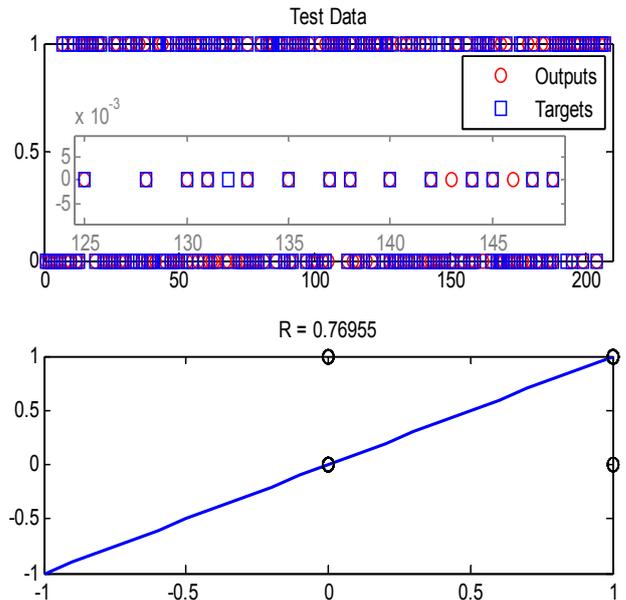


Fig. 16. Results of testing the ANFIS network based on TLBO

7. Conclusion

The main concentration of this paper was on the detection methods of bank credit cards' frauds. According to the derived results of implementation of ANFIS-PSO and ANFIS-TLBO algorithms on credit cards' data for the fraud detection and comparison with other methods, one has observed that the algorithm trained based on TLBO had a better accuracy and a narrower error margin. This is due to the fact that the TLBO algorithm has fewer parameters to be tuned. Also, the error comparison results derived from the TLBO, PSO, and gradient descent algorithm showed that in equal iteration number, the TLBO algorithm presents better results. The execution speed of two algorithms was another fact to be considered; PSO and TLBO algorithms were slower than the gradient descent, but this was not important when dealing with real time systems, because the network was trained a single time. Finally, it could be seen that the TLBO algorithm can train the network parameters with a considerable accuracy and just based on its own information, and this is all the contrary of gradient descent and least squares methods which have very high calculation complexities.

References

- [1] Singh, S. P.; Shukla, S. S. P.; Rakesh, N.; Tyagi, V., "Problem Reduction In Online Payment System Using Hybrid Model" *International Journal of Managing Information Technology* (2011).
- [2] Raj, S.; Porti, A., "Analysis on Credit Card Fraud Detection Methods", *International Conference on Computer* (2011).
- [3] Ganji, V.; Mannem, S., "Credit card fraud detection using anti-k nearest neighbor algorithm", *International Journal on Computer Science and Engineering* (2012).
- [4] Zareapoor, M.; Seeja, K. R.; Afshar, M., "Analysis of Credit Card Fraud Detection Techniques: based on Certain Design Criteria", *International Journal of Computer Applications* (2012).
- [5] Benson Edwin Raj, S.; Annie Portia, A., "Analysis on Credit Card Fraud Detection Methods", *International Conference on Computer, Communication and Electrical Technology* (2011).
- [6] King, J.; Pun, F., "Improving Credit Card Fraud Detection using a Meta-Learning Strategy", *Chemical Engineering and Applied Chemistry University of Toronto* (2011).
- [7] Olszewski, D., "Fraud detection using self-organizing map visualizing the user profiles", *Knowledge-Based Systems* pp. 324–334 (2014).
- [8] Huang, S. Y.; Tsaih, R. H.; Yu, F., "Topological pattern discovery and feature extraction for fraudulent financial reporting", *Expert Syst. Appl.* 41 pp. 4360–4372 (2014).
- [9] Rao, R. V.; Savsani, V. J.; Vakharia, D. P., "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems", *Computer-Aided Design* 43.3 pp. 303–315 (2011).
- [10] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems", *Information Sciences* 183.1 (2012) 1–15.
- [11] Jang, J. S. R., "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Trans. Sys., Man and Cybernetics*, Vol. 23, No.3 (1993).
- [12] Huang, S. Y.; Tsaih, R. H.; Yu, F., "Topological pattern discovery and feature extraction for fraudulent financial reporting", *Expert Systems with Applications*. Vol. 41 pp. 4360–4372 (2014).
- [13] Kamaloo, E.; Saniee Abadeh, M., "An artificial immune system for extracting fuzzy rules in credit scoring", *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)* pp. 1–8, Barcelona (2010).
- [14] Wu, D. D.; Olson, D., "Enterprise risk management: a DEAVaR approach in vendor selection", *International Journal of Production Research*, vol. 48, no. 16 pp. 4919–4932 (2010).
- [15] Kamaloo, E.; Saniee Abadeh, M., "Credit Risk Prediction Using Fuzzy Immune Learning", *Hindawi Publishing Corporation, Advances in Fuzzy Systems* pp. 11–22 (2014).
- [16] [https://archive.ics.uci.edu/ml/datasets/Statlog+\(German+Credit+Data\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)).
- [17] [https://archive.ics.uci.edu/ml/datasets/Statlog+\(Australian+Credit+Approval\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval)).