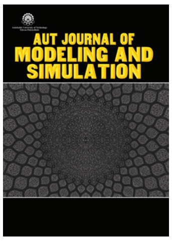

AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 51(1) (2019) 67-80
DOI: 10.22060/miscj.2019.13906.5085

A Combined Learning Approach for Credit Scoring Using Adaptive Hierarchical Mixture of Experts: Iranian Banking Industry

D. Dadmohammadi, A. Ahmadi*

Department of Industrial Engineering & Management Systems, Amirkabir University of Technology, Tehran, Iran.

ABSTRACT: Traditional methods for granting credit to loan applicants are based on personal judgment. Nevertheless, the current financial crisis alongside the efforts of banks and financial institutes for decreasing the percentage of overdue loans emphasis the importance of Credit Scoring (CS) models. This paper provides a credit scoring model by means of Modular Neural Network (MNN) established upon combined hybrid-ensemble learning. The proposed model is composed of four powerful neural networks that construct collectively the Adaptive Hierarchical Mixture of Experts (AHME). Training process is a hybrid way for learning the modular model and adaption to the CS model based on the modulation of learning rules specific to each module and particular HME online learning algorithm. Binary Particle Swarm Optimization (BPSO), using Taguchi reasoning scheme for tuning the governing parameters, is also applied for reducing dimensionality and decomposing the problem among the various modules. The proposed model's performance is compared with that of Multi-Layer Perceptron (MLP) and Laterally Connected Neural Network (LCNN) models. The aforementioned models are evaluated using the data obtained from one of the Iranian banks. Results demonstrate that the AHME outperforms other methods in terms of prediction accuracy as well as the Area Under the ROC Curve (AUC) and the Mean Squared Error (MSE) rate.

Review History:

Received: 3 January 2018
Revised: 24 May 2018
Accepted: 28 May 2019
Available Online: 1 June 2019

Keywords:

Credit Scoring
Hybrid-Ensemble Learning
Adaptive Hierarchical Mixture of Experts (AHME)
Binary Particle Swarm Optimization (BPSO)
Iranian Banking Industry

1. Introduction

Today, process of globalization and competition between financial institutions in local and global markets has growingly increased the necessity to enhance and revise systems in financial and capital enterprises throughout the world. It is thus natural that banks are no exceptions. Making a balance between supply and demand in banking resources and loans, managing and reducing delayed payments, and getting rid of security-based system are more incorporated into discussions that highlight the necessity for the implementation of rating system in banking system than anything else. For this reason, given the growing credit volume of financial sector, many of different CS models have been developed by banks and researchers in order to measure the credit of loan applicants (Limsombunchai et al. 2005). Banks and other financial organizations are seeking profit in order to make money for their shareholders. Loan granting process is one of the principle services which still considers the base of the income to such organizations. When wrong decisions, regarding the loan application, are taken; credit risk arises. If credit risk evaluation is wrong, defaulters may increase and it may cause banks' bankruptcy (Kambal et al. 2013).

CS models are based on statistical techniques, operational research, Artificial Intelligence (AI) methods, and data mining. The most popular techniques are traditional models based on statistical analysis as well as advanced techniques. A variety of models and methods are available in order to appraise bank customers, including Linear Discriminant Analysis (LDA), Linear Regression Analysis (LRA), Multivariate Adaptive Regression Splines (MARS), Classification and Regression Tree (CART), Support Vector Machine (SVM), Genetic Algorithm (GA), and Artificial Neural Network (ANN), though they are not limited to these methods only (Akkoç 2012; Ayouche et al. 2017; Kambal et al. 2013; Li et al. 2013; Limsombunchai et al. 2005; Ong et al. 2005).

The CS is a binary classification that classifies or rates customers into predefined "good" and "bad" credit applicants. Therefore, the task of CS process is to rate borrowers to approximate the ability of refunding the associated loan. This process uses quantitative measures of past loans to predict future loans (Ayouche et al. 2017). Therefore, in order to examine credit applications, new techniques should be developed to predict credits more thoroughly. A well-designed model should carefully achieve a high accuracy of classification into account in order to classify new available applicants or customers as honouring or dishonouring clients (Han et al. 2013).

Although a large number of new approaches have been suggested, many other issues should be taken into

*Corresponding author's email: abbas.ahmadi@aut.ac.ir

consideration to increase the accuracy of CS (Ong et al. 2005). Henli and Hend summarized the underlying statistical methods; the methods seemed relatively easy for implementation. They were able to produce correct and easily interpreted results. However, statistical methods have plenty of problems including less resistance, hypothesis based quality, low adjustability with new environment, and inefficiency in large-scale problems (Han et al. 2013; Kambal et al. 2013). Therefore, researchers often seek streamlined methods to tackle the above-mentioned problems.

Back in two decades ago, ANN was concluded as an important alternative to financial prediction studies, and it has received the attention of many researchers due to its high capability to learn nonlinear phenomena as well as high prediction accuracy. Unlike statistical methods, ANN needs no assumption (Akkoç 2012). On the one hand, ANN has been open to criticism of researchers, amongst which we can refer to long learning process and lower prediction accuracy for problems including many variables and their inability to break down problems into sub-problems in order to minimize big problems such as CS models of banks and financial institutions (Rafiee Vahid & Ahmadi 2016). On the other hand, due to their broadness, easiness, and effectiveness, they offer extraordinary computational abilities that help to discover various competitive hypotheses simultaneously (Kiruthika, 2015). Their disadvantage and advantage have led researchers to use some algorithms simultaneously in order to remove their deficiencies. In this regard, Modular NNs (MNNs) are relatively new combined algorithms developed by ANN and some evolutionary algorithms concurrently. MNNs are made up of subnetworks that may be separated according to categories based on different structures and functionalities; each subnetwork is combined with another. Each subnetwork represents a different neural network, which can perform individual subtasks. Various learning algorithms can also be mixed with each other. Better training of neural network is achieved by combining the best learning algorithms for a special task. Basically, different approaches to modularization used in integration with each other to get an optimum combination of a hybrid network structure and a learning algorithm (Yarushev & Averkin 2017).

The accuracy of classification is very important and sensitive in CS. Because failure to consider this axiom would cause irreparable damages to banks, we improve the accuracy of CS model through a hybrid model and the dimension reduction process. In feature subset selection problem, prediction accuracy of the selected subset depends on the size of the subset and the selected features as well. This difficulty necessitates developing efficient heuristic technique to manage the computational complexity as well as to induce models by high prediction accuracy (Unler & Ulper 2010). Therefore, the novel approach is to use an adaptive hierarchical hybrid modular model called AHME along with BPSO. It reduces the dimensions of input data and optimizes the structure of experts in order to perform CS of customers for one of the Iranian state banks.

As a matter of fact, the innovation of research lies in both the use of AHME model for CS of customers and the new method for how to train the hierarchical network. Training the hierarchical networks including HME is often undertaken along with the use of Expectation-Maximization algorithm (EM) or online algorithm using techniques of recursion estimation theory (Jordan and Jacobs 1994), which involves lengthy and difficult probable complicated computation. It causes learning process to take a lot of time. For this reason, by using a hybrid-ensemble learning approach, training algorithm of each module and online algorithm of HME training as well as addressing the foregoing deficiency, we could achieve a high rate of accuracy in order to predict customer's credit behaviour via CS model.

In the proposed method, in addition to reducing a large quantity of intricate computations, probable interpretation of model resulting from EM method is obtained. Accordingly, four powerful neural networks can be used as a combined model for CS problem, together with BPSO algorithm to build a strong hierarchical network. To decompose the model amongst modules, dimensions of input data on each module is broken via BPSO algorithm and the structure of each expert is optimized, in the first phase. The innovation lying in this paper is to use a super powerful hierarchical network for special problem of bank customers' CS, as well as the new approach to train this type of network.

The rest of this paper is organized as follows. Initially, we review the literature and previous studies. Then, AHME model and its training process will be explained. Later on, the applied dataset and Taguchi method are expressed. Accordingly, the results of applying the proposed model are presented and compared with those of other models. Finally, paper findings and future research directions are provided.

2. Background

Research in the field of bank customer's CS is divided into two major groups; studies and research conducted based on statistical methods and those conducted based on AI techniques and relevant approaches. Although there are no convincing results suggesting which one is decisively better, given the review of literature conducted in both foregoing areas, the second class has received greater attention of researchers. In this paper, the focus of the literature review in the field of CS problems is directed to AI methods and relevant approaches including the combined multiple classifiers, i.e. ensemble learning (Rafiee Vahid & Ahmadi 2016).

In the beginning, DA and regression were the only methods used in the field of CS models. Winigton utilized logistic regression for credit scoring for the first time. Gerabowsky and Tally used DA and Probit for scoring applicants of main chain stores in the US. Data mining techniques, which have been recently developed including neural networks, Genetic Programming (GP), and SVM, can properly carry out classification task. Additionally, these methods also achieved better performance than traditional statistical methods. Limosbanchi et al. used Logit model and two kinds of ANN, i.e. Probabilistic Neural Network (PNN) and MLP, in order to estimate the scoring model of agricultural loans in Thailand.

The results indicated that PNN model is generally more powerful to predict accurately in comparison with other two models (Limsombunchai et al. 2005; Wang et al. 2016).

Since AI approaches including ANNs, GA, and expert systems have been designed and introduced, their application in CS and financial studies have become widespread and have been developing rapidly. Ang et al. used GP to inventively and autonomously determine sufficient discriminant functions and valid features simultaneously. They utilized two numerical samples to compare rate of error with other CS models including ANN, Decision Tree (DT), rough sets, and logistic regression. According to the results, they concluded that GP can outperform other models (Ong et al. 2005).

In most studies, researchers compare ANN with traditional statistical methods such as DA, LR, Probit regression, Naive Bayes (NB), Classification and Regression Tree (CART), and KNN. ANN achieved better performance than these techniques, hence it was considered to be the proper alternative to these conventional techniques in credit scoring (Kambal et al. 2013). These studies have highlighted the use of several ensemble classifiers in building credit-scoring models. Tsai and Wu (2008) tested the performance of single NN classifier and compared it with multiple and diversified NN classifiers. The results showed that there is no dominant classifier. West et al. (2005) used three NN-ensemble strategies (cross-validation, bagging and boosting), the results showed that NN-ensembles are more accurate, robust and superior than single NN. Nanni and Lumini (2009) investigated the performance of several ensemble classifiers. The obtained results showed superiority of ensemble in terms of classification accuracy. Twala (2010) investigated five classifiers with different noise levels of attributes, and tried to improve the accuracy of their ensembles. The results revealed that ensemble classifiers are more accurate at different noise levels (Alaraj et al. 2014).

Hasieh developed behavioural rating models with respect to credit cards data by means of Self-Organizing Map (SOM). In this research, bank clients were classified in three major beneficial groups, and the results of the study can be used in the development of marketing strategies. In another study, it was concluded that cluster analysis can promote the performance of CS models with respect to ANN (Akkoç 2012).

In recent years, ensemble classifiers have been suggested for improving the performance of CS models. The key idea of ensemble classifiers is to integrate a number of classifiers into a single multiple classifier. Ju et al. found that combination of ANN and SVM can work better than single classifiers. Similarly, in a study by Nani and Lumini, ANN was determined to be the best single classifier, but the best performance as a whole was obtained from the quasi-classifier group with a Lowenberg's neural network model (Shayeghi et al. 2010). Hasieh and Howang developed CS models of ensemble classifiers after they separated Germany's credit data in good, bad, and marginal classes with cluster analysis. Finlay, they compared the performance of multiple classifiers, and found that error trimmed boosting outperforms the entire multiple classifiers regarding England's credit data (Akkoç 2012). Disay, Krook and Overstreet examined MLP, MOE, and linear DA, and logistic regression with respect to scoring applicants of credits in credit union industry. Their methodology included two-part field data of Cross Validation (CV) obtained from three credit unions with the assumption that identical expenditures are both good and bad for credit risks. They concluded that ANN models were slightly better than logistic regression models (Ong et al. 2005).

Unler & Upler (2016) investigated the feature subset selection problem for the binary classification problem using logistic regression model. They developed a modified discrete PSO algorithm for the feature subset selection problem. They compared their approach by two other competitive heuristic strategies namely Tabu Search (TS) and Scatter Search (SS), using publicly available datasets and demonstrated the effectiveness of their proposed methodology. The discrete PSO approach has recently gained more attention for solving the feature subset selection problem in terms of both classification accuracy and computational performance.

Babaoglu et al. (2010) explored the efficiency of BPSO and Genetic Algorithm (GA) techniques as feature selection models on determination of Coronary Artery Disease (CAD) based on Exercise Stress Testing (EST) data. Their dataset having 23 features was obtained from patients who had performed EST and coronary angiography. Classification results of feature selection technique using BPSO and GA were compared with each other and also with the results of the whole features using simple SVM model. The results indicated that feature selection technique using BPSO is more successful than feature selection technique using GA on determining CAD.

For all the discussed methods, either in statistical techniques or in AI ones, it is observed that high accuracy along with model's low error rate is a key and important issue that researchers are unanimously agreed upon. It is evident that subsequent to statistical methods, AI methods particularly neural networks have received a great deal of interest in the field of CS problems. As per our knowledge, it can be declared that almost no study has been carried out in the application of the hierarchical combination of ANNs for bank customers' CS, but the application of this network is vastly utilized in signal processing, time series analysis, and the estimation of speech quality. Therefore, for the above reasons, it is difficult to say which method can offer the best result when using diverse dataset in different nations. Accordingly, efficiency and accuracy of CS model can be tangibly boosted using a reliable and powerful model such as AHME. Moreover, BPSO is selected to enhance the efficiency of feature selection mode.

3. Proposing hybrid-ensemble learning based modular model

The classification of customers is done by the network which has modular based training and structure. A recursive process in such architectures uses separate models for estimating various parts of a problem. The general approach is

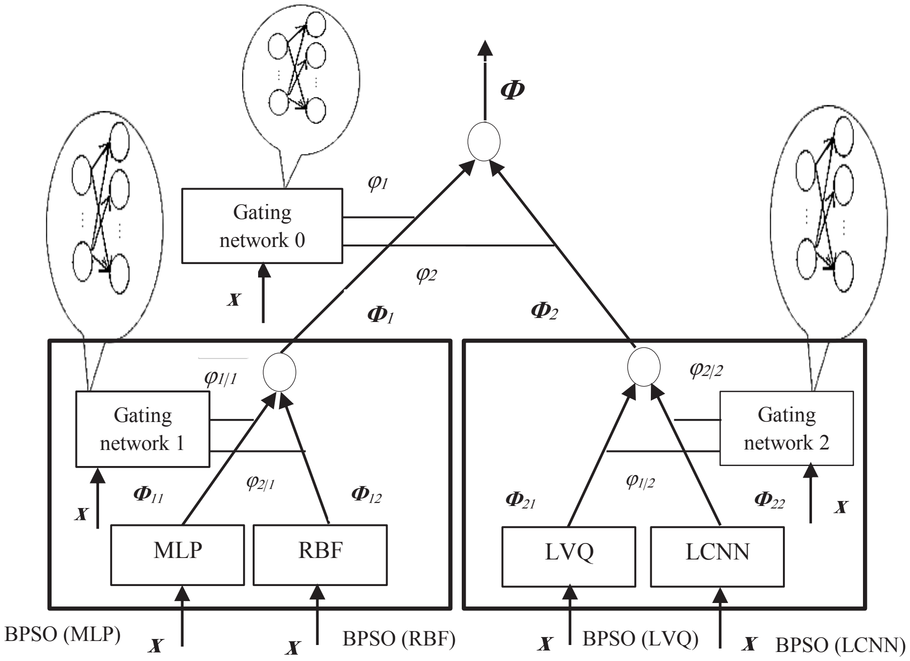

Fig. 1. Topology of the proposed modular model

to divide the problem into a set of sub-problems and allocate a set of experts to each sub-problem. Different approaches use different methods and outputs of experts to divide the problem into a sub-problem and calculate the best solution (Waterhouse & Robinson 1994).

The proposed model, named Adaptive HME (AHME) has modular structure and is composed of the ANNs. In fact, it combines the learning rules of each module and the HME's original learning rule. All the classifiers, including the gating and the expert networks are trained concurrently. Usually, the modular structure is formed either on the data or on the features of the data. On the other hand, in one approach, the data is broken and is used in different modules of the network; in another approach, the subsets of features are chosen and introduced to different modules of the network.

In this paper, the second approach is considered. Three gating networks, which function to integrate the outputs of each level, are generalized linearly. The expert networks are feed-forward ANNs. Accordingly, the following ANNs are used:

1. MLP neural network (Karray et al. 2004):

2. RBF neural network (Ahmadi et al. 2010; Tran & Duong 2017)

3. Laterally Connected Neural Network (LCNN) (Kothari and Agyepong 1996)

4. Learning Vector Quantizer (LVQ) Neural Network (Tang et al. 2007)

The main reason of applying these networks is their reputation and power in classifying problems of previous studies done in this field. Of course, the LCNN is selected because of its novelty in the field of CS models.

The way the expert networks are placed alongside each other is based on their ability in dealing with the underlying problems. This assumption causes the equilibrium to be established at the output of left and right modules. In this study, due to high volume of computing, we assume the same structure for all expert networks in all layers. All networks, including the gating and the experts have input units according to the number of features (customers' characteristics). The number of output units in the expert networks is the same as the number of categories, while it is equal to the number of the expert networks in each module in gating networks (Valdovinos et al. 2006, Alaraj et al. 2014). The hidden layer equals to unity and has variable number of neurons regarding the review of related works. Furthermore, each network's hidden layer structure is optimized through the several runs of individual network.

4. The AHME model training process

4.1. Particle Swarm Optimization (PSO)

The PSO algorithm is an evolutionary and meta-heuristic search algorithm that uses two strategies to find the best local and global solution. Two strategies are exploration and exploitation. The first one is the ability of expanding search space, whilst the next is the ability of finding the optimum throughout a good solution (Behesti et al. 2015).

PSO method begins with a population of particles in M-dimensional space and proceeds through a number of iterations to find an optimal solution. Each particle k □[1… R] is distinguished from other particles via velocity vector and its position, denoted by s_{km} and p_{km} , respectively. In order to select a new velocity, each particle considers three components; previous velocity, individual best position, and global best position. The individual best and global positions are called $p_{km}{}^{b}$ and $p_{km}{}^{*}$, respectively (Ahmadi et al. 2012). Accordingly, new velocity vector and positions are updated as follows:

$$\boldsymbol{s}_{km}(t+1)=w\boldsymbol{s}_{km}(t)+c_1.r_1\left(\boldsymbol{p}^{b}_{km}(t)-\boldsymbol{p}_{km}(t)\right) + c_2.r_2\left(\boldsymbol{p}^{*}_{m}(t)-\boldsymbol{p}_{km}(t)\right) \tag{۱}$$

$$\boldsymbol{p}_{km}(t+1)=\boldsymbol{p}_{km}(t)+\boldsymbol{s}_{km}(t+1) \tag{۲}$$

where w is the inertia weight, c_1 and c_2 are cognitive and social components, respectively; r_1 and r_2 are generated randomly using a uniform distribution in interval [0,1]. In addition to c_1 and c_2 parameters, implementation of initial algorithm requires a limit for vector velocity i.e., s_{max} (Chuang et al. 2011; Netjinda et al. 2015; Unler and Alper 2010; Hamada & Hasan 2018).

4.1.1. BPSO algorithm

For binary space, researchers adjusted PSO for searching in binary spaces by using a sigmoid convertor to convert velocities into zero and one to induce the values of particle positions to take up zero or one values. For selecting a subset of features, we determine the position of a particle as a binary vector. Accordingly, m represents the total number of features in the main dataset (Unler & Alper 2010; Vieira et al. 2013). The binary vector is $p_{km}^{(t)} = (p_{k1}(t), p_{k2}(t),\ldots,p_{km}(t))$ where $p_{km}(t)=1$ if feature m is included in the feature subset; zero, otherwise (Unler and Alper 2010; Vieira et al. 2013). Therefore, the equation of updating position in continuous PSO is replaced by following equations:

$$sigmoid(\boldsymbol{s}_{km}(\boldsymbol{t}))=\frac{1}{1+\boldsymbol{e}^{-\boldsymbol{s_{km}(t)}}} \tag{۳}$$

where $s_{km}(t)$ indicates the probability that the mth bit in $p_{km}(t)$ is one. Now, the updating equation of the position of the particle in the BPSO is as follows:

$$\boldsymbol{p}_{km}(t)=\begin{cases} 1, & if\ \boldsymbol{r_3}<sigmoid(\boldsymbol{s}_{km}(t)) \\ 0, & otherwise \end{cases} \tag{۴}$$

$$m=1,\ldots,M,$$

where r_3 is a uniform random number in interval [0,1].

Inertia weight has a direct impact on diversity. As the most common application of continuous and binary PSO implementation, it begins with a large quantity and then reduces in a fixed rate as the algorithm proceeds (Unler & Alper 2010). For each iteration of the algorithm, it is kept updated by:

$$w^{t+1}=w_{max}-\frac{(w_{max}-w_{min})t}{Q} \tag{۵}$$

where w_{max} and w_{min} are the limits on the w and Q is the maximum number of iterations.

The process of data introduction to the network starts by revealing the structure of expert networks and the gating one. In this model, features are distributed among the four expert networks based on the “soft split” method (Waterhouse and Robinson 1994). To do this, as shown in Fig. 1, firstly the dimension reduction or feature selection process via the BPSO algorithm (Babaoglu, et al. 2010; Vieira et al. 2013) for each network is done; then, the obtained features are introduced to AHME (Vieira et al. 2013; Xue et al. 2014). Meanwhile, the structure of each expert would be optimized. A pseudocode for exerting BPSO procedure onto the each expert network is provided in the following algorithm(s).

4.2. AHME learning

The topology of AHME network, shown in Fig. 2, is a tree where gating networks are placed in non-terminal nodes of the tree. They receive input vector x and generate scalar outputs that are fraction of a unit in each node (Jordan and Jacobs 1994; Waterhouse and Robinson 1994). Expert networks are placed as the leaves of the tree. Each expert network generates the output vector Φ_{ij} for each input vector which continues up the tree and is combined by the outputs of gating network. Expert network (i, j) generates the output Φ_{ij} as the generalized linear function from the input x:

$$\boldsymbol{\Phi}_{ij}=\boldsymbol{f}(\upsilon_{ij}\boldsymbol{x})\quad \boldsymbol{i}.\boldsymbol{j}=\mathbf{1}.\mathbf{2}. \tag{۶}$$

where “□” _”ij” and f (.) are selected as an initial weight matrix and a continuous non-linear function, respectively. The expert network outputs are clarified as the log chance of “success” under a Bernoulli distribution (Jordan & Jacobs 1994). The gating networks at next level are generalized linear. Intermediate variables ζ_{ij} are defined as follows:

$$\zeta_{ij}=\boldsymbol{v}_{ij}^{T}\boldsymbol{x}. \tag{۷}$$

like before, ν_{ij} is selected as an initial weight matrix of gating networks. Then, the output of each gating network at second level is:

$$\varphi_{j|i}=\frac{\exp(\zeta_{ij})}{\sum_k \exp(\zeta_{ik})} \tag{۸}$$

where $\varphi_{j|i}$ is the output of jth unit in the ith gating network at the second level of the network structure. It should be noted that the $\varphi_{j|i}$ s are positive and summation of them for each x is equal to one. Finally, at the top level of the network, we define intermediate variables ζ_i for gating network 0 as before:

$$\zeta_{i}=\boldsymbol{v}_{i}^{T}\boldsymbol{x}. \tag{۹}$$

where ν_i is a weight vector and the top level gating network ith output is actually the “softmax” function of the Eq. (9) (Jordan & Jacobs 1994):

$$\varphi_i=\frac{exp(\zeta_i)}{\sum_k exp(\zeta_k)}. \tag{۱۰}$$

Algorithm 1: Pseudocode for MLP, RBF, LVQ and LCNN training based on BPSO

```
initialize a swarm of size $R$
repeat
  for each particle $k \in [1, \dots, R]$
            if $p_{km}(t)$>rand then
      $p_{km}(t)$=1 else
      $p_{km}(t)$=0
    end if
    for each bit $m \in [1, \dots, M]$
        if $p_{km}(t)$=0 then
          $0 \leftarrow \boldsymbol{x}_i \; where \; i \in [main \; dataset]$
        end if
    end for
train the MLP network using back propagation learning algorithm,
train the RBF network using two-stage learning strategy algorithm,
train the LVQ network using LVQ1 learning algorithm and
train the LCNN network using straight forward back propagation learning algorithm with $\boldsymbol{x}_i$
  do update position and velocity
     if MSE($p_{km}(t+1)$)<MSE($p^b_{km}(t)$) then
       $p^b_{km}(t+1) \leftarrow p_{km}(t+1)$
     end if
   end for
$p^*_m\,(t+1) \;\min_{p^b_{km}(t)} \{MSE(p^b_{km}(t)) | k \in [1, \dots, R]\}$
until termination condition is met
obtain the selected features and optimum structure of the MLP, RBF, LVQ and LCNN
```

As before, φ_is are always positive and because of their nature, their total for each x equals one. It can be also determined as a fraction of the input space. At each non-terminal point, as illustrated in Fig. 2, the output vector is the weighted output of the expert networks. Thereupon, in the second layer of the two level trees, the output at the ith non-terminal node is defined as follows:

$$\boldsymbol{\Phi}_i = \sum_j \varphi_{j|i} \boldsymbol{\Phi}_{ij}. \qquad (۱۱)$$

The top-level output of the tree is as following equation:

$$\boldsymbol{\Phi} = \sum_i \boldsymbol{\varphi_i} \boldsymbol{\Phi_i}. \qquad (۱۲)$$

Note that both φ and "Φ" depend on input x, even if the total output is a non-linear function of x (Jordan & Jacobs 1994).

The AHME training process can be interpreted as a maximum likelihood estimation. Accordingly, the total probability of the production γ from x is actually combination of the probabilities of the production γ from the component distributions apiece, where the combined components are polynomial probabilities. Therefore, they are as follows:

$$P(\gamma|\boldsymbol{x}, \vartheta) = \sum_i \varphi_i(\boldsymbol{x}, \boldsymbol{\nu}_i) \sum_j \varphi_{j|i}(\boldsymbol{x}, \boldsymbol{\nu}_i) P(\gamma|\boldsymbol{x}, \vartheta_{ij}) \qquad (۱۳)$$

The output y is a discrete random variable in binary classification problem having possible outcomes of "success" and "failure". Hence, the $P(\gamma|x, □_{ij})$ in the above equation is usually considered Bernoulli distribution. In this case,

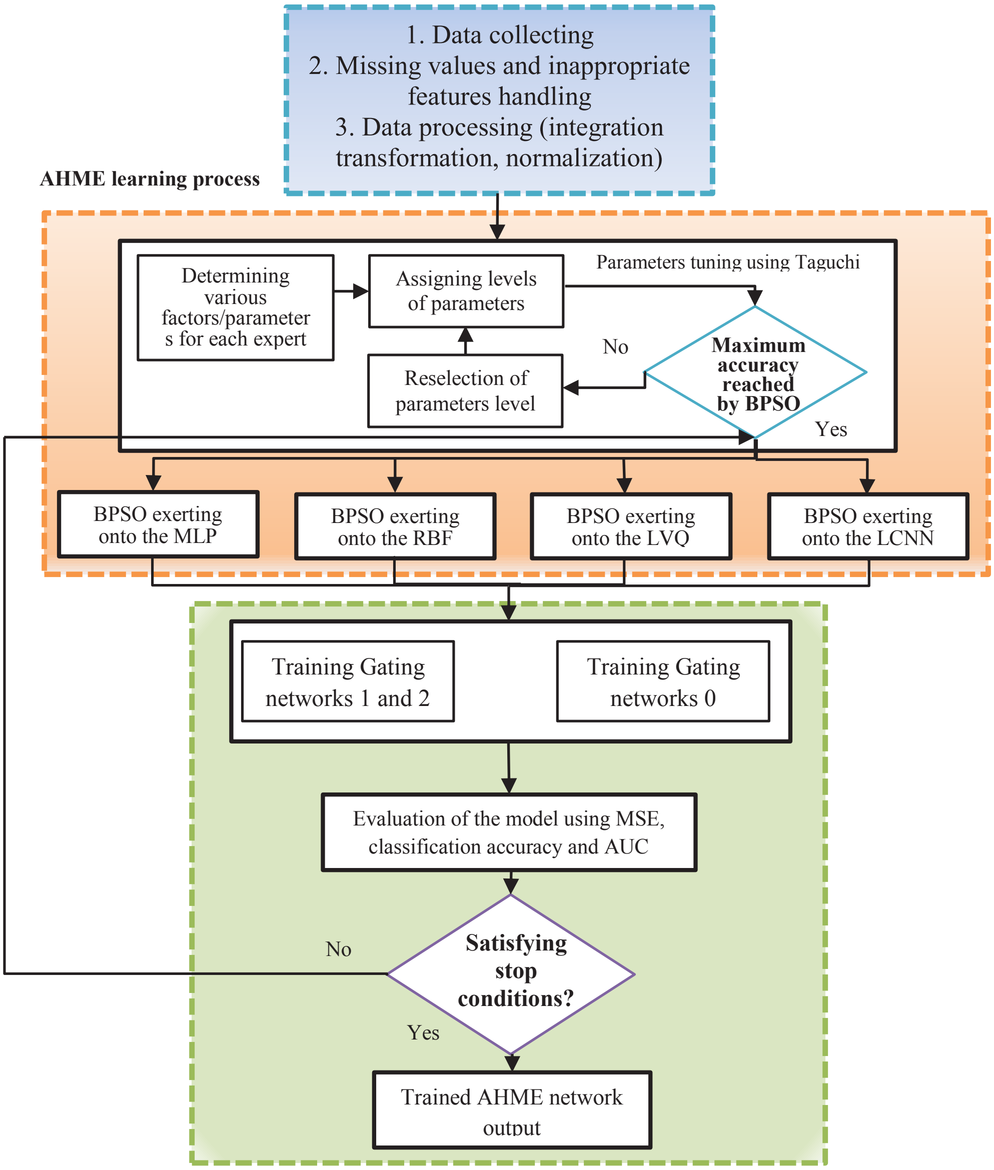
1. Data collecting
2. Missing values and inappropriate features handling
3. Data processing (integration transformation, normalization)
AHME learning process
Determining various factors/parameters for each expert
Assigning levels of parameters
Parameters tuning using Taguchi
No
Maximum accuracy reached by BPSO
Reselection of parameters level
Yes
BPSO exerting onto the MLP
BPSO exerting onto the RBF
BPSO exerting onto the LVQ
BPSO exerting onto the LCNN
Training Gating networks 1 and 2
Training Gating networks 0
Evaluation of the model using MSE, classification accuracy and AUC
No
Satisfying stop conditions?
Yes
Trained AHME network output

Fig. 2. A hybrid-ensemble learning credit scoring system

the mean Φ_{ij} is the conditional probability of grouping the input as "success". Equation (13) by replacing the Bernoulli distribution will be as follows:

$$P(\gamma|\boldsymbol{x},\vartheta)=\sum_i \varphi_i \sum_j \varphi_{j|i}\, \boldsymbol{\Phi}_{ij}^{\gamma}(1-\ \boldsymbol{\Phi}_{ij})^{1-\boldsymbol{\gamma}}. \qquad (۱۴)$$

Thus, the log likelihood of a dataset $[x^{(t)}, \gamma^{(t)}]$ is achieved with respect to the log of the product for N distributions of the above equation, which results in the subsequent expression:

$$l(\boldsymbol{\vartheta};\boldsymbol{x})=\sum_t \ln \sum_i \varphi_i^{(t)} \sum_j \varphi_{j|i}^{(t)} P_{ij}(\gamma^{(t)}). \qquad (۱۵)$$

Now, the purpose is the minimization of this function. There are various methods for which one can refer to the EM, least-squares algorithm and gradient descent. In this paper, the gradient descent method is used for training the network (Jordan & Jacobs 1994).

By differentiating $l(□;x)$, the gradient descent learning rule will be obtained for the weight matrix υ_{ij} :

$$\Delta \upsilon_{ij}=\ \omega \sum_t \lambda_i^{(t)} \lambda_{j|i}^{(t)} (\gamma^{(t)}-\boldsymbol{\Phi}^{(t)})\boldsymbol{x}^{(t)T} \qquad (۱۶)$$

where "ω" is a network learning rate. This learning rule for the ith weight vector in the gating network at the highest level is obtained as:

$$\Delta \boldsymbol{\nu}_i=\ \omega \sum_t (\lambda_i^{(t)}-\varphi_i^{(t)})\boldsymbol{x}^{(t)} \qquad (۱۷)$$

Similarly, that is for the jth weight vector in the gating network at the other levels as follows:

$$\Delta \boldsymbol{v}_{ij}=\omega \sum_t (\lambda_{j|i}^{(t)}-\varphi_{j|i}^{(t)})\boldsymbol{x}^{(t)}. \qquad (۱۸)$$

In equations (16), (17) and (18), λ_i and $\lambda_{j|i}$ are the posterior probabilities by using Bayes' rule at the nodes of the tree given by:

$$\lambda_i=\frac{\varphi_i \sum_j \varphi_{j|i} P_{ij}(\gamma)}{\sum_i \varphi_i \sum_j \varphi_{j|i} P_{ij}(\gamma)} \qquad (۱۹)$$

and

$$\lambda_{j|i}=\frac{\varphi_{j|i}\mathrm{P}_{\mathrm{ij}}(\gamma)}{\sum_{\mathrm{j}} \varphi_{j|i}\mathrm{P}_{\mathrm{ij}}(\gamma)} \qquad (۲۰)$$

It is worth mentioning that φ_i and $\varphi_{j|i}$ are the prior probabilities because they are calculated on the basis of the input x without any knowledge of the related target output γ. A posterior probability is also delineated once, based on both the known target input and output (Jordan & Jacobs 1994; Waterhouse & Robinson 1994).

Now, according to the previous sections, input vector x is the same for all networks. The task of gating networks is learning from a suitable harmonious fusion of expert networks for each input vector. The training procedure is performed on the basis of a hybrid approach. Expert and gating networks undergo training simultaneously, in the sense that hybrid learning process begins after application of BPSO algorithm in each expert network and the selection of superior features for the network. Through this way, each expert network based on algorithms specialized for each expert network, i.e. back propagation algorithm for MLP network, two-stage learning strategy for RBF network, straight forward back-propagation algorithm for LCNN, and LVQ1 algorithm for LVQ network yields the initial output of each expert. The gating networks in right order together with expert on the basis of forward learning algorithm of each network produce the output. In what follows, using Eq. (11) coming from each of non-terminal points of the tree in the second layer and Eq. (12), the final output of network is obtained at the highest level of the tree. Up to this point, a training epoch is past for AHME learning, and the process of weight updating and network training begins by checking the error rate of MSE network learning. Afterward, posterior probabilities (Eq. (19) and (20)) are calculated by using the outputs from previous stage. Then, the update of the weights for gating networks is undertaken by Eq. (17) and (18) for ith and jth weight vectors at final and lower level of tree, respectively. Together by learning the gating networks, the weight update for expert networks is performed by algorithms specialized for each one, and the weights of network are adjusted, and the process iterates until the conditions for ceasing training are met. The above-mentioned procedure along with the learning process is entirely shown in Fig. 2.

Table 1.
Total view of dataset state.

No. of Samples	Good Credit	Bad Credit
108	82	26

5. Empirical procedure

5.1. Description of real-world credit dataset

Used dataset belongs to various corporate customers of a well-known Iranian state bank. They are related to corporate

Algorithm 2: Hybrid learning algorithm for AHME model

1. Provide the input vector $\boldsymbol{x}$ for each input data based on its corresponding features to expert and gating networks.
2. Get the outputs of expert networks and gating 1 and 2 based on the rules related to each network.
3. Compute the $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ according to the equation (11).
4. Compute the output of gating network 0 according to the related rules.
5. Compute the total output of the network based on the equation (12).
6. Compute the error value according to the total output and MSE function.
7. Compute the posterior probabilities for each data using the current parameters values.
8. Update the gating networks' weights using equations (17) and (18).
9. Update the expert networks' weights using the learning rules of each network.
10. Repeat the steps 1 to 9 for other samples.
11. Check if the cumulative MSE has been less than a certain value, then the network is trained; otherwise, repeat the training process for another epoch.

customers, who have higher importance than others. An overview of the used dataset can be seen in Table 1.

5.2. Research variables

The research variables are as follows:

1. Dependent variable (q): Given that credit customers have been divided into two categories of good and bad credit, this variable can own two states of zero and one. If a customer has defaulted in refunding the loans and has had delayed paying back the loans, q is equal to zero. If loans refund are done in due date, q is equal to one.

2. Independent variables: They include those variables that influence on customer's credit risk or in other words, affect the dependent variable of model. Therefore, according to the conducted investigations, a set of variables that can somehow influence on dependent variable of the model are:
1. Working capital (current liabilities minus current assets): x1,
2. Current liability ratio (current liabilities to assets): x2,
3. Equity ratio (equity to total assets): x3,
4. Asset ratio (current asset to total assets): x4,
5. Acid test ratio (current assets minus inventory to current liability): x5,
6. Cash flow ratio (cash and deposits to current liability): x6,
7. Cash and deposits to total assets: x7,
8. Current ratio (current asset to current liability): x8,
9. Liability ratio (total liabilities to total assets): x9,
10. Profit margin (net profit to sales): x10,
11. Return on Asset (net profit to total assets): x11,
12. Retained profit to total assets: x12,
13. Liability to equity ratio: x13,
14. Total bank loans to total assets: x14,
15. Total bank loans to total liabilities: x15,
16. Net sales to total assets: x16,
17. Inventory to net sales: x17,
18. Current liabilities to net sales: x18.

5.3. Tuning the parameters of BPSO by Taguchi method

The distinctive feature of Taguchi method is to guide experiments using Orthogonal Array (OA) technique and implement statistical analysis by applying the Signal to Noise Ratio (SNR). The SNR, applied to investigate the impact of the noise factors on the robustness of a system, is a performance measure for assessing the quality of a product or a process. By optimizating SNR, a product or a process will have better robustness and the effect of the noise factors on it will be minimized actually (Wang et al. 2017).

For the problem under consideration, the aim is to determine the optimized parameters of BPSO algorithm used for ANNs to perform dimensional reduction process and select the best features of dataset taken for each expert network. Factors that make a significant difference to performance should be selected. The characteristic of Taguchi method is the ability to take account of uninfluential variables, even if they were viewed as influential in the beginning of the optimization process. We take five parameters into account in our experiments; the five parameters or factors are displayed by A to E, for the sake of simplicity. The relevant parameters and level of each operational parameter are listed in Table 2. For our experiments, the matrix L16 is selected which represents 16 tests with five four-level factors. OA selected is displayed in Table 3. Response variable for every expert network was chosen as a fitness function of BPSO which is equal to the difference between the mean of classification accuracy and the mean of MSE in training process. Therefore, the approximate value of response close to the number one offers a better solution.

Some of the parameters that remained fixed during the experiments are: swarm size=30, iterations=600. The calculated SNRs are summarized in Table 3.

The response of each test is shown by R1 to R16, and the average rate of response for each factor is computed at each level. Considering the subject matter, for each factor and different levels, the inputs of response table and vector of response will be produced. Response graphs are exhibited by different factors in Fig. 3 and Fig. 4.

Table 4 includes ranks based on Delta statistic that compares the relative size of effects. Delta statistic is the difference between the maximum and minimum mean for each factor. We use the means of level in response table in order to determine which level of each factor yields the best result.

Obviously, wmin, [c1,c2], wmin and smax have the greatest effect on the SNR in MLP, RBF, LCNN and LVQ, respectively. From response table or response vector, the optimized level of each factor can be predicted as a level with the maximum value of SNR. The means of each level are shown in Table 3, in that SNRs are maximized when the optimized structure of BPSO parameters is determined for

Table 2
Factors and the levels of different parameters.

Factor	Corresponding parameter of BPSO	Level			
		1	2	3	4
A	$[c_1,c_2]$	[2,2]	[2,1.5]	[1.5,1]	[0.5,1.5]
B	w_{max}	1	0.9	0.8	0.7
C	w_{min}	0.4	0.3	0.2	0.1
D	W	1	0.9	2	1.5
E	s_{max}	0.8	1.1	1.5	2

Table 3
Response table for SNR ratios and matrix experiments with L16OA (For different Networks)

Experiment Number	Factor				Response	SNR			
	A	B	C	D		MLP	RBF	LCNN	LVQ
1	1.0	0.4	1.0	0.8	R1	-2.12920	-3.82498	-3.82363	-2.38601
2	0.9	0.3	0.9	1.1	R2	-2.15699	-3.93906	-3.96370	-2.39173
3	0.8	0.2	2.0	1.5	R3	-2.14142	-3.75912	-3.95137	-2.39173
4	0.7	0.1	1.5	2.0	R4	-2.13475	-3.83713	-3.84254	-2.39402
5	1.0	0.3	2.0	2.0	R5	-2.14698	-3.91312	-3.87912	-2.39288
6	0.9	0.4	1.5	1.5	R6	-2.13698	-3.96644	-3.84254	-2.38487
7	0.8	0.1	1.0	1.1	R7	-2.10150	-3.85471	-3.85607	-2.39173
8	0.7	0.2	0.9	0.8	R8	-2.13253	-3.94589	-3.98154	-2.39173
9	1.0	0.2	1.5	1.1	R9	-2.12476	-3.70839	-3.87912	-2.39288
10	0.9	0.1	2.0	0.8	R10	-2.12809	-3.77789	-3.80073	-2.38944
11	0.8	0.4	0.9	2.0	R11	-2.14253	-3.86555	-3.90631	-2.39402
12	0.7	0.3	1.0	1.5	R12	-2.16368	-3.84254	-3.98428	-2.38944
13	1.0	0.1	0.9	1.5	R13	-2.13920	-3.96507	-3.82498	-2.38487
14	0.9	0.2	1.0	2.0	R14	-2.11811	-3.84659	-3.96781	-2.39173
15	0.8	0.3	1.5	0.8	R15	-2.15477	-3.87504	-3.84389	-2.38716
16	0.7	0.4	2.0	1.1	R16	-2.09708	-3.89950	-3.86013	-2.39173

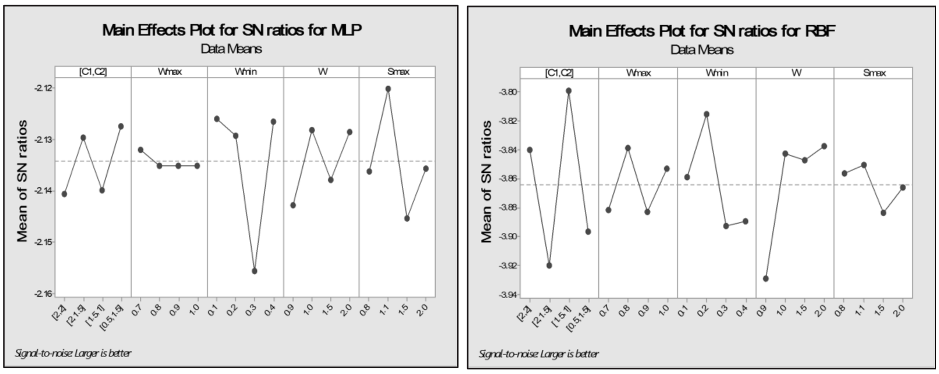

Fig. 3. Response graph for MLP & RBF network

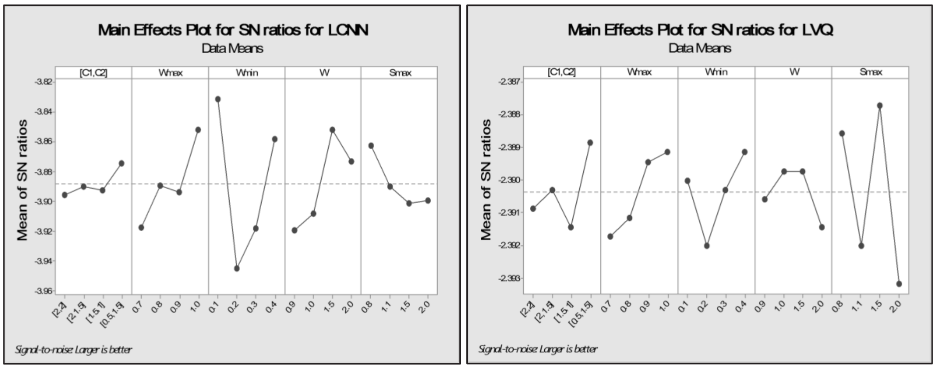

Fig. 4. Response graph for LCNN & LVQ network

Table 4.
Level of factor providing the best results.

		$[c_1,c_2]$	$w_{\max}$	$w_{\min}$	W	$s_{\max}$
MLP	Delta	0.013	0.003	0.030	0.015	0.025
	Rank	4	5	1	3	2
RBF	Delta	0.121	0.044	0.077	0.091	0.033
	Rank	1	4	3	2	5
LCNN	Delta	0.021	0.065	0.114	0.067	0.038
	Rank	5	3	1	2	4
LVQ	Delta	0.003	0.003	0.003	0.002	0.005
	Rank	3	4	2	5	1

Table 5.
Predicted best strategy parameters (for different networks)

	MLP	RBF	LCNN	LVQ
Factor (Level)	A (4) B (1) C (1) D (2) E (2)	A (3) B (2) C (2) D (4) E (2)	A (4) B (4) C (1) D (3) E (1)	A (4) B (4) C (4) D (2 or 3) E (3)

each network. The values of relevant parameters are shown in Table 5.

6. Experimental results and analysis

In this section, we examine the performance of the proposed model and its results are compared with MLP and LCNN models. The model's results are categorized based on efficiency and accuracy prediction, so the best model is selected. It should be noted that learning for the entire data was undertaken on the basis of 5-fold CV algorithm.

In phase one, the selected features for each module of the ANN were obtained according to the nature of the modular model. Results for each expert network are given in Table 6 separately. Omitted features are distinguished by ×.

In the next step and before providing the final results, prior probabilities (impact amount of each networks in training process) and posterior probabilities (the most important learning and network convergence factors), are shown in Table 7. (Jordan & Jacobs 1994; Waterhouse & Robinson 1994).

* In fact, prior probabilities are gating networks' outputs.

* φi and φj|i are interpreted as prior because they are only computed based on input x without any knowledge of the target value related to the output y.

* Numbers related to the prior probabilities indicate which network has the most contribution in modular learning of the AHME network.

* Numbers related to the posterior probabilities in the

Table 6.
Selected features in each module of the AHME

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
MLP		×									×	×	×	×	×		×	×
RBF										×	×		×	×	×	×		
LCNN	×			×		×		×	×					×	×			
LVQ								×	×	×	×			×		×		

Table 7.
Prior and posterior probabilities of AHME

Posterior Probabilities						Prior Probabilities			
λ_i		$\lambda_{j\|i}$				LVQ	LCNN	RBF	MLP
0.9936	0.0066	0.974	0.0251	1	0	0.9750	0.0250	0.0757	0.9243

Table 8.
Results for the proposed approach.

Prediction Accuracy of Training	Error Rate of Training	Prediction Accuracy of Testing	Error Rate of Testing
0.951220	0.0481	0.846154	0.1319

Table 9.
Results of different models

Model	Prediction Accuracy of Training	Prediction Accuracy of Testing	Error Rate of Training	Error Rate of Testing
MLP	0.936539	0.738632	0.0613	0.2328
LCNN	0.867631	0.723354	0.1079	0.1952

nodes of the tree are related to the gating network 0, 1 and 2 from left to right, respectively.

According to the obtained features of each network, data by selected features is presented to the networks. The performance of the proposed approach is given in Table 8 according to the testing and training data.

Table 9 shows the results of MLP and LCNN with 8 neurons in hidden layer. Final prediction of networks is given based on the training and test datasets.

The results of MLP and LCNN, their convergence and convergence of the AHME network in terms of training error rate are shown in Fig. 5. As it can be seen from the diagram trend, the AHME network, through very high precision than other models, has been able to do the anticipation during the 30 epochs only. It is clear that the AHME is a superior model based on both criteria of model complexity and time-solving according to Fig. 5.

ROC curve for three existing models based on the two sets of training and testing is drawn in Fig. 6 and Fig. 7. Based on these diagrams, AUC for training set of the AHME, LCNN and MLP models is 0.9993, 0.8680 and 0.9105 and for testing set is 0.6316, 0.5921 and 0.6222 respectively provided in Table 10.

We ranked the models in order to select and present the superior one due to the accuracy, AUC and error rate. As mentioned before, the ROC curves and their AUC were used for better comparison of performance. It is obvious that whatever the AUC is closer to one, the network model would have better performance. Accordingly, the AHME model has the best performance as shown in Fig. 8.

The MLP network could achieve the next rank and the LCNN model placed at the last rank because of acquiring worst values of two measures.

7. Conclusions and future directions

In this paper, a novel model for bank corporate customers' CS named AHME was proposed. Applying the hybrid learning rule instead of common learning rules gave the superb results compared with the MLP and LCNN credit scoring model. The proposed model is efficient due to the high efficiency on one hand and the very low error rate on the other, amongst others in the same field. The diagram of error rate comparison of each network was used in order to survey the model's efficiency for separating the customers into good credit and bad credit. Additionally, the analysis of the ROC curve was also used. The results confirm that the AHME

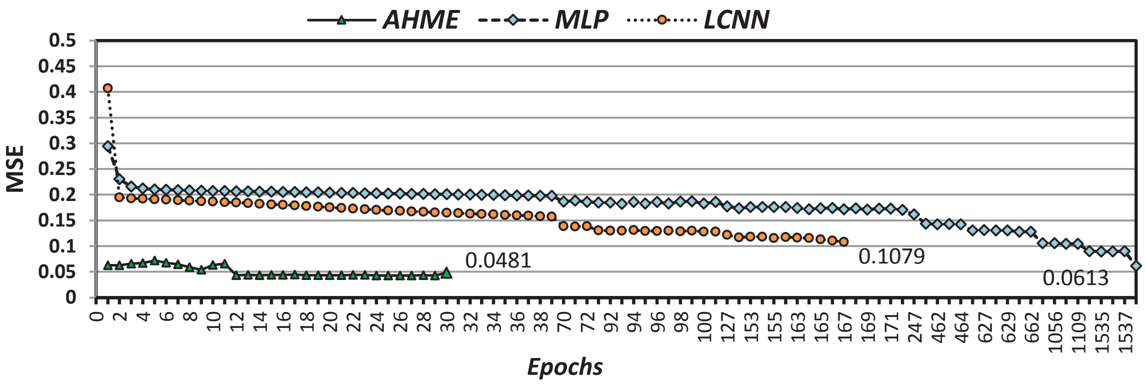

Fig. 5. Comparison diagram of error rate and convergence trend for all models

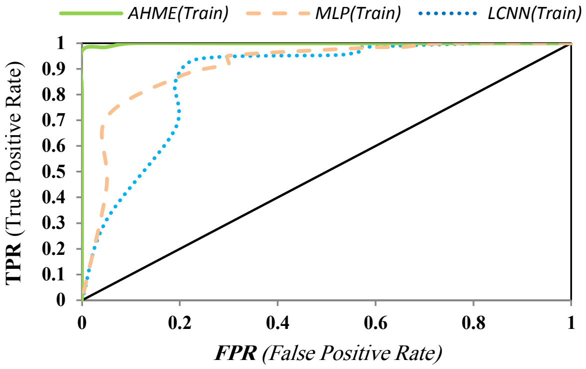

Fig. 6. ROC curve for training set of the models

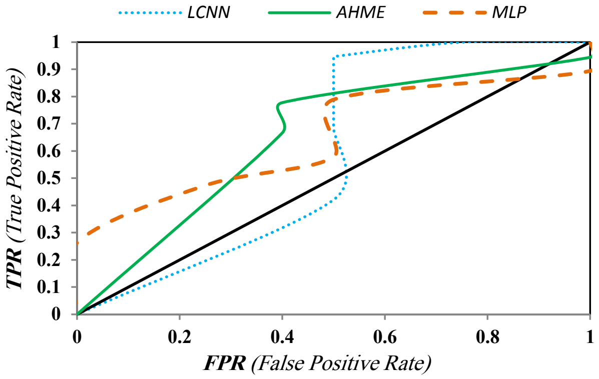

Fig. 7. ROC curve for testing set of the models

model is more efficient than others in the mentioned field. It is also more reliable for real world issues and its training-solving time is lower than other models significantly. It also indicates that the AHME could converge within 30 epochs. Actually, the MLP and LCNN does not have enough ability for CS the real-world problems like this study. Obviously, the premier model acts as a bidder to banks and the final decision is up to the authorities and assessors.

As for future research, one can focus on the CS of chequebook applicants. This paper was used for classifying the customers into two groups, namely good credit and bad credit. However, the central bank of Iran uses four categories including good, Non-Performing Loan (NPL) (who have two to six months debts), delayed due date (who have six to 18 months debts) and doubtful due date (who have more than 18 months debts) customers. Extending the proposed approach to four-class is another opportunity for future research. Future studies could also be extended by (a) using different credit datasets with different sizes and attributes for extra validation, (b) using different ensemble methods and strategies to learn more diversified sets (c) using different machine learning methods such as SVM and DT.

Table 10
AUC obtained for three models.

Model	AUC	
	Training data	Testing data
LCNN	0.8680	0.5921
AHME	0.9993	0.6316
MLP	0.9105	0.6222

References

Ahmadi, A., Fakhri, K., & Kamel, M. S. 2010. “Flocking Based Approach for Data Clustering.” Natural Computing 9(3): 767–91.

Ahmadi, A., Fakhri, K., & Kamel, M. S. 2012. “Model Order Selection for Multiple Cooperative Swarms Clustering Using Stability Analysis.” Information Sciences 182(1): 169–83.

Akkoç, S. 2012. “An Empirical Comparison of Conventional Techniques, Neural Networks and the Three Stage Hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) Model for

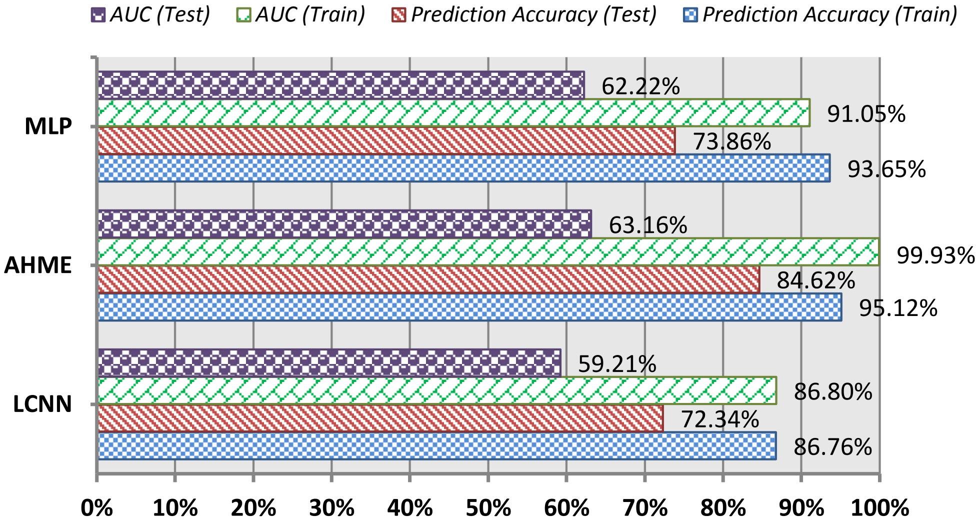

Fig. 8. The results of the three models considering the prediction accuracy and the AUC based on the training and testing set

Credit Scoring Analysis: The Case of Turkish Credit Card Data." European Journal of Operational Research 222(1): 168–78.

Alaraj, M., Abbod, M., & Hunaiti, Z. 2014. "Evaluating Consumer Loans Using Neural Networks Ensembles." International Conference on Machine Learning, Electrical and Mechanical Engineering (ICMLEME), http://dx.doi.org/10.15242/IIE.E0114084.

Ayouche, S., Aboulaich, R., & Ellaia, R. 2017. "Partnership Credit Scoring Classification Probem: A Neural Network Approach." International Journal of Applied Engineering Research, 12(5): 693–704.

Babaoglu, İ., Findik, O., & Ülker, E. 2010. "A Comparison of Feature Selection Models Utilizing Binary Particle Swarm Optimization and Genetic Algorithm in Determining Coronary Artery Disease Using Support Vector Machine." Expert Systems with Applications 37(4): 3177–83.

Behesti, Z., Shamsuddin, S., M., & Hasan, S. 2015. "Memetic binary particle swarm optimization for discrete optimization problems." Information Sciences Journal, 299: 58–84.

Chuang, L.,Y., Hsiao, C., J., & Yang, C., H. 2011. "An Improved Binary Particle Swarm Optimization with Complementary Distribution Strategy for Feature Selection." In International Conference on Machine Learning and Computing, 244–48.

Hamada, M., & Hasan, M. 2018. "Artificial Neural Networks and Particle Swarm Optimization Algorithms for Preference Prediction in Multi-Criteria Recommender Systems." Informatics Journal, 5(2):25.

Jordan, M., I., & Jacobs, R., A. 1994. "Hierarchical Mixtures of Experts and the EM Algorithm." Neural computation 6(2): 181–214.

Kambal, E., Osman, I., Taha, M., Mohammed, N., & Mohammed, S. 2013. "Credit Scoring Using Data Mining Techniques with Particular Reference to Sudanese Banks." Inernational Confrance on Computer, Electrrical and Electronics Engineering (ICCEEE), 378 - 383.

Karray, F., O., & De Silva, C. 2004. Soft Computing and Intelligent Systems Design: Theory, Tools, and Applications. Pearson/Addison Wesley. https://books.google.com/books?id=mqYw-Xig0IsC.

Kiruthika & Dilsha, M. 2015. "A Neural Network Approach for Microfinance Credit Scoring." Journal of Statistics and Management Systems, 18:1-2, 121-138.

Kothari, R., & Agyepong, K. 1996. "On Lateral Connections in Feed-Forward Neural Networks." In Neural Networks, 1996., IEEE International Conference on, IEEE, 13–18.

Limsombunchai, V., Gan, C., & Lee, M. 2005. "An Analysis of Credit Scoring for Agricultural Loans in Thailand." American Journal of Applied Sciences 2(8): 1198.

Netjinda, N., Achalakul, T., & Sirinaovakul, B. 2015. "Particle Swarm Optimization Inspired by Starling Flock Behavior." Applied Soft Computing 35: 411–22.

Ong, C., S., Huang, J., J., & Tzeng, G., H. 2005. "Building Credit Scoring Models Using Genetic Programming." Expert Systems with Applications 29(1): 41–47.

Rafiee Vahid, P., & Ahmadi, A. 2016. "Modelling Corporate Customers Credit Risk Considering the Ensemble Approaches in Multiclass Classification : Evidence from Iranian Corporate Credits." To appear in The Journal of Credit Risk.

Shayeghi, H., Shayanfar, H., A., & Azimi, G. 2010. "A Hrbrid Particle Swarm Optimization Back Prppagation Algorithm for Short Term Load Forcasting." International Journal on Technical and Physical Problems of Engineering(IJTPE), 4(2): 12–22.

Tang, Q., Liu, B., H., Chen, Y., Q., & Ding, J., J. 2007. "Application of LVQ Neural Network Combined with the Genetic Algorithm in Acoustic Seafloor Classification." Chinese Journal of Geophysics 50(1): 291–98.

Tran, K., & Duong, T. 2017. "The Application of Radial Basis Function (RBF) Neural Network for Mechanical Fault Diagnosis of Gearbox." Future Technologies Conference (FTC), 145-149.

Unler, A., & Alper, M. 2010. "A Discrete Particle Swarm Optimization Method for Feature Selection in Binary Classification Problems." European Journal of Operational Research 206(3): 528–39.

Valdovinos, R., M., & Sanchez, J., S. 2006. "Ensembles of Multilayer Perceptron and Modular Neural Networks for Fast and Accurate Learning." In Artificial Intelligence, 2006. MICAI'06.

Vieira, S., M., Mendonça, L., F., Farinha, G., J., & Sousa, J., M., C. 2013. "Modified Binary PSO for Feature Selection Using SVM Applied to Mortality Prediction of Septic Patients." Applied Soft Computing Journal 13(8): 3494–3504.

Wang, H., Geng, Q., & Qiao, Z. 2017. "Parameter Tuning of Particle Swarm Optimization by Using Taguchi Method and Its Application to Motor Design." 4th IEEE International Conference on Information Science and Technology, Shenzhen, 722-726.

Wang, P. 2016. "Credit Scoring Model: A Combination of Genetic Programming and Deep Learning." IOP Conference Series: Materials Science and Engineering, V(269), https://doi.org/10.1088/1757-899X/269/1/012056.

Waterhouse, S., R., & Robinson, A., J. 1994. "Classification Using Hierarchical Mixtures of Experts." In Neural Networks for Signal Processing [1994] IV. Proceedings of the 1994 IEEE Workshop, IEEE, 177–86.

Xue, B., Zhang, M., & Browne, W., N. 2014. "Particle Swarm Optimisation for Feature Selection in Classification: Novel Initialisation and Updating Mechanisms." Management Science and Engineering, 7(3), 81-85.

Yarushev, S., & Averkin, A. 2017. "Time series analysis based on the biologically inspired modular approach." Procedia Computer Science Journal, V(120): 843–853.

HOW TO CITE THIS ARTICLE

D. Dadmohammadi and A. Ahmadi, A Combined Learning Approach for Credit Scoring Using Adaptive Hierarchical Mixture of Experts: Iranian Banking Industry, AUT J. Model. Simul., 51(1) (2019): 67 - 80.

DOI: 10.22060/miscj.2019.13906.5085