Expert Systems with Applications 39 (2012) 2606–2620

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A novel hybrid classification model of artificial neural networks and multiple linear regression models

Mehdi Khashei *, Ali Zeinal Hamadani, Mehdi Bijari

Department of Industrial Engineering, Isfahan University of Technology, Isfahan, Iran

ARTICLE INFO

Keywords:
Classification
Pattern recognition
Artificial neural networks (ANNs)
Multiple linear regression models (MLR)
Discriminant analysis (DA)

ABSTRACT

The classification problem of assigning several observations into different disjoint groups plays an important role in business decision making and many other areas. Developing more accurate and widely applicable classification models has significant implications in these areas. It is the reason that despite of the numerous classification models available, the research for improving the effectiveness of these models has never stopped. Combining several models or using hybrid models has become a common practice in order to overcome the deficiencies of single models and can be an effective way of improving upon their predictive performance, especially when the models in combination are quite different. In this paper, a novel hybridization of artificial neural networks (ANNs) is proposed using multiple linear regression models in order to yield more general and more accurate model than traditional artificial neural networks for solving classification problems. Empirical results indicate that the proposed hybrid model exhibits effectively improved classification accuracy in comparison with traditional artificial neural networks and also some other classification models such as linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), *K*-nearest neighbor (KNN), and support vector machines (SVMs) using benchmark and real-world application data sets. These data sets vary in the number of classes (two versus multiple) and the source of the data (synthetic versus real-world). Therefore, it can be applied as an appropriate alternate approach for solving classification problems, specifically when higher forecasting accuracy is needed.

1. Introduction

Classification is an important area of research that concerned with assigning an object to one of a set of classes, based upon attributes of that object. The performance of the classification process is dependent on how well the discriminant function for the specific problem performs. A discriminant function is developed to minimize the misclassification rate, according to some given samples of input and output vector couples, which are referred to as training data set. This discriminant function is then used for classifying new observations into previously defined groups and for testing the accuracy of the classification. Classification problems have been examined in fields as diverse as business, medicine, biology, image recognition, etc. and using of these models has become more indispensable in aforementioned areas, especially in business and finance. Several different classification approaches have been proposed in the literature since the earliest work of Fisher (1936). The classification approaches are generally categorized in two main categories, linear and nonlinear approaches.

Linear classification approaches partition the input space into a collection of disjoint regions, separated by linear decision boundaries. Notable examples of linear classification techniques that have been widely used in classification include those by multiple linear regression (MLR), linear discriminant analysis (LDA), logistic regression, separating hyper planes, etc. These classification techniques work well when the classes are linearly separable. However, in many real world problems the data may not be linearly separable and also data are very closely spaced and therefore a highly nonlinear decision boundary is required in order to separate the data (Satapathy et al., 2009). Several classes of nonlinear classification techniques have been proposed in the literature in order to overcome the linear limitation of the linear classification techniques. These techniques include those by the classical techniques such as quadratic discriminant analysis (QDA), *K*-nearest neighbor (KNN), etc. and artificial neural networks approaches such as neural trees, multilayer perceptrons (MLPs), probabilistic neural networks (PNNs), support vector machines (SVMs), etc.

Artificial neural networks are one of the most accurate and widely used classification techniques that have enjoyed fruitful applications in many areas. Several distinguishing features of artificial neural networks make them valuable and attractive for

* Corresponding author. Tel.: +98 311 3912550 1; fax: +98 311 3915526.
E-mail address: khashei@in.iut.ac.ir (M. Khashei).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.08.116

classification tasks. First, as opposed to the traditional model-based techniques, artificial neural networks are data-driven self-adaptive methods in that there are few a priori assumptions about the models for problems under study. Second, artificial neural networks can generalize. After learning the data presented to them (a sample), artificial neural networks can often correctly infer the unseen part of a population even if the sample data contain noisy information. Third, artificial neural networks are universal functional approximators. It has been shown that a network can approximate any continuous function to any desired accuracy. Finally, artificial neural networks are nonlinear (Khashei, Hejazi, & Bijari, 2008).

Given the advantages of artificial neural networks, it is not surprising that this methodology has attracted overwhelming attention in classification (Maulik & Mukhopadhyay, 2010). Artificial neural networks have been found to be a viable contender to various traditional classification models in many different areas (Dubois, Bohling, & Chakrabarti, 2007; Kara & Okandan, 2007). Castellani and Rowlands (2009) address the design and the training of a multilayer perceptron classifier for identification of wood veneer defects from statistical features of wood sub-images. Kruzlicova et al. (2009) demonstrate the possibility of using artificial neural networks for the Slovak white wines classification. Olmez and Dokur (2003) propose using the artificial neural networks in order to handle the heart sounds classification problems and to increase the classification performance.

Banerjee, Kiran, Murty, and Venkateswarlu (2008) present an artificial neural system for classification and identification of Anopheles mosquito species based on the information content of ribosomal DNA sequences. Acharya, Bhat, Iyengar, Rao, and Dua (2003) deal with the classification of certain diseases using artificial neural network and fuzzy equivalence relations. The heart rate variability is used as the base signal from which certain parameters are extracted and presented to the artificial neural network for classification. Guven and Kara (2006) concentrate on the diagnosis of subnormal eye through the analysis of electro-oculography (EOG) signals with using of the artificial neural network. Fingerprints classification (Nagaty, 2001), cervical cancer classification (Qiu, Tao, Tan, & Wu, 2007), protein structure classification (Karci & Demir, 2009), gait classification in post-stroke patients (Kaczmarczyk, Wit, Krawczyk, & Zaborski, 2009), bankruptcy prediction problem (Pendharkar, 2005), Tamil documents classification (Rajan, Ramalingam, Ganesan, Palanivel, & Palaniappan, 2009) are some other successful applications of the artificial neural networks in comparison with other those of the traditional classification models.

Although artificial neural networks are flexible computing frameworks and universal approximators that can be applied to a wide range of forecasting problems with a high degree of accuracy, their performance in some specific situations such as linear problems is inconsistent (Khashei & Bijari, 2010). In the literature, several papers are devoted to comparing artificial neural networks with linear models. Despite of the several studies, which have shown artificial neural networks are significantly better than the conventional linear models and their results considerably and consistently more accurately, some other studies have reported inconsistent results (Zhang, Patuwo, & Hu, 1998). Some researchers believe that in some specific situations where artificial neural networks perform worse than linear statistical models, the reason may simply be that the data is linear without much disturbance, therefore; cannot be expected that artificial neural networks to do better than linear models for linear relationships (Khashei & Bijari, 2010). However, for whatever reason, using artificial neural networks to model linear problems have yielded mixed results; and hence, it is not wise to apply neural networks blindly to any type of data.

Both multiple linear regression and artificial neural networks models have achieved successes in their own linear or nonlinear domains. However, none of them is a universal model that is suitable for all circumstances. The approximation of the multiple linear regression models to complex nonlinear problems as well as artificial neural networks to model linear problems may be totally inappropriate, and also, in problems that consist both linear and nonlinear correlation structures. Using hybrid models or combining several models has become a common practice in order to overcome the limitations of each component model (Khashei, Bijari, & Raissi, 2009). The basic idea of these multi-model approaches is the use of each component model's unique capability to better capture different patterns in the data. In addition, since it is difficult to completely know the characteristics of the data in a real problem, hybrid methodology that has both linear and nonlinear modeling capabilities can be a good strategy for practical use.

In the literature, different combination techniques have been proposed in order to overcome the deficiencies of single classification models and yield more accurate results (Hur & Kim, 2008). The combination techniques can be generally categorized in two main categories, competitive and cooperative architectures. In a competitive architecture, the aim is to build appropriate modules to represent different parts, and to be able to switch control to the most appropriate model. In a cooperative modular, the aim is to combine models to build a complete picture from a number of partial solutions. The assumption is that a model may not be sufficient to represent the complete behavior of a under study system.

In recent years, several hybrid classification models have been proposed, using artificial neural networks and applied to the classification problems with good performance. Chakraborty (2009) proposes an integrated approach for classification and variable selection using the Bayesian *K*-nearest neighbor and stochastic search variable selection technique for simultaneous cancer classification and gene selection. This model provides a full probabilistic treatment for *K*-nearest neighbor along with adaptive variable selection. Connolly, Granger, and Sabourin (2010) propose an adaptive classification system (ACS) that combines a fuzzy ART-MAP neural network classifier suitable for incremental learning, and a dynamic particle swarm optimization (DPSO) algorithm capable of finding and tracking several local optima in the optimization space, for video-based face recognition. Ostermark, 2000 proposes a flexible hybrid genetic fuzzy neural network (GFNN) algorithm for multigroup classification problems that combines genetic computation with those on fuzzy neural networks. Aci, Inan, Avci, and neighbor (2010) propose a hybrid method by using *K*-nearest neighbor, Bayesian models and genetic algorithms in order to achieve successful results on classifying by eliminating data that make difficult to learn.

Polat and Gunes (2009) propose a novel hybrid classification system based on C4.5 decision tree classifier and one-against-all approach to classify the multi-class problems in order to improve the classification accuracy in the case of multi-class classification problems. Tagluk, Akin, and Sezgin (2010) describe a new hybrid model by combining wavelet transforms and artificial neural networks to classify sleep apnea syndrome (SAS). Wang, Li, Zhang, Gui, and Huang (2010) propose a novel ensemble method which combines base probabilistic neural networks (PNNs) classifiers with neighborhood rough set model based gene reduction. Pendharkar (2001) propose a hybrid evolutionary-neural approach for binary classification that incorporates a special training data over-fitting minimizing selection procedure for improving the prediction accuracy on holdout sample. This approach integrates parallel global search capability of genetic algorithms (GAs) and local gradient-descent search of the back-propagation algorithm. Sinha and Fieguth (2006) propose a new neuro-fuzzy classifier that

combines neural networks and concepts of fuzzy logic for the classification of defects by extracting features in segmented buried pipe images.

In this paper, the multiple linear regression models and artificial neural networks, which are one of the most accurate and widely used linear and nonlinear classification techniques; respectively, are combined together in order to construct a new hybrid model of neural networks to overcome the linear deficiency of these models and yield a more accurate classification model than traditional artificial neural networks. In our proposed model, the multiple linear regression (MLR) models are applied in order to magnify the linear components of the attributes that may not be completely modeled by neural network and generate the necessary data from the attributes for using in artificial neural networks. Therefore, in the first phase of the proposed model, the linear components of attributes are summarized in the new attribute using a multiple linear regression model for better modeling by neural network. Then, in the second phase, a neural network is used in order to model and classify data using original attributes and a generated linear attribute by multiple linear regression. Six well-known benchmark and real-world data sets—the Ripley synthetic data set, the Pima Indian Diabetes data set, the Fisher iris data set, the Forensic glass data set, the Japanese credit data set, and the Gene expression data set—are used in this paper in order to show the appropriateness and effectiveness of the proposed model for classification tasks. The rest of the paper is organized as follows. In the next section, the basic concepts and modeling approaches of the artificial neural networks (ANNs) and some other used classification models in this paper are briefly reviewed. In Section 3, the formulation of the proposed model is introduced. In Section 4, a comparative assessment of all approaches using benchmark data sets is presented. The performance results of proposed model for two sets of real-world applications are discussed in Section 5. Our concluding remarks are presented in Section 6.

2. Classification approaches

In this section, the basic concepts and modeling approaches of the artificial neural networks (ANNs), support vector machines (SVMs), *K*-nearest neighbor (KNN), quadratic discriminant analysis (QDA), and linear discriminant analysis (LDA) models for classification are briefly reviewed.

2.1. Linear discriminant analysis (LDA)

Linear discriminant analysis (LDA) is a very simple and effective supervised classification method with wide applications. The basic theory of linear discriminant analysis is to classify compounds by dividing an *n*-dimensional descriptor space into two regions that are separated by a hyperplane that is defined by a linear discriminant function. Discriminant analysis generally transforms classification problems into functions that partition data into classes, thus reducing the problem to the identification of a function. The focus of discriminant analysis is on determining this functional form and estimating its coefficients. In linear discriminant analysis, this function is assumed to be linear. The linear discriminant function was introduced by Fisher (1936). Fisher's linear discriminant function (Fisher, 1936) works by finding the mean of the set of attributes for each class and using the mean of these means as the boundary. The function achieves this by projecting the attribute points onto the vector that maximally separates their class means and minimizes their within-class variance. Fisher's linear discriminant function can be written:

$$X'S^{-1}(\overline{X}_2-\overline{X}_1)-1/2(\overline{X}_2+\overline{X}_1)'S^{-1}(\overline{X}_2-\overline{X}_1)>c, \tag{1}$$

where X is the vector of the observed values, $\overline{X}_i(i=1,2)$, is the mean of values for each group, S is the sample covariance matrix of all variables, and c is the cost function. If the misclassification cost of each group is considered equal, c is set to zero. A member is classified into one group if the result of the equation is greater than c (or zero) and into the other if less than c (or zero). A result equal to c indicates that a sample cannot be classified into either class based on the features used in the analysis.

The linear discriminant function distinguishes between two classes. If a data set has more than two classes, the process must be broken down into multiple two-class problems. The linear discriminant function was found for each class versus all samples that were not of that class (one-versus-all). Final class membership for each sample was determined by the linear discriminant function that produced the highest value. Linear discriminant analysis is optimal when the variables are normally distributed with equal covariance matrices. In this case, the linear discriminant function is in the same direction as the Bayes optimal classifier (Billings & Lee, 2002). The linear discriminant is known to perform well on moderate sample sizes when compared to more complex methods (Ghiassi & Burnley, 2010). As a straightforward mathematical function, requiring nothing more complicated than matrix arithmetic, the linear discriminant is relatively simple to perform. The assumption of linearity in the class boundary, however, limits the scope of application for linear discriminant analysis. Real-world data frequently cannot be separated by a linear boundary. When boundaries are nonlinear, the performance of the linear discriminant may be inferior to other classification methods.

2.2. Quadratic discriminant analysis (QDA)

Quadratic discriminant analysis (QDA), first introduced by Smith (1947), is another distance based classifier which is very similar to the linear discriminant function classifier. In fact, quadratic discriminant analysis is an extended of the linear discriminant function. Both discriminant functions assume that the values of each attribute in each class are normally distributed, however, the discriminant score between each sample and each class is calculated using the sample variance–covariance matrix of each class separately rather than the overall pooled matrix and so is a method that takes into account the different variance of each class. On the other hand, in linear discriminant analysis it is assumed that the covariance matrices of the groups are equal, whereas quadratic discriminant analysis makes no such assumption. When the covariance matrices are not equal, the boundary between the classes will be a hyper-conic and in theory the use of quadratic discriminant analysis will result in better discrimination and classification rates. However, due to the increased number of additional parameters that need to be estimated, it is quite possible that the classification by quadratic discriminant analysis is worse than that of linear discriminant analysis (Malhotra, Sharma, & Nair, 1999). The quadratic discriminant is found by evaluating the equation:

$$X'\left(S_1^{-1}-S_2^{-1}\right)X+2\left(\overline{X}_2'S_2^{-1}-\overline{X}_1'S_1^{-1}\right)X - \left[\overline{X}_2'S_2^{-1}\overline{X}_2-\overline{X}_1'S_1^{-1}\overline{X}_1+Ln(|S_2|/S_1|)\right]>c. \tag{2}$$

The same conditions apply to the nature of c and classification in the case that the result is equal to c or zero. As with the linear discriminant, the quadratic discriminant function distinguishes between two classes. For multiple class data sets, this was handled the same as for linear discriminant analysis. The size of the differences in variances determines how much better the quadratic discriminant function will perform than the linear discriminant. For large variance differences, the quadratic discriminant excels when

compared to the linear discriminant. Additionally, of the two, only the quadratic discriminant can be used when population means are equal (Marks & Dunn, 1974). Although more broadly applicable than the linear discriminant, the quadratic discriminant is less resilient under non-optimal conditions. The quadratic discriminant can behave worse than the linear discriminant for small sample sizes. Additionally, data that is not normally distributed results in a poorer performance by the quadratic discriminant, when compared to the linear discriminant.

Marks and Dunn (1974) found the performance of the quadratic discriminant function to be more sensitive to the dimensions of the data than the linear discriminant, improving as the number of attributes increases to a certain optimal number, then rapidly declining. Linear and nonlinear discriminant functions are the most widely used classification methods. This broad acceptance is due to their ease of use and the wide availability of tools. Both, however, assume the form of the class boundary is known and fits a specific shape. This shape is assumed to be smooth and described by a known function. These assumptions may fail in many cases. In order to perform classification for a wider range of real-world data, a method must be able to describe boundaries of unknown, and possibly discontinuous, shapes.

2.3. K-nearest neighbor (KNN)

The *K*-nearest neighbor (KNN) model is a well-known supervised learning algorithm for pattern recognition that first introduced by Fix and Hodges (1951), and is still one of the most popular nonparametric models for classification problems (Fix & Hodges, 1951, 1952). *K*-nearest neighbor assumes that observations which are close together are likely to have the same classification. The probability that a point x belongs to a class can be estimated by the proportion of training points in a specified neighborhood of x that belong to that class (Fix & Hodges, 1951). The point may either be classified by majority vote or by a similarity degree sum of the specified number (k) of nearest points. In majority voting, the number of points in the neighborhood belonging to each class is counted, and the class to which the highest proportion of points belongs is the most likely classification of x. The similarity degree sum calculates a similarity score for each class based on the *K*-nearest points and classifies x into the class with the highest similarity score. Due to its lower sensitivity to outliers, majority voting is more commonly used than the similarity degree sum (Chaovalitwongse, 2007). In this paper, majority voting is used for the data sets.

In order to determine which points belong in the neighborhood, the distances from x to all points in the training set must be calculated. Any distance function that specifies which of two points is closer to the sample point could be employed (Fix & Hodges, 1951). The most common distance metric used in *K*-nearest neighbor is the Euclidean distance (Viaene, Derrig, Baesens, & Dadene, 2002). The Euclidean distance between each test point f_t and training set point f_s, each with n attributes, is calculated using the equation:

$$d = \left[(f_{t1} - f_{s1})^2 + (f_{t2} - f_{s2})^2 + \cdots + (f_{tn} - f_{sn})^2\right]^{1/2}. \quad (3)$$

In general the following steps are performed for the *K*-nearest neighbor model (Yildiz, Altilar, & Akademik Bilisim, 2008):

(1) Chosen of k value.
(2) Distance calculation.
(3) Distance sort in ascending order.
(4) Finding k class values.
(5) Finding dominant class.

One challenge to use the *K*-nearest neighbor is to determine the optimal size of k, which acts as a smoothing parameter. A small k will not be sufficient to accurately estimate the population proportions around the test point (Enas & Choi, 1986). A larger k will result in less variance in probability estimates but the risk of introducing more bias (Viaene et al., 2002). K should be large enough to minimize the probability of a non-Bayes decision, but small enough that the points included give an accurate estimate of the true class. Enas and Choi (1986) found that the optimal value of k depends upon the sample size and covariance structures in each population, as well as the proportions for each population in the total sample. For cases in which the differences in the covariance matrices and the difference between sample proportions were either both small or both large, Enas and Choi (1986) found that the optimal k to be $N^{3/8}$, where N is the number of samples in the training set. When there was a large difference between covariance matrices and a small difference between sample proportions, or vice versa, Enas and Choi (1986) determined $N^{2/8}$ to be the optimal value of k.

This model presents several advantages (Berrueta, Alonso-Salces, & Heberger, 2007):

(i) Its mathematical simplicity, which does not prevent it from achieving classification results as good as (or even better than) other more complex pattern recognition techniques.
(ii) It is free from statistical assumptions, such as the normal distribution of the variables.
(iii) Its effectiveness does not depend on the space distribution of the classes.

In additional, when the boundaries between classes cannot be described as hyper-linear or hyper-conic, *K*-nearest neighbor performs better than the linear and quadratic discriminant functions. Enas and Choi (1986) found that the linear discriminant performs slightly better than *K*-nearest neighbor when population covariance matrices are equal, a condition that suggests a linear boundary. As the differences in the covariance matrices increases, *K*-nearest neighbor performs increasingly better than the linear discriminant function (Enas & Choi, 1986).

However, despite of the all advantages cited for the *K*-nearest neighbor models, they also have some disadvantages. *K*-nearest neighbor model cannot work well if large differences are present in the number of samples in each class. *K*-nearest neighbor provides poor information about the structure of the classes and of the relative importance of each variable in the classification. Furthermore, it does not allow a graphical representation of the results, and in the case of large number of samples, the computation can become excessively slow. In addition, *K*-nearest neighbor model much higher memory and processing requirements than other methods. All prototypes in the training set must be stored in memory and used to calculate the Euclidean distance from every test sample. The computational complexity grows exponentially as the number of prototypes increases (Muezzinoglu & Zurada, 2006).

2.4. Support vector machines (SVMs)

Support vector machines (SVM) is a new pattern recognition tool theoretically founded on Vapnik's statistical learning theory (Vapnik, 1998). Support vector machines, originally designed for binary classification, employs supervised learning to find the optimal separating hyperplane between the two groups of data. Having found such a plane, support vector machines can then predict the classification of an unlabeled example by asking on which side of the separating plane the example lies. Support vector machine acts as a linear classifier in a high dimensional feature space originated

by a projection of the original input space, the resulting classifier is in general non-linear in the input space and it achieves good generalization performances by maximizing the margin between the two classes. In the following we give a short outline of construction of support vector machine.

Consider a set of training examples as follows:

$$\{(x_i, y_i)\} \quad x_i \in R^n,\ y_i \in \{+1, -1\}; \quad i = 1, 2, \ldots, m, \tag{4}$$

where the x_i are real n-dimensional pattern vectors and the y_i are dichotomous labels. Support vector machine maps the pattern vectors $x \in R^n$ into a possibly higher dimensional feature space $(z = \phi(x))$ and construct an optimal hyperplane $w \cdot z + b = 0$ in feature space to separate examples from the two classes. For support vector machine with L1 soft-margin formulation, this is done by solving the primal optimization problem as follows:

$$\begin{aligned} &\text{Min } \frac{1}{2}||w|| + C\sum_{i=1}^{m}\xi_i \\ &\text{s.t. } y_i(w \cdot z_i + b) \geqslant 1 - \xi_i x \quad \xi_i \geqslant 0, i = 1, 2, \ldots m, \end{aligned} \tag{5}$$

where C is a regularization parameter used to decide a tradeoff between the training error and the margin, and $\xi_i (i = 1, 2, \ldots, m)$ are slack variables. The above problem is computationally solved using the solution of its dual form:

$$\begin{aligned} &\text{Max}_\alpha \quad \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j k(x_i, x_j) \\ &\text{s.t} \qquad \sum_{i=1}^{m}\alpha_i y_i = 0; \quad 0 \leqslant \alpha_i \leqslant C,\ i = 1, 2, \ldots m, \end{aligned} \tag{6}$$

where $k(x_i, x_i) = \phi(x_i) \cdot \phi(x_j)$ is the kernel function that implicitly define a mapping ϕ. The resulting decision function is:

$$f(x) = sgn\left\{\sum_{i=1}^{m}\alpha_i y_i k(x_i, x) + b\right\}. \tag{7}$$

All kernel functions have to fulfill Mercer theorem (Song & Tang, 2005); however, the most commonly used kernel functions are polynomial kernel and radial basis function kernel, respectively.

$$k(x_i, x_j) = \left(a(x_i, x_j) + b\right)^d, \tag{8}$$

$$k(x_i, x_j) = \exp\left(-g\|x_i, x_j\|^2\right). \tag{9}$$

Support vector machines differ from discriminant analysis in two significant ways. First, the feature space of a classification problem is not assumed to be linearly separable. Rather, a nonlinear mapping function (also called a kernel function) is used to represent the data in higher dimensions where the boundary between classes is assumed to be linear (Duda, Hart, & Stork, 2001). Second, the boundary is represented by support vector machines instead of a single boundary. Support vectors run through the sample patterns which are the most difficult to classify, thus the sample patterns that are closest to the actual boundary (Duda et al., 2001). Overfitting is prevented by specifying a maximum margin that separates the hyper plane from the classes. Samples which violate this margin are penalized. The size of the penalty is a parameter often referred to as C (Brown et al., 2000; Christianini & Shawe-Taylor, 2000).

2.5. Artificial neural networks (ANNs)

Artificial neural networks (ANNs) are computer systems developed to mimic the operations of the human brain by mathematically modeling its neuro-physiological structure. Artificial neural networks have been shown to be effective at approximating complex nonlinear functions (Zhang, 2001). For classification tasks, these functions represent the shape of the partition between classes. In artificial neural networks, computational units called neurons replace the nerve cells and the strengths of the interconnections are represented by weights, in which the learned information is stored. This unique arrangement can acquire some of the neurological processing ability of the biological brain such as learning and drawing conclusions from experience. Artificial neural networks combine the flexibility of the boundary shape found in K-nearest neighbor with the efficiency and low storage requirements of discriminant functions. Like the K-nearest neighbor, artificial neural networks are data driven; there are no assumed model characteristics or distributions, as is the case with discriminant analysis (Berardi & Zhang, 1999).

Single hidden layer feed forward network is the most widely used model form for modeling, forecasting, and classification (Silva, Marques, & Alexandre, 2008). The model is characterized by a network of three layers of simple processing units connected by acyclic links (Fig. 1). The relationship between the output (y) and the inputs $(x_1, x_2, \ldots, x_p)$ has the following mathematical representation:

$$y_t = w_0 + \sum_{j=1}^{q} w_j \cdot g\left(w_{0,j} + \sum_{i=1}^{p} w_{i,j} \cdot x_{t,i}\right) + \varepsilon_t, \tag{10}$$

where $w_{i,j} (i = 0, 1, 2, \ldots, p, j = 1, 2, \ldots, q)$ and $w_j (j = 0, 1, 2, \ldots, q)$ are model parameters often called connection weights; p is the number of input nodes; and q is the number of hidden nodes. Data enters the network through the input layer, moves through hidden layer, and exits through the output layer. Each hidden layer and output layer node collects data from the nodes above it (either the input layer or hidden layer) and applies an activation function. Activation functions can take several forms. The type of activation function is indicated by the situation of the neuron within the network. In the majority of cases input layer neurons do not have an activation function, as their role is to transfer the inputs to the hidden layer. The logistic and hyperbolic functions are often used as hidden layer and output transfer function for classification problems that are shown in Eqs. (11) and (12), respectively. Other transfer functions can also be used such as linear and quadratic, each with a variety of modeling applications.

$$Sig(x) = \frac{1}{1 + \exp(-x)}. \tag{11}$$

$$Tanh(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)}. \tag{12}$$

The simple network given by (10) is surprisingly powerful in that it is able to approximate the arbitrary function as the number of hidden nodes when q is sufficiently large. In practice, simple network structure that has a small number of hidden nodes often

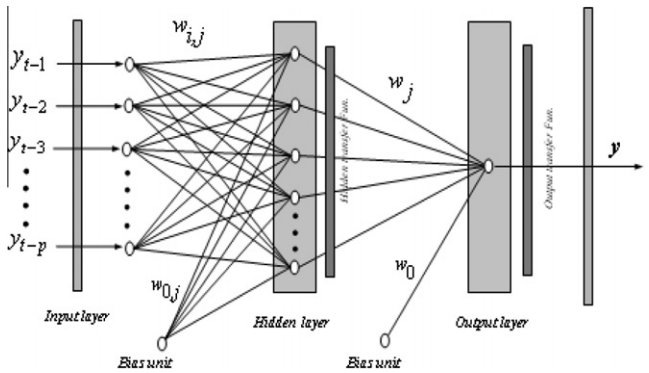

Fig. 1. Neural network structure ($N^{(p\text{-}q\text{-}1)}$.).

works well in out-of-sample forecasting. This may be due to the overfitting effect typically found in the neural network modeling process. An overfitted model has a good fit to the sample used for model building but has poor generalizability to data out of the sample.

There exist many different approaches such as the pruning algorithm, the polynomial time algorithm, the canonical decomposition technique, and the network information criterion for finding the optimal architecture of an artificial neural network. These approaches can be generally categorized as follows (Khashei & Bijari, 2010): (i) Empirical or statistical methods that are used to study the effect of internal parameters and choose appropriate values for them based on the performance of model. The most systematic and general of these methods utilizes the principles from Taguchi's design of experiments. (ii) Hybrid methods such as fuzzy inference where the artificial neural network can be interpreted as an adaptive fuzzy system or it can operate on fuzzy instead of real numbers. (iii) Constructive and/or pruning algorithms that, respectively, add and/or remove neurons from an initial architecture using a previously specified criterion to indicate how artificial neural network performance is affected by the changes. The basic rules are that neurons are added when training is slow or when the mean squared error is larger than a specified value. In opposite, neurons are removed when a change in a neuron's value does not correspond to a change in the network's response or when the weight values that are associated with this neuron remain constant for a large number of training epochs. (iv). Evolutionary strategies that search over topology space by varying the number of hidden layers and hidden neurons through application of genetic operators and evaluation of the different architectures according to an objective function (Benardos & Vosniakos, 2007).

Although many different approaches exist in order to find the optimal architecture of an artificial neural network, these methods are usually quite complex in nature and are difficult to implement (Zhang et al., 1998). Furthermore, none of these methods can guarantee the optimal solution for all real forecasting problems. To date, there is no simple clear-cut method for determination of these parameters and the usual procedure is to test numerous networks with varying numbers of hidden units, estimate generalization error for each and select the network with the lowest generalization error (Hosseini, Luo, & Reynolds, 2006).

Once a network structure is specified, the network is ready for training a process of parameter estimation. The parameters are estimated such that the cost function of neural network is minimized. Cost function is an overall accuracy criterion such as the following mean squared error:

$$E = \frac{1}{N}\sum_{n=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{n=1}^{N}\left(y_t - \left(w_0 + \sum_{j=1}^{Q} w_j g\left(w_{0j} + \sum_{i=1}^{P} w_{i,j} y_{t-i}\right)\right)\right)^2, \quad (13)$$

where, N is the number of error terms. This minimization is done with some efficient nonlinear optimization algorithms other than the basic backpropagation training algorithm (Rumelhart & McClelland, 1986), in which the parameters of the neural network, $w_{i,j}$, are changed by an amount $\Delta w_{i,j}$, according to the following formula:

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}}, \quad (14)$$

where, the parameter η is the learning rate and $\partial E/\partial w_{i,j}$ is the partial derivative of the function E with respect to the weight $w_{i,j}$. This derivative is commonly computed in two passes. In the forward pass, an input vector from the training set is applied to the input units of the network and is propagated through the network, layer by layer, producing the final output. During the backward pass, the output of the network is compared with the desired output and the resulting error is then propagated backward through the network, adjusting the weights accordingly. To speed up the learning process, while avoiding the instability of the algorithm, Rumelhart and McClelland (1986) introduced a momentum term δ in Eq. (6), thus obtaining the following learning rule:

$$\Delta w_{i,j}(t+1) = -\eta \frac{\partial E}{\partial w_{i,j}} + \delta \Delta w_{i,j}(t), \quad (15)$$

The momentum term may also be helpful to prevent the learning process from being trapped into poor local minima, and is usually chosen in the interval [0; 1]. Finally, the estimated model is evaluated using a separate hold-out sample that is not exposed to the training process.

3. Formulation of the hybrid proposed model

Despite the numerous classification models available, the accuracy is fundamental to many decision processes, and hence, never research into ways of improving the effectiveness of the classification models been given up. Many researchers have combined the predictions of multiple classifiers to produce a better classifier, which has been reported to improve performance (Chen, Lin, & Chou, 2010). The effectiveness of a hybrid relies on the extent to which its classifiers make different errors, or are error independent. Errors come from four aspects, that is, different data sampling methods, different parameter settings, different classifiers, and different combination strategies (Amanda, 1999). By means of the combined predictions of several classifiers, a better performance than that of any of the individual classifiers is sought. Breiman refers to multiple experts of classifiers that have demonstrated the potential to reduce the generalization error of a classifier model from 5% to 70%. In order words, multiple classifiers may provide more accurate classification results than single classifier (Breiman, 1999).

In this paper, a novel hybrid classification model of artificial neural networks is proposed in order to yield more accurate results using the multiple linear regression models. The main aim of the proposed model is to use the unique advantages of the multiple linear regression models in linear modeling in order to overcome the linear modeling limitation of the traditional artificial neural networks. Therefore, in the first phase of the proposed model a multiple linear regression model is used in order to magnify the linear components in the attributes for better using by neural network in the second phase. Then the magnified linear components are summarized in a new attribute as L ($n+1$th attribute). The main goal of using the multiple linear regression models is to evaluate the relationship between attributes as independent or predictor variables and class value as dependent variable. This is done by fitting a straight line to a number of observations. Specifically, a line is produced so that the squared deviations of the observed points from that line are minimized. Thus this procedure is generally referred to as least squares estimation (Sahoo, Schladow, & Reuter, 2009). Mathematically, if the class value is linearity dependent on the values of their attributes, then a multiple regression model is as follows:

$$L = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n = \sum_{i=0}^{n} \alpha_i x_i, \quad (16)$$

where $x_i (i = 0, 1, 2, \ldots, n)$ are attributes and $\alpha_i (i = 0, 1, 2, \ldots, n)$ are unknown coefficients that are estimated by the least squares method. Then, in the second phase of the proposed model.

Then, in the second phase, a neural network is used in order to jointly model both linear and nonlinear structures and classify

using original attributes and a generated linear attribute by multiple linear regression as follows:

$$y_t = w_0 + \sum_{j=1}^{q} w_j \cdot g\left(w_{0,j} + \sum_{i=1}^{n} w_{i,j} \cdot x_{t,i} + w_{n+1,j} \cdot x_{t,n+1}\right) + \varepsilon_t = \sum_{j=0}^{q} w_j \cdot g\left(w_{0,j} + \sum_{i=1}^{n+1} w_{i,j} \cdot x_{t,i}\right) + \varepsilon_t, \quad (17)$$

where, $g(w_{0,0} + \sum_{i=1}^{n+1} w_{i,0} \cdot x_{t,i}) = 1$, $w_{i,j}(i = 0, 1, 2, \ldots, n+1, j = 0, 1, 2, \ldots, q)$ and $w_j(j = 0, 1, 2, \ldots, q)$ are connection weights, $n+1$ is the number of the all attributes (input nodes), and q is the number of hidden nodes. Although, in this paper, the proposed model is only used for constructing hybrid model by multilayer perceptrons (MLP) for classification purpose, this methodology can be generally applied to a wide range of artificial neural networks such as support vector machines (SVMs), general regression neural networks (GRNNs), probabilistic neural networks (PNNs), etc. for modeling, casual forecasting, and classification purposes.

3.1. Proposed model for two-class classification

The output of the proposed model is continuous. Classification problems differ in that its output is discrete. However, classification can also be viewed as the process of drawing a partition between classes. The proposed model can be used to approximate a function that identifies this partition. Our proposed model does not assume the shape of the partition, unlike the linear and quadratic discriminant analysis. In contrast to the *K*-nearest neighbor method, the proposed model does not require storage of training data. Once the model has been trained, it performs much faster than *K*-nearest neighbor because it does not need to iterate through individual training samples. The proposed model does not require experimentation and final selection of a kernel function and a penalty parameter as is required by the support vector machines. Our proposed model solely relies on a training process in order to identify the final classifier model. Finally, the proposed model dose not has the mixed results in linear problems and also, in problems that consist both linear and nonlinear correlation structures as traditional artificial neural networks.

In order to apply the proposed model to classification, certain modifications to the model needed to be made. As with other classification models (with the exception of *K*-nearest neighbor), the output of the proposed model is continuous, while classification requires discrete results. Similar to other models, continuous output of the proposed model is converted to a discrete class by assigning a sample to the class to which the output was closest. Each class is assigned a numeric value. The difference between the output and each numeric value is then calculated, and the sample is put in the class with which its output has the smallest difference. In proposed model for two-class classification problems, the values of $\{-1, +1\}$ and $\{0, +1\}$ are respectively considered as class values, when the hyperbolic and logistic functions are used as output transfer functions of the proposed model. However, in case of using the linear transfer function for output layer of the proposed model may be better to apply the values of $\{-10, +10\}$ or $\{-100, +100\}$ as class values. The larger class values expand small differences in the output, helping the model to become more sensitive to variations in the input.

3.2. Hierarchical proposed model for multiple class classification

In this section, the hierarchical proposed model for multiple class classification is introduced. The additional complexity inherent in multiple class classification problems presents a challenge to many classification models. An approach that has been commonly used in order to improve multiple class performance of classifiers is hierarchical models. Common hierarchical classifiers include hierarchical neural networks and decision trees. Decision trees often identify one or more attributes used to split data at each node, until each sample reaches a leaf node at which it is assigned to a class (Amasyali & Ersoy, 2008). The hierarchical neural networks may come in a variety of architectures. Lee and Ersoy (2007) describe a hierarchical neural network in which difficult training samples are rejected from the first layer and used for training in subsequent layers. Kim, Kehtarnavaz, Yeary, and Thornton (2003) first branch classes that can be easily separated, then use feature selection to build additional branches for the remaining classes, determining which features to use at each node of the tree. Finally, neural networks are used at each node to improve the performance of the tree.

Reasons for using hierarchical classifiers focus on reducing complexity. Porter and Liu (1996) describe hierarchical classifiers as a subset of modular classifiers. They suggest that modular classifiers often arise when a combination of factors include a large number of classes, classes have difficult shapes (are not compact, convex, or connected), classes do not have distinct boundaries, boundaries are highly nonlinear, and misclassification of some points carries a high penalty. Lee and Ersoy (2007) describe hierarchical classification as a way in which to detect data which are more difficult to classify in order to classify these data differently.

In this paper, three different approaches, namely "one versus one", "one versus rest", and "one versus all" are examined in order to develop a hierarchical version of the proposed model following the aforementioned reasoning. In all of these approaches are postulated that if a class is removed from the data set, the remaining classes may become easier to classify without the influence of the removed class. On the other hand, in order to simplify the training and improve the performance, multiple class classification problems can be broken down into several two-class data sets. The modeling cost of the "one versus one" approach in order to develop of the hierarchical proposed model is too high. For a case of k classes, the "one versus one" approach needs $\binom{k}{2} = \frac{k!}{(k-2)!2!} = \frac{k(k-1)}{2}$ two-class classifiers, while the "one versus rest", and the "one versus all"; approaches only need $k-1$ and k two-class classifiers. Therefore using the "one versus one" approach for developing of the hierarchical proposed model is not reasonable.

In the "one versus rest" approach, for a case of k classes, a class from these k classes is first considered as a category, and the rest $k-1$ classes as another category, and a two-class classifier is constructed. Next, this class is excluded, and then the described process is repeated for a case of $k-1$ classes. On the other hand, a class from remaining $k-1$ classes is considered as a category, and the rest $k-2 = (k-1)-1$ classes as another category, and a second two-class classifier is constructed, and so on and so forth till the last two-class classifier is constructed. In this way, $k-1$ two-class classifier must be constructed in all for a case of k classes. The "one versus all" approach is similar to the "one versus rest" approach with a bit difference. In the "one versus all" approach, for a case of k classes, a class from these k classes is also considered as a category, and the rest $k-1$ classes as another category, and a two-class classifier is constructed; however, this class is not excluded. In this way, k two-class classifier must be constructed in all for a case of k classes.

4. Comparative assessment of benchmark data sets

The classification performance of the proposed model is compared with the linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), *K*-nearest neighbor (KNNN), support vector machine (SVM), and traditional artificial neural network

(ANN) models using four well-known benchmark data sets. Each data set chosen for use in the benchmark comparisons has been used in multiple published studies to assess classification performance of various classification models. Two of these have two classes, and two contain multiple (three and four) classes. Each data set is divided into a training set and a test set, and each model is applied accordingly. The classification error rate for each class is calculated and is presented, as well as the percent improvement in error rate for the proposed model.

4.1. The benchmark two-class data sets

As classification involves the partitioning of space between two or more sets of data, many classification models are easiest to use when distinguishing between two classes. Multiple classes add complexity by requiring the creation of multiple partitions. Two class data sets directly demonstrate the effectiveness of the partition created by each classification model. The data sets used for benchmarking include a synthetic data set by Ripley (1994) and real-world data set of diabetes diagnosis among Pima Indians (Asuncion & Newman, 2007). Both of these data sets have been widely used in the published literature to assess performance of various classification models.

4.1.1. Ripley synthetic data set

The Ripley synthetic data set is created by Ripley for his evaluation of the use of feed-forward, back-propagation neural networks for classification (Ripley, 1994). The data set consists of 1250 samples with two attributes. Two hundred and fifty samples are used for training, and one thousand samples are used for testing. The two classes are equally represented in the data set. Each class consists of two normally distributed populations, so the full set of each attribute for each class is not normally distributed. According to the previous works, each classifier is trained on the first 250 samples, and then tested it on the remaining 1000 samples.

In order to obtain the optimum network architecture of the proposed model, based on the concepts of artificial neural networks design (Khashei, 2005) and using pruning algorithms in *MATLAB 7* package software, different network architectures are evaluated to compare the ANNs performance. The best fitted network which is selected, and therefore, the architecture which presented the best accuracy with the test data, is composed of three inputs, four hidden and one output neurons (in abbreviated form, $\boldsymbol{N}^{(3\text{-}4\text{-}1)}$). The structure of the best-fitted network is shown in Fig. 2. The weights and biases of the best-fitted network for the Ripley synthetic data set are given in Table 1. The misclassification percentages of the each model and improvement percentages of the proposed model in comparison with those of other classification models for the Ripley synthetic data set in both training and test data sets are summarized in Tables 2 and 3, respectively.

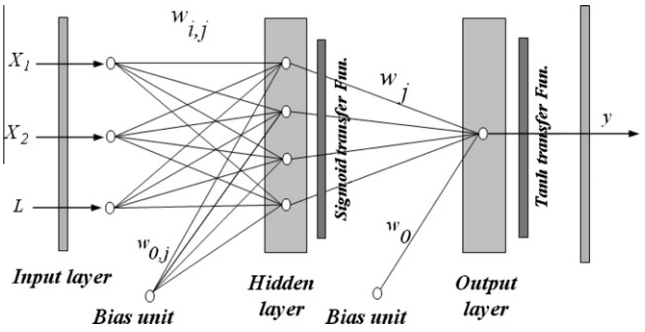

Fig. 2. Structure of the best fitted network (Ripley synthetic data set), $\boldsymbol{N}^{(3\text{-}4\text{-}1)}$.

Table 1
Weights and biases of the final network for Ripley synthetic data set $\boldsymbol{N}^{(3\text{-}4\text{-}1)}$.

Input weights			Hidden weights	Biases	
$w_{i,1}$	$w_{i,2}$	$w_{i,3}$	w_j	$w_{0,j}$	w_0
−24.2025	−46.6937	105.5367	−3.3376	−18.0292	−24.2988
88.6777	115.3515	−136.229	0.96526	1.0934	
0.22549	−0.16326	0.7589	26.995	0.005918	
45.6907	43.1422	18.2927	12.927	−44.7388	

4.1.2. Pima Indian Diabetes data set

The Pima Indian Diabetes data set is collected by the National Institute of Diabetes and Digestive and Kidney Diseases and consists of diabetes diagnoses (positive or negative) and attributes of female patients who are at least 21 years old and of Pima Indian heritage (Asuncion & Newman, 2007). The eight attributes represent (1) the number of times pregnant, (2) the results of an oral glucose tolerance test, (3) diastolic blood pressure (mm Hg), (4) triceps skin fold thickness (mm), (5) 2-h serum insulin (micro U/ml), (6) body mass index (weight in kg/(height in m)^2), (7) diabetes pedigree function, and (8) age (year). The data set consists of 768 samples, about two third of which have a negative diabetes diagnosis and one third with a positive diagnosis. The data set is randomly split into equal size training and test sets of 384 samples each.

Based on the previous works, each classifier is trained on the half of the data set, and then tested it on the remaining half of the data set. In a similar fashion, by using pruning algorithms in *MATLAB7* package software, the best fitted network which is selected, is composed of nine inputs, three hidden and one output neurons ($\boldsymbol{N}^{(9\text{-}3\text{-}1)}$). The structure of the best-fitted network is shown in Fig. 3. The weights and biases of the best-fitted network for the Pima Indian Diabetes data set are given in Table 4. The misclassification percentages of the each model and improvement percentages of the proposed model in comparison with those of other classification models for the Pima Indian Diabetes data set in both training and test data sets are also summarized in Tables 2 and 3, respectively.

4.1.3. Comparison with other classification models for benchmark two-class data sets

In the Ripley synthetic data classification case, several different architectures of artificial neural network are designed and examined. An architecture composed of two inputs, three hidden and one output neurons ($\boldsymbol{N}^{(2\text{-}3\text{-}1)}$) as employed by Ripley (1994) has been found to be the most accurate among all other ANN architectures, with a 9.4% error rate. However, our proposed model outperforms this model on the test portion of the data set, with an error rate of 8.9%, an improvement of 5.32% compared to the best traditional neural network results of 9.4%. In addition, based on the nature of the data set, it is expected that linear and quadratic discriminant analysis will not be optimal classifiers as these models are the best when all their attributes are normally distributed. This expectation is confirmed by the performance of each. The linear discriminant analysis and quadratic discriminant analysis models have the error rate of 10.8% and 11.7% on the test portion of the data set that show the classification rate 17.59% and 23.93% worse than our proposed model, respectively. The support vector machine improves slightly on linear discriminant analysis and quadratic discriminant analysis, with a 10.1% error rate. However, this error rate is an 11.88% higher than the proposed model on the test portion of the data set. In contrast of other classification models, the proposed model cannot outperform the *K*-nearest neighbor model on the test portion of the data set. Our proposed model misclassifies 9.2% and 8.9% of the training and test samples, an improvement of 16.36% and −11.25% compared to the best *K*-nearest neighbor results of 11.0% and 8.0%, respectively.

Table 2
Two-class benchmark data sets classification results.

Model	Classification error (%)			
	Ripley synthetic data set		Pima Indian Diabetes data set	
	Train	Test	Train	Test
Linear discriminant analysis (LDA)	14.4	10.8	26.6	21.9
Quadratic Discriminant Analysis (QDA)	14.4	11.7	23.7	28.1
K-nearest neighbor (KNN)	11.0	8.0	23.4	24.7
Support vector machines (SVM)	13.6	10.1	9.9	30.0
Artificial neural networks (ANN)	N/A	9.4	18.8	25.3
Hybrid proposed model	9.2	8.9	17.9	19.1

Table 3
Percentage improvement of the proposed model in comparison with those of other classification models (two-class cases).

Model	Improvement (%)			
	Pima Indian Diabetes data set		Ripley synthetic data set	
	Train	Test	Train	Test
Linear discriminant analysis (LDA)	36.11	17.59	32.71	12.79
Quadratic discriminant analysis (QDA)	36.11	23.93	24.47	32.03
K-nearest neighbor (KNN)	16.36	−11.25	23.50	22.67
Support vector machines (SVM)	32.35	11.88	−80.81	36.33
Artificial neural networks (ANN)	*N/A*	5.32	4.79	24.51

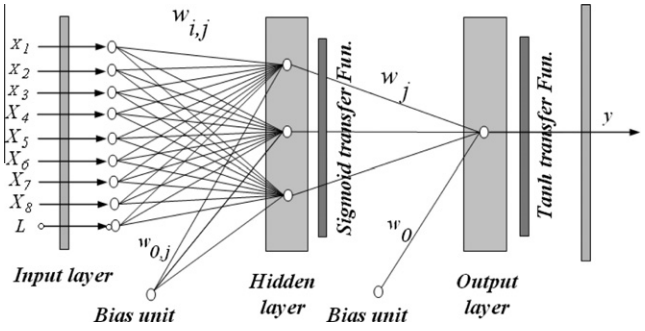

Fig. 3. Structure of the best fitted network (Pima Indian Diabetes data set), $\boldsymbol{N}^{(9\text{-}3\text{-}1)}$.

In the Pima Indian Diabetes data classification case, our proposed model has the lowest error on the test portion of the data set in comparison to other those used models for the Pima Indian Diabetes data set, with a misclassification rate of 19.1%. As previous case, several different architectures of artificial neural network are designed and examined. The best performing architecture for a traditional artificial neural network produces a 25.3% error rate, which proposed model improves by 36.33%. Linear discriminant analysis performs second best with an error rate of 21.9%, a classification rate 12.79% worse than the proposed model. Quadratic discriminant analysis misclassifies 28.1% of the test samples, which is also a 32.03% worse than the proposed model. As *K*-nearest neighbor scores can be sensitive to the relative magnitude of different attributes, all attributes are scaled by their *z*-scores before using *K*-nearest neighbor model (Ghosh, 2006). The best *K*-nearest neighbor, with a $K = 13$ has error rates of 24.7% that is a 22.67% higher than the proposed model error. The support vector machine model produces an error rate of 30.0%. The proposed model improves upon these by 36.33% for the support vector machine model.

For both binary classification benchmark data sets, proposed model performs better than the traditional artificial neural network. The improvement varies from 5.32% to 24.51% in comparison to the neural network for the Ripley synthetic data set to the Pima Indian Diabetes data set. In addition, the performance of the proposed model is overall better than support vector machine and also other traditional classification models such as linear discriminant analysis and quadratic discriminant analysis for both the Ripley synthetic and the Pima Indian Diabetes data sets. These results suggest that the proposed model can perform well on data sets with a variety of characteristics.

4.2. Multiple class data sets

Real-world applications frequently require classification into more than two classes. Classification into multiple classes is far more complex than two-class classification. Many classification methods determine the partition between two sets of data. Additional classes require adjustments to the classification methods that frequently result in higher error rates. For linear and quadratic discriminant analysis, the $n - 1$ one-versus-all models (where n is the number of classes) and classified each sample as described above.

4.2.1. Fisher iris data set

The Fisher iris data set is perhaps the oldest and most widely used classification data set. The data set is named for Fisher, who used it in his seminal 1936 paper on linear discriminant analysis

Table 4
Weights and biases of the final network for Pima Indian Diabetes data set $\boldsymbol{N}^{(9\text{-}3\text{-}1)}$.

Input weights									Hidden weights	Biases	
$w_{i,1}$	$w_{i,2}$	$w_{i,3}$	$w_{i,4}$	$w_{i,5}$	$w_{i,6}$	$w_{i,7}$	$w_{i,8}$	$w_{i,9}$	w_j	$w_{0,j}$	w_0
0.81342	−3.4378	−1.0066	−1.1853	1.341	−0.2356	1.2539	−0.50397	0.53024	1.9858	2.0411	
3.1515	3.1223	1.5813	2.9963	8.0098	−1.9961	−2.7061	−1.9061	9.054	3.9523	−11.393	2.0397
−0.43714	−1.221	−0.62315	−0.8798	0.42667	0.072862	0.41524	−0.90402	0.08204	−2.9929	1.2636	

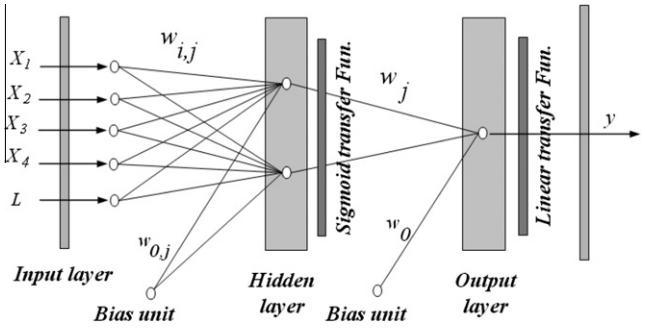

Fig. 4. Structure of the best fitted network (Fisher iris data set), $\boldsymbol{N}^{(5\text{-}2\text{-}1)}$.

(Fisher, 1936). The data set consists of 150 samples, split evenly between three classes. The classes represent three species of iris: Iris Setosa, Iris Versicolour, and Iris Virginica. Each iris is characterized by four attributes, (1) sepal length, (2) sepal width, (3) petal length, and (4) petal width. The data set is randomly divided into 75 training and 75 test samples.

Similar to the previous works, each classifier is trained on the half of the data set, and then tested it on the other half of the data set. In a similar fashion, by using pruning algorithms in *MATLAB7* package software, the best fitted network which is selected, is composed of five inputs, two hidden and one output neurons ($\boldsymbol{N}^{(5\text{-}2\text{-}1)}$). The structure of the best-fitted network is shown in Fig. 4. The weights and biases of the best-fitted network for the Fisher iris data set are given in Table 5. The misclassification percentages of the each model and improvement percentages of the proposed model in comparison with those of other classification models for the Fisher iris data set in both training and test data sets are also summarized in Tables 6 and 7, respectively. It must be noted that the "one versus all" approach is applied in order to construct the hierarchical proposed model for the Fisher iris data set due to better results in comparison with "one versus rest" approach.

4.2.2. Forensic glass data set

The Forensic glass data set is also used by Ripley in his paper on neural networks for classification (Ripley, 1994). The original data set has a size of 214 samples, but Ripley regroup and omit some sample, resulting in 185 samples in four classes: (i) window float glass, (ii) window non-float glass, (iii) vehicle window glass, and (iv) other glass. The window float glass and window non-float glass classes are largest with 70 and 76 samples, respectively. The vehicle window glass class is smallest with 17 samples, and the other glass consists of 22 samples. The attributes used to characterize each sample consist of (1) refractive index (RI), (2) weight percent of the sodium oxide (Na), (3) weight percent of the magnesium oxide (Mg), (4) weight percent of the aluminum oxide (Al), (5) weight percent of the silicon oxide (Si), (6) weight percent of the potassium oxide (K), (7) weight percent of the calcium oxide (Ca), (8) weight percent of the barium oxide (Ba), and 9) weight percent of the iron oxide (Fe).

According to Ripley's work, the Forensic glass data set is randomly divided into a training set of 89 and a test set of 96 samples. In a similar fashion, by using pruning algorithms in *MATLAB7* package software, the best fitted network which is selected, is composed of ten inputs, four hidden and one output neurons ($\boldsymbol{N}^{(10\text{-}4\text{-}1)}$). The structure of the best-fitted network is shown in Fig. 5. The weights and biases of the best-fitted network for the Forensic

Table 5
Weights and biases of the final network for Fisher iris data set $\boldsymbol{N}^{(5\text{-}2\text{-}1)}$.

Input weights					Hidden Weights	Biases	
$w_{i,1}$	$w_{i,2}$	$w_{i,3}$	$w_{i,4}$	$w_{i,5}$	w_j	$w_{0,j}$	w_0
−2.9495	−5.2286	−4.8253	2.8299	1.5681	−11.4813	9.7718	3.693
8.136	0.39887	3.59	−4.0272	8.2306	7.7432	−0.34014	

Table 6
Multiple-class benchmark data sets classification results.

Model	Classification error (%)			
	Fisher iris data set		Forensic glass data set	
	Train	Test	Train	Test
Linear discriminant analysis (LDA)	10.7	12.0	39.3	32.3
Quadratic discriminant analysis (QDA)	1.3	1.3	71.9	72.9
K-nearest neighbor (KNN)	2.7	2.7	0.0	29.2
Support vector machines (SVM)	0.0	1.3	28.1	30.2
Artificial neural networks (ANN)	N/A	4.6	N/A	33.0
Hybrid proposed model (non-hierarchical)	0.0	2.7	30.11	31.52
Hybrid proposed model (hierarchical)	0.0	1.3	24.6	26.8

Table 7
Percentage improvement of the proposed model in comparison with those of other classification models (multiple-class cases).

Model	Improvement (%)			
	Fisher iris data set		Forensic glass data set	
	Train	Test	Train	Test
Linear discriminant analysis (LDA)	100.00	89.17	37.40	17.03
Quadratic discriminant analysis (QDA)	100.00	0.00	65.79	63.24
K-nearest neighbor (KNN)	100.00	51.85	N/A	8.22
Support vector machines (SVM)	N/A	0.00	12.46	11.26
Artificial neural networks (ANN)	N/A	71.74	N/A	18.79

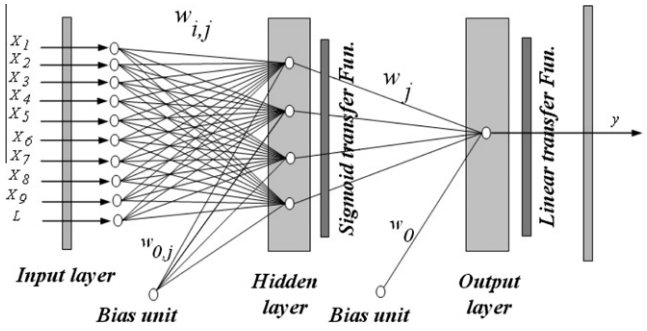

Fig. 5. Structure of the best fitted network (Forensic glass data set), $N^{(10\text{-}4\text{-}1)}$.

glass data set are given in Table 8. The misclassification percentages of the each model and improvement percentages of the proposed model in comparison with those of other classification models for the Forensic glass data set in both training and test data sets are also summarized in Tables 6 and 7, respectively. The "one versus all" approach is also applied in order to construct the hierarchical proposed model for the Forensic glass data set due to better results.

4.2.3. Comparison with other classification models for benchmark multiple class data sets

In the Fisher iris data classification case, similar to the previous section, several different architectures of artificial neural network are designed and examined. An architecture composed of four inputs, two hidden and one output neurons ($N^{(4\text{-}2\text{-}1)}$) as examined by Curram, Mingers, and networks (1994) has been found to be the most accurate among all other neural network architectures, with a 4.6% error rate on the test portion of the data set. However, the hierarchical proposed model outperforms this model on the test portion of the data set, with an error rate of 1.3%, an improvement of 71.74%. The hierarchical proposed model and support vector machine performs the best on both training and test samples with a 0.0% and 1.3% error rate each in the training and test samples, respectively. The quadratic discriminant analysis also performs best with a 1.3% error rate on the test portion of the data set (misclassifying only one sample); however, its performance on the training portion of the data set is worse than the hierarchical proposed model and support vector machine models. The non-hierarchical proposed model and *K*-nearest neighbor both have a 2.7% error rate on the test portion of the data set; however, the error rate of the non-hierarchical proposed model on the training portion of the data set is better than the *K*-nearest neighbor model. In a similar fashion, for *K*-nearest neighbor, all attributes are scaled by their *z*-scores before using *K*-nearest neighbor model. Linear discriminant analysis performs worst in training and test samples with an error rate of 10.7% and.12.0%, respectively.

In the Forensic glass data classification case, our hierarchical proposed model also has the lowest error on the test portion of the data set in comparison to other those used models for the Forensic glass data set, with a misclassification rate of 26.8%. As previous case, several different architectures of artificial neural network are designed and examined. The best performing architecture composed of nine inputs, six hidden and one output neurons ($N^{(9\text{-}6\text{-}1)}$) as designed by Ripley (1994) for a traditional artificial neural network, produces a 33.0% error rate. However, this performance is a 18.79% lower than the hierarchical proposed model. In additional, the classification error rate for linear discriminant analysis is 32.3%. The hierarchical proposed model improves upon this by 17.03%.

The Forensic glass data set presents additional challenges for quadratic discriminant analysis because the barium and iron composition attributes only have values of zero for certain classes, resulting in means and variances of zero. The inverse of these variances is therefore undefined. As the quadratic discriminant function requires the inverse covariance matrix for each class, these attributes have to be excluded for the quadratic discriminant analysis. Quadratic discriminant analysis, handicapped by the removal of two attributes, misclassified 72.9% of the samples, performing 63.24% worse than the hierarchical proposed model. *K*-nearest neighbor has the next lowest error rate with 29.2% misclassified, 8.22% higher than that of the hierarchical proposed model. The support vector machine an error rate of 30.2%, 11.26% higher than hierarchical proposed model.

For both multiple class classification benchmark data sets, the hierarchical proposed model performs better than the traditional artificial neural network. The improvement varies from 71.74% to 18.79% in comparison to the neural network for the Fisher iris data set to the Forensic glass data set. In addition, the hierarchical proposed model performs as well as or better than support vector machine and also other traditional classification models such as linear discriminant analysis, quadratic discriminant analysis, and *K*-nearest neighbor for both examined data sets. These results again show that the proposed model produces consistently good results in a variety of cases.

5. Comparative assessment of real-world application data sets

In order to examine real-world applications of the proposed model, data sets from actual real-life experiments are applied that are studied by other researchers. The results for a real-world application involving a two-class data set and another using a multiple class data set are presented. The classification performance of the proposed model is also compared to linear discriminant analysis, quadratic discriminant analysis, *K*-nearest neighbor, artificial neural network, and support vector machines for these data sets.

5.1. Japanese credit data set

In this section, in order to show the appropriateness and effectiveness of the proposed model for two-class real-life data set classification, the Japanese credit data set is also used. Tsai (2008) examined the effectiveness of neural networks and support vector machines algorithms in financial decisions. Tsai tested several data sets involving credit decisions and bankruptcy. These data sets included the Japanese credit data set. This data set includes 690

Table 8
Weights and biases of the final network for Forensic glass data set $N^{(10\text{-}4\text{-}1)}$.

Input weights										Hidden weights	Biases	
$w_{i,1}$	$w_{i,2}$	$w_{i,3}$	$w_{i,4}$	$w_{i,5}$	$w_{i,6}$	$w_{i,7}$	$w_{i,8}$	$w_{i,9}$	$w_{i,10}$	w_j	$w_{0,j}$	w_0
−2.2896	3.5977	−4.3527	8.0505	2.543	−4.3536	−10.362	−11.5388	−1.399	10.6445	−1.0635	−5.3076	3.7307
1.5935	7.9005	3.4825	−0.4458	1.8219	0.99987	−8.2148	0.094395	1.3817	1.2482	−1.945	0.71878	
−13.906	−11.065	0.61151	−6.0034	11.7224	7.6782	18.9919	−16.2885	3.5179	25.458	−1.649	−0.3740	
13.9653	−1.7118	3.2723	13.4774	−4.1294	−1.5671	−2.5803	−6.1778	−1.046	0.2082	2.6253	−4.5509	

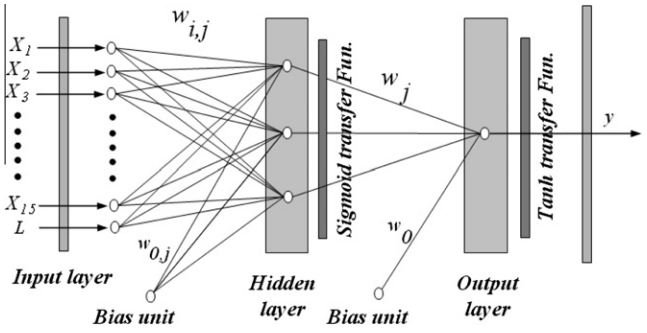

Fig. 6. Structure of the best fitted network (Japanese credit data set), $\boldsymbol{N}^{(16\text{-}3\text{-}1)}$.

samples with 15 attributes. Tsai (2008) varied several parameters of the classification techniques, including the split between training and test data (30–70%, 50–50%, and 70–30%), the nonlinear mapping function for the support vector machine (linear, polynomial, and radial basis) with varying parameters specific to the mapping function, and the number of hidden nodes and learning epoch for the neural networks. For the support vector machines, Tsai did not vary C, instead using a default parameter of $C = 1$. In this paper, the 50–50% randomly split of the Japanese credit data set with first 50% in the training set and second 50% in the test set is used. Similar to the previous section, by using pruning algorithms in *MATLAB7* package software, the best fitted network which is selected, is composed of sixteen inputs, three hidden and one output neurons ($\boldsymbol{N}^{(16\text{-}3\text{-}1)}$). The structure of the best-fitted network is shown in Fig. 6. The weights and biases of the best-fitted network for the Japanese credit data set are given in Table 9. The misclassification percentages of the each model and improvement percentages of the proposed model in comparison with those of other classification models for the Japanese credit data set in both training and test data sets are also summarized in Tables 10 and 11, respectively.

5.2. Gene expression data set

Gene expression data set is another real-life data set, used in order to show the appropriateness and effectiveness of the proposed model for multiple class real-life data set classification. Liang and Keleman (2005) examine the use of a regularized neural network for classification of genes into expression patterns based on a series of microarray measurements at seven time points. These expression patterns are useful in studying the function of individual genes, as genes that are expressed at similar times are often involved in the same or related functions. The neural networks tested by Liang and Keleman (2005) are feed-forward, back-propagation neural networks with a single hidden layer with five to twenty neurons. The regularization is carried out in order to smooth the data and involved modifying the cost function with a penalty term for complexity. Liang and Keleman (2005) use a data set derived experimentally by Chu (1998). The full data set include 6118 genes; however, only 477 of these are classified into seven temporal expression patterns. These expression patterns are based on the time at which the gene becomes most active and labeled as Metabolic, Early I, Early II, Early Middle, Middle, Middle Late, and Late. The raw data included measurements of green signal, green background, red signal, and red background at seven time points: 0, 0.5, 3, 6, 7, 9, and 11.5 h. As described by Liang and Keleman (2005), the data for each time point is transformed as follows:

Table 9
Weights and biases of the final network for Japanese credit data set $\boldsymbol{N}^{(16\text{-}3\text{-}1)}$.

Input weights				Hidden weights	Biases	
	1	2	3	w_j	$w_{0,j}$	w_0
$w_{i,1}$	51.2081	−20.8674	−36.6992			
$w_{i,2}$	−73.023	−96.1825	−20.6655			
$w_{i,3}$	190.8709	−8.5857	48.5775	31.3847	−31.6826	
$w_{i,4}$	−73.3384	27.4435	−8.294			
$w_{i,5}$	−55.9658	−15.0496	−35.3142			
$w_{i,6}$	4.8612	79.9365	58.6241			
$w_{i,7}$	−28.8495	49.2018	−61.7827			
$w_{i,8}$	−65.4581	15.0332	35.4456	−1.6164	−18.5726	−38.5538
$w_{i,9}$	146.8952	−9.0266	−24.1631			
$w_{i,10}$	17.1367	−33.4932	57.9254			
$w_{i,11}$	−27.5656	13.8619	−3.7064			
$w_{i,12}$	−38.9883	31.3838	−9.7578			
$w_{i,13}$	−88.7485	−28.4904	−24.5764			
$w_{i,14}$	8.6577	71.6517	−59.0827	39.3017	−9.6762	
$w_{i,15}$	39.0782	−2.9183	187.172			
$w_{i,16}$	83.523	−41.5886	104.3039			

Table 10
Real-world data sets classification results.

Model	Classification error (%)			
	Japanese credit data set		Gene expression data set	
	Train	Test	Train	Test
Linear discriminant analysis (LDA)	12.3	13.8	14.3	17.2
Quadratic discriminant analysis (QDA)	10.7	15.6	9.7	13.4
K-nearest neighbor (KNN)	11.3	13.8	7.8	10.2
Support vector machines (SVM)	13.2	13.8	4.1	5.7
Artificial neural networks (ANN)	N/A	11.4	5.3	18.0
Hybrid proposed model (non-hierarchical)	11.3	8.1	5.3	6.3
Hybrid proposed model (hierarchical)	N/A	N/A	1.9	3.8

Table 11
Percentage improvement of the proposed model in comparison with those of other classification models (real-world cases).

Model	Improvement (%)			
	Japanese credit data set		Gene expression data set	
	Train	Test	Train	Test
Linear discriminant analysis (LDA)	8.13	41.30	86.71	77.91
Quadratic discriminant analysis (QDA)	−5.61	48.08	80.41	71.64
K-nearest neighbor (KNN)	0.00	41.30	75.64	62.75
Support vector machines (SVM)	14.39	41.30	53.66	33.33
Artificial neural networks (ANN)	N/A	28.95	64.15	78.89

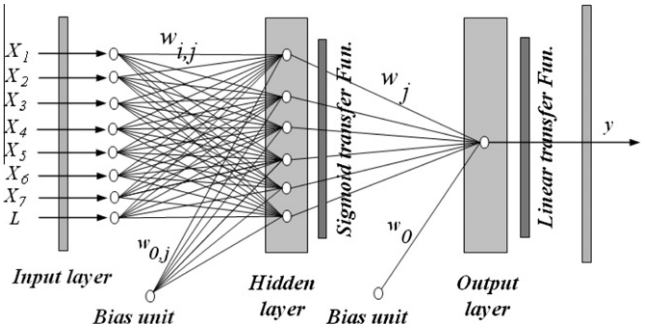

Fig. 7. Structure of the best fitted network (Gene expression data set), $\boldsymbol{N}^{(8\text{-}6\text{-}1)}$.

$$X_t = \mathrm{Log}[(\text{Red signal} - \text{Red background})/(\text{Green signal} - \text{Green background})], \quad (18)$$

As with the other data sets, the Gene expression data set is randomly divided into training and test data sets. As Liang and Kleman (2005) that use two third of the samples for training and one third for testing, we do the same. In the process of splitting the data, we find that one of the classes, late, has only five samples. As this is too few samples for the classification models to work properly, the classes are combined together based on the most similarities in their attribute values in four classes. The resulting four classes is Early (Early I and Early II), Middle (Early-Mid and Middle), Late (Mid-Late and Late), and Metabolic. Similar to the previous sections, by using pruning algorithms in *MATLAB7* package software, the best fitted network which is selected, is composed of eight inputs, six hidden and one output neurons ($\boldsymbol{N}^{(8\text{-}6\text{-}1)}$). The structure of the best-fitted network is shown in Fig. 7. The weights and biases of the best-fitted network for the Gene expression data set are given in Table 12. The misclassification percentages of the each model and improvement percentages of the proposed model in comparison with those of other classification models for the Gene expression data set in both training and test data sets are also summarized in Tables 10 and 11, respectively. Similar to the previous section, the hierarchical proposed model is constructed by the "one versus all" approach for better results in comparison with "one versus rest" approach.

5.3. Comparison with other classification models for real-world application data sets

In the Japanese credit iris data classification case, similar to the previous section, several different architectures of artificial neural network are designed and examined. An architecture composed of fifteen inputs, sixteen hidden and one output neurons ($\boldsymbol{N}^{(15\text{-}16\text{-}1)}$) as designed by Tsai (2008) has been found to be the most accurate among all other neural network architectures, with a 11.4% error rate on the test portion of the data set. However, the proposed model outperforms this model on the test portion of the data set, with an error rate of 8.1%, an improvement of 28.95%. In addition, the proposed model performs the best on both training and test samples, except for training error rate of the quadratic discriminant analysis, with an 11.3% and 8.1% error rate in the training and test samples, respectively. Linear discriminant analysis, *K*-nearest neighbor, and the support vector machine all perform second best with an error rate of 13.8% on the test portion of the data set, 41.30% higher than the proposed model error rate. Quadratic discriminant analysis has an error rate of 15.6%, which is 48.08% higher than that of the proposed model.

In the Gene expression data classification case, our hierarchical proposed model also has the lowest error on the training and test portions of the data set in comparison to other those used models for this data set, with a misclassification rate of 1.9% and 3.8% in the training and test samples, respectively. As previous case, several different architectures of artificial neural network are designed and examined. The best performing architecture for a traditional artificial neural network produces a 5.3% and 18.0% error rate in training and test samples, which hierarchical proposed model improves by 64.15% and 78.89%, respectively. Support vector machine performs second best with an error rate of 5.7% on the test portion of the data set, 33.33% higher than the hierarchical proposed model error rate. *K*-nearest neighbor has an error rate of 10.2%, 62.75% higher than that for the hierarchical proposed model. Quadratic discriminant and linear discriminant analysis have error rates of 13.4% and 17.2%, which is 71.64% and 77.91% higher than that of the hierarchical proposed model, respectively. However, these results, similar to the previous sections, indicate that the proposed model produces consistently good results in a variety of real cases.

Table 12
Weights and biases of the final network for Gene expression data set $\boldsymbol{N}^{(8\text{-}6\text{-}1)}$.

Input weights								Hidden weights	Biases	
$w_{i,1}$	$w_{i,2}$	$w_{i,3}$	$w_{i,4}$	$w_{i,5}$	$w_{i,6}$	$w_{i,7}$	$w_{i,8}$	w_j	$w_{0,j}$	w_0
−3.3017	65.5952	59.11	−57.8499	−148.207	−2.4866	17.5391	23.754	0.92044	−4.3516	3.0259
−16.7609	113.1719	51.6045	−24.2982	−43.9624	−59.7476	−92.5946	21.8746	−1.5096	34.7111	
8.3691	−159.653	−85.9385	−60.1087	−45.8549	100.6706	169.834	17.7931	−3.0818	−38.401	
3.054	6.1253	5.9243	6.566	4.3777	2.0008	3.2012	5.2264	−0.49173	7.7327	
−21.9816	33.3968	24.1211	−3.4951	−37.0104	−14.1262	2.2978	−1.0702	−0.53938	8.4801	
−0.43547	−0.49018	5.9225	2.0089	4.3482	3.5945	2.5461	7.0595	1.5091	5.4826	

6. Conclusions

Classification plays an important role in many applications related to artificial intelligence in the sense of predictive decision in information processing. These applications spanned a wide range of research fields including business, medicine, biology, image recognition, data mining, etc. Many researches in classification have been argued that the performance improves in combined models. In hybrid models, the aim is to reduce the risk of using an inappropriate model by combining several models to reduce the risk of failure and obtain results that are more accurate. Typically, this is done because the underlying process cannot easily be determined. The motivation for combining models comes from the assumption that either one cannot identify the true data generating process or that a single model may not be sufficient to identify all the characteristics of the time series.

In this paper, a new hybrid model of artificial neural networks is proposed as an alternative model for classification problems using the multiple linear regression models. The main aim of the proposed model is to use the unique advantages of the multiple linear regression models in linear modeling in order to overcome the linear modeling deficiency of the traditional artificial neural networks. The proposed model consists of two phases, (i) summarizing the linear components in the attributes in a new attribute for better modeling by neural networks, and (ii) classifying data by a neural network using original attributes and a generated linear attribute by multiple linear regression. Six well-known benchmark (synthetic and real-life) and real-world data sets—the Ripley synthetic data set, the Pima Indian Diabetes data set, the Fisher iris data set, the Forensic glass data set, the Japanese credit data set, and the Gene expression data set—are used in this paper in order to show the appropriateness and effectiveness of the proposed model for both two-class and multiple-class classification tasks. The obtained results of the two-class problems indicate that the proposed model to be superior to all alternative models for both synthetic and real-life benchmark data sets.

In order to solve multiple-class problems, in this paper a hierarchical version of the proposed model is developed by examining three different approaches including ''one versus one'', ''one versus rest'', and ''one versus all''. Among these approaches, the ''one versus all'' approach yield more accurate results and apply for constructing the hierarchical version of the proposed model. Empirical results for this group of problems indicate that the hierarchical proposed model consistently outperforms traditional multilayer perceptrons and other models used in this paper such as linear discriminant analysis, quadratic discriminant analysis, *K*-nearest neighbor, and support vector machines.

Acknowledgments

The authors wish to express their gratitude to Dr. Gholam Ali Raissi Ardali, Department of Industrial Engineering, Isfahan University of Technology, who greatly helped us.

References

Acharya, U., Bhat, P., Iyengar, S. S., Rao, A., & Dua, S. (2003). Classification of heart rate data using artificial neural network and fuzzy equivalence relation. *Pattern Recognition, 36*, 61–68.

Aci, M., Inan, C., & Avci, M. (2010). A hybrid classification method of *k* nearest neighbor Bayesian methods and genetic algorithm. *Expert Systems with Applications, 37*, 5061–5067.

Amanda, J. C. (1999). *Combining artificial neural nets: Ensemble and modular multinet systems*. London: Springer.

Amasyali, M., & Ersoy, O. (2008). Cline: A new decision-tree family. *IEEE Transactions on Neural Networks, 19*(2), 356–363.

Asuncion, A., & Newman, D. (2007). *UCI machine learning repository*. Irivine, CA: University of California, School of Information and Computer Science, 2007.

Banerjee, A., Kiran, K., Murty, U., & Venkateswarlu, Ch. (2008). Classification and identification of mosquito species using artificial neural networks. *Computational Biology and Chemistry, 32*, 442–447.

Benardos, P. G., & Vosniakos, G. C. (2007). Optimizing feed-forward artificial neural network architecture. *Engineering Applications of Artificial Intelligence, 20*, 365–382.

Berardi, V., & Zhang, G. P. (1999). The effect of misclassification costs on neural network classifiers. *Decision Sciences, 30*(3), 659–668.

Berrueta, L., Alonso-Salces, R., & Heberger, K. (2007). Supervised pattern recognition in food analysis. *Journal of Chromatography A, 1158*, 196–214.

Billings, S., & Lee, K. (2002). Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. *Neural Networks, 15*(2), 262–270.

Breiman, L. (1999). Prediction games and arcing algorithm. *Neural Computation, 11*, 1493–1517.

Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., et al. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America 97*(1), 262–267.

Castellani, M., & Rowlands, H. (2009). Evolutionary artificial neural network design and training for wood veneer classification. *Engineering Applications of Artificial Intelligence, 22*, 732–741.

Chakraborty, S. (2009). Simultaneous cancer classification and gene selection with Bayesian nearest neighbor method: An integrated approach. *Computational Statistics and Data Analysis, 53*, 1462–1474.

Chaovalitwongse, W. (2007). On the time series *k*-nearest neighbor classification of abnormal brain activity. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 37*(6).

Chen, S., Lin, S., & Chou, S. (2010). Enhancing the classification accuracy by scatter-search-based ensemble approach. *Applied Soft Computing* xxx, xxx–xxx, 2010.

Christianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines*. Cambridge University Press.

Chu, S. (1998). The transcriptional program of sporulation in budding yeast. *Science, 282*, 699–705.

Connolly, J., Granger, E., & Sabourin, R. (2010). An adaptive classification system for video-based face recognition. Information Sciences xxx, xxx–xxx.

Curram, S., & Mingers, J. (1994). Neural networks decision tree induction and discriminant analysis: An empirical comparison. *The Journal of the Operational Research Society, 45*(4), 440–450.

Dubois, M., Bohling, G., & Chakrabarti, S. (2007). Comparison of four approaches to a rock facies classification problem. *Computers & Geosciences, 33*, 599–617.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern classification*. New York: John Wiley & Sons, Inc..

Enas, G., & Choi, S. (1986). Choice of the smoothing parameter and efficiency of *k*-nearest neighbor. *Computers and Mathematics with Applications, 12*, 235–244.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7*, 465–475.

Fix, E., & Hodges, J. (1951). *Discriminatory analysis – Nonparametric discrimination: Consistency properties*. Project No. 21-49-004, Report No. 4, Contract No. AF 41(128)-31, USAF School of Aviation, Randolph Field, Texas.

Fix, E., & Hodges, J. (1952). *Discriminatory analysis – Nonparametric discrimination: Small sample performance*. Project No. 21-49-004, Report No. 11, Contract No. AF 41(129)-31, USAF School of Aviation, Randolph Field, Texas.

Ghiassi, M., & Burnley, C. (2010). Measuring effectiveness of a dynamic artificial neural network algorithm for classification problems. *Expert Systems with Applications, 37*, 3118–3128.

Ghosh, A. K. (2006). On optimum choice of *K* in nearest neighbor classification. *Computational Statistics & Data Analysis, 50*, 3113–3123.

Guven, A., & Kara, S. (2006). Classification of electro-oculogram signals using artificial neural network. *Expert Systems with Applications, 31*, 199–205.

Hosseini, H., Luo, D., & Reynolds, K. J. (2006). The comparison of different feed forward neural network architectures for ECG signal diagnosis. *Medical Engineering & Physics, 28*, 372–378.

Hur, J., & Kim, J. (2008). A hybrid classification method using error pattern modeling. *Expert Systems with Applications, 34*, 231–241.

Kaczmarczyk, K., Wit, A., Krawczyk, M., & Zaborski, J. (2009). Gait classification in post-stroke patients using artificial neural networks. *Gait & Posture, 30*, 207–210.

Kara, S., & Okandan, M. (2007). Atrial fibrillation classification with artificial neural networks. *Pattern Recognition, 40*, 2967–2973.

Karci, A., & Demir, M. (2009). Estimation of protein structures by classification of angles between a-carbons of amino acids based on artificial neural networks. *Expert Systems with Applications, 36*, 5541–5548.

Khashei, M. (2005). *Forecasting the Esfahan steel company production price in Tehran metals exchange using artificial neural networks (ANNs)*. Master of Science Thesis, Isfahan University of Technology, 2005.

Khashei, M., & Bijari, M. (2010). An artificial neural network (p, d, q) model for time series forecasting. *Expert Systems with Applications, 37*, 479–489.

Khashei, M., Bijari, M., & Raissi, GH A. (2009). Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (ANNs). *Neurocomputing, 72*, 956–967.

Khashei, M., Hejazi, S. R., & Bijari, M. (2008). A new hybrid artificial neural networks and fuzzy regression model for time series forecasting. *Fuzzy Sets and Systems, 159*, 769–786.

Kim, N., Kehtarnavaz, N., Yeary, M., & Thornton, S. (2003). DSP-based hierarchical neural network modulation signal classification. *IEEE Transactions on Neural Networks, 14*(5), 1065–1071.

Kruzlicova, D., Mocak, J., Balla, B., Petka, J., Farkova, M., & Havel, J. (2009). Classification of Slovak white wines using artificial neural networks and discriminant techniques. *Food Chemistry, 112*, 1046–1052.

Lee, J., & Ersoy, O. (2007). Consensual and hierarchical classification of remotely sensed multispectral images. *IEEE Transactions on Geoscience and Remote Sensing, 45*(9), 2953–2963.

Liang, Y., & Keleman, A. (2005). Temporal gene expression classification with regularised neural network. *International Journal of Bioinformatics Research and Applications, 14*, 399–413.

Malhotra, M., Sharma, S., & Nair, S. (1999). Decision making using multiple models. *European Journal of Operational Research, 114*, 1–14.

Marks, S., & Dunn, O. (1974). Discriminant functions when covariance matrices are unequal. *Journal of the American Statistical Association, 69*, 555–559.

Maulik, U., & Mukhopadhyay, A. (2010). Simulated annealing based automatic fuzzy clustering combined with ANN classification for analyzing microarray data. *Computers & Operations Research, 37*, 1369–1380.

Muezzinoglu, M., & Zurada, J. (2006). RBF-based neurodynamic nearest neighbor classification in real pattern space. *Pattern Recognition, 39*, 747–760.

Nagaty, K. (2001). Fingerprints classification using artificial neural networks: a combined structural and statistical approach. *Neural Networks, 14*, 1239–1305.

olmez, T., & Dokur, Z. (2003). Classification of heart sounds using an artificial neural network. *Pattern Recognition Letters, 24*, 617–629.

Ostermark, R. (2000). A hybrid genetic fuzzy neural network algorithm designed for classification problems involving several groups. *Fuzzy Sets and Systems, 114*, 311–324.

Pendharkar, P. (2001). An empirical study of design and testing of hybrid evolutionary-neural approach for classification. *Omega, 29*, 361–374.

Pendharkar, P. C. (2005). A threshold-varying artificial neural network approach for classification and its application to bankruptcy prediction problem. *Computers & Operations Research, 32*, 2561–2582.

Polat, K., & Gunes, S. (2009). A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems. *Expert Systems with Applications, 36*, 1587–1592.

Porter, W., & Liu, W. (1996). Fuzzy HMC classifiers. *Information Sciences, 94*, 151–178.

Qiu, X., Tao, N., Tan, Y., & Wu, X. (2007). Constructing of the risk classification model of cervical cancer by artificial neural network. *Expert Systems with Applications, 32*, 1094–1099.

Rajan, K., Ramalingam, V., Ganesan, M., Palanivel, S., & Palaniappan, B. (2009). Automatic classification of Tamil documents using vector space model and artificial neural network. *Expert Systems with Applications, 36*, 10914–10918.

Ripley, B. (1994). Neural networks and related methods for classification. *Journal of the Royal Statistical Society Series B – Methodological, 56*(3), 409–456.

Rumelhart, D., & McClelland, J. (1986). *Parallel distributed processing*. Cambridge, MA: MIT Press,.

Sahoo, G., Schladow, S., & Reuter, J. (2009). Forecasting stream water temperature using regression analysis, artificial neural network, and chaotic non-linear dynamic models. *Journal of Hydrology, 378*, 325–342.

Satapathy, S., Murthy, J., Prasad Reddy, P., Misra, B., Dash, P., & Panda, G. (2009). Particle swarm optimized multiple regression linear model for data classification. *Applied Soft Computing, 9*, 470–476.

Silva, L., Marques, J., & Alexandre, L. A. (2008). Data classification with multilayer perceptrons using a generalized error function. *Neural Networks, 21*, 1302–1310.

Sinha, S. K., & Fieguth, P. W. (2006). Neuro-fuzzy network for the classification of buried pipe defects. *Automation in Construction, 15*, 73–83.

Smith, C. A. (1947). Some examples of discrimination. *Annals of Eugenics, 13*, 272–282.

Song, J., & Tang, H. (2005). Support vector machines for classification of homo-oligomeric proteins by incorporating subsequence distributions. *Journal of Molecular Structure: THEOCHEM, 722*, 97–101.

Tagluk, M., Akin, M., & Sezgin, N. (2010). Classification of sleep apnea by using wavelet transform and artificial neural networks. *Expert Systems with Applications, 37*, 1600–1607.

Tsai, C. (2008). Financial decision support using neural networks and support vector machines. *Expert Systems, 25*(4), 380–393.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.

Viaene, S., Derrig, R., Baesens, B., & Dadene, G. (2002). A comparison of state-of-the art classification techniques for expert automobile insurance claim fraud detection. *The Journal of Risk and Insurance, 69*(3), 373–421.

Wang, S., Li, X., Zhang, S., Gui, J., & Huang, D. (2010). Tumor classification by combining PNN classifier ensemble with neighborhood roughest based gene reduction. *Computers in Biology and Medicine, 40*, 179–189.

Yildiz, T., Yildirim, S., & Altilar, D. (2008). *Spam filtering with parallelized KNN algorithm*. Akademik Bilisim, 2008.

Zhang, G. P. (2001). An investigation of neural networks for linear time-series forecasting. *Computers and Operations Research, 28*, 1112–1183.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting, 14*, 35–62.