Iterative Least Squares Functional Networks Classifier

Emad A. El-Sebakhy, Ali S. Hadi, and Kanaan A. Faisal

***Abstract*—This paper proposes unconstrained functional networks as a new classifier to deal with the pattern recognition problems. Both methodology and learning algorithm for this kind of computational intelligence classifier using the iterative least squares optimization criterion are derived. The performance of this new intelligent systems scheme is demonstrated and examined using real-world applications. A comparative study with the most common classification algorithms in both machine learning and statistics communities is carried out. The study was achieved with only sets of second-order linearly independent polynomial functions to approximate the neuron functions. The results show that this new framework classifier is reliable, flexible, stable, and achieves a high-quality performance.**

***Index Terms*—Functional networks, minimum description length, statistical pattern recognition.**

I. Introduction

PATTERN classification problem is a supervised learning multidisciplinary research problem. It occurs in a wide range of human activity. At its broadest, the term could cover any activity in which some decision is made on the basis of currently available information, and a pattern classification procedure is a method for repeatedly making such judgments in new situations. The function of pattern classification is to categorize an unknown pattern into a distinct class based upon a suitable similarity measure. Several research studies for classification have been proposed in the literature, but studies have shown that so far no algorithm uniformly outperforms all others in terms of accuracy and quality [12], [15], [17]. There is no consensus in the literature as to which one performs better than the others. Comparative studies among the most common machine learning and statistics classifiers have been carried out in [8] and highlighted both their advantages and disadvantages drawbacks.

Recently, the scientists have been supporting the hybrid classifier, which combines more than one classifier to get better performance [2], [10]. [12], [18], [20], [22]. The main contribution of this paper is to propose a new intelligent systems classifier called *unconstrained functional network using the iterative least squares scheme* for learning and select the best model based on the minimum description length (MDL) criterion. As it is illustrated as follows, this new framework will be able to get over some of the most common drawbacks of the existing classifiers. In addition, it is much easier to use and, as compared to existing methods, it takes much less computations. Furthermore, it uses both domain knowledge and data knowledge to build a decision.

The rest of this paper is organized as follows. Functional networks background is introduced in Section II. The iterative least squares functional networks classifier methodology is explained in Section III. Comparison with the most common classifiers is investigated using real-world application in Section IV. Section V contains both conclusion and some ideas for future work.

Manuscript received September 16, 2005; revised April 23, 2006 and October 11, 2006; accepted November 8, 2006. This work was supported by the Cornell University, Ithaca, NY; King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia; and Egyptian and Educational Bureau, Washington D.C.

E. A. El-Sebakhy and K. A. Faisal are with the Computer Science, College of Computer Science and Engineering, Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dahran 31261, Saudi Arabia (e-mail: dodi5@ccse.kfupm.edu.sa; faisal@ccse.kfupm.edu.sa).

A. S. Hadi is with the Statistical Sciences Department, Cornell University, Ithaca NY 14850 USA (e-mail: ali-hadi@cornell.edu; ahadi@aucegypt.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2007.891632

II. Functional Networks

Functional networks were recently introduced as a generalization of the standard neural networks, which deal with general functional models instead of sigmoidal-like ones. The neuron functions are learned instead of weights as in the standard neural networks. Unlike neural networks, both inputs and outputs need not be normalized. Herein, we give the fundamentals of functional networks, and for details, one may refer to the publications of [4]–[6] and [9].

As it can be seen in Fig. 1, a functional network consists of the following: 1) several layers of storing units, one layer for containing the input data (x_i; $i = 1, \ldots, 4$), another for containing the output data (x_7) and none, one, or several layers to store intermediate information (x_5 and x_6); 2) one or several layers of processing units that evaluate a set of input values and delivers a set of output values (f_i); and 3) a set of directed links. Generally, functional networks extend the standard neural networks by allowing neuron functions f_i to be not only true multiargument and multivariate functions, but to be different and learnable, instead of fixed functions. In addition, the neuron functions in functional networks are unknown functions from a given family, such as *polynomial, exponential,* and *Fourier*, etc., to be estimated during the learning process. Furthermore, functional networks allow connecting neuron outputs, forcing them to be coincident.

III. Functional Networks Classifier

The goal in the pattern classification problem is to learn the probability π_{ik} that the object i falls in one of the predefined

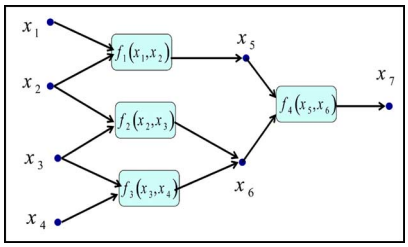

Fig. 1. Functional network architecture: an example.

classes $\{A_k\}_{k=0}^{c-1}$. Based on this definition, we assume that the initial functional network classification model can be written as

$$\pi_{ik} = h_k(\mathbf{x}_i, \boldsymbol{\Theta}_k), \qquad i = 1, \ldots, n \tag{1}$$

where $\mathbf{x}_i \equiv (x_{i1}, \ldots, x_{ip})$ is the ith observation in a given training set $D = \{(\mathbf{x}, y)\}$, where $\mathbf{x} \in \Re^p$ and $y \in \Re$. The matrix $\boldsymbol{\Theta}_k = [\theta_{1k}, \ldots, \theta_{pk}]^T$ represents the functional network parameters needed to be learned. The probabilities in (1) are subject to two constraints

$$h_k(\mathbf{x}_i, \boldsymbol{\Theta}_k) \geq 0 \quad \text{and} \quad \sum_{k=0}^{c-1} h_k(\mathbf{x}_i, \boldsymbol{\Theta}_k) = 1. \tag{2}$$

Therefore, the functions $h_k(\mathbf{x}_i, \boldsymbol{\Theta}_k)$ are to be learned based on the available data, bearing in mind the restrictions on $h_k(\mathbf{x}_i, \boldsymbol{\Theta}_k)$ in (2). Commonly used parametric models for $h_k(\mathbf{x}_i, \boldsymbol{\Theta}_k)$ are as follows.

1) The cumulative distribution function (CDF)

$$h_k(\mathbf{x}_i, \boldsymbol{\Theta}_k) = F^{-1}(\mathbf{x}^T \boldsymbol{\Theta}_k) \tag{3}$$

where $F(.)$ is an appropriate *cdf* and $\boldsymbol{\Theta}_k$ are the parameters of $F(.)$.

2) The *sigmoidal* function

$$h_k(\mathbf{x}_i, \boldsymbol{\Theta}_k) = S(\mathbf{x}^T \boldsymbol{\Theta}_k) \tag{4}$$

where $S(.)$ is any sigmoidal function satisfying, possibly after normalization, the constraints in (2).

Therefore, when dealing with the classification problem using functional networks, we assume that the probability π_{ik} can be written as

$$\pi_{ik} = \mathbf{p}\left(g_k(\mathbf{x}_i, \boldsymbol{\Theta}_k)\right) \tag{5}$$

where $g_k(\mathbf{x}_i, \boldsymbol{\Theta}_k)$ are unknown, but unrestricted functions to be learned from the data, and the function $\mathbf{p}(.)$ must satisfy the probability conditions. Since functional networks do not make any assumptions about the function $g_k(\mathbf{x}_i, \boldsymbol{\Theta}_k)$, then it can be known or unknown in form AND/OR in the parameters. It could be linear or nonlinear. However, since we approximate $g_k(\mathbf{x}_i, \boldsymbol{\Theta}_k)$ by a family of linearly independent functions, one may think of functional networks here as *semiparametric* procedures. We choose one of the most common discrepancy measures to learn the neuron functions $g_k(\mathbf{x}_i, \boldsymbol{\Theta}_k)$. In this paper, we are going to choose the iterative least squares optimization criterion with the class indicator variables and the semiparametric procedures. This will lead to fast convergence and efficient results for the output model. However, one can choose different loss functions.

A. Learning Algorithm

The learning procedures in the unconstrained functional networks classifier is equivalent to approximate the neuron functions using some families of linearly independent elementary functions $\Psi_j = \{\psi_{r_j}(x_{ij}); r_j = 1, 2, \ldots, m_{jk}\}$, that is

$$g_k(\mathbf{x}_i, \boldsymbol{\Theta}_k) \doteq \sum_{r_1}^{m_{1k}} \cdots \sum_{r_p}^{m_{pk}} c_{r_1 r_2 \ldots r_p} \psi_{r_1}(x_{i1}) \ldots \psi_{r_p}(x_{ip})$$

where $c_{r_1} \ldots r_p$ are the parameters in the network model [4]–[6]. We note that, in this approach, the class category y_k is immediately learned using $E(y_k|x_i) = \pi_{ik}$, for $i = 1, \ldots, n$; and $k = 0, \ldots, c-1$. Here, we assume that the class category has a multinomial prior distribution. Hence, the goal is to classify an object x_i to one and only one of these known classes. Therefore, we choose the model

$$y_{ik}|x_i = \pi_{ik} + \varepsilon_{ik} \tag{6}$$

where y_{ik} is an indicator variable, which is a $(c+1)$ binary variable coded 0 or 1 to indicate the group membership of an observation, that is

$$y_{ik} = \begin{cases} 1, & \text{if } x_i \in A_k \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $\sum_{k=0}^{c-1} y_{ik} = 1$, for $i = 1, \ldots, n$, regardless the value of y. These binary variables are introduced only to clarify the minimized function and are not used in the actual functional network procedures. The *iterative least squares* loss error function is written as

$$\text{Min}\left\{\mathbf{Q} = \sum_{i=1}^{n}\left[y_{ik} - \left(\frac{e^{g_k(\mathbf{x}_i, \theta_{jk})}}{1 + \sum_{k=1}^{c-1} e^{g_k(\mathbf{x}_i, \theta_{jk})}}\right)\right]^2\right\}$$

subject to $||\theta_{jk}|| = 1$, for $j = 1, \ldots, (m+1)^p$. The minimum is obtained by solving the system of equations $\partial l(\theta_{jk})/\partial c_{r_1 \ldots r_p} = 0$ in $c_{r_1 \ldots r_p}$

$$\frac{\partial \mathbf{Q}}{\partial \theta_{jk}} = \sum_{i=1}^{n} [2\pi_{ik}(y_{ik} - \pi_{ik})(\pi_{ik} - 1)] \frac{\partial g_k(\mathbf{x}_i, \theta_{jk})}{\partial \theta_{jk}}.$$

Therefore, we obtain a system of equations that can be solved using the iterative numerical method. We note that the number of parameters in the unconstrained functional network model is $\prod_{j=1}^{p} m_{jk}$, which is a very large number, and is computationally expensive. One way to reduce the parameters in the functional network is to write the neuron functions $g_k(\mathbf{x}_i, \boldsymbol{\Theta}_k)$ as

$$g_k(\mathbf{x}_i, \boldsymbol{\Theta}_k) \doteq \sum_{r=1}^{2^p - 1} \prod_{j=1}^{p} g_{rjk}(x_{ij}) \tag{8}$$

where $g_{rjk}(x_{ij})$ are unknown functions, which subsume the unknown parameters $\boldsymbol{\Theta}_k$. The unknown neuron functions

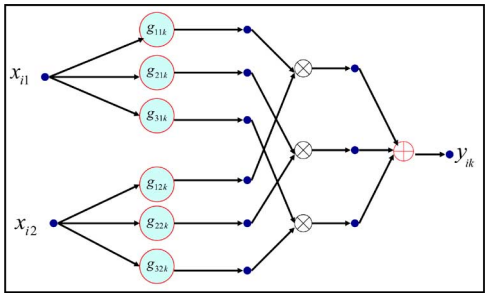

Fig. 2. Unconstrained functional networks classifier with only two feature variables architecture.

$g_{rjk}(x_{ij})$ can be approximated using a family of linearly independent functions $\Psi_{rj} = \{\psi_{rjl}(x_{ij}) : l = 1, m_{rjk}, r = 1, \ldots, 2^p - 1\}$, that is

$$g_{rjk}(x_{ij}) \doteq \sum_{l=1}^{m_{rjk}} a_{rjkl}\psi_{rjl}(x_{ij}). \tag{9}$$

Therefore, we only need to learn the parameters $\{a_{rjkl}\}$. For more simplification, the neuron functions $g_k(\mathbf{x}_i, \theta_{jk})$ in (8) can be expanded as

$$\begin{aligned} g_k(\mathbf{x}_i, \boldsymbol{\Theta}_k) \doteq & \sum_{j=1}^{p} g_{jjk}(x_{ij}) + \sum_{r=LL_1}^{UL_2} g_{rj_1k}(x_{ij_1})\, g_{rj_2k}(x_{ij_2}) \\ & + \ldots + \sum_{r=LL_{p-1}}^{UL_p} g_{rj_1k}(x_{ij_1}) \ldots g_{rj_pk}(x_{ij_p}), \\ & j_1, j_2, \ldots, j_p = 1, \ldots, p \\ & \text{such that } j_1 \neq j_2 \neq \ldots \neq j_p \end{aligned} \tag{10}$$

where $LL_s = \sum_{j=0}^{s} \binom{p}{j}$ and $UL_t = \sum_{j=1}^{t} \binom{p}{j}$; for all $s = 1, \ldots, p-1$ and $t = 2, \ldots, p$. Fig. 2 shows the architecture of the corresponding functional network classification model (6) for two feature variables only, say $\mathbf{x}_1$ and $\mathbf{x}_2$. Suppose that each one of the neuron functions $g_{rjk}(x_{ij})$ is approximated using the sets of known linearly independently families. Now, using the uniqueness conditions as in [3] and [4], the neuron function $g_{11k}(x_{i1})$ is the only neuron that contains the constant term. The rest of the neurons do not contain the constant term. Therefore, we obtain a system of nonlinear equations of size $(m+1)^p \times (m+1)^p$. The resultant optimization problem can be solved numerically. Thus, for a given training set D, the decision functions $g_k(.)$ on $\mathbf{x}_i \in \Re^p$ can be learned from this training sample and then we actually predict the response variable y for any given test sample $\mathbf{x}^* \in \Re^p$. The unconstrained *functional networks with iterative least squares* (FN-ILS) classifier and the corresponding learning procedures for p predictor variables are summarized in Fig. 4.

B. Model Selection and Validation

Once the learning process has finished and the neuron functions of the functional network have been computed, it is important to check the quality of the resulting model. A computer implementation of model selection can be done in several ways; see [10] and [16] for more details. The best functional network classifier is chosen based on the smallest value of the description length. The form of the description length for the classification problem is defined as

$$L(\boldsymbol{\Theta}_k) = \frac{m\log(nk)}{2} + \frac{nk}{2}\log\left(\frac{1}{nk}\sum_{i=1}^{n}\varepsilon_i^2(\boldsymbol{\Theta}_k)\right)$$

for $k = 0, \ldots, c-1$, where m and k are the number of elements in the family and the category levels, respectively. We note that the description length has two terms. The first term is a penalty for including too many parameters in the functional network model. The second term measures the quality of the functional network model fitted to the training set. Therefore, the best model is the model with the smallest value of $L(\boldsymbol{\Theta}_k)$.

The final step in the development and implementation of the classification model is the judgment and evaluation of the quality and capability of the fitted model (validation step). To achieve this step, we compute the following quality measures.

1) Compute both *correct classification rate (CCR)* and *average squared classification error (ASCE)* that are defined as

$$\text{CCR} = \frac{\sum_{k=0}^{c-1} CC_k}{n} \quad \text{ASCE} = \frac{\sum_{k=0}^{c-1}[n_k - CC_k]^2}{n} \tag{11}$$

where CC_k is the number of correctly classified observations and n_k is the number of observations in class k. The model with the highest correct classification rate is the one with the better performance. The smaller ASCE, the better classifier performance is. In general, we use both CCR and ASCE to judge the functional network classifier performance, and compare it with the one of the most common classifiers in literatures. The better classifier is the one with the highest CCR value and the smallest ASCE value.
2) *Time of execution*: It is the time needed to execute the classifier till obtaining the final classification model. The lower computation cost is the better classifier.
3) *The minimum description length (MDL)* criterion: As explained previously, the best model is the one with the smallest MDL value.
4) *Number of parameters*: A model with more parameters is able to lead to a better fit, but it can be less practical than another with smaller number of parameters. Therefore, knowing number of parameters in the final model will lead to better judgement.

From the previous illustrations, we conclude that the unconstrained functional networks classifier gets over the "black box," the option chosen by both hidden layers and neurons, and the limited ability to explicitly identify possible causal relationships problems in the standard neural networks. In addition, the learned parameters in functional networks have significant meaning and the final functional networks model is determined by solving a system of equations; then, it does not have the difficulty of local optima in both feedforward neural network (FFN) [2], [21] and radial basis function (RBF)[13], [14], [22]. The performance of the functional networks classifier does not have critical parameter, such as the k-*value* and metric function to measure the distance between data item as in the k-nearest-neighbor technique (KNN). The performance of the new classifier is investigated against the most common existing classifiers

- The *K-nn* classifier: The best k is $n^{\frac{3}{8}}$;
- The *SVM* classifier:
  - Gauessian and Sigmoidal Kernel;
  - Parameters, $\sigma = 0.5$, $\gamma = 0.5$, and $\delta = 1$;
  - A constant $M = 100$, satisfying $0 \leq \lambda_{i,k} \leq M$;
- The *MLP* classifier:
  - Use *newff()*; with two hidden layer ;
  - # hidden nodes is $min\{2p + 2, \frac{n}{10}\}$;
  - Activation functions: *purelin*, *logsig*, *tansig*;
  - Use 3000 iteration and a tolerance 0.01;
- The *FN-ILS* classifier:
  - Use t$q = 2$, polynomial degree;
- Both *PPN* and *RBF* classifier:
  - One parameter: $\sigma = \frac{d_{max}}{\sqrt{k}}$;
  - where d_{max} is the maximum Euclidean distance between the selected centers;
  - k is the number of the centers;
- No parameters for *MLR* and MDA classifiers.

Fig. 3. Parameters for different classifiers.

in both machine learning and statics communities using both real-world applications and simulation studies. For the sake of both space and simplicity, we recorded only these investigations on one real-world application as it is shown in Section IV.

IV. Real Application and Comparative Studies

A. Initializations

The data sets we use here come from the University of California at Irvine website.[1] All the computations are implemented using MATLAB V6 under Pentium IV personal computer.

We use the *stratified sampling* and cross-validation techniques to make sure that we get the same proportion from each group as in the original data. We repeat the estimation and validation processes for $N = 500$ times; then, we compute all the quality measures explained in Section III for all classifiers. Next, we summarize the results by computing the average, the standard deviation, and the coefficient of variation of each quality measure over these 500 runs.

We draw two graphs: one for the mean of CCR versus its standard deviation over the 500 runs, and the other for the mean of the ASCE versus mean of MDL. These graphs in Fig. 6 help us to decide which classifier is better in its performance. In both plots, each classifier is represented by a symbol. In the graph of the average of CCR versus its standard deviation, a good classifier should appear in the upper left corner of the graph. In the graph of average of MDL versus average of ASCE, a good classifier should appear in the bottom left of the plot. In addition, corresponding to these graphs, we summarize the results in Tables I and II. In these tables, the highest CCRs are given in boldface.

B. Input Parameters for Classifiers

We investigate both constrained functional networks classifier and existing classifiers on distinct databases, such as thalassemias screening database. Different classifiers require different sets of input parameters; we select the input parameters that give the best performance of the classifier. The configuration for each classifier and how it can be used during the process are illustrated in Fig. 3.

[1]URL: ftp://ftp.ics.uci.edu/pub/machine-learning-databases.

(1.) *Input*:
- Data: $D = \{(\mathbf{y}_i, \mathbf{x}_i)\}; \mathbf{x}_i \in \Re^p,; \; i = 1, \ldots, n$ drawn from c distinct classes,

(2.) *Initialization*:
- α and the degree of approximation, q.
- Linearly independent function, $\{\psi_{rjl}(x_{ij})\}$.
- Compute the matrix, $\mathbf{W}$ of size $n \times (q+1)^p$.
- Probabilities: $\prod = [\prod_1, \ldots, \prod_c]$; where $\prod_k = [\pi_{1k}, \ldots, \pi_{nk}]^T$; $\pi_{ik} = \frac{n_k}{n}$.
- Coded matrix $\mathbf{Y} = (y_{ik})$ of size $n \times c - 1$.
- Unknown matrix: $\widehat{\mathbf{\Theta}} = (\hat{\Theta}_{ik})$ of size $n \times c - 1$.
- Compute $\mathbf{\Theta}$ using $y_{ik} \mid x_i = \pi_{ik} + \varepsilon_{ik}$.

(3.) *Computational Process*:
For $k = 0, 1, \ldots, c$, compute the following:
- The vector $\hat{\pi}_{ik} = \dfrac{e^{g_k(\mathbf{x}_i, \Theta_k)}}{1 + \sum_{k=1}^{c-1} e^{g_k(\mathbf{x}_i, \Theta_k)}}$.
- The Iterative Least Squares Loss function: $\mathbf{Q} = \sum_{i=1}^{n} [\hat{y}_{ik} - \hat{\pi}_{ik}]^2$.
- The vector $\hat{y}_{ik}$, and confusion matrix (CM).
- The vector, $\theta_k = [\theta_{1k}, \ldots, \theta_{nk}]^T$.
- The value of $SSE = \sum_{i=1}^{n} [y_{ik} - \hat{\pi}_{ik}]^2$.
- The Log-Likelihood function $l(\theta_{jk})$ $\sum_{i=1}^{n} \left[\sum_{k=1}^{c-1} y_{ik}\, g_k(\mathbf{x}_i, \Theta_k) - ln\left(1 + \sum_{k=1}^{c-1} e^{g_k(\mathbf{x}_i, \Theta_k)}\right)\right]$.
- The standard errors (*S.e.*), that is defined as: $S.e = \sqrt{[I(\hat{\theta}_{ik})]^{-1}}$, where $I(\hat{\theta}_{ik})$ is a matrix of size $c\,(m+1)^p \times c(m+1)^p$,
- The *Z-test*, and odd-ratio (OR_k), where $Z_{test} = \dfrac{\hat{\theta}_k}{S.e.}$; $OR_k = e^{\hat{\theta}_k}$.
- The *p-value*: $[1 - c.d.f(Z_{test})]$,
- The degrees of freedom: $[(q+1)^p - 1]\; [c - 1]$.
- The log-likelihood at no coefficients (L_0).
- The value of *G-Statistic*: $G = 2(L_0 - L)$.

(4.) *Building the unconstrained classifier*:
- Select the Model with the smallest $L(\mathbf{\Theta}_k)$.
- Compute Quality measures:
  - ◇ CCR; ASCE; Num.Para., and Exec.Time

(5.) *Validate and use the functional networks classifier*
(6.) *end*.

Fig. 4. Learning algorithm of the functional networks with iterative least squares (FN-ILS) classifier.

C. Thalassemias Data

We apply all classifiers to thalassemias data. Thalassemias are genetic defects that are commonly found in many parts of the world including Africa, the Far East, and the Mediterranean regions [1]. In order to know how many people suffer from this disease, the heterogenous population should be screened. A first level analysis using hemocromocytometric data and a second level examination (total HbA2, globin chain synthesis, and genetic analysis), should be carried out to identify a and h thalassemia carriers. The data set consists of 304 clinical records of eight grade students based on a thalassemia screening carried out by the Ozieri Hospital, Italy. The data has 13 predictor variables, and three categories with 196 from normal cases $(y = 0)$, 81 from a cases $(y = 1)$, and 27 from h cases $(y = 2)$.

In [11] and [19], the training of the computer to classify patients was used, which consisted of 304 clinical records based on a thalassemia screening. The predictor variables, which were considered relevant for the classification were RBC, Hb, Ht, and

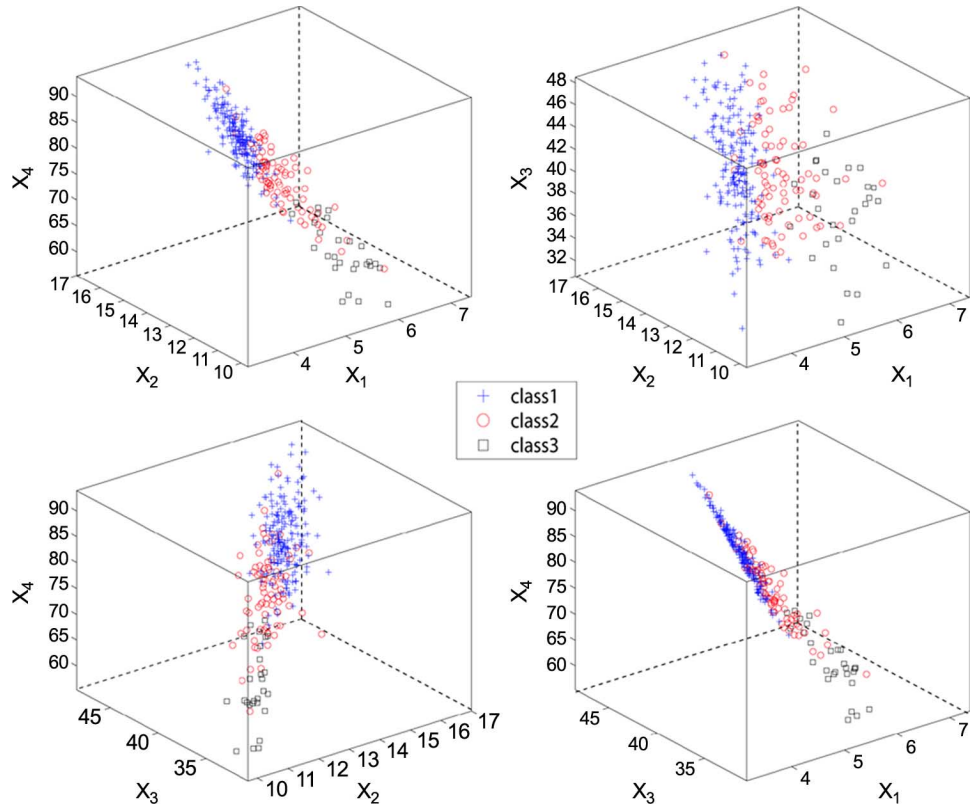

Fig. 5. Scatter plot matrix of the used thalassemias screening database.

TABLE I
THALASSEMIAS DATA: THE INTERNAL VALIDATION RESULTS WITH FOUR PREDICTORS

Classifier	COR-OBS1	COR-OBS2	COR-OBS3	MISS-OBS1	MISS-OBS2	MISS-OBS3	CCR
LogReg	182	57	24	14	24	3	0.865
DiscAnal	160	74	27	36	7	0	0.859
KNN	195	32	9	1	49	18	0.776
SVM	190	78	26	6	3	1	**0.967**
PNN	185	65	20	11	16	7	0.882
RBFN	187	57	16	9	24	11	0.855
FFN	192	76	26	4	5	1	**0.967**
FNBF-ILS	192	77	25	4	4	2	**0.964**

MCV [note: RBC—red blood cell count (106/Al); Hb—hemoglobin (g/dl); Ht—hematocrit (%); and MCV—mean corpuscular volume (fl)].

One can see from the 3-D scatter plot in Fig. 5, that there is some overlap between the two groups. For the internal validation purpose, we summarize the output of both existing classifiers and functional networks backward-forward-iterative least squares (FNBF-ILS) in Table I. The columns 2–4 of Table I contain the number of correctly classified observations in each class. The last column contains the correct classification rate; the highest value is in boldface. The remaining columns contain the number of misclassified observations in each class. From Table I, we observe that KNN, RBF networks, and discriminant analysis (DA) give the lowest CCR value among all classifiers. The FNBF-ILS with $q = 2$ and support vector machines (SVMs) classifiers have the highest CCR. The FNBF-ILS performance is stable in both 2-D and high dimensions. The MLR, probabilistic neural network (PNN), and KNN classifiers perform more or less the same.

To evaluate the performance of all the classifiers on thalassemias data, we divide the given data into testing and training sets. The training set consists of 108 observations (with 55 normal cases, 44 a, and 9 h) and the testing set is made up of 196 observations (141 normal cases, 37 a, and 18 h). Note that the training data has less observations than the testing data, which represents a challenge for all methods. The quality measure results and the scatter plot graphs are shown in Table II and Fig. 6, respectively.

Fig. 6 shows two scatter plots, where each of the eight classifiers is represented by a symbol. The first plot is the average of CCR versus its standard deviation. A good classifier appears in the upper left corner of the graph. The second plot is the average of MDL versus the mean of ASCE. A good method appears in the bottom left corner of the plot.

From Table II and Fig. 6, we observe, for example, that the FFN, logistic regression, SVM, and KNN classifiers have the worst performance in the external validation. The functional networks classifier (FNBF-ILS) is giving the highest values of the average CCR, but with less time of computations. The FFN and FNBF-ILS classifiers have the highest execution time. This is due to their optimization technique for learning the neural functions. All other classifiers, logistic regression, linear discrimi-

TABLE II
THALASSEMIAS DATA: THE EXTERNAL VALIDATION RESULTS

Classification Method	No. Parameter		MDL		Time of Exec.		CCR		ASCE	
	mean	StDev	mean	StDev	mean	StDev	mean	StDev	mean	StDev
Logistic Regression	5	0	-495.109	32.883	0.876	0.051	0.782	0.035	1.014	0.341
Linear Disc Anal.	5	0	-615.915	40.473	0.006	0.008	0.822	0.032	0.811	0.404
KNN	5	0	-586.607	27.906	0.038	0.018	0.799	0.026	0.822	0.228
SVM	5	0	-447.576	31.445	0.110	0.024	0.746	0.031	1.183	0.309
PNN	3	0	-639.327	30.175	1.218	0.077	0.837	0.021	0.623	0.178
RBFN	3	0	-610.906	42.285	1.339	0.056	0.834	0.024	0.562	0.171
FFN	3	0	-537.019	145.088	54.334	25.188	0.768	0.125	1.987	3.647
FNBF-ILS	13	1.873	-689.619	41.689	9.444	4.432	**0.876**	0.021	0.301	0.108

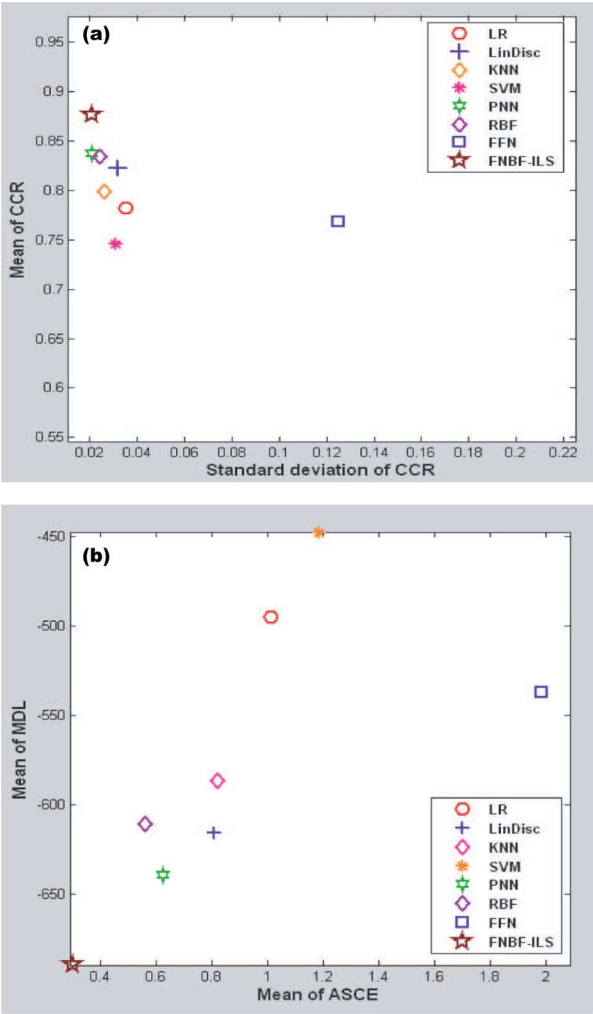

Fig. 6. External validation results: (a) average of CCR versus σ_{CCR} and (b) average of MDL versus average of ASCE.

nant analysis, PNN, and RBF are having close values of the average CCR, and their performances more or less are the same.

V. CONCLUSION

Based on the results obtained from the real-world applications, we observe that the new classifier using the family of linearly independent polynomial functions with degree at most two outperforms the most common existing classifiers in both internal and external validation. In addition, it is giving the highest values of the average correct classification rate and less time of computations. On the other hand, the FNNs classifier has the highest execution time. This is due to its optimization technique for learning the neuron functions. All the other classifiers, logistic regression, linear discriminant analysis, PNN, and RBF have close values of the average correct classification rat and their performance are more or less the same.

We conclude that the functional networks classifier gets over the "black box," the option chosen by both hidden layers and neurons, and the limited ability to explicitly identify possible causal relationships problems in the standard neural networks. In addition, the learned parameters in functional networks have significant meaning and the final functional networks model is determined by solving a system of equations. It did not have the difficulty of local optima as in both feedforward neural networks and RBFs. Furthermore, if the neuron functions are linear, then it leads to the logistic function, but without need to make the linearity assumption *a priori*. If they are nonlinear, it leads to the sigmoidal function, which is the function used in the standard neural network. Therefore, functional networks can be thought of as *semiparametric* for both the logistic regression and generalization of the standard neural network.

Finally, we can say that this new framework can be considered a building stone in the decision making learnable techniques and the results can also be applied in different business, science, engineering, bioinformatics, and other industrial world wide applications. We suggest that for future work, we do more simulation work with different families of linearly independent functions and different kind of models beside the mixture model with different criterion of learning the functional networks classifier.

ACKNOWLEDGMENT

The authors would like to thank the associate editor and the anonymous reviewers for their reading and providing many helpful comments.

REFERENCES

[1] G. Amendoliaa, M. Cossub, B. Ganaduc, G. L. Golosioa, and G. Masalaa, "A comparative study of k-nearest neighbour, support vector machine and multi-layer perceptron for thalassemia screening," *Murac Chemometrics Intell. Lab. Syst.*, vol. 69, pp. 13–20, 2003.
[2] P. Le Callet, C. Viard-Gaudin, and D. Barba, "A convolutional neural network approach for objective video quality assessment," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1316–1327, Sep. 2006.
[3] E. Castillo and R. Ruiz-Cobo, *Functional Equations and Modeling in Science and Engineering*. New York: Marcel Dekker, 1992.
[4] E. Castillo, "Functional networks," *Neural Process. Lett.*, vol. 7, no. 3, pp. 151–159, 1998.
[5] E. Castillo, A. Cobo, J. Gutiérrez, and E. Pruneda, *Introduction to Functional Networks With Applications: A Neural Based Paradigm*. Norwell, MA: Kluwer, 1998.

[6] E. Castillo, A. Cobo, R. Gómez-Nesterkín, and A. S. Hadi, "A general framework for functional networks," *Networks*, vol. 35, pp. 70–82, 1999.
[7] S. Dreiseitl, L. Ohno-Machado, H. Kittler, S. Vinterbo, H. Billhardt, and M. Binder, "A comparison of machine learning methods for the diagnosis of pigmented skin lesions," *J. Biomed. Inf.*, vol. 34, no. 3, pp. 28–36, 2001.
[8] S. Dreiseitla and L. Ohno-Machadob, "Logistic regression and artificial neural network classification models: A methodology review," *J. Biomed. Inf.*, vol. 35, pp. 352–359, 2002.
[9] E. El-Sebakhy, "Functional networks training algorithm for statistical pattern recognition," in *Proc. IEEE 9th Int. Symp. Comput. Commun.*, 2004, vol. 1, pp. 92–97.
[10] B. Fei and J. Liu, "Binary tree of SVM: A new fast multiclass training and classification algorithm," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 696–704, May 2006.
[11] J. Haematol, "British committee for standards in haematology," *Guideline: The Laboratory Diagnosis of Haemaoglobinopathies*, vol. 101, pp. 783–792, 1998.
[12] M. E. Mavroforakis and S. Theodoridis, "A geometric approach to support vector machine (SVM) classification," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 671–682, May 2006.
[13] K. Z. Mao and G. Huang, "Neuron selection for RBF neural network classifier based on data structure preserving criterion," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1531–1540, Nov. 2005.
[14] Y. O. Member, S. Hwang, Y. Ou, C. Chen, and Z. Chen, "Data classification with radial basis function networks based on a novel Kernel density estimation algorithm," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 225–234, Jan. 2005.
[15] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," *Philos. Trans. Roy. Soc. London A, Math. Phys. Sci.*, vol. 209, pp. 415–446, 1999.
[16] J. Li, M. T. Manry, P. L. Narasimha, and C. Yu, "Feature selection using a piecewise linear network," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1101–1115, Sep. 2006.
[17] T. Mitchell, "Machine learning and data mining," *Commun. ACM*, vol. 4, pp. 2–11, 1999.
[18] X. Pan, X. Ye, and S. Zhang, "A hybrid method for robust car plate character recognition," *Eng. Appl. Artif. Intell.*, vol. 18, no. 8, pp. 963–972, 2005.
[19] J. Pathol, "Thalassemia working party of the British committee for standards in haematology task force," *Guidelines for Investigations of the Alpha and Beta Thalassemia Traits*, vol. 47, pp. 289–295, 1994.
[20] K. Polat, S. Sahan, and S. Gne, "A novel hybrid method based on artificial immune recognition system (AIRS) with fuzzy weighted preprocessing for thyroid disease diagnosis," *Expert Syst. Appl.*, 2006, to be published.
[21] J. Wu, Z. Lin, and P. Hsu, "Function approximation using generalized adalines," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 541–558, May 2006.
[22] Z. R. Yang, "A novel radial basis function neural network for discriminant analysis," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 604–612, May 2006.

Emad A. El-Sebakhy received the B.Sc. (*cum laude*) and M.Sc. degrees in mathematics and computational sciences from Mansoura University, Egypt, in 1990 and 1994, respectively, and the Ph.D. degree in computer science and statistics from Cornell University, Ithaca, NY, in 2004.

He is an Assistant Professor of Computer Science at the Information and Computer Science Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. He has been affiliated with Department of Mathematics and Computational Science, Faculty of Science at Mansoura University as an Assistant Professor of Computer Science since August 2004. He is the leader of the computational intelligence in bioinformatics research group at KFUPM. He has achieved numerous research projects with ARAMCO oil and gas company. He is the leader of the computational intelligence and its applications in bioinformatics, engineering, oil and gas industry, and software engineering business research group at the College of Computer Science and Engineering at KFUPM. His research interests include computational intelligence, machine learning, data mining, functional networks, neural networks, fuzzy systems, fuzzy rule-based systems, genetic algorithms, soft computing, and computational intelligence applications.

Dr. Sebakhy has been a member of both artificial intelligence and data mining technical programs and organizing committees of several international IEEE conferences.

Ali S. Hadi received the B.S. degree from Ain Shams University, Egypt, and the M.Sc. and Ph.D. degrees in statistics and computer information system from the New York University, NY, in 1982, 1984, respectively.

He is a Professor Emeritus and former Chairman of the Department of Social Statistics at Cornell University's School of Industrial and Labor Relations. He is the Stephen H. Weiss Presidential Fellow, Cornell University. He is also a member of Cornell's Graduate Field of Statistics and the Department of Statistical Sciences. Currently, he is a Professor of Mathematical, Statistical, and Computational Sciences at the Department of Mathematics, the American University, Cairo, Egypt, where he is the Vice Provost and the Director of Graduate Studies and Research. He has written five books, four book chapters, and more than 75 articles. He has taught graduate and undergraduate courses in statistics, mathematics, expert systems, and artificial intelligence; and has won several excellence-in-teaching awards. He is a Coeditor of the *Journal of Economic and Social Research*, a past Editor of the *Newsletter of the International Environmetrics Society*, an Associate Editor of *Environmetrics*, and an Associate Editor of *TEST: Journal of the Spanish Statistical and Operations Research Society*. His research interests include linear and nonlinear models, multivariate analysis, parameter and quantize estimation, robust statistics and outlier detection, statistical computing and graphics, expert systems, uncertainty in artificial intelligence, Bayesian, Markov, and neural network models, environmetrics, matrix algebra, and simulation.

Dr. Hadi is a Fellow of the American Statistical Association and an Elected Member of the International Statistical Institute, a Fellow of the Royal Statistical Society, a Member of the Institute of Mathematical Statistics, and a Member of the International Association for Statistical Computing.

Kanaan A. Faisal received the B.Sc. degree in systems engineering, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 1980, the M.Sc. degree in computer science from the University of Minnesota, Minneapolis, in 1986, and the Ph.D. degree in computer science, Washington University, Seattle, in 1990.

Currently, he is an Assistant Professor in the Information and Computer Science Department and the Chairman of the Department of Information and Computer Science, King Fahd University of petroleum and Minerals. He is the Chair of the Intelligence Systems Research Group in the Information and Computer Science Department at KFUPM. He is the KFUPM member who is responsible for designing most of the information and computer science ABET courses. His research interests include artificial intelligence, machine learning, data mining, database, and computer education.

Dr. Faisal is a member of the Computational intelligence in Bioinformatics Research Group at KFUPM. He organized several international IEEE conferences on the KFUPM campus.