Neural Comput & Applic (2010) 19:317–328
DOI 10.1007/s00521-009-0310-y

ORIGINAL ARTICLE

A hybrid genetic based functional link artificial neural network with a statistical comparison of classifiers over multiple datasets

Satchidananda Dehuri · Sung-Bae Cho

Received: 17 June 2008 / Accepted: 5 October 2009 / Published online: 24 October 2009

Abstract This paper proposed a hybrid genetic based functional link artificial neural network (HFLANN) with simultaneous optimization of input features for the purpose of solving the problem of classification in data mining. The aim of the proposed approach is to choose an optimal subset of input features using genetic algorithm by eliminating features with little or no predictive information and increase the comprehensibility of resulting HFLANN. Using the functionally expanded of selected features, HFLANN overcomes the nonlinearity nature of problems, which is commonly encountered in single-layer neural networks. The features like simplicity of the architecture and low computational complexity of the network encourage us to use it in classification task of data mining. Further, the issue of statistical tests for comparison of algorithms on multiple datasets, which is even more essential to typical machine learning and data mining studies, has been all but ignored. In this work, we recommend a set of simple, yet safe and robust parametric and nonparametric tests for statistical comparisons of HFLANN with FLANN and RBF classifiers over multiple datasets by an extensive simulation studies.

Keywords Classification · Data mining · Genetic algorithm · FLANN · RBF

S. Dehuri (✉) · S.-B. Cho
Soft Computing Laboratory,
Department of Computer Science,
Yonsei University, 262 Seongsanro,
Sudaemoon-gu, Seoul 120-749, Korea
e-mail: satchi.lapa@gmail.com

S.-B. Cho
e-mail: sbcho@yonsei.ac.kr

List of symbols

Ω	Universal set of individuals
M	Number of classes
X	Number of patterns
N	Number of datasets used for experimental studies
K	Number of algorithms (both proposed and used for comparisons)
P_1^j	Performance of the *j*th algorithms on the *i*th dataset
$\bar{P}$	Mean performance difference of algorithms
σ_i	Standard deviation of the *i*th algorithms over multiple datasets
σ_p	Variance of the difference between two means
R_{pos}	Summation of all positive ranks
R_{neg}	Summation of all negative ranks
R_{s}	The smallest rank among R_{pos} and R_{neg}
α	Level of significance
Z	*z*-Distributions
N	Original set of features
D	Selected set of features
T	Number of iterations
E	Error criterion
T_1	Test set 2
T_2	Test set 1
T	Training/test set 1/2
τ	Tradeoff between criteria

1 Introduction

For the past two decades, there have been a lot of studies focused on the classification problem in the field of data mining [1, 2]. The general goal of data mining is to extract knowledge from large gamut of data, it is important to bear in mind some desirable properties of discovered knowledge. The discovered knowledge should be highly predictive and comprehensible. The relative importance of each of these properties, which can be considered as quality criteria to evaluate discovered knowledge, depends strongly on several factors, such as the kind of data mining task being solved, the application domain, and the user. However, since this work focuses on classification task of data mining, it is important that the discovered knowledge have a high predictive accuracy, even though in many cases, the comprehensibility tends to be more important than predictive accuracy.

Knowledge comprehensibility is usually important for at least two related reasons. First, the knowledge discovery process usually assumes that the discovered knowledge will be used for supporting a decision to be made by a human user. Second, if the discovered knowledge is not comprehensible to the user, he/she will not be able to validate it, hindering the interactive aspect of the knowledge discovery process, which includes knowledge validation and refinement. In addition to giving importance on predictive accuracy of the proposed method, an equal importance is also given to the comprehensibility, which is another considerable criterion. In this paper, we are measuring the comprehensibility of the proposed method by reducing the architectural complexity. As we know, the architectural complexity of functional link artificial neural network (FLANN) [3] is directly proportional to the number of features and the functions in hand for expansion of the given feature value. For reducing the architectural complexity, we first select a few subsets of features (i.e., feature selection [4]) and then applying the usual procedure of function expansion and training by back propagation learning. The steps from selection to learning are accomplished by hybridization of FLANN with genetic algorithms (GAs) [5] and, therefore, we named it as hybrid FLANN (HFLANN).

Traditional statistical classification procedures such as discriminant analysis are built on the Bayesian decision theory [6]. In these procedures, an underlying probability distribution must be assumed in order to calculate the posterior probability upon which the classification decision is made. One major limitation of the statistical models is that they work well only when the underlying assumptions are satisfied. The efficiency of these methods depends to a large extent on the various assumptions or conditions under which the models are developed. Users must have a good knowledge of both data properties and model capabilities before the models can be successfully applied.

Neural networks [7] have emerged as an important tool for classification. The recent vast research activities in neural classification have established that neural networks are a promising alternative to various conventional classification methods. The ANNs are capable of generating complex mapping between the input and the output space, and thus these networks can form arbitrarily complex nonlinear decision boundaries.

Pao et al. [8] have given a direction that their proposed FLANN may be conveniently used for function approximation and can be extended for pattern classification with faster convergence rate and lesser computational load than an multi-layer perceptron (MLP) structure. The FLANN is basically a flat network, and the need of the hidden layer is removed, and hence the learning algorithm used in this network becomes very simple. The functional expansion effectively increases the dimensionality of the input vector, and hence the hyper planes generated by the FLANN provide greater discrimination capability in the input pattern space. Although many types of neural networks can be used for classification purposes [7], we choose feed forward multi-layer networks or multi-layer perceptrons (MLPs) as a benchmark method for comparison. Even though it has a complex architecture and long training time, it is most widely studied and used neural network for classification. In addition, we used FLANN with gradient descent method for classification of our previous work [3] for comparison. Although FLANN with back propagation learning gives promising results but if the number of features and the functions to be used for expansion is large, then the complexity of the architecture increases, and hence it became less comprehensible. Another important point is no matter how intelligent the FLANN is, it will fail to predict the unknown sample if it is applied to low quality data. Hence, to improve the capability of the FLANN for accurate prediction and its comprehensibility, we hybridized with genetic algorithms (GAs). Therefore, we named it as HFLANN. This method not only has practical time complexity, but also achieves good performance.

Over the last years, the machine learning and data mining community has become increasingly alert of the need for statistical validation of the results. This can be ascribed to the maturity of the area, increasing the number of real-life applications and the availability of open algorithmic frameworks that make it easy to develop new algorithms or modify the existing, and compare them among themselves.

The rest of this paper is organized as follows. In Sect. 2, we have discussed the background materials very quickly. Section 3 provides our proposed HFLANN for classification. In Sect. 4, we have presented the experimental studies and a parametric and nonparametric statistical comparative performance with other classifiers like RBF and FLANN trained by back propagation learning. Section 5 concludes the article.

2 Background

In this section, we will discuss the basic background material required for a deep understanding of the proposed method. The section is divided into five subsections, namely, basic working principle of genetic algorithms, a formal model of functional link artificial neural network, the importance of feature selection, classification task of data mining, and the statistical test for comparison of classifiers.

2.1 Genetic algorithms

In this section, we review the function of genetic algorithms (GAs) [5]. GAs are stochastic search algorithms characterized by the fact that a number N of potential solutions (called individuals $I_k \in \Omega$, where Ω represents the space of all possible individuals) of the optimization problem simultaneously sample the search space. This population $P = \{I_1, I_2, \ldots, I_N\}$ is modified according to the natural evolutionary process: after initialization, selection $S{:}I^N \rightarrow I^N$, and recombination $\mathcal{R}{:}I^N \rightarrow I^N$ are executed in a loop until some termination criterion is reached. Each run of the loop is called a generation, and $P(t)$ denotes the population at generation t.

The selection operator is intended to improve the average quality of the population by giving individuals of higher quality a higher probability to be copied into the next generation. Selection thereby focuses on the search of promising regions in the search space. The quality of an individual is measured by a fitness function $f{:}P \rightarrow R$. Recombination and mutation change the genetic material in the population in order to obtain new points in the search space. Figure 1 depicts the steps that are performed in GA.

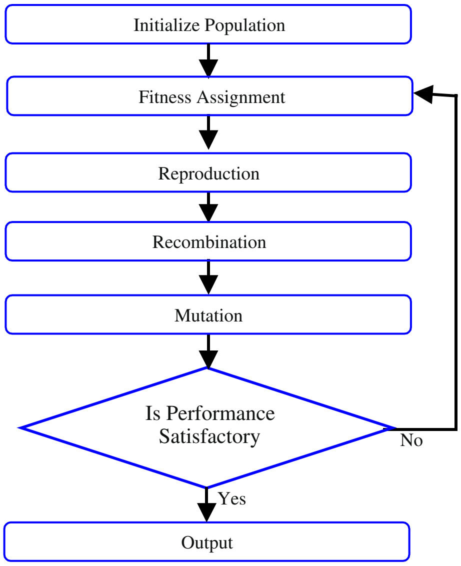

Fig. 1 Flow diagram of genetic algorithms

2.2 Functional link artificial neural networks

In general, the models that we use to solve complex classification problems are multi-layer neural network. There are many algorithms to train the neural network models. However, the models being complex in nature, one single algorithm cannot be claimed as best for training to suit different scenarios of the complexities of real-life problems. Depending on the complexities of the problems, the number of layers and number of neurons in the hidden layer need to be changed. As the number of layers and the number of neurons in the hidden layer increases, training the model becomes further complex. Very often, different algorithms fail to train the model for a given problem set.

To overcome the complexities associated with multi-layer neural network, a single-layer neural network can be considered as an alternative approach. But the single-layer neural network being linear in nature very often fails to map the complex nonlinear problems. The classification task in data mining is highly nonlinear in nature. Therefore, for solving such problems in single-layer feed forward artificial neural network is almost an impossible task.

In order to bridge the gap between the linearity in the single-layer neural network and the highly complex and computationally intensive multi-layer neural network, the FLANN architecture with back propagation learning for classifications is suggested [3]. The FLANN architecture uses a single-layer feed forward neural network to overcome the linear mapping, functionally expands the input vector. Figure 2 shows the simple architecture of our previously proposed FLANN with gradient descent.

The given set of patterns is fed to the input layer and is expanded in hidden layer, and then the weighted sum is fed to the single neuron of the output layer. The weights are optimized by the back propagation learning during the process of training.

The set of functions considered for function expansion may not be always suitable for mapping the nonlinearity of the complex task. In such cases, few more functions may be incorporated to the set of functions considered for expansion of the input dataset. However, dimensionality of many problems itself are very high and further increasing the dimensionality to a very large extent may not be an appropriate choice. So, it is advisable and also a new research direction to choose a small set of alternative functions, which can map the function to the desired extent with an output of significant improvement.

2.3 Feature selection

Feature selection is one of the very important preprocessing tasks of data mining and knowledge discovery in databases. It is obvious that the quality of discovered

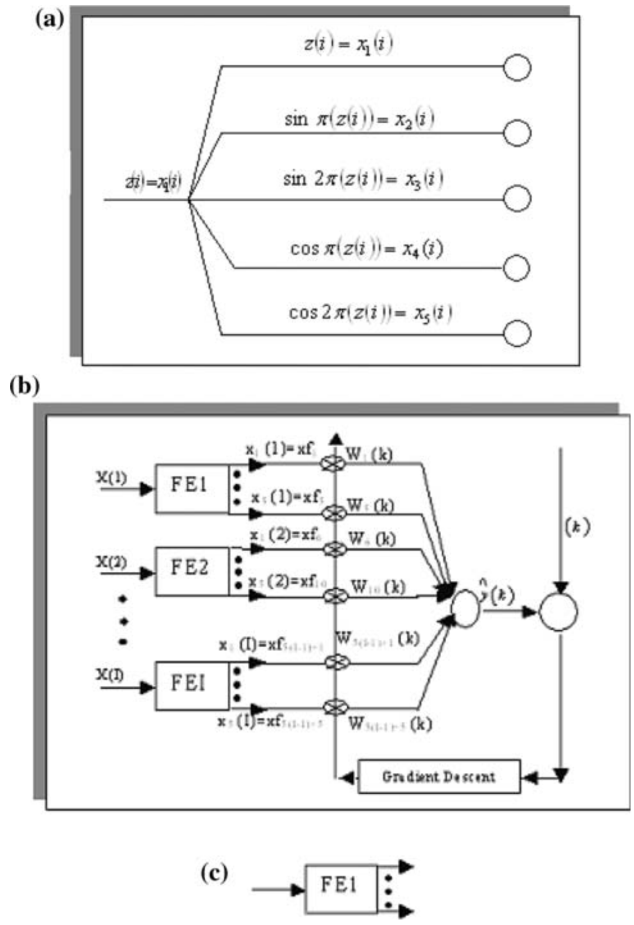

Fig. 2 **a** functional expansion of the input feature, **b** FLANN model for classification, **c** block representation of **a**

knowledge strongly depends on the quality of data being mined. No matter how intelligent a data mining algorithm is, it will fail to discover high quality knowledge if it is applied to low quality data. This has motivated the development of several feature selection algorithms. The main goal of the feature selection is to select a subset of relevant feature out of all available features of the data being mined.

In general, feature selection can be visualized as the selection of a subset of features that will reduce the probability of misrecognition in the operational (classification) phase. A feature selection scheme based on the availability of a set of labeled samples from each of the predefined set of classes is referred to as feature selection in a supervised environment. But in practice, one often comes across situations where the samples are unlabeled or at best imperfectly labeled. Again, feature selection is very important for machine learning due to its potential of speeding up and reducing the costs of the followed stage of concept learning or instance classification and improving the performance of the learned results. Therefore, how to select the optimal feature subset to describe a learning system is always regarded as a key technology in the domain of machine learning.

Furthermore, among the different categories of feature selection algorithms, the genetic algorithm (GA) [5] is a rather recent development. GA-based feature selection is very essential because of the following reasons. Suppose there are 'm' numbers of features in the data being mined. Then, the total number of candidate feature subsets is 2^m, which is the size of search space of the feature selection grows exponentially with the number of features.

The GA is biologically inspired and has many mechanisms mimicking natural evaluation. It has a great deal of potential in scientific and engineering optimization on search problems. The pioneering work by Siedlecki and Sklansky [9] demonstrated evidence for the superiority of GA compared to representative classical algorithms. Subsequently, many literatures were published that have shown advantages of GAs for feature selection. Other heuristic techniques like genetic programing [10] and PSO [11] are also used synergistically for optimizing both the features and classification accuracy, but in this work, we hybridized GA with FLANN to obtain a near optimal set of features with high classification accuracy.

2.4 Classification

The digital revolution has made digitized information easy to capture and fairly inexpensive to store. With the development of computer hardware and software and the rapid computerization of business, huge amount of data have been collected and stored in databases. The rate at which such data stored is growing at a phenomenal rate. As a result, traditional ad-hoc mixtures of statistical techniques and data management tools are no longer adequate for analyzing this vast collection of data.

Raw data is rarely of direct benefit. Its true value is predicated on the ability to extract information useful for decision support or exploration and understanding the phenomenon governing the data source. In most domains, data analysis was traditionally a manual process. One or more analysts would become intimately familiar with the data and, with the help of statistical techniques, provide summaries and generate reports. In effect, the analyst acted as a sophisticated query processor. However, such an approach rapidly breaks down as the size of data grows and the number of dimensions increases. When the scale of data manipulation, exploration and inferencing goes beyond human capacities, people look to computing technologies for automating the process.

All these have prompted the need for intelligent data analysis methodologies, which could discover useful knowledge from data. The term KDD [2] refers to the overall process of knowledge discovery in databases. Data mining is a particular step in this process, involving the application of specific algorithms for extracting patterns (models) from data. Supervised pattern classification is one of the important tasks of data mining.

Supervised pattern classification can be viewed as a problem of generating appropriate class boundaries, which can successfully distinguish the various classes in the feature space. In real-life problems, the boundaries between different classes are usually nonlinear. It is known that using a number of hyperplanes, one can approximate any nonlinear surface. Hence, the problem of classification can be viewed as searching for a number of linear surfaces that can appropriately model the class boundaries while providing minimum number of misclassified data points.

The goal of pattern classification is to assign input patterns to one of a finite number, *M,* of classes. In the following, it will be assumed that input patterns consist of static input vectors $\boldsymbol{x}$ containing X elements or continuous valued real numbers denoted x_1, x_2, …, $\boldsymbol{x}_{X.}$ Elements represent measurements of features selected to be useful for distinguishing between classes. Input patterns can be viewed as points in the multidimensional space defined by the input feature measurements. The purpose of a pattern classifier is to partition this multidimensional space into decision regions to indicate which class an input belongs to.

Application of a pattern classifier first requires selection of features that must be tailored separately for each problem domain. Features should contain information required to distinguish between classes, be insensitive to irrelevant variability in the input, and also be limited in number to permit efficient computation of discriminant functions and to limit the amount of training data required. Good classification performance requires selection of effective features and also selection of a classifier that can make good use of those features with limited training data, memory, and computing power. Following feature selection, classifier development requires collection of training and test data, and separate training and test or use phases. During the training phase, a limited amount of training data and a priori knowledge concerning the problem domain is used to adjust parameters and/or to learn the structure of the classifier. During the test phase, the classifier designed from the training phase is evaluated on new test data by providing a classification decision for each input pattern. Classifier parameters and/or structure may then be adapted to take advantage of new training data or to compensate for nonstationary inputs, variation in internal components, or internal faults. Further evaluations require new test data.

It is important to note that test data should never be used to estimate classifier parameters or to determine classifier structure. This will produce an overly optimistic estimate of the real error rate. Test data must be independent data that is only used to assess the generalization of a classifier, defined as the error rate on never-before-seen input patterns. One or more uses of test data, to select the best performing classifier or the appropriate structure of one type of classifier, invalidate the use of that data to measure generalization.

2.5 Statistical test for comparison of classifiers

One of the goals of this paper is the study of the statistical tests that could be used for comparing two or more classifiers on multiple datasets. Assume that we have tested k different algorithms on N datasets. Let $P_i^j,\ 1 \le i \le N,\ 1 \le j \le k$ be the performance score of the *j*th algorithm on *i*th dataset. The task is to decide whether, based on the values of P_i^j, the algorithms are statistically different or not (i.e., whether HFLANN statistically different from FLANN and RBF or not).

In this section, we shall examine several known and less-known statistical tests such as paired *t*-test, Wilcoxon signed ranks test [12] and study their suitability for our purpose from the point of what they actually measure and of their safety regarding the assumptions they make about the data.

2.5.1 Paired t-test

A common way to test whether the difference between two classifiers results over various datasets is nonrandom to compute a paired *t*-test, which checks whether the average difference in their performance over the datasets is significantly different from zero.

Let P_i^1 and P_i^2 be the performance scores of two classifiers on *i*th out of N datasets. The paired *t*-test is computed as follows: construct the null hypothesis and follow the following steps.

1. Calculate the mean difference between two classifiers over all the datasets i.e., $\bar{P} = \bar{P}^1 - \bar{P}^2$, where $\bar{P}^1 = \sum_{i=1}^{N} P_{i}^{1}$ and $\bar{P}^2 = \sum_{i=1}^{N} P_{i}^{1}$.
2. Calculate σ_1^2 and σ_2^2, where $\sigma_1^2 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(P_i^1 - \bar{P}^1\right)^2}$ and $\sigma_2^2 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(P_i^2 - \bar{P}^2\right)^2}$
3. Calculate the variance of the difference between the two means as follows $\sigma_p = \sqrt{\frac{1}{N}\left(\sigma_1^2 + \sigma_2^2\right)}$.
4. Calculate the required *t*-value, $t\_value = \frac{\bar{P}}{\sigma_P}$.

Enter the *t*-table (2 $N - 1$) degrees of freedom; choose the level of significance required (normally $p = 0.05$), and read the *t*-value. Then, the decision is whether the null hypothesis is accepted or rejected based on the tests statistics support to the null hypothesis.

2.5.2 Wilcoxon signed ranks test

The Wilcoxon signed ranks test [12] is a nonparametric alternative to the paired *t*-test, which ranks the differences in performances of two classifiers for each dataset, ignoring the signs and compares the ranks for the positive and the negative differences.

Let P_i be the difference between the performance scores of the two classifiers on ith out of N datasets. The differences are ranked according to their absolute values; average ranks are assigned in case of ties. Let R_{pos} be the sum of ranks for the datasets on which the second algorithm outperformed the first, and R_{neg} be the sum of ranks for the first algorithm outperformed the second. Ranks of $P_i = 0$ are split evenly among the sums; if there are an odd number of them, one is ignored:

$$R_{\text{pos}} = \sum_{P_i > 0} \text{rank}(Pi) + \frac{1}{2}\sum_{P_i = 0} \text{rank}(Pi) \tag{1}$$

$$R_{\text{neg}} = \sum_{P_i < 0} \text{rank}(P_i) + \frac{1}{2}\sum_{P_i = 0} \text{rank}(P_i) \tag{2}$$

Let R_s be the smaller of the sums, $R_s = \min\{R_{\text{pos}}, R_{\text{neg}}\}$. For a large number of datasets, the statistics $z = \left(R_s - \frac{1}{4}(N(N+1))\middle/\sqrt{\frac{1}{24}(N(N+1)(2N+1))}\right)$ is distributed approximately normally. With $\alpha = 0.05$, the null hypothesis can be rejected if z is smaller than -1.96.

The Wilcoxon signed ranks test is more sensible than the t-test. It assumes commensurability of differences, but only quantitatively; greater differences still count more, which is probably desired, but the absolute magnitudes are ignored. From the statistical point of view, the test is safer since it does not assume Gaussian distributions. Also, the outliers have less effect on the Wilcoxon than on the t-test.

The Wilcoxon test assumes continuous differences P_i; therefore, they should not be rounded to, say, one or two decimals since this would decrease the power of the test due to a high number of ties.

When the assumptions of paired t-test are met, the Wilcoxon signed ranks test is less powerful than the paired t-test. On the other hand, when the assumptions are violated, the Wilcoxon test can be even more powerful than the t-test.

3 Proposed method

The proposed HFLANN is a single hidden-layer artificial neural network (ANN) with a genetically optimized set of features. It has the capability of generating complex decision regions by nonlinear enhancement of hidden nodes referred to as functional links. Figure 3 shows the topological structure of the HFLANN. The proposed method is characterized by a set of FLANN with a different subset of features. The initial input of the network is same as the number of input variables of the data domain.

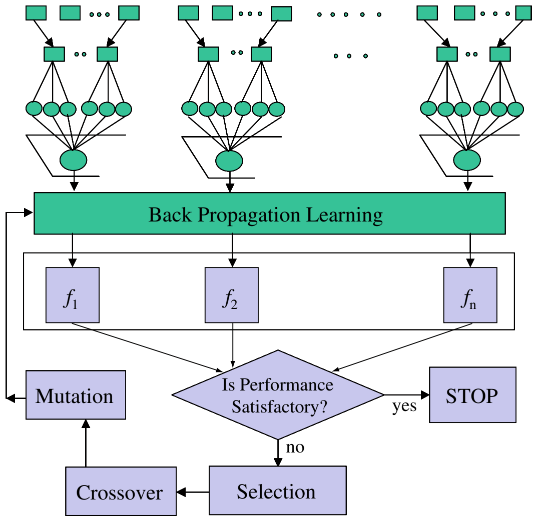

Fig. 3 Topological structure of the HFLANN

Let n be the number of original features of the data domain. The number of features selected to become a chromosome of the genetic population is m $d \le n$. The m varies from chromosomes to chromosomes of the genetic population (i.e., $1 \le d \le n$). For simplicity, let us see how a single chromosome with d features is working cooperatively for HFLANN.

In this work, we have used the general trigonometric function for mapping the d feature from one form to another form of higher dimension. However, one can use a function that is very close to the underlying distribution of the data, but it requires some prior domain knowledge. In this work, we are taking five functions out of which four are trigonometric and one is linear (i.e., keeping the original form of the feature value). Out of the four trigonometric functions, two are sine and two are cosine functions. In the case of trigonometric functions, the domain is feature values and range is a real number lies between [−1,1]. It can be written as

$$f : D \rightarrow R^{[-1,1] \cup \{x\}} \tag{3}$$

where $D = \{x_{i1}, x_{i2}, \ldots, x_{id}\}$, and d is known as the number of features.

In general, let us take $f_1, f_2, \ldots, f_k$ be the number of functions used to expand each feature value of the pattern. Therefore, each input pattern can now be expressed as

$$\begin{aligned} \vec{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{id}\} &\rightarrow \{\{f_1(x_{i1}), f_2(x_{i1}), \ldots, f_k(x_{i1})\}, \ldots, \\ &\times \{f_1(x_{id}), f_2(x_{id}), \ldots, f_k(x_{id})\}\} \\ &= \{\{y_{11}, y_{21}, \ldots, y_{k1}\}, \ldots, \{y_{1d}, y_{2d}, \ldots, y_{kd}\}\}, \end{aligned} \tag{4}$$

The weight vector between hidden layer and output layer is multiplied with the resultant sets of nonlinear outputs and are fed to the output neuron as an input. Hence, the weighted sum is computed as follows:

$$s = \sum_{j=1}^{m} y_{ij}.w_j, \quad i = 1, 2, \ldots, X$$
$$\text{and m be the total number of expanded features} \tag{5}$$

The network has the ability to learn through back propagation learning. The training requires a set of training data, i.e., a series of input and associated output vectors. During the training, the network is repeatedly presented with the training data and the weights adjusted by back propagation learning from time to time till the desired input–output mapping occurs.

Hence, the estimated output is computed by the following metric:

$$\hat{y}_i(t) = f(s_i), \quad i = 1, 2, \ldots, X.$$

The error $e_i(t) = y_i(t) - \hat{y}_i(t), \quad i = 1, 2, \ldots, X$ be the error obtained from the ith pattern of the training set. Therefore, the error criterion function can be written as,

$$E(t) = \sum_{i=1}^{X} e_i(t) \tag{6}$$

and our objective is to minimize this function by gradient decent approach until $E \leq \varepsilon$.

This process is repeated for each chromosomes of the GA, and then based on the performance, each chromosome will be assigned a fitness value. Using that fitness value, the usual process of GA is executed until some good topology with high predictive accuracy is achieved.

3.1 High level algorithms for HFLANN

The specification of the near optimal HFLANN architecture and related parameters can be obtained by both genetic algorithms and back propagation learning, as it is explained in the following. Evolutionary algorithms of genetic type are stochastic search and optimization methods. Principally, based on computational models of fundamental process, such as reproduction, recombination, and mutation. An algorithm of this type begins with a set (population) of estimates (genes) called individuals (chromosomes) appropriately encoded. Each one is evaluated for its fitness in solving the classification task of data mining. During each iteration (algorithm time-step), the most-fit individuals are allowed to make and bear offspring.

3.1.1 Individual representation

For the evolutionary process, the length of each particle is n (i.e., the upper bound of a feature vector). Figure 4 shows the structure of a chromosome that is used for design of HFLANN. Each cell of the chromosome contains binary value either 0 or 1. The cell value controls the activation (the value of 1 is assigned) or deactivation (the value of 0 is assigned) of the functional expansion for individuals.

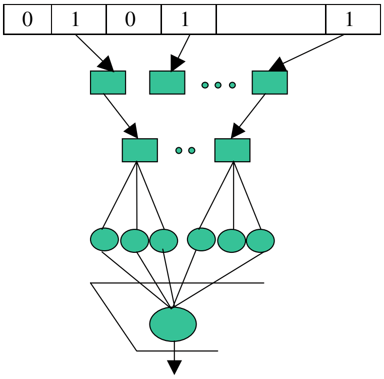

Fig. 4 Individual representation with its associated FLANN topology

3.1.2 Objective function

During evolution, each individual measures its effectiveness by the error criterion function using Eq. 6, and the predictive accuracy is assigned as it corresponding fitness.

The major steps of HFLANN can be described as follows:

```
1.  DIVISION OF DATASET
    Divide the dataset into two parts: training and testing
2.  RANDOM INITIALIZATION
    Initialize each individual randomly from the domain {0,1}.
3.  REPEAT
4.  FOR THE POPULATION
FOR each sample of the training set
MAPPING OF INPUT PATTERN
        Map each pattern from low to high dimension, i.e. expand each feature
        value according to the predefined set of functions.
CALCULATE the weighted sum and feed as an input to the node of the output
layer.
CALCULATE the error and accumulate it.
BACK PROPAGATION LEARNING
        Minimize the error by back propagation learning.
ASSIGN THE FITNESS
5.  FOR THE POPULATION
    5.1 Perform Roulette Wheel Selection to obtain the better chromosomes.
6.  FOR THE POPULATION
Perform recombination
Mutation
7.  UNTIL < Maximum Iteration is Reached>
```

If we look very closely, this algorithm is not only selecting the optimal set of features, but also evolving a set of FLANN architecture. Therefore, we can say this is a type of evolving FLANN. However, in this work, we are not taking into account of optimizing the architecture from all aspects (i.e., topological structure as well as weights). Hence, instead of a multi-objective function optimization, we are only optimizing the uni-objective, i.e., known as predictive accuracy of the HFLANN.

4 Experimental studies

The performance of the EFLANN model was evaluated using a set of five public domain datasets like IRIS, WINE, PIMA, BUPA Liver Disorders, ECOLI, GLASS, HOUSING, LED7, LYMPHA, and ZOO from the University of California at Irvine (UCI) machine learning repository [13]. In addition, we have taken VOWEL dataset for show the performance of HFLANN to classify six overlapping vowel classes [14]. We have compared the results of HFLANN with other competing classification methods such as radial basis function network (RBF) and our previously proposed FLANN with gradient descent.

This section is divided into three subsections. Section 4.1 discusses the nature and characteristics of the dataset being classified. Section 4.2 discusses the parameter set up required for the experiment. The comparative performance of the model is demonstrated in Sect. 4.3 with a discussion. The classification performance of HFLANN with chromosome knowledge incorporation is presented in Sect. 4.4. Finally, the statistical tests are analyzed theoretically in Sect. 4.5.

4.1 Description of the datasets

Let us briefly discuss the datasets, which we have taken for our experimental setup.

IRIS Dataset: This is the most popular and simple classification dataset based on multivariate characteristics of a plant species (length and thickness of its petal and sepal) divided into three distinct classes (Iris Setosa, Iris Versicolor and Iris Virginica) of 50 instances each. One class is linearly separable from other two; the later are not linearly separable from each other. In a nutshell, it has 150 instances and 5 attributes. Out of 5 attributes, four attributes are predicting attributes and one is goal attribute. All the predicting attributes are real values.

WINE Dataset: These dataset are resulted from a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. In classification context, this is a well-posed problem with well-behaved class structures. The total number of instances is 178, and it is distributed as 59 for class 1, 71 for class 2 and 48 for class 3. The number of attributes is 14 including class attribute, and all 13 are continuous in nature. There are no missing attribute values in this dataset.

PIMA Indians Diabetes Database: This database is a collection of all female patients of at least 21 years of age of PIMA Indian heritage. It contains 768 instances, 2 classes of positive and negative and 9 attributes including the class attribute. The attribute contains either integer or real values. There are no missing attribute values in the dataset.

BUPA Liver Disorders: This dataset related to the diagnosis of liver disorders and created by BUPA Medical Research, Ltd. It consists of 345 records, 7 attributes including the class attribute. The class attribute is repeated with only two class values for entire database. The first 5 attributes are all blood tests, which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. Each record corresponds to a single male individual.

ECOLI: This dataset describes about the protein localization sites. It contains 336 instances, 7 predictive attributes with no missing values and one class attribute. The samples are distributed into 8 classes, and the class distribution is highly unbalanced.

GLASS: The glass identification dataset contains 214 instances and 11 attributes (including an Id#) plus the class attribute (whose domain contains 6 values). All the attribute values are continuous, and no one contain missing values.

VOWEL: This dataset consists of 871 patterns with 6 overlapping vowel classes and three input features. All entries are integers.

HOUSING: The Boston housing data concerns housing values in suburbs of Boston. There are 506 samples and 13 continuous attributes (including class attributes) and 1 binary valued attribute with no missing values.

LED7: The LED display dataset contain 7 attributes and user chooses the number of instances. No attribute contains missing values.

LYMPHOGRAPHY: This dataset contains 148 instances and 19 attributes (including the class attribute) with no missing values. The classes are distributed into 4 classes.

ZOO: This dataset contains 101 instances of zoo information. The number of attributes is 18 (animal name, 15 Boolean attribute, and 2 numeric attributes). There is no missing value in the domain of the attributes, and it contains 7 types of zoo.

Table 1 presents a summary of the main characteristics of the databases that have been used in this study. The

Table 1 Summary of the dataset used in simulation studies

Sl. No.	Dataset	Instances	Attribute	Classes
1	IRIS	150	4	3
2	WINE	178	13	3
3	PIMA	768	8	2
4	BUPA	345	6	2
5	ECOLI	336	7	8
6	GLASS	214	9	6
7	VOWEL	871	3	6
8	HOUSING	506	13	5
9	LED7	300	7	10
10	LYMPHOGRAPHY	148	18	4
11	ZOO	101	16	7

second column of this table gives the dataset name, while other columns indicate, respectively, the number of instances, the number of attributes, and number of classes.

4.2 Parameter setup

For evaluating the proposed algorithm, the following user defined parameters and protocols related to the dataset need to be set beforehand.

A twofold cross validation is carried out for all the dataset by randomly dividing the dataset into two parts (datasets1. dat and dataset2.dat). Each of these two sets was alternatively used either as a training set or test set.

The quality of each individual is measured by the predictive performance obtained during training. It is also very important to set the optimal values of the following parameters to reduce the local optimality. The parameters are described as follows:

Population size: The size of the population denoted as |P| = 50 is fixed for all the datasets. We have chosen 50 to avoid under and over fit during the training. The larger the number of individuals, the more number of computation time is required, and the performance of the system will slow down.

Stop Criteria: The iteration is fixed to 1,000 for all the datasets.

Length of the individuals is fixed to n, where n is the number of input features. The probability for crossover is 0.7 and mutation is 0.02.

4.3 Comparative performance

The predictive performance obtained from HFLANN for the earlier mentioned datasets was compared with the results obtained from FLANN with back propagation learning and radial basis function network (RBF). Table 2 summarizes the average training and test performances of HFLANN and compared with FLANN and RBF.

From Table 2, we can easily verified that except BUPA case in all other dataset on an average, the proposed method is giving promising results in both training and test cases. In the case of BUPA, FLANN is performing better. Table 3 illustrates a fair comparative performance of the proposed algorithm by using maximum predicative value obtained in training and test set.

Table 4 shows the percentage of relevant feature selected for each of the datasets during training of HFLANN.

Figure 5 shows a graphical view of the percentage of feature selected. The X-axis represents the datasets, and Y-axis represents the percentage of active bits in the optimal chromosome obtained during the training.

Table 2 Average comparative performance of HFLANN, FLANN, and RBF

Dataset	Algorithms	Training performance	Testing performance
IRIS	HFLANN	98.0001	97.3335
	FLANN	96.6665	96.6665
	RBF	38.5000	38.5000
WINE	HFLANN	99.4380	90.4495
	FLANN	97.1910	88.7640
	RBF	85.3935	79.2130
PIMA	HFLANN	80.7290	72.1355
	FLANN	79.5570	72.1355
	RBF	77.4740	76.0415
BUPA	HFLANN	77.6820	69.2785
	FLANN	77.9725	69.2800
	RBF	71.0125	66.9530
ECOLI	HFLANN	55.1670	50.8020
	FLANN	49.9625	47.3075
	RBF	31.1780	26.1100
GLASS	HFLANN	63.5565	51.5075
	FLANN	60.7510	50.3800
	RBF	48.9865	34.6440
VOWEL	HFLANN	40.4395	38.1965
	FLANN	27.9250	24.7220
	RBF	25.2555	24.3250
HOUSING	HFLANN	82.2130	72.5295
	FLANN	76.4825	69.7630
	RBF	67.1940	65.4150
LED7	HFLANN	30.8110	27.5280
	FLANN	22.4185	19.7000
	RBF	20.2820	16.5720
LYMPHOGRAPHY	HFLANN	97.2970	77.0270
	FLANN	91.8920	74.3245
	RBF	85.1350	72.2927
ZOO	HFLANN	99.0385	86.1850
	FLANN	97.1155	85.1645
	RBF	96.1540	81.0830

4.4 Knowledge incorporation in measure of the predictive accuracy

Let n be the total number of features in the dataset; T_1 denote the number of feature selected using the training set 1 and testing set 2; T_2 denote the number of feature selected using training set 2 and testing set 1.

Notations and their Meaning:

$|n|$ represent the total number of features in the dataset.
$|T_1|$ denote the total number of selected features in test set 2.
$|T_2|$ denote the total number of selected features in test set 1.

Table 3 Comparative performance w.r.t maximum training performance

Dataset	Training/test performance	HFLANN	FLANN	RBF
IRIS	TR/Te	98.667/97.333	98.667/97.333	57.333/48.000
WINE	TR/Te	100/91.011	97.753/93.258	86.517/82.022
PIMA	TR/Te	81.51/72.656	80.208/72.656	78.125/77.604
BUPA	TR/Te	77.907/70.349	78.488/70.93	71.676/68.208
ECOLI	TR/Te	59.829/54.701	52.137/52.137	38.462/27.434
GLASS	TR/Te	63.81/57.143	60.952/55.046	53.211/38.095
VOWEL	TR/Te	40.708/41.88	33.628/28.205	27.434/25.641
HOUSING	TR/Te	85.375/77.075	79.842/71.542	70.356/66.088
LED7	TR/Te	34.188/35.398	33.333/27.434	29.06/19.469
LYMPH.	TR/Te	97.297/78.378	94.595/77.027	86.486/75.676
ZOO	TR/Te	100/87.755	100/85.714	100/84.615

Tr Training, *Te* Test

Table 4 Percentage of feature selected

Dataset	Percentage of feature selected
IRIS	75
WINE	46.1538
PIMA	75
BUPA	75
ECOLI	57.1429
GLASS	55.5556
VOWEL	33.333
HOUSING	76.9231
LED7	57.1429
LYMPHOGRAPHY	38.8889
ZOO	31.25

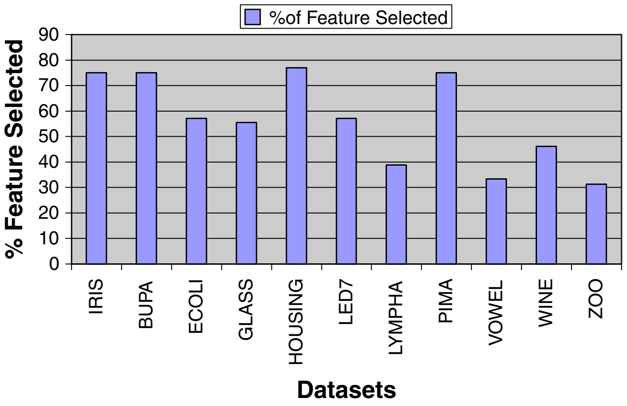

Fig. 5 Percentage of optimal set of selected features

The fitness of the chromosome with respect to T_1 is

$$f(T_1) = \frac{|PA| \times |n| - \tau \times |T_1|}{|n|} \tag{7}$$

Similarly the fitness of the chromosome with respect to T_2 is

$$f(T_2) = \frac{|PA| \times |n| - \tau \times |T_2|}{|n|} \tag{8}$$

In general, we write

$$f(T) = |PA| - \frac{\tau \times |T|}{|n|} \tag{9}$$

where $|PA|$ represent the predictive accuracy, and τ represent the tradeoff between two criteria, and its value is 0.01.

Table 5 shows the predictive accuracy using Eqs. 7 and 8 of the HFLANN by incorporating a kind of knowledge of each chromosome optimally selected with respect to test set 1 and test set 2.

Table 6 shows the performance of hit percentage in training set 1 and training set 2 and its corresponding individual with active number of bits. From this table, one can take the conclusion that whether the explicit knowledge incorporation will be important in classifier or not.

Figure 6 shows the predictive performance of HFLANN by incorporating individual knowledge with $\tau = 0.01$ for training set 1 and training set 2 with respect to their corresponding active bits of the individual.

4.5 *T*-Test and Wilcoxon signed ranks test paired *t*-test

We have tested the proposed method (HFLANN) with FLANN and RBF using the *t*-test individually for training and testing performance scores. In order to test the significance of our algorithm over to FLANN and RBF, let us first construct the null hypothesis. The null hypothesis is that there is no difference between the average performance of HFLANN versus FLANN and HFLANN versus RBF.

4.5.1 HFLANN versus FLANN

Null hypothesis: means are equal.

t_value = 0.3833 with degree of freedom is 20.

The chosen level of significance is 0.05, and the tabulated value is 2.09. As the calculated t_value is less than the tabulated value, we reject the null hypothesis i.e., the proposed algorithm is significantly better than FLANN.

Table 5 Predictive accuracy of HFLANN by knowledge incorporation with $\tau = 0.01$

Dataset	*N*	P.A. test set 1 chromosome	P.A. test set 2 chromosome
IRIS	4	95.9925	97.3260
WINE	13	89.8826	91.0064
PIMA	8	71.6125	72.6548
BUPA	6	70.3343	68.2013
ECOLI	7	54.6939	46.8973
GLASS	9	57.1374	45.8642
VOWEL	3	34.5063	41.8733
HOUSING	13	67.9771	77.0673
LED7	7	19.6551	35.3923
LYMPH	18	78.3724	75.6721
ZOO	16	87.7494	84.6119

Table 6 Performance in training set 1 and training set 2

Dataset	Training set1/ training set 2	Hit percentage in training set 1	Hit percentage in training set 2
IRIS	3/3	95.9925	97.3260
WINE	6/7	89.8826	91.0064
PIMA	7/6	71.6125	72.6548
BUPA	5/5	70.3343	68.2013
ECOLI	4/5	54.6939	46.8973
GLASS	7/5	57.1374	45.8642
VOWEL	1/1	34.5063	41.8733
HOUSING	10/9	67.9771	77.0673
LED7	4/2	19.6551	35.3923
YMPH	7/10	78.3724	75.6721
ZOO	5/9	87.7494	84.6119

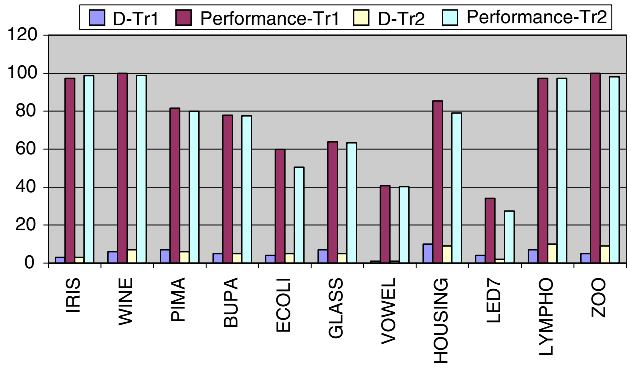

Fig. 6 HFLANN performance on training set 1 and training set 2 with respect to the individual active bits

4.5.2 *HFLANN versus RBF*

Null hypothesis: means are equal.

*t*_value=1.4751 with degree of freedom is 20.

The chosen level of significance is 0.05, and the tabulated value is 2.09. As the calculated *t*_value is less than the tabulated value, we reject the null hypothesis i.e., the proposed algorithm is significantly better than RBF.

4.5.3 *Wilcoxon signed ranks test*

Like paired *t*-test, we will test the proposed method HFLANN with FLANN and RBF separately because it can compare two algorithms at a time over multiple datasets. Here, we are trying to reject the null hypothesis that both algorithms perform equally well. The ranks are assigned from the lowest to the highest absolute difference, and the equal differences are assigned average ranks. Tables 7 and 8 show the classification performance of HFLANN versus FLANN and HFLANN versus RBF and their corresponding ranks considering the training set.

The sum of the ranks for positive difference is $R_{\mathrm{pos}} = 65$ and the sum of the ranks for the negative difference is $R_{\mathrm{neg}} = 1$. According to the table of exact critical values for the Wilcoxon's test, for a confidence level of $\alpha = 0.05$ and

Table 7 Performance score and ranks of HFLANN versus FLANN

Dataset	HFLANN	FLANN	Difference	Rank
IRIS	98.0001	96.6665	1.3336	3
WINE	99.4380	97.1910	2.2470	5
PIMA	80.7290	79.5570	1.1700	2
BUPA	77.6820	77.9728	−0.2905	1
ECOLI	55.1670	49.9625	5.2045	7
GLASS	63.5565	60.7510	2.8055	6
VOWEL	40.4395	27.9250	12.5145	11
HOUSING	82.2130	76.4825	5.7305	9
LED7	30.8110	22.4185	8.3925	10
LYMPHO.	97.2970	91.8920	5.4050	8
ZOO	99.0385	97.1155	1.9230	4

Table 8 Performance score and ranks of HFLANN versus RBF

Dataset	HFLANN	RBF	Difference	Rank
IRIS	98.0001	38.5000	59.5001	11
WINE	99.4380	85.3935	14.0445	6
PIMA	80.7290	77.4740	3.2550	2
BUPA	77.6820	71.0125	6.6695	3
ECOLI	55.1670	31.1780	23.9890	10
GLASS	63.5565	48.9865	14.5700	7
VOWEL	40.4395	25.2555	15.1840	9
HOUSING	82.2130	67.1940	15.0190	8
LED7	30.8110	20.2820	10.5290	4
LYMPHO.	97.2970	85.1350	12.1620	5
ZOO	99.0385	96.1540	2.8845	1

$N = 11$ datasets, the difference between the classifiers is significant if the smaller of the sums is equal or less than 11. We, therefore, reject the null hypothesis.

The sum of the ranks for positive difference is $R_{\text{pos}} = 66$ and the sum of the ranks for the negative difference is $R_{\text{neg}} = 0$. According to the table of exact critical values for the Wilcoxon's test, for a confidence level of $\alpha = 0.05$ and $N = 11$ datasets, the difference between the classifiers is significant if the smaller of the sums is equal or less than 11. We, therefore, reject the null hypothesis.

Hence, we can conclude that in both the cases, the proposed algorithm is significantly different from 0.

5 Conclusions

In this paper, we have evaluated the proposed method HFLANN for the task of classification in data mining by giving an equal importance to the selection of optimal set of features and classification accuracy. The HFLANN model functionally maps the selected set of feature value from lower to higher dimension. The experimental studies demonstrated that the classification performance of HFLANN model is promising. In almost all cases, the results obtained with the HFLANN proved to be better than the best results found by its competitor like RBF and FLANN with back propagation learning. Further, we theoretically and empirically analyzed parametric (*t*-test) and nonparametric (Wilcoxon signed rank test) tests that can be used for comparing classifiers over multiple datasets. The architectural complexity is low, whereas training time is little bit costly as compared to FLANN. As we know, one of the most important criteria of data mining is, how comprehensible the model is? If the architectural complexity increases, then the comprehensibility decreases. Therefore, from this aspect, we can claim that the proposed model can fit in data mining task of classification.

Acknowledgments Dr. S. Dehuri would like to thank Department of Science and Technology, Govt. of India, vide letter number SR/BY/E-07/2007 dated 03-01-2008 for financial support under the BOYSCAST Program 2007–2008 and the financial support of BK21 Project, Korea's Ministry of Education and Human Research Development. Prof. S. B. Cho would like to thank the financial support of BK21 Project initiated by Korea's Ministry of Education and Human Resource Development.

References

1. Kriegel H-P et al (2007) Future trends in data mining. Data Min Knowl Disc 15(1):87–97
2. Fayyad UM, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery: an overview. In: Fayyad UM, Piatetsky-Shapiro G, Smyth P (eds) Advances in knowledge discovery and data mining. AAAI Press, Menlo Park, CA, pp 1–34
3. Misra BB, Dehuri S (2007) Functional link artificial neural network for classification task in data mining. J Comput Sci 3(12):948–955
4. Oh Il-S et al (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 20(11):1424–1437
5. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Morgan Kaufmann, Los Altos
6. Duda RO, Hart PE, Stork DG (2001) Pattern classification. Wiley, New York
7. Zhang GP (2000) Neural networks for classification: a survey. IEEE Trans Syst Man Cybern Part C Appl Rev 30(4):451–461
8. Pao Y-H et al (1992) Neural-net computing and intelligent control systems. Int J Control 56(2):263–289. doi:10.1080/00207179208934315
9. Siedlecki W, Skalansky J (1988) On automatic feature selection. Int J Pattern Recognit Artif Intell 2(2):197–220. doi:10.1142/S0218001488000145
10. Koza JR (1994) Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge
11. Chang C-G et al (2007) Application of particle swarm optimization based BP neural network on engineering project risk evaluating. In: Proceedings of 3rd international conference on natural computation (ICNC 2007), pp 750–754
12. Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics 1:80–83. doi:10.2307/3001968
13. Blake CL, Merz CJ UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html
14. Pal SK, Majumdar DD (1977) Fuzzy sets and decision making approaches in vowel and speaker recognition. IEEE Trans Syst Man Cybern 7:625–629