Fuzzy Decision Tree, Linguistic Rules and Fuzzy Knowledge-Based Network: Generation and Evaluation

Sushmita Mitra, *Senior Member, IEEE*, Kishori M. Konwar, and Sankar K. Pal, *Fellow, IEEE*

***Abstract*—A fuzzy knowledge-based network is developed based on the linguistic rules extracted from a fuzzy decision tree. A scheme for automatic linguistic discretization of continuous attributes, based on quantiles, is formulated. A novel concept for measuring the goodness of a decision tree, in terms of its compactness (size) and efficient performance, is introduced. Linguistic rules are quantitatively evaluated using new indices. The rules are mapped to a fuzzy knowledge-based network, incorporating the frequency of samples and depth of the attributes in the decision tree. New fuzziness measures, in terms of class memberships, are used at the node level of the tree to take care of overlapping classes. The effectiveness of the system, in terms of recognition scores, structure of decision tree, performance of rules, and network size, is extensively demonstrated on three sets of real-life data.**

***Index Terms*—Classification, decision tree, fuzzy ID3, knowledge-based network, rule evaluation, rule generation, soft computing.**

I. Introduction

THE concept of decision trees was popularized by Quinlan with ID3 [1], which stands for *Interactive Dichotomizer 3*. Systems based on this approach use an information theoretic measure of entropy for assessing the discriminatory power of each attribute. The most important feature of decision trees is their capability to break down a complex decision-making process into a collection of simpler decisions and thereby, providing an easily interpretable solution [2]. ID3 is a popular and efficient method of decision—making for classification of *symbolic* data and is generally not suitable in cases where numerical values are to be operated upon. Since most real life problems deal with nonsymbolic (numeric, continuous) data, they must be discretized prior to attribute selection. Classification and Regression Trees (CART) [3] and C4.5 [4], however, do not require prior partitioning. Here the thresholds are dynamically computed depending on the conditions along a path, and often result in the multiple use of a particular attribute with different thresholds. This can lead to an increased accuracy at the cost of reduced comprehensibility. Another problem with ID3 is that it cannot provide any information about the intersection region where the pattern classes are overlapping.

The fusion of fuzzy sets with decision trees enables one to combine the uncertainty handling and approximate reasoning capabilities of the former with the comprehensibility and ease of application of the latter. This enhances the representative power of decision trees *naturally* with the knowledge component inherent in fuzzy logic, leading to better robustness, noise immunity, and applicability in uncertain/imprecise contexts. Fuzzy decision trees [5] assume that all domain attributes or linguistic variables have pre-defined fuzzy terms, determined in a data-driven manner using fuzzy restrictions. The information gain measure, used for splitting a node, is modified for fuzzy representation and a pattern can have nonzero match to one or more leaves. Techniques for the design of fuzzy decision trees have been reported in literature [5]–[12].

Ichihashi *et al.* [6] extract fuzzy reasoning rules viewed as fuzzy partitions. An algebraic method to facilitate incremental learning is also employed. Xizhao and Hong [7] discretize continuous attributes using fuzzy numbers and possibility theory. Pedrycz and Sosnowski [8], on the other hand, employ context-based fuzzy clustering for this purpose. Yuan and Shaw [9] induce a fuzzy decision tree by reducing classification ambiguity with fuzzy evidence. The input data is fuzzified using triangular membership functions around cluster centers obtained using Kohonen's feature map [13]. Wang *et al.* [10] present optimization principles of fuzzy decision trees based on minimizing the total number and average depth of leaves, proving that the algorithmic complexity of constructing a minimum tree is NP-hard. Fuzzy entropy and classification ambiguity are minimized at node level, and fuzzy clustering is used to merge branches.

Decision trees and neural networks are the most commonly used nonparametric tools for pattern classification. While in decision trees the number of tuples becomes smaller as the path between the root node and a new node increases, the decision boundaries of the neural net are formed by considering all the available input tuples as a whole. Hence a neural net can be expected to generate fewer rules, but with larger number of antecedent conditions [14]. In recent years enormous work has been done in an attempt to combine the advantages of neural networks and decision trees [15]–[17].

Determination of the optimal size of an artificial neural network (ANN) is a problem of considerable importance, as this has a significant impact on the effectiveness of its performance. In

Manuscript received September 20, 2000; revised July 13, 2001 and August 15, 2002.

S. Mitra and S. K. Pal are with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India (e-mail: sushmita@isical.ac.in, sankar@isical.ac.in).

K. M. Konwar was with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India. He is now with the University of Connecticut, Storrs, CT 06268 USA.

Digital Object Identifier 10.1109/TSMCC.2002.806060

general, it is desirable to have small networks. This is because increasing the number of hidden nodes/links may improve the approximation quality of an ANN at the expense of deteriorating its generalization capability (due to the resulting redundancy). One way of improving the generalization behavior of an ANN is to use knowledge-based networks [18], [19], which consider crude domain knowledge to generate the initial network architecture that is later refined in the presence of training data. Fuzzy knowledge-based networks [20], [21] typically incorporate fuzziness at the network level, using fuzzy neural networks. This manner of automatically generating the optimal network architecture helps in reducing the search space and time while the network traces the solution. Decision trees can be used for this purpose.

The present article describes the formulation of a fuzzy knowledge-based network using the principle of a fuzzy decision tree. Quantitative measures are defined to evaluate the effectiveness of the fuzzy decision tree and the linguistic rules. The novel concept of tree evaluation, in terms of its compactness and performance, enables extraction of only meaningful (less ambiguous) rules. A smaller/compact tree is more efficient both in terms of storage and time requirements, tends to generalize better to unknown test cases, and leads to the generation of more comprehensible linguistic rules. This results in the generation of a compact (less redundant) fuzzy knowledge-based network. Quantitative evaluation of the linguistic rules not only minimizes human intervention, but also provides aids for knowledge discovery. A measure "Coverage" is also introduced in this regard.

Discretization of continuous attributes, based on the distribution of pattern points in the feature space, is made in linguistic terms using quantiles. Unlike other fuzzy decision trees [5], this discretization to boolean form helps in reducing the computational complexity while preserving the linguistic nature of the decision in rule form. New fuzziness measures, in terms of class memberships, are used at the node level of the tree to take care of overlapping classes. Pruning is used to minimize noise, resulting in a smaller decision tree with more efficient classification. The extracted rules are mapped onto a fuzzy knowledge-based network. Unlike [15]–[17], the frequency of samples (representative of a rule) and the depth of the attributes in the decision tree are incorporated during the mapping.

The effectiveness of the system is exhaustively demonstrated on three sets of real-life data, *viz., Vowel, Wisconsin Breast Cancer* and *Balance scale*.

II. Fuzzy ID3

First, we present the classical ID3 algorithm. This is followed by incorporation of fuzziness at the input, output, and node levels, to handle different forms of uncertainty. Finally, a new metric, called $\mathcal{T}$-measure, is developed to evaluate the decision tree both in terms of performance and size.

A. ID3 Algorithm

ID3 uses an information-theoretic approach. The procedure is that at any point, one examines the feature that provides the greatest gain in information or, equivalently, the greatest decrease in entropy. Entropy is defined as $p \log_2 p$, where probability p is determined on the basis of frequency of occurrence.

The general case is that of N labeled patterns partitioned into sets of patterns belonging to classes C_i, $i = 1, 2, 3, \ldots, l$. The population in class C_i is n_i. Each pattern has n features and each feature can take on two or more values. The ID3 prescription for synthesizing an efficient decision tree can be stated as follows [22]:

Step 1) *Calculate initial value of entropy*

$$\text{Entropy} = \sum_{i=1}^{l} -\left(\frac{n_i}{N}\right) \log_2 \left(\frac{n_i}{N}\right) = \sum_{i=1}^{l} -p_i \log_2 p_i. \quad (1)$$

Step 2) *Select that feature which results in the maximum decrease in entropy (gain in information), to serve as the root node of the decision tree.*

Step 3) *Build the next level of the decision tree providing the greatest decrease in entropy.*

Step 4) *Repeat Steps 1 through 3*. Continue the procedure until all subpopulations are of a single class and the system entropy is zero.

At this stage, one obtains a set of leaf nodes (subpopulation) of the decision tree, where the patterns are of a single class. Note that there can be some nodes which cannot be resolved any further.

B. Incorporation of Fuzziness

Input attributes are automatically discretized in linguistic terms, based on the distribution of pattern points in the feature space. Different forms of fuzzy entropy are computed at the node level, in terms of class membership, to take care of overlapping classes. Pruning is used to minimize noise, resulting in a smaller decision tree with more efficient classification.

1) Input Representation: Any input feature value is described in terms of some combination of overlapping membership values in the linguistic property sets *low* (L), *medium* (M) and *high* (H). An n-dimensional pattern $\mathbf{F}_i = [a_1, a_2, \ldots, a_n]$ is represented as a $3n$-dimensional vector [23]

$$\mathbf{F}_i = \Big[\mu_{\text{low}}(a_1)(\mathbf{F}_i), \mu_{\text{medium}}(a_1)(\mathbf{F}_i), \mu_{\text{high}}(a_1)(\mathbf{F}_i), \ldots, \mu_{\text{high}}(a_n)(\mathbf{F}_i)\Big] \quad (2)$$

where the μ values indicate the membership functions of the corresponding linguistic functions *low*, *medium* and *high* along each feature axis. Each μ value is then discretized, using a threshold (generally 0.5), to enable a convenient mapping in the ID3 framework. This discretization to boolean form speeds up computation by reducing the complexity of the search space. However the linguistic flavor of the attributes is retained, thereby enabling the extraction of more user-friendly *natural* rules that are then mapped to the fuzzy knowledge-based network.

When the input feature is numerical, we divide it into three partitions (with range [0, 1]) using only two parameters P_{j1} and P_{j2} as depicted in Fig. 1. Features in linguistic and set forms can also be handled. Note that, unlike [23], we do not consider

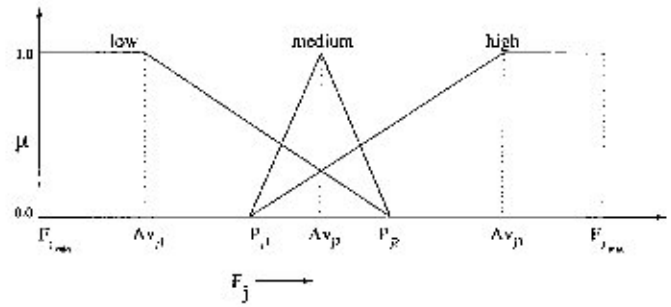

Fig. 1. Linguistic input membership functions.

the arithmetic mean but use *quantiles* or *partition values*[1] [24] in order to minimize the influence of extreme values or noisy patterns.

Conventional ID3 algorithm, using (1).

Let $F_{j_{max}}$ and $F_{j_{min}}$ denote the maximum and minimum values encountered along feature F_j considering N training patterns $F_{1j}, F_{2j}, \ldots, F_{Nj}$. Let these patterns be sorted in the ascending order of their values along the jth axis. The first quantile (P_{j1}) is the value of F_j that exceeds one-third of the measurements and is less than the remaining two-thirds. The second quantile (P_{j2}) is the value of F_j that exceeds two-thirds of the measurements and is less than the remaining one-third. In order to determine the two quantiles, we divide the measurements into a number of small class intervals of equal width δ and count the corresponding class frequencies f_i. The position of the kth partition value (here quantile, as $k = 1, 2$ for three partitions) is calculated as

$$P_{jk} = l_i + \frac{R_k - cf_{i-1}}{f_i} \cdot \delta \tag{3}$$

where l_i is the lower limit of the ith class interval, $R_k = N \cdot k/3$ is the rank of the kth partition value, and cf_{i-1} is the cumulative frequency of the immediately preceding class interval, such that $cf_{i-1} < R_k < cf_i$. Then, in Fig. 1, we have $Av_{j1} = (F_{j_{min}} + P_{j1})/2$, $Av_{j2} = (P_{j1} + P_{j2})/2$, and $Av_{j3} = (P_{j2} + F_{j_{max}})/2$.

The membership values of a pattern along the jth axis, in the corresponding three-dimensional linguistic space of (2), is computed as

$$\mu_{\text{low}}(a_j)(\mathbf{F}_i) = \begin{cases} 1, & \text{for } F_{ij} < Av_{j1} \\ \frac{P_{j2} - F_{ij}}{P_{j2} - Av_{j1}}, & \text{for } Av_{j1} \le F_{ij} < P_{j2} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$\mu_{\text{medium}}(a_j)(\mathbf{F}_i) = \begin{cases} 0, & \text{for } F_{ij} < P_{j1} \\ \frac{Av_{j2} - F_{ij}}{Av_{j2} - P_{j1}}, & \text{for } P_{j1} < F_{ij} < Av_{j2} \\ \frac{F_{ij} - P_{j2}}{P_{j2} - Av_{j2}}, & \text{for } Av_{j2} \le F_{ij} < P_{j2} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

$$\mu_{\text{high}}(a_j)(\mathbf{F}_i) = \begin{cases} 0, & \text{for } F_{ij} < P_{j1} \\ \frac{F_{ij} - P_{j1}}{Av_{j3} - P_{j1}}, & \text{for } P_{j1} \le F_{ij} < Av_{j3} \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

[1]Quantiles or partition values are the values of a variate which divide the total frequency into a number of equal parts

2) Output Membership and Fuzzy Entropy: Consider an l-class problem domain. The membership of the ith pattern in class k, lying in the range [0, 1], is defined as [23]

$$\mu_{ik}(\mathbf{F}_i) = \frac{1}{1 + \left(\frac{z_{ik}}{f_d}\right)^{f_e}} \tag{7}$$

where z_{ik} is the weighted distance of the training pattern $\mathbf{F}_i$ from class C_k, and the positive constants f_d and f_e are the denominational and exponential fuzzy generators controlling the amount of fuzziness in the class membership set.

Fuzziness is incorporated into the ID3 algorithm at the node level by modifying the conventional decision function, with classical Shannon entropy, by the inclusion of different fuzzy measures. The fuzzy entropy considers the membership of a pattern to a class and helps enhance the discriminative power of an attribute. In order to reduce the effect of noise or exceptions, a node is pruned depending on the number of patterns reaching it. For this purpose, a threshold t is defined as a lower bound on the fraction of patterns allowed in an existing node.

Let us now provide the different fuzzy entropy/fuzziness measures, denoted as *cases a, b, d* respectively, investigated at the node level of the decision tree. Note that μ_{ij}, the membership of the jth pattern to the ith class, is calculated by (7) and p_i is the *a priori* probability of the ith class. Comparison is provided with *cases c* [6], *cases e* [25] and *cases f* [22].

Case a:

$$\text{Entropy} = -\sum_{i=1}^{l} p_i \log_{32} p_i - \frac{1}{N} \sum_{i=1}^{l} \sum_{j=1}^{N} [\mu_{ij} \log_2 \mu_{ij} + (1 - \mu_{ij}) \log_2 (1 - \mu_{ij})]. \tag{8}$$

The first term on the right is the classical entropy of (1), while the second term corresponds to fuzzy entropy [23].

Case b: Same as *Case a*, but without pruning.

Case c [6]:

$$\text{Entropy} = -\sum_{i=1}^{l} \frac{\sum_{j=1}^{N} \mu_{ij}}{N} \log_{32} \frac{\sum_{j=1}^{N} \mu_{ij}}{N}. \tag{9}$$

This is a normalized version of fuzzy entropy, with no classical entropy component involved.

Case d:

$$\text{Entropy} = -\sum_{i=1}^{l} \frac{\sum_{j=1}^{N} \mu_{ij}}{N} \log_n \frac{\sum_{j=1}^{N} \mu_{ij}}{N} + \frac{1}{N} \sum_{i=1}^{l} \sum_{j=1}^{N} [\mu_{ij} \log_2 \mu_{ij} + (1 - \mu_{ij}) \log_2 (1 - \mu_{ij})]. \tag{10}$$

This is an amalgamation of the two forms of fuzzy entropy, the first term on the right corresponding to (9) and the second term relating to the fuzzy entropy part of (8).

Case e [25]:

$$\text{Entropy} = -\sum_{i=1}^{l} p_i \log_2 p_i - \frac{1}{N}\sum_{x=1}^{l}\sum_{j=1}^{N} \min(\mu_{ij}, 1-\mu_{ij}). \tag{11}$$

Here the first term on the right is the classical entropy of (1), while the second term corresponds to a fuzzy measure of the ambiguity present.

Case f: Conventional ID3 algorithm, using(1).

C. Performance Measure for Decision Tree

Decision trees generated by different fuzzy entropy measures may vary in size and structure, and this influences the performance of both the tree and the rules extracted from it. In order to evaluate the efficiency of a decision tree we propose the *T-measure*, keeping in view the following issues.

- The less the depths of the leaf nodes of the tree, the better it is since it takes less time to reach a decision.
- The existence of unresolved terminal nodes is undesirable.
- The distribution of labeled leaf nodes at different depths affects the performance of the tree; a tree whose frequently accessed leaf nodes are at lower depths is more efficient in terms of time.

Definition II.1: The **T-measure**, T, for a decision tree is defined as

$$T = \frac{2n - \sum_{i=1}^{N_{leaves}} w_i d_i}{2n - 1} \tag{12}$$

where

$$w_i = \begin{cases} \frac{n_i}{N}, & \text{for a resolved leaf node} \\ \frac{2N_i}{N}, & \text{otherwise} \end{cases} \tag{13}$$

n is the number of binary attributes of a pattern, d_i is the depth of a leaf node, N_{leaves} is the number of terminal (leaf/unresolved) nodes, N is the total number of pattern in the training set and N_i is the total number of training set patterns that percolate down to the i th leaf node.The value of T lies in the interval [0, 1). A value 0 for T is undesirable and a value close to 1 signifies a good decision tree.

Now we demonstrate the evaluation of the T-measure with an example. Consider a two-class problem, with two-dimensional patterns. Let Fig. 2 depict two decision trees generated by two different algorithms. For the decision tree in Fig. 2(a),

$$T = \frac{2\times 2 - 0.5\times 1 - 0.4\times 2 - 0.2\times 2}{2\times 2 - 1} = 0.77$$

while for the decision tree in Fig. 2(b) one obtains

$$T = \frac{2\times 2 - 0.5\times 2 - 0.4\times 1 - 0.2\times 2}{2\times 2 - 1} = 0.73.$$

Hence, we observe that the first decision tree is better than the second, since the fraction of patterns in the node at depth *one* is more in the first case.

Theorem: The value of T-measure lies within 0 and 1, i.e., $0 \le T < 1$.

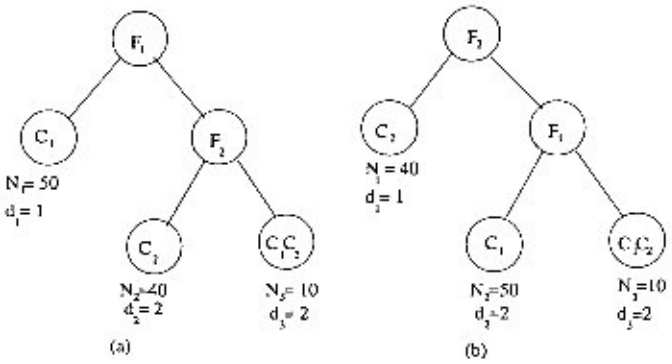

Fig. 2. Example demonstrating T–measure computation.

Proof: Let us first establish the upper limit. By (13), we have

$$w_i \ge \frac{N_i}{N},\ i = 1, 2, \ldots N_{leaves} \tag{14}$$

and

$$d_i \ge 1,\ i = 1, 2, \ldots, N_{leaves}. \tag{15}$$

Hence

$$\sum_{i=1}^{N_{leaves}} w_i d_i > \sum_{i=1}^{N_{leaves}} \frac{N_i}{N}.$$

Since $\sum_{i=1}^{N_{leaves}} N_i = N$, one obtains

$$\sum_{i=1}^{N_{leaves}} w_i d_i \ge 1.$$

So

$$2n - \sum_{i=1}^{N_{leaves}} w_i d_i < 2n - 1$$

$$i.e.,\ T = \frac{2n - \sum_{i=1}^{N_{leaves}} w_i d_i}{2n - 1} < 1. \tag{16}$$

Now we check the lower bound for T. We have

$$w_i < \frac{2N_i}{N},\ i = 1, 2, \ldots, N_{leaves} \tag{17}$$

and

$$d_i < n,\ i = 1, 2, \ldots, N_{leaves}. \tag{18}$$

Hence

$$\sum_{i=1}^{N_{leaves}} w_i d_i \le \sum_{i=1}^{N_{leaves}} \frac{2N_i n}{N} = 2n$$

$$i.e.,\ 0 \le 2n - \sum_{i=1}^{N_{leaves}} w_i d_i$$

$$i.e.,\ 0 \le \frac{2n - \sum_{i=1}^{N_{leaves}} w_i d_i}{2n - 1} \quad T. \tag{19}$$

Thus one obtains

$$0 \le T < 1. \tag{20}$$

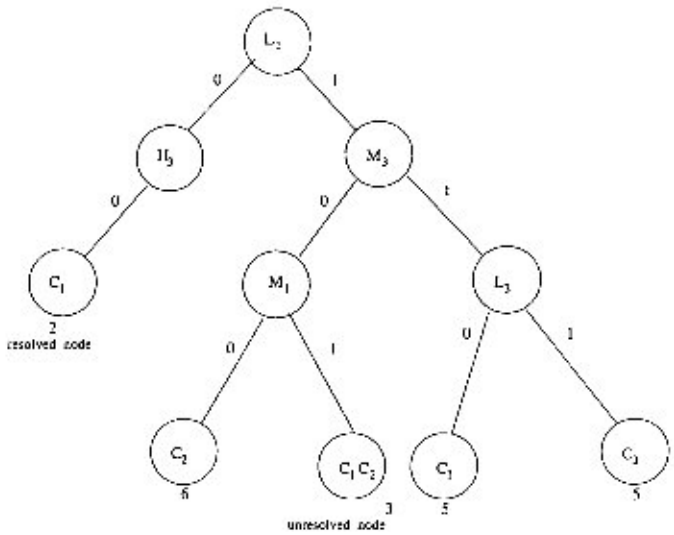

Fig. 3. Sample decision tree for rule generation.

III. Rule Generation and Evaluation

Here we explain the algorithm for extracting domain knowledge, in the form of rules, using the decision tree generated by the fuzzy ID3. Let us consider the leaf nodes only. The path from the root to a leaf can be traversed to generate the rule corresponding to a pattern from that class. In this manner, one obtains a set of rules for all the pattern classes, in the form of intersection of the features/attributes encountered along the traversal paths. The ith attribute is marked as A_i or $\overline{A_i}$ depending on whether the traversal is made along the right or left branch. Each rule is marked by its frequency, that is the number of pattern points reaching this leaf node. Note that each leaf node that has pattern points corresponding to only one class is termed *resolved*.

A. Example

The scheme of extracting the rules from the decision tree is demonstrated with an example. Suppose the training set consists of 21 patterns, from three pattern classes, with three features F_1, F_2 and F_3. After splitting each feature into the three linguistic variables *low*, *medium*, and *high* by (2), one obtains the nine-dimensional symbolic features L_1, M_1, H_1, L_2, M_2, H_2, L_3, M_3, H_3. Let the sample decision tree be shown in Fig. 3, and the extracted rules be

1) $\overline{L_1} \wedge \overline{H_3} \rightarrow C_1 : 2$,
2) $L_1 \wedge \overline{M_3} \wedge \overline{M_1} \rightarrow C_2 : 6$,
3) $L_1 \wedge \overline{M_3} \wedge M_1 \rightarrow C_1 C_2 : 3$,
4) $L_1 \wedge M_3 \wedge \overline{L_3} \rightarrow C_1 : 5$,
5) $L_1 \wedge M_3 \wedge L_3 \rightarrow C_3 : 5$.

B. Quantitative Measures for Rule Evaluation

Now we provide a set of indices for quantitatively evaluating the extracted rules. New measures to estimate the ambiguity/confusion and coverage of these rules are designed in the context of decision trees. Let N be an $l \times l$ matrix whose (i, j)th element n_{ij} indicates the number of patterns actually belonging to class i but classified as class j.

Definition III.1: Accuracy: It is the correct classification percentage, provided by the rules on a test set defined as n_{ic}/n_i, where n_i is equal to the number of points in class i and n_{ic} of these points are correctly classified.

Definition III.2: User's Accuracy: If n'_i points are found to be classified into class i, then user's accuracy (U) is defined as $U = n_{ic}/n'_i$. This gives a measure of the confidence that a classifier attributes to a region as belonging to a class. In other words, it denotes the level of purity associated with a region.

Definition III.3: Kappa [26]: The coefficient of agreement called "kappa" measures the relationship of beyond chance

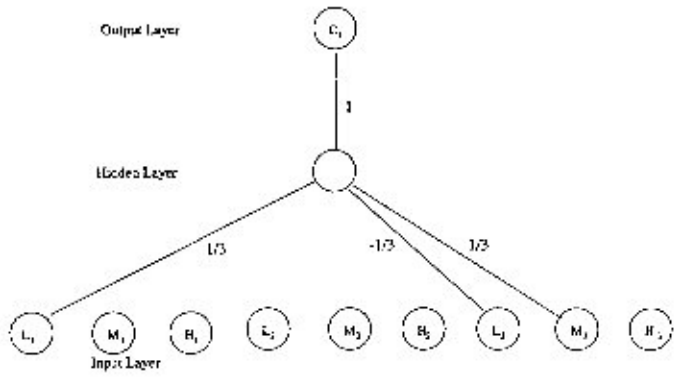

Fig. 4. Weight encoding using Model I.

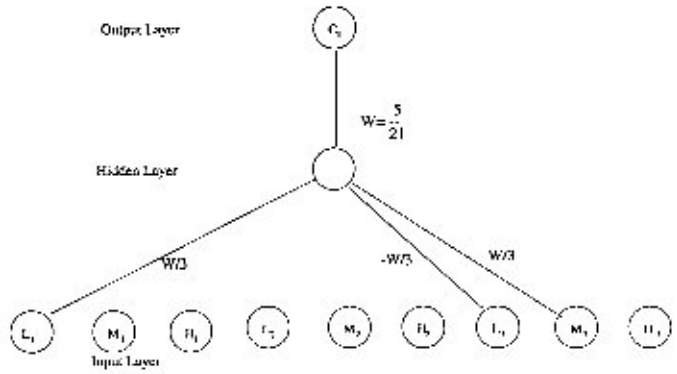

Fig. 5. Weight encoding using Model II.

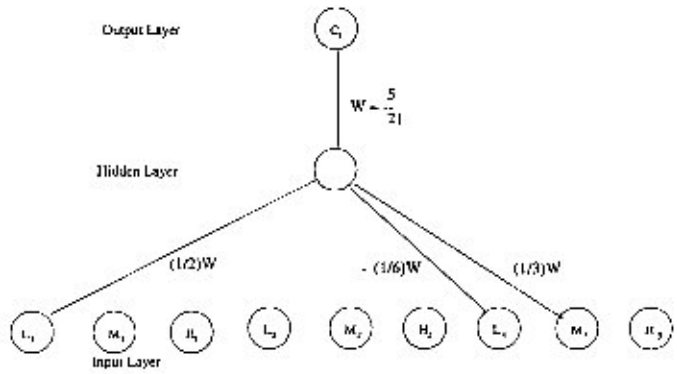

Fig. 6. Weight encoding using Model III.

TABLE I
PERFORMANCE OF FUZZY ID3 ON VOWEL DATA

Case	Train set (%)	Recognition scores (%)													T
		Training							Testing						
		ə	a	i	u	e	o	Net	ə	a	i	u	e	o	Net	
a	10	79.2	69.9	89.0	83.6	81.5	72.2	81.1	65.4	50.3	85.4	82.9	70.9	68.7	72.8	.70
	20	84.7	66.9	89.7	87.6	70.3	65.6	77.6	69.4	59.3	88.8	79.8	67.8	62.0	71.9	.69
	30	77.7	61.7	91.4	87.3	69.7	63.5	75.9	72.5	59.0	87.9	84.7	63.2	65.4	72.6	.68
	40	78.0	56.2	90.1	86.9	69.4	63.2	74.7	74.8	49.2	91.8	89.0	59.3	58.8	71.0	.67
	50	79.7	57.3	94.2	91.1	61.8	64.3	74.9	73.1	60.0	91.3	89.4	58.8	66.5	73.1	.67
b	10	89.3	64.5	89.8	87.4	72.3	60.7	77.2	68.6	50.3	82.9	84.8	68.3	60.7	70.7	.53
	20	90.3	53.4	92.0	87.9	68.5	52.9	73.5	71.5	53.9	88.2	84.3	62.1	51.8	68.8	.53
	30	88.6	62.5	94.1	89.6	55.0	43.4	69.9	74.2	54.2	88.0	85.1	53.1	39.4	64.5	.53
	40	91.2	53.6	95.5	92.6	46.6	31.8	65.6	81.6	55.4	92.4	88.7	46.9	28.3	62.8	.53
	50	90.1	52.6	94.9	91.6	41.8	25.9	63.0	82.4	55.3	93.7	92.0	40.3	25.5	61.6	.53
c	10	87.2	71.1	91.3	84.3	73.9	70.0	79.5	61.7	51.7	87.0	82.4	67.2	61.6	70.4	.70
	20	82.4	57.8	89.1	87.8	73.7	62.2	76.1	65.1	58.0	89.4	81.5	63.5	57.6	70.0	.69
	30	79.0	62.4	91.4	86.4	66.0	62.8	74.8	69.8	54.6	90.0	85.0	59.3	60.9	70.4	.70
	40	76.5	55.5	91.9	86.5	65.5	63.1	74.1	72.0	48.1	91.9	89.4	59.5	57.4	70.6	.69
	50	79.9	51.3	93.1	87.8	61.2	62.9	73.2	70.1	54.9	90.0	84.6	64.5	68.3	73.1	.67
d	10	83.4	58.7	87.8	86.0	78.7	70.2	79.0	65.2	47.3	87.0	86.8	66.1	59.0	70.3	.71
	20	82.3	62.4	88.5	80.3	71.7	68.7	76.3	71.5	51.5	86.9	84.5	66.2	62.1	71.5	.71
	30	76.9	57.7	89.5	82.7	68.2	64.0	74.2	71.0	56.7	90.8	84.7	57.2	60.9	70.3	.69
	40	79.8	55.8	94.2	90.2	60.7	65.6	74.5	72.3	57.5	89.7	84.8	60.5	60.9	71.1	.69
	50	75.0	55.5	94.5	89.3	59.9	66.1	74.1	72.9	50.6	91.7	86.8	56.8	61.1	70.3	.69
e	10	31.5	12.9	78.3	71.0	69.7	77.4	66.0	29.2	27.5	77.1	74.0	64.3	73.0	63.6	.64
	20	43.4	15.4	80.8	68.3	54.3	75.3	61.3	32.7	23.9	87.7	71.1	50.1	85.2	64.5	.61
	30	38.0	9.1	80.6	68.1	44.8	75.0	58.1	41.5	20.4	82.5	72.8	44.3	85.1	62.3	.60
	40	31.1	16.3	82.7	70.0	39.4	79.3	58.9	30.9	22.2	81.5	72.6	38.5	86.9	60.4	.60
	50	27.1	14.7	83.0	70.9	41.3	80.1	58.8	31.0	18.5	81.1	72.1	39.1	88.2	60.7	.60
f	10	28.4	15.5	79.3	65.8	74.3	74.5	63.3	13.2	24.5	68.7	63.7	61.0	48.8	52.7	.64
	20	17.0	16.9	79.4	54.2	61.5	73.3	57.7	19.7	23.5	68.2	65.4	51.0	51.4	51.4	.61
	30	13.5	6.1	82.8	60.7	52.3	77.7	57.1	10.8	19.8	76.4	72.1	47.1	32.2	48.1	.60
	40	14.4	7.7	78.4	61.3	56.7	76.6	57.2	9.7	21.9	76.6	72.6	44.8	34.5	48.4	.60
	50	14.3	7.9	79.6	60.9	52.7	76.3	56.5	5.3	18.1	74.5	72.8	39.8	25.5	44.5	.59

agreement to expected disagreement. It uses all the cells in the confusion matrix, not just the diagonal elements. The estimate of kappa (K) is the proportion of agreement after chance agreement is removed from consideration. The kappa value for class $i(K_i)$ is defined as

$$K_i = \frac{n \cdot n_{ii} \quad n_i \cdot n'_i}{n \cdot n'_i - n_i \cdot n'_i}. \tag{21}$$

The numerator and denominator of overall kappa are obtained by summing the respective numerators and denominators of K_i separately over all classes.

Definition III.4: Confusion [27]: This measure quantifies the goal that the "*confusion should be restricted within minimum number of classes*." This property is helpful in higher level decision making. Let $\dot{n}_{ij}$ be the mean of all n_{ij} for $i \neq j$. Then we define

$$\text{Conf} = \frac{\text{Card}\{n_{ij} : n_{ij} > \dot{n}_{ij}, i \neq j\}}{l} \tag{22}$$

for an l class problem. The lower the value of $Conf$, less is the number of classes between which confusion occurs.

Definition III.5: Coverage: We define it as the ratio between the total number of patterns associated with the rules corresponding to resolved leaf nodes, and the total number of patterns in all the rules and hence the terminal (resolved and/or unresolved) nodes.

When the rules can perfectly classify all the patterns, coverage is 1, and when they cannot classify any pattern then it is 0. For example, from Fig. 3 we have

$$\text{Coverage} = \frac{2+6+5+5}{2+6+3+5-5} = \frac{18}{21}.$$

IV. Mapping of Rules to Neural Network Architecture

In this section we describe a new way of mapping the extracted rules to generate an optimal fuzzy knowledge-based neural network. Unlike other approaches [15]–[17], the frequency of samples (representative of a rule) and the depth of the attributes in the corresponding decision tree are taken into consideration during the mapping.

Before going into the details of knowledge encoding, let us first introduce the different parameters of a multilayer perceptron (MLP). The output of a neuron in any layer (h) of an MLP, other than the input layer ($h \neq 0$), is $y_j^h = 1/[1 + \exp(-\sum_i y_i^{h-1} w_{ji}^{h-1})]$, where y_i^{h-1} is the state of the ith neuron in the preceding ($h-1$)th layer and w_{ji}^{h-1} is the weight of the connection from the ith neuron in layer ($h-1$) to the jth neuron in layer (h). For nodes in the input layer, y_j^0 corresponds to the jth component of the input vector. Note that $x_j^h = \sum_i y_i^{h-1} w_{ji}^{h-1}$. The $3n$-dimensional input vector of (2) is clamped at the input layer to the input nodes $[y_1^0, y_2^0, \ldots, y_{3n}^0]$. Here $y_1^0, \ldots, y_{3n}^0$ refer to the activation values of the $3n$ neurons in the input layer. The l-dimensional output vector, in terms of class membership values (μ) of patterns by (7), is clamped at the l nodes in the output layer of the MLP. During training, the weights are updated by backpropagating errors with respect to these membership values such that the contribution of uncertain/ambiguous pattern vectors is automatically reduced.

The details of the different knowledge encoding schemes, mapping the rules extracted from the decision tree, are described here. Let r_{ki} be the ith rule for class C_k with frequency f'_{ki}. Each rule is mapped using a single hidden node, modeling the conjunct, that connects the attributes corresponding to the appropriate pattern class. Therefore, one generates at least l hidden nodes in a single hidden layer for an l-class problem. For simplicity, rules involving only one class (pertaining to leaves) are selected and those corresponding to unresolved nodes of the decision tree are discarded. If there are two rules for a single class C_k, then that rule with the highest frequency is considered. Hence we use only l hidden nodes to model l classes. This constraint can of course be relaxed to incorporate other rules, *albeit* at the cost of increasing the size and computational complexity of the resultant network. The sample rules generated from Fig. 3 thus reduce to

1) $L_1 \wedge \overline{M}_3 \wedge \overline{M}_1 \rightarrow C_2$: 6,
2) $L_1 \wedge M_3 \wedge \overline{L}_3 \rightarrow C_1$: 5,
3) $L_1 \wedge M_3 \wedge L_3 \rightarrow C_3$: 5.

These rules are used to initially encode an MLP, that then learns in the presence of training data. It is to be noted that these rules just serve as representatives, describing the major characteristics of the pattern classes, and as the starting point of the MLP, for further learning. The representative rulebase, therefore, need not be too detailed/accurate; rather, a crude knowledge is sufficient to initiate the training procedure. This is the reason for sacrificing accuracy at the expense of simplicity at the decision tree level, by pruning the nodes and limiting the size of the extracted rulebase. The generalization aspect and other intricacies of the decision boundary are handled after the network mapping phase, during neural learning.

TABLE II
Quantitative Measures For Evaluating Rules in Vowel Data

Case	Train set (%)	Accuracy (%)	User's Accuracy (%)	Kappa	Confusion	Coverage
a	10	63.20	72.67	0.67	2.30	0.80
	20	62.70	73.15	0.67	2.37	0.78
	30	59.74	75.47	0.70	2.37	0.72
	40	59.62	77.14	0.72	2.20	0.73
	50	60.06	75.40	0.70	2.34	0.75
b	10	60.33	71.45	0.65	2.54	0.76
	20	60.14	73.58	0.67	2.24	0.79
	30	60.52	77.63	0.72	2.26	0.73
	40	59.46	79.36	0.75	2.38	0.69
	50	60.15	80.65	0.77	2.06	0.72
c	10	70.81	87.14	0.85	1.75	0.78
	20	66.90	84.60	0.81	2.04	0.75
	30	65.59	84.08	0.81	2.37	0.75
	40	62.62	84.20	0.81	2.32	0.72
	50	60.92	83.00	0.80	2.14	0.70
d	10	71.55	86.45	0.83	1.84	0.77
	20	65.26	84.80	0.82	2.07	0.74
	30	62.04	81.42	0.78	2.27	0.72
	40	62.22	82.82	0.80	2.26	0.71
	50	59.22	83.22	0.80	2.20	0.68
e	10	60.69	77.22	0.72	2.19	0.71
	20	59.70	71.80	0.67	1.89	0.60
	30	53.59	74.64	0.70	1.93	0.54
	40	53.39	73.95	0.69	1.99	0.52
	50	53.27	72.71	0.68	1.92	0.52
f	10	60.36	63.00	0.57	2.13	0.73
	20	57.15	66.00	0.60	2.29	0.60
	30	57.10	66.00	0.61	2.11	0.54
	40	55.72	67.00	0.61	1.97	0.50
	50	55.67	67.00	0.61	2.16	0.47

A. Model I

The weight w_{ki}^1, between output node k (class C_k) and hidden node i (rule r_{ki}), is set at $f'_{ki} + \varepsilon$, where ε is a small random number; and $f''_{ki} = 1$. The weight $W_{iA_j}^0$ between attribute A_j (L_j or M_j or H_j) and hidden node i is clamped to $(w_{ki}^1/\text{Card}(r_{ki})) - \varepsilon$. Here $\text{Card}(r_{ki})$ indicates the number of features/attributes encountered along the traversal path from the root to the leaf containing the pattern corresponding to rule r_{ki} a of class C_k. In other words, $\text{Card}(r_{ki})$ is the number of operands in the conjunct of rule r_{ki} for class C_k. An example illustrating this scheme is provided in Fig. 4 for class C_1. The superscript indicates the layer of the neural net under consideration, the values 1 and 0 corresponding to the hidden-output and input-hidden layers respectively.

B. Model II

Here a factor $W = f'_{ki}/(\sum_k f'_{ki})$ is used to indicate the importance of a rule for a particular class C_k, among all rules determining the whole network. The scheme for mapping weight $w_{iA_j}^0$ is the same as in Model I. An example illustrating this scheme is provided in Fig. 5.

TABLE III
COMPARATIVE PERFORMANCE OF KNOWLEDGE-ENCODED MLPS FOR VOWEL DATA

	Train set (%)	Recognition scores (%)													No. of link	No. of cycle
		Training							Testing							
		ə	a	i	u	e	o	Net	ə	a	i	u	e	o	Net	
	10	72.2	98.2	89.1	89.3	99.4	98.1	93.3	35.9	87.0	81.0	82.7	81.4	84.5	78.9	90.5	242
M	20	56.6	91.4	78.2	84.1	98.1	98.0	87.5	31.8	86.5	74.6	75.0	89.8	92.7	79.6	91.2	203
L	30	38.6	90.8	66.2	75.7	97.7	98.8	82.6	28.4	85.3	68.2	75.0	94.3	94.1	79.4	91.4	106
P	40	23.8	89.2	62.5	60.8	96.5	97.7	76.8	22.8	87.0	65.1	66.7	95.0	96.4	77.6	91.8	27
M	10	57.0	86.6	85.3	91.0	97.6	97.9	89.4	27.7	84.4	80.4	80.6	85.3	87.0	79.0	65.6	197
o	20	42.7	88.0	79.0	81.2	96.3	97.9	85.6	25.2	84.6	77.0	78.2	90.2	91.9	79.7	65.6	102
d	30	24.9	89.5	76.3	78.1	95.5	97.3	83.0	18.8	83.0	75.3	75.5	93.0	95.3	80.0	65.2	40
I	40	14.9	87.4	74.0	75.9	96.3	98.9	81.5	10.1	82.8	73.8	73.3	93.4	97.9	78.9	67.7	35
M	10	11.2	80.6	74.5	71.4	92.2	98.3	78.3	27.8	84.1	80.4	80.6	84.8	86.2	78.7	66.5	35
o	20	49.2	90.6	84.7	90.4	97.6	97.9	88.9	25.9	84.5	77.0	78.2	90.0	91.9	79.7	65.2	28
d	30	16.5	89.2	77.1	78.7	95.4	97.6	82.7	18.0	83.2	75.3	75.0	93.0	95.2	79.4	63.1	197
II	40	3.4	85.2	71.6	73.9	92.7	98.6	78.6	9.4	81.6	73.7	73.3	93.2	98.2	78.7	66.1	102
M	10	51.7	92.0	85.9	86.6	97.3	94.3	88.7	28.0	86.2	78.5	83.6	84.1	84.6	78.5	63.6	102
o	20	37.7	91.6	80.5	84.7	95.7	98.0	86.0	26.0	86.0	77.2	77.4	90.6	93.3	80.3	68.3	50
d	30	25.9	87.6	75.5	77.5	96.3	98.5	82.9	18.6	84.5	74.1	76.7	92.9	96.4	80.2	66.5	35
III	40	22.8	83.0	75.3	75.1	94.9	97.8	80.8	17.7	84.7	73.3	73.6	92.5	95.6	79.0	64.2	25

C. Model III

Here, as in Model II, a factor $W = f'_{ki}/(\sum_k f'_{ki})$ is used to indicate the importance of a rule for a particular class C_k among all rules determining the whole network. But the scheme for mapping weight $w^0_{iA_j}$ depends on the importance of feature A_j in the corresponding decision tree. While constructing the tree, the feature associated with a node is chosen on the basis of maximum information gain. Hence the attributes/features ought to be given weightage in descending order of their appearance in the decision tree. Consider Fig. 3. We note that the attributes are selected in the order L_1, M_3, I_{-3} for class C_1. So the weight $w^0_{A_1 I_{-3}}$ is assigned a value $(\{2[\mathrm{Card}(r_{ki}) \quad i+1]\}/(\mathrm{Card}(r_{ki})[\mathrm{Card}(r_{ki})+1]))W$. It is to be noted that

$$\sum_{i=1}^{\mathrm{Card}(r_{ki})} \frac{2\,[\mathrm{Card}\,(r_{ki}) - i + 1]}{\mathrm{Card}\,(r_{ki})\,[\mathrm{Card}\,(r_{ki}) + 1}\Bigg| W = W. \qquad (23)$$

An example illustrating this scheme is provided in Fig. 6.

V. IMPLEMENTATION AND RESULTS

The system was implemented on three sets of real-life data, *viz.*, *Vowel* data (available in http://www.isical.ac.in/~sushmita/patterns), and the *Wisconsin breast cancer* and *Balance scale* data [28]. Different sizes of training sets are selected at random and, in each case, the remaining data is kept aside as the test set. All results are averaged over 40 runs. The threshold t for pruning a node of the decision tree is set at 0.2 after several experiments. The stability of this choice of t has been verified for different datasets. The performance of the fuzzy ID3, extracted rules and the knowledge-encoded MLP's are provided in each case.

It is generally observed, from Tables I and IV, Table VII, that the value of T decreases with an increase in size of the training data. This is because an increase in training set-size results in a tree of greater depth, with an increased possibility of larger number of unresolved nodes, leading to a lower value of T by (12)–(13). In general, *cases a, c, d* (fuzzy entropy by (8)–(10)) are found to perform better than *case e*[25] and *case f* (classical entropy).

The fuzzy ID3 (Tables I and IV, Table VII) is used to extract rules (Tables II and V, Table VIII), which are then used for generating fuzzy knowledge-based networks (Tables III and VI, Table IX). The classification performance is provided for all three stages. It is observed that generally, the knowledge-based networks result in the better performance in terms of network size and recognition scores. This is natural since the crude domain knowledge is encoded and further refined here, in the presence of training data.

A. Vowel Data

Here we present some results demonstrating the effectiveness of the fuzzy ID3 algorithm on a set of 871 Indian Telugu vowel sounds [29], collected by trained personnel. These were uttered by three male speakers in the age group of 30 to 35 years, in a Consonant-Vowel-Consonant context. The data set has three features; F_1, F_2 and F_3 corresponding to the first, second, and third vowel format frequencies obtained through

TABLE IV
PERFORMANCE OF FUZZY ID3 ON CANCER DATA

Case	Train set (%)	Recognition scores (%) Training 1	Training 2	Training Net	Testing 1	Testing 2	Testing Net	T
	10	99.3	70.9	89.9	94.6	66.7	84.7	.95
	20	100.0	74.8	91.4	99.6	68.0	88.4	.94
a	30	100.0	72.6	90.4	100.0	64.5	87.3	.94
	40	100.0	73.5	91.0	100.0	67.9	88.7	.94
	50	100.0	72.0	90.2	100.0	71.8	90.1	.94
	10	97.5	96.5	96.8	96.5	90.0	94.2	.51
	20	100.0	93.9	97.8	99.4	84.2	94.0	.51
b	30	100.0	93.0	97.6	100.0	85.1	94.9	.51
	40	100.0	92.8	97.4	100.0	86.3	95.1	.51
	50	100.0	95.0	98.2	100.0	85.6	94.8	.51
	10	98.2	93.1	96.5	96.4	83.3	91.8	.96
	20	99.8	93.0	97.5	100.0	83.9	94.3	.96
c	30	99.2	91.4	96.6	99.8	91.9	97.0	.97
	40	100.0	92.6	97.4	100.0	91.1	96.9	.96
	50	100.0	95.1	98.3	100.0	90.7	96.7	.96
	10	97.4	90.5	95.1	94.0	85.8	91.1	.96
	20	99.6	93.3	97.5	99.5	84.8	94.3	.96
d	30	100.0	93.2	97.6	100.0	90.8	96.6	.96
	40	100.0	92.9	97.6	100.0	92.4	97.3	.96
	50	100.0	92.6	97.5	100.0	89.2	96.1	.96
	10	98.4	67.5	87.8	96.0	70.3	90.2	.96
	20	99.0	66.2	87.4	95.8	81.4	90.7	.96
e	30	99.1	65.7	87.3	97.6	71.7	88.4	.96
	40	98.8	67.4	88.0	96.5	86.3	92.8	.96
	50	98.9	66.2	87.2	97.7	73.8	89.4	.96
	10	95.6	66.5	86.7	87.5	83.7	86.2	.95
	20	96.0	70.7	87.8	92.4	84.1	89.5	.95
f	30	96.0	68.7	87.0	92.5	84.6	89.8	.95
	40	95.4	70.3	86.5	92.8	86.8	90.7	.94
	50	95.1	70.4	86.4	93.1	87.4	91.2	.94

TABLE V
QUANTITATIVE MEASURES FOR EVALUATING RULES IN CANCER DATA

Case	Train set (%)	Accuracy (%)	User's Accuracy (%)	Kappa	Confusion	Coverage
	10	89.19	89.19	0.82	1.45	1.00
	20	86.25	86.25	0.72	1.50	1.00
a	30	86.91	86.91	0.76	1.50	1.00
	40	84.73	84.73	0.70	1.48	1.00
	50	86.55	86.55	0.75	1.50	1.00
	10	82.06	82.06	0.54	1.50	1.00
	20	89.46	89.46	0.77	1.50	1.00
b	30	90.33	90.33	0.79	1.50	1.00
	40	88.95	88.95	0.75	1.50	1.00
	50	90.58	90.58	0.79	1.50	1.00
	10	90.22	90.22	0.76	1.55	1.00
	20	93.64	93.64	0.84	1.57	1.00
c	30	90.30	90.30	0.74	1.52	1.00
	40	90.22	90.22	0.76	1.52	1.00
	50	86.75	86.75	0.69	1.48	1.00
	10	93.90	93.90	0.85	1.57	1.00
	20	91.58	91.58	0.79	1.50	1.00
d	30	90.81	90.81	0.77	1.50	1.00
	40	91.46	91.46	0.80	1.52	1.00
	50	87.63	87.63	0.73	1.50	1.00
	10	87.35	87.35	0.76	1.48	1.00
	20	83.49	83.49	0.67	1.48	1.00
e	30	86.91	86.91	0.76	1.50	1.00
	40	86.65	86.65	0.76	1.50	1.00
	50	80.68	80.68	0.60	1.45	1.00
	10	85.95	85.95	0.73	1.55	1.00
	20	86.88	86.88	0.74	1.50	1.00
f	30	85.00	85.00	0.70	1.48	1.00
	40	87.22	87.22	0.76	1.50	1.00
	50	87.06	87.06	0.72	1.50	1.00

spectrum analysis of the speech data. Fig. 7 shows a 2D projection of the 3D feature space of the six vowel classes *a, i, u, e, o* in the F_1-F_2 plane, for ease of depiction. The boundaries of the classes in the given data set are ill-defined (fuzzy).

Table I provides the recognition scores (%) and 'T'-values for the different *cases a–f* of (8)–(11) and (1), over both the training and test sets. It is observed that fuzzy ID3 with the entropy measure *a* gives the best generalization performance, in terms of score (%), over the test set. On the other hand, *case d* gives the highest values for 'T' followed by *cases c* and *a*. This implies that the entropy term of (10) generates a tree of least overall depths, followed by those of (9) and (8) respectively.

The various quantitative measures are computed in Table II for the rules generated in *cases a–f*. The *User's Accuracy* and *Kappa* of the rules is approximately greater than 80% and 0.8 respectively for *cases c* and *d*. Hence the fuzzy entropy of (9)–(10) result in better accuracy. The *Coverage* is poorer for *cases e, f*, while the *Confusion* is found to be lower in *case e*. Thus the fuzzy measure of (11) leads to a smaller number of classes between which confusion (misclassification) occurs, at the expense of poorer *coverage* of classification.

A comparative performance of the knowledge encoded MLP's, using mapping schemes of *models I–III* with fuzzy measures of *case a* at node level, and the conventional MLP (no encoding) are provided in Table III. Six hidden nodes are used in the network. It is observed that the knowledge-based MLP encoded by *model III* [(23)] provides higher recognition scores over test set. All three knowledge-encoded models are of smaller size and require less number of training cycles. This implies that consideration of the frequency of samples and the attribute depths in the decision tree, during mapping, retains more meaningful information for neural net design.

TABLE VI
COMPARATIVE PERFORMANCE OF KNOWLEDGE-ENCODED MLPS FOR CANCER DATA

	Train set (%)	Recognition scores (%) Training 1	2	Net	Testing 1	2	Net	No. of links	No. of cycles
MLP	10	98.9	99.8	99.2	97.0	95.0	96.3	20.7	21
	20	98.6	96.5	97.9	96.8	95.1	96.2	32.5	26
	30	98.4	99.0	98.6	96.7	96.1	96.5	33.4	26
	40	98.4	98.4	98.4	97.0	94.9	96.3	38.0	44
Model I	10	98.0	100.0	99.1	96.7	96.8	96.7	32.2	32
	20	98.8	99.8	99.2	96.9	95.5	96.5	31.5	33
	30	98.4	99.3	98.7	96.8	93.1	95.5	34.6	57
	40	98.7	98.5	98.6	96.8	94.9	96.1	36.8	81
Model II	10	98.3	99.6	98.8	96.7	95.6	96.3	33.0	28
	20	98.3	99.4	98.6	96.9	95.0	96.5	36.6	32
	30	98.2	99.3	98.6	96.7	95.6	96.4	38.8	57
	40	98.2	98.7	98.4	96.9	94.4	96.0	35.3	80
Model III	10	98.3	100.0	98.9	96.8	96.3	96.6	31.3	28
	20	98.3	99.9	98.9	96.6	96.3	96.5	31.8	32
	30	98.4	99.0	98.7	96.6	96.5	96.2	33.8	57
	40	98.6	98.4	98.5	97.2	94.5	96.2	36.5	81

B. Wisconsin Breast Cancer Data

The *Breast Cancer* data [28], [30] consists of 699 patterns with nine input features, corresponding to cytological characteristics of human breast tissues, *viz., Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli and Mitoses,* having continuous values in the range [1, 10], for two output classes *Benign* and *Malignant* (referred to as 1 and 2 in the sequel). Two hidden nodes are used for network mapping.

Tables IV–VI provide the different results. It is observed from Table IV that entropy measures *c* and *d* lead to the best overall performance, both in terms of recognition scores and T -value, followed by *cases e* and *f*. In this aspect, it is analogous to Table I where the fuzzy entropy of (9)–(10) perform better. Although the classification performance of *case b* is higher than *a* (its pruned version), the value of T is poor in the former. This is natural since a pruned tree (*case a*) has a lower depth and hence higher T. As in Table II, here Table V demonstrates that the rules generated in *cases c* and *d* have higher overall *Accuracy*. It is interesting to note that the *Coverge* for classification of the rules is perfect in all cases. It is seen from Table VI that, here, there is no significant gain in using knowledge-encoded MLP's. This is perhaps because the $27 \times 2 \times 2$ network does not have much scope for improvement with only two hidden nodes involved, and there already exists reasonably good classification prior to the MLP tuning.

C. Balance Scale Data

The Balance scale data [28] consists of 625 instances generated to model psychological experimental results. There are four

TABLE VII
PERFORMANCE OF FUZZY ID3 ON BALANCE DATA

Case	Train set (%)	Recognition scores (%) Training 1	2	3	Net	Testing 1	2	3	Net	T
a	10	92.4	73.9	89.6	89.8	84.9	70.7	74.6	79.4	.81
	20	89.7	76.0	83.7	86.0	84.6	81.2	70.7	80.7	.78
	30	86.1	85.3	79.6	83.2	82.1	86.4	78.8	80.9	.78
	40	86.4	83.0	81.8	84.4	82.3	85.8	79.4	81.1	.78
	50	83.3	84.9	79.8	81.9	80.2	85.6	78.7	79.8	.76
b	10	96.1	71.0	91.6	92.4	86.8	68.9	75.7	80.2	.52
	20	92.6	70.7	81.2	85.6	83.6	73.7	70.9	76.9	.52
	30	85.4	80.6	73.9	79.7	80.9	76.0	72.2	76.0	.52
	40	83.3	81.2	70.6	77.3	79.3	78.6	69.3	74.7	.52
	50	81.5	84.9	65.3	74.3	76.4	89.0	62.3	70.9	.52
c	10	95.6	70.8	86.9	90.1	83.6	69.8	67.9	75.2	.52
	20	92.5	77.7	77.5	84.5	86.2	64.5	68.1	76.0	.52
	30	87.8	85.1	72.3	80.6	86.2	63.8	68.2	75.9	.52
	40	81.9	83.4	69.3	76.1	82.3	75.7	65.0	73.9	.53
	50	79.9	82.9	68.6	75.1	78.6	86.4	62.2	71.7	.52
d	10	90.4	74.4	84.9	89.8	83.3	77.0	72.0	77.6	.52
	20	91.5	77.0	79.6	85.2	84.9	63.4	70.5	76.4	.52
	30	82.2	85.5	74.2	78.9	82.3	72.1	70.4	75.9	.52
	40	81.0	84.4	70.5	76.6	81.9	73.9	66.5	74.1	.52
	50	78.4	85.6	65.4	73.0	76.5	84.5	63.9	72.4	.52
e	10	61.8	50.0	80.0	71.1	75.6	32.0	61.4	65.4	.75
	20	79.5	36.2	55.5	64.7	74.1	35.6	66.5	67.5	.71
	30	70.0	45.8	73.4	69.8	70.2	42.1	71.8	68.7	.70
	40	70.7	29.1	72.6	68.5	69.5	31.8	70.6	67.0	.68
	50	71.4	30.4	73.4	69.2	67.6	34.2	76.1	68.5	.69
f	10	64.6	46.6	80.0	71.0	64.3	46.9	66.5	63.8	.75
	20	64.0	48.7	69.8	65.7	66.9	51.5	72.1	68.1	.72
	30	68.3	40.0	76.0	69.6	69.0	46.0	72.6	68.8	.72
	40	68.8	38.9	74.3	69.0	66.2	45.2	76.4	69.1	.70
	50	69.3	39.2	80.9	72.7	65.7	39.2	75.9	68.2	.70

numeric attributes corresponding to the *left weight, left distance, right weight* and *right distance*, and three output classes, viz., *tip right, tip left* and *balanced* (referred to as 1, 2 and 3 in the sequel). We use three hidden nodes during network mapping.

Tables VII–IX provide the various results. It is observed from Table VII that the entropy measure *a* provides the best results in terms of recognition scores over test set, and T-value. *Cases e* and *f* provide moderate values for T (around 0.7), but have poorer recognition scores (less than 70%). On the other hand, *cases b, d* and *c* have moderate classification performance (around 75%) at the expense of very low values for T. Hence the fuzzy entropy of (8)–(10) provide better classification. Table VIII shows that the rules extracted in *case a* have better overall performance in terms of *Accuracy* and *Coverage*, while the Confusion in misclassification is maximum for the unpruned *case b*. It is seen from Table IX that, in general, the knowledge-encoded MLPs fare better than the conventional MLP in terms of both size and training time.

TABLE VIII
QUANTITATIVE MEASURES FOR EVALUATING RULES IN BALANCE DATA

Case	Train set (%)	Accuracy (%)	User's Accuracy (%)	Kappa	Confusion	Coverage
	10	84.76	91.72	0.87	1.73	0.92
	20	78.80	89.93	0.85	1.65	0.86
a	30	75.48	88.93	0.83	1.80	0.83
	40	70.26	89.15	0.84	1.85	0.84
	50	73.44	88.62	0.84	1.77	0.80
	10	86.13	97.41	0.96	2.38	0.88
	20	77.06	98.47	0.97	2.62	0.77
b	30	70.13	99.75	1.00	2.93	0.70
	40	66.04	99.60	0.99	2.92	0.66
	50	62.34	100.00	1.00	3.00	0.62
	10	81.78	89.37	0.83	1.78	0.90
	20	77.04	90.34	0.85	1.68	0.84
c	30	73.96	88.70	0.83	1.75	0.81
	40	74.14	87.71	0.81	1.58	0.82
	50	73.37	85.93	0.79	1.68	0.83
	10	73.94	87.01	0.81	1.52	0.82
	20	77.12	89.68	0.84	1.68	0.84
d	30	74.52	87.98	0.81	1.60	0.82
	40	74.10	88.05	0.84	1.70	0.81
	50	73.94	87.01	0.81	1.52	0.82
	10	67.51	71.78	0.53	1.36	0.91
	20	78.92	88.45	0.82	1.70	0.87
e	30	75.59	89.20	0.84	1.75	0.83
	40	74.68	87.90	0.83	1.77	0.83
	50	74.36	87.30	0.82	1.80	0.83
	10	78.00	89.08	0.83	1.72	0.85
	20	77.49	88.34	0.83	1.70	0.86
f	30	75.86	88.42	0.83	1.70	0.84
	40	73.17	88.18	0.83	1.82	0.80
	50	67.51	71.78	0.53	1.36	0.91

TABLE IX
COMPARATIVE PERFORMANCE OF KNOWLEDGE-ENCODED MLPS FOR BALANCE DATA

	Train set (%)	Recognition scores (%)								No. of links	No. of cycles
		Training				Testing					
		1	2	3	Net	1	2	3	Net		
	10	99.6	88.0	99.8	98.3	90.9	15.6	86.2	82.8	46.2	309
MLP	20	97.0	35.3	97.2	91.8	95.2	12.3	92.7	87.5	47.8	331
	30	97.6	25.8	96.5	91.8	95.2	14.3	91.9	87.8	48.4	307
	40	97.0	28.3	95.0	91.3	97.2	10.2	93.5	89.6	47.1	312
	10	98.5	64.4	97.6	95.0	87.8	17.0	88.6	82.6	38.7	277
Model	20	97.8	38.2	97.0	92.8	94.8	15.4	92.6	87.4	36.2	272
I	30	98.1	26.8	96.5	91.2	97.0	7.1	94.2	88.7	37.9	252
	40	98.2	14.8	97.3	91.3	96.8	6.6	94.3	88.8	38.3	247
	10	98.3	64.9	98.2	92.7	98.0	20.3	96.3	91.3	33.4	277
Model	20	97.4	37.4	97.1	92.0	98.0	19.7	97.1	91.2	33.0	272
II	30	99.1	17.9	94.5	90.5	96.0	12.7	92.9	88.0	33.4	252
	40	97.5	15.8	97.0	91.0	96.3	15.1	93.6	88.6	40.1	247
	10	97.8	61.2	99.0	94.8	88.1	11.8	89.7	82.7	39.4	277
Model	20	97.1	37.2	95.0	91.9	94.3	18.1	91.9	87.3	37.0	272
III	30	98.1	20.6	95.7	91.3	96.0	18.5	94.1	89.1	35.7	252
	40	97.6	18.1	94.0	90.0	97.1	9.3	93.6	88.9	36.8	247

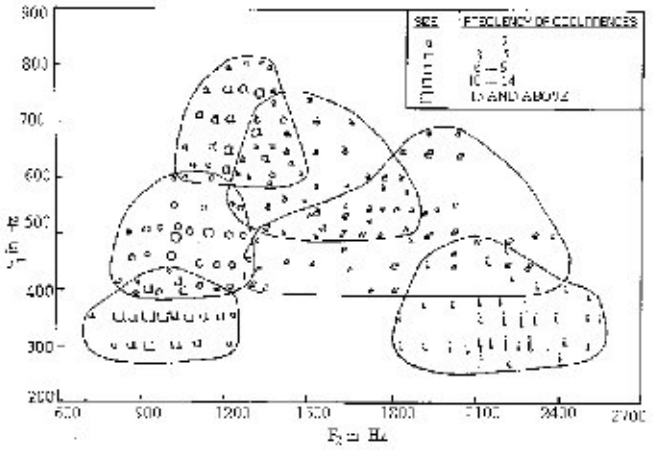

Fig. 7. Vowel diagram in F'-F_2 plane.

VI. CONCLUSIONS AND DISCUSSION

Some issues related to the design of a fuzzy knowledge-based network, based on linguistic rules extracted from a fuzzy decision tree, have been dealt with in this article. Major contributions include

1) developing a new scheme for automatic linguistic discretization of continuous attributes using quantiles;
2) introducing the novel concept of a quantitative measure T to evaluate the goodness of the decision tree, in terms of its compactness and performance;
3) evaluating quantitatively the extracted linguistic rules with some new indices;
4) mapping the linguistic rules to a fuzzy knowledge-based network, incorporating frequency of samples and depth of attributes in the decision tree;
5) using new fuzziness measures at node level of the tree, to handle overlapping classes.

Effectiveness of the system has been exhaustively demonstrated on three sets of real-life data, *viz., Vowel, Wisconsin Breast Cancer* and *Balance scale*. Knowledge encoding using linguistic rules extracted from the fuzzy decision tree generally enhances the performance of the knowledge-based system in terms of both network compactness and recognition scores. It is typically observed that the value of 'T' decreases with an increase in size of the training data. This is because an increase in training set size leads to the consideration of a larger number of both noisy and good samples during the decision tree generation. The former influences the formation of a tree of greater depth, with an increased possibility of larger number of unresolved nodes, leading to a lower value of 'T'. In general, *cases a, c, d* [fuzzy entropy at node level of tree, by (8)–(10)] performs better than *case e* [25] and *case f* (classical entropy).

The automatic fuzzy partitioning of the feature space to overlapping linguistic terms has been made using quantiles. One can, of course, introduce more partitions corresponding to each feature. It was observed that in addition to increasing the computational complexity, this does not always induce a higher accuracy. Increasing the granulation/partitioning at certain selected *interesting* regions of the feature space however, is an issue currently being investigated.

REFERENCES

[1] J. R. Quinlan, "Induction on decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[2] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst, Man, Cybern.*, vol. 21, pp. 660–674, 1991.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks/Cole, 1984.

[4] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

[5] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, pp. 1–14, Jan. 1998.

[6] H. Ichihashi, T. Shirai, K. Nagasaka, and T. Miyoshi, "Neuro fuzzy ID3: A method of inducing fuzzy decision trees with linear programming for maximizing enrtopy and algebraic methods," *Fuzzy Sets Syst.*, vol. 81, no. 1, pp. 157–167, 1996.

[7] W. Xizhao and J. Hong, "On the handling of fuzziness for continuous-valued attributes in decision tree generation," *Fuzzy Sets Syst.*, vol. 99, pp. 283–290, 1998.

[8] W. Pedrycz and A. Sosnowski, "Designing decision trees with the use of fuzzy granulation," *IEEE Trans. Syst., Man, Cybern. A*, vol. 30, pp. 151–159, Mar. 2000.

[9] Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets Syst.*, vol. 69, pp. 125–139, 1995.

[10] X. Wang, B. Chen, G. Qian, and F. Ye, "On the optimization of fuzzy decision trees," *Fuzzy Sets Syst.*, vol. 112, pp. 117–125, 2000.

[11] I. J. Chiang and J. Y. J. Hsu, "Integration of fuzzy classifiers with decision trees," in *Proc. Asian Fuzzy Syst. Symp.*, 1996, pp. 266–271.

[12] I. Hayashi, T. Maeda, A. Bastian, and L. C. Jain, "Generation of fuzzy decision trees by fuzzy ID3 with adjusting mechanism of and/or operators," in *Proc. Int. Conf. Fuzzy Syst.*, 1998, pp. 681–685.

[13] T. Kohonen, *Self-Organization and Associative Memory*. Berlin, Germany: Springer-Verlag, 1989.

[14] H. J. Lu, R. Setiono, and H. Liu, "Effective data mining using neural networks," *IEEE Trans. Knowledge Data Eng.*, vol. 8, pp. 957–961, 1996.

[15] I. K. Sethi, "Entropy nets: From decision trees to neural networks," *Proc. IEEE*, vol. 78, pp. 1605–1613, 1990.

[16] I. Ivanova and M. Kubat, "Initialization of neural networks by means of decision trees," *Knowledge-Based Syst.*, vol. 8, pp. 333–344, 1995.

[17] R. Setiono and W. K. Leow, "On mapping decision trees and neural networks," *Knowledge-Based Syst.*, vol. 12, pp. 95–99, 1999.

[18] L. M. Fu, "Knowledge-based connectionism for revising domain theories," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 173–182, 1993.

[19] G. G. Towell and J. W. Shavlik, "Knowledge-based artificial neural networks," *Artif. Intell.*, vol. 70, pp. 119–165, 1994.

[20] S. Mitra, R. K. De, and S. K. Pal, "Knowledge-based fuzzy MLP for classification and rule generation," *IEEE Trans. Neural Networks*, vol. 8, pp. 1338–1350, 1997.

[21] M. Banerjee, S. Mitra, and S. K. Pal, "Rough fuzzy MLP: Knowledge encoding and classification," *IEEE Trans. Neural Networks*, vol. 9, pp. 1203–1216, 1998.

[22] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley, 1989.

[23] S. K. Pal and S. Mitra, *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*. New York: Wiley, 1999.

[24] G. R. Davies and D. Yoder, *Business Statistics*. London: Wiley, 1937.

[25] P. K. Singal, S. Mitra, and S. K. Pal, "Incorporation of fuzziness in ID3 and generation of network architecture," *Neural Comput. Applicat.*, vol. 10, pp. 155–164, 2001.

[26] G. H. Rosenfeld and K. Fitzpatrick-Lins, "Coefficient of agreement as a measure of thematic classification accuracy," *Photogrammetric Eng. Remote Sensing*, vol. 52, pp. 223–227, 1986.

[27] S. K. Pal, S. Mitra, and P. Mitra, "Rough fuzzy MLP: Modular evolution, rule generation and evaluation," *IEEE Trans. Knowledge Data Eng.*, vol. 15, 2003, to be published.

[28] C. Blake and C. Merz. (1998) UCI Repository of Machine Learning Databases. Dept. Inform. Comput. Sci., Univ. California, Irvine. [Online]. Available: http://www.ics.uci.edu/~mlearn/ MLRepository.html.

[29] S. K. Pal and D. D. Majumder, "Fuzzy sets and decision making approaches in vowel and speaker recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 625–629, 1977.

[30] O. L. Mangasarian and W. H. Wolberg, "Cancer diagnosis via linear programming," *SIAM News*, vol. 23, pp. 1–18, 1990.

Sushmita Mitra (M'99–SM'01) received the B.Sc. (Hons.) degree in physics and the B.Tech and M.Tech. degrees in computer science from the University of Calcutta, Calcutta, India, in 1984, 1987, and 1989, respectively, and the Ph.D. degree in computer science from Indian Statistical Institute, Calcutta, in 1995.

From 1992 to 1994, she was with the European Laboratory for Intelligent Techniques Engineering, Aachen, Germany, as a German Academic Exchange Service (DAAD) Fellow. Since 1995, she has been an Associate Professor with the Indian Statistical Institute, Calcutta, which she joined in 1989. She was a Visiting Professor at Meiji University, Japan, in 1999. Her research interests include data mining, pattern recognition, fuzzy sets, artificial intelligence, neural networks, and soft computing.

Dr. Mitra received the *National Talent Search Scholarship* (1978–1983) from the National Council for Educational Research and Training, India, the IEEE TNN Outstanding Paper Award in 1994, and the CIMPA-INRIA-UNESCO Fellowship in 1996. She is a co-author of the book *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing Paradigm* (New York: Wiley, 1999) and has about 50 research publications.

Kishori M. Konwar received the M.Sc. degree in physics from the Indian Institute of Technology, Kanpur, India, in 1998 and the M.Tech. degree in computer science from the Indian Statistical Institute, Calcutta, India, in 2000. Presently, he is pursuing the Ph.D. degree at the University of Connecticut, Storrs.

Sankar K. Pal (M'81–SM'84–F'93) received the M.Tech. and Ph.D. degrees in radio physics and electronics in 1974 and 1979, respectively, from the University of Calcutta, Calcutta, India. In 1982, he received another Ph.D. degree in electrical engineering, along with the DIC degree, from Imperial College, University of London, London, U.K.

He is a Professor and Distinguished Scientist at the Indian Statistical Institute, Calcutta. He is also the Founding Head of Machine Intelligence Unit. He worked at the University of California, Berkeley, and the University of Maryland, College Park, from 1986 to 1987 as a Fulbright Postdoctoral Visiting Fellow; at the NASA Johnson Space Center, Houston, TX, from 1990 to 1992 and 1994 as a Guest Investigator under the NRC-NASA Senior Research Associateship Program; and at the Hong Kong Polytechnic University, Hong Kong in 1999 and 2000 as a Visiting Professor. His research interests include pattern recognition, image processing, data mining, soft computing, neural nets, genetic algorithms, and fuzzy systems. He is a co-author/co-editor of eight books including *Fuzzy Mathematical Approach to Pattern Recognition* (New York: Wiley, 1986) and *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing* (New York: Wiley, 1999), and he has about 300 research publications. He was an Associate Editor of the *Pattern Recognition Letters*, the *International Journal on Pattern Recognition and Artificial Intelligence, Neurocomputing, Applied Intelligence, Information Sciences, Fuzzy Sets and Systems, Fundamental Informaticae*, the *International Journal on Image and Graphics*, and the *International Journal of Approximate Reasoning*.

Dr. Pal served as a Distinguished Visitor of the IEEE Computer Society (USA) for the Asia-Pacific Region from 1997 to 1999 for delivering lectures in Australia, Singapore, and China. He is a Fellow of the Third World Academy of Sciences, Italy, International Association for Pattern Recognition, USA, and all four National Academies for Science/Engineering in India. He received the 1990 S. S. Bhatnagar Prize (which is the most coveted award for a scientist in India), the 1993 Jawaharlal Nehru Fellowship, the 1993 Vikram Sarabhai Research Award, the 1993 NASA Tech Brief Award, the 1994 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award, the 1995 NASA Patent Application Award, the 1997 IETE—Ram Lal Wadhwa Gold Medal, the 1998 Om Bhasin Foundation Award, the 1999 G. D. Birla Award for Scientific Research, the 2000 Khwarizmi International Award (First winner) from the Islamic Republic of Iran, the 2001 Syed Husain Zaheer Medal from Indian National Science Academy, and the 2001 FICCI Award for Engineering and Technology from the Federation of Indian Chamber of Commerce and Industries, India. He was an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS from 1994 to 1998, a Member of the Executive Advisory Editorial Board, of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, and a Guest Editor of many journals, including IEEE COMPUTER.