NOVEL HYBRID RULE NETWORK BASED ON TS FUZZY RULES

Feng Guo, Lin Lin*, Xiaolong Xie*, Bin Luo**

Abstract: A novel hybrid rule network based on TS fuzzy rules is proposed to resolve the problems of fuzzy classification and prediction. The proposed model learns by using genetic algorithm and is able to cover the whole distribution regions of the samples. In the learning process: (1) fuzzy intervals of each dimension of the samples are partitioned evenly; (2) computing intervals (CIs) are established based on the even intervals; (3) linear weighted model of several normal probability distributions is used to describe the sample probability distribution on CIs; (4) membership degree of each CI is learnt to evaluate the importance of each CI, avoiding the problem that the optimal intervals are difficult to cover the original sample spaces; (5) dynamic rule selection mechanism is used to dynamically combine a small number of optimal rules linearly to achieve nonlinear approximation, reducing the computation load.

Three experiments are performed: the experiments on Iris and Mackey-Glass chaotic time series show that HRN can achieve satisfactory results and is more effective in terms of generalization ability, whereas the experiment on exhaust gas temperature demonstrates that HRN can predict the EGT of aero engine effectively.

Key words: *Hybrid rule network, dynamic rule selection mechanism, fuzzy classification, prediction*

Received: November 6, 2013 **DOI:** 10.14311/NNW.2015.25.005
Revised and accepted: February 20, 2015

1. Introduction

Because the ability to handle the complex problems with strong nonlinearity or high degree of uncertainty, the fuzzy models have been widely employed in many real fields, such as system identification [24], automatic control [3, 26], pattern recognition [26], data mining [1], prediction [10], etc. Fuzzy model is proved to be a powerful model in complex system modeling [25]. There are at least two different kinds of rule-based fuzzy models, including the Mamdani fuzzy model [21] and the Takagi-Sugeno (TS) fuzzy model [24]. Recently, the TS fuzzy model has got more attention than the Mamdani fuzzy model, because the TS fuzzy model can approximate the complex nonlinear system with fewer rules and higher accuracy [13, 15].

*Feng Guo, Lin Lin – Corresponding author, Xiaolong Xie, Bin Luo, School of Mechatronics Engineering, Harbin Institute of Technology, Harbin, China. E-mail: waiwaiyl@163.com

The design of fuzzy modeling based on the input–output data can be divided into two parts: structure identification and parameter identification. Structure identification is used to generate both the antecedent part and the consequent part of the fuzzy rules, whereas parameter identification is used to generate the parameters of membership functions in the antecedent part and of linear function in the consequent part. The methods proposed for structure identification consist of heuristics [24], clustering-based method [5], neural networks [2], etc. And the methods proposed for parameter identification cover least-square method [24], gradient descent [13], genetic algorithm (GA) [12], etc.

In fact, the performance of typical fuzzy model is greatly dependent on the prior knowledge of expertise, which is used to make decisions about how to partial the input data. But sometimes the prior knowledge cannot help to cluster the input data, and it probably generates a larger number of fuzzy rules. The redundant fuzzy rules can neither improve the accuracy nor enhance the explanatory. Therefore, the effective methods for fuzzy clustering should be proposed to enhance the performance of fuzzy model. The literature [27] has summarized three kinds of fuzzy clustering methods to determine the number of fuzzy rules: increasing or merging clusters from the initial clustering number, mountain clustering or subtractive clustering, and trade-off among several clustering validity indexes. But it is difficult to obtain the accurate and reasonable result directly using them, and they may decrease the robustness of the model. On the other hand, in the typical fuzzy model, once the clustering is completed, the antecedent part of fuzzy rules is confirmed, so the rules would keep the same pattern for all the input data.

To solve the problem, many researches were progressed. Some methods combine GA and TS fuzzy model, for instance, the TS-group method of data handing algorithm [29] employs rules fusion and rules combination to optimize the fuzzy rules. Some methods use support vector regression, for instance, the incremental smooth support vector regression [14] reduces the number of fuzzy rules through more forms of membership functions. Some researchers even presented new system form, for instance, the habitually linear evolving TS fuzzy model [16] controls the number of rules by an adaptive threshold on the error. However, these methods cannot promise the consequent part of the fuzzy rules to be appropriate for each input data.

In this research, a novel rule-based fuzzy network model named *hybrid rule network* (HRN) is proposed to solve the problems of fuzzy classification and prediction. HRN is mainly established by the nodes which represent fuzzy subspaces (different clusters of the input data) and are linked to others, with the thought of generating rules based on the TS fuzzy rule. There are three stages to use HRN after initializing the fuzzy subspaces for one sample. Firstly, HRN dynamically adjusts the features of all fuzzy subspaces by using the weights for the input vector of the sample. Secondly, in the rule-space consisting of all the probable rules, HRN builds the optimal rule set for the sample using the dynamic rule selection mechanism (DRSM). Thirdly, HRN obtains the input vector of the sample according to the optimal rule set. In the learning process, many samples are used, and HRN collects all the errors between the input vector and output of the samples to obtain the fitness. Once the fitness satisfies the threshold, the global optimal solution can be obtained.

This paper is organized as follows. In Section 2, the construction of HRN is described including structure analysis and node analysis. In Section 3, the learning algorithm of HRN is discussed including chromosome coding, DRSM, and fitness function. In Section 4, three experiments are presented: using Iris samples to validate the fuzzy classification capability, using the Mackey-Glass chaotic time series samples to validate the prediction capability, and using EGT samples to show the performance of HRN in the actual engineer field. In Section 5, conclusion remarks are summarized, and open problems are presented.

2. Construction of HRN

2.1 Structure of HRN

The samples used in HRN are described as follows:

$$\begin{gathered} \text{samples} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i, \mathbf{y}], \\ \mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ij}]^{\mathrm{T}}, \\ \mathbf{y} = [y_1, y_2, \ldots, y_j], \\ i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, m. \end{gathered} \tag{1}$$

where n is the dimension of the input vector of a sample, m is the capacity of the samples, x_{ij} is the i-th variable in the input vector of j-th sample and y_j is the output vector of the j-th sample.

As mentioned in Section 1, HRN is a network model established to solve the problems of fuzzy classification and prediction, which is based on the thought of forming the neural network nodes and constructing the rules of TS fuzzy model. The core of HRN is the nodes taking the information about the corresponding fuzzy set.

In a typical fuzzy model, a rule is defined as

$$\begin{aligned} \mathrm{R}^n: \quad & \text{If } x_1 \text{ is } A_1^n \text{ and } x_2 \text{ is } A_2^n \text{ and } \ldots \text{and } x_i \text{ is } A_i^n \\ & \text{then } y^n = a_1^n x_1 + \cdots + a_i^n x_i + b^n, \end{aligned} \tag{2}$$

where R^n is the n-th fuzzy rule, A_i^n is the fuzzy set of x_i in rule R^n, y^n is the output of R^n, a_i^n is the real coefficient corresponding to x_i, b^n is a compensation value. The structure of a typical TS fuzzy model is shown in Fig. 1.

For the j-th sample, $W_j^1, W_j^2, \ldots, W_j^n$ are rule weights for R^1, R^2, ..., and R^n, which are obtained according to some clustering method, and the calculated sample's output $y_j = W_j^1 y_j^1 + W_j^2 y_j^2 + \cdots + W_j^n y_j^n$ is a linear weighted sum by each rule's output.

As we can see in Fig. 1, the rules keep the same pattern and are composed of the same fuzzy sets for each sample. In another words, the rules are predetermined in the typical fuzzy model, so the generation of the rules may not show the rules-selected completeness which represents the capability to select the rules from all the rules built by different combinations of the fuzzy sets. Therefore, it is difficult to guarantee that each sample is calculated through the optimal rules, and thus the typical fuzzy model may not obtain the satisfactory result sometimes. To avoid

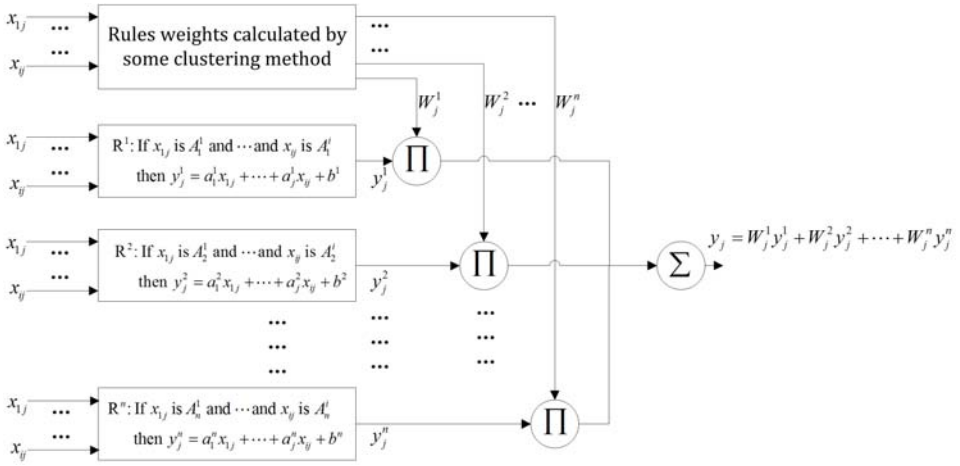

Fig. 1 *Structure of the typical TS fuzzy model.*

that deficiency, the structure of HRN is proposed, based on TS fuzzy rules. This structure is shown in Fig. 2.

A_i^k is the k-th fuzzy set represented as the interval of the i-th dimension of the current sample, $f_i^k(x)$ is the membership function on the interval A_i^k and whose parameter is x_{ij}, ω_i^k is the weight used to modify the membership degree calculated by $f_i^k(x)$, a_i^k is the real coefficient corresponding to interval A_i^k, the implication of which is same to that of a_i^n in Eq. (2), y_j^k is the j-th sample's output obtained by the k-th rule that built through the node taking interval A_i^k (the detail is discussed in Section 2.2), W^k is the weight of y_j^k, B is a compensation matrix, the element of which is used to generate b^n in Eq. (2).

Therefore, HRN is supposed to consist of a $k \times$ n nodes-matrix with links and a scalar matrix. Each node in the nodes-matrix is composed of an interval, a membership function, a weight, and a real coefficient.

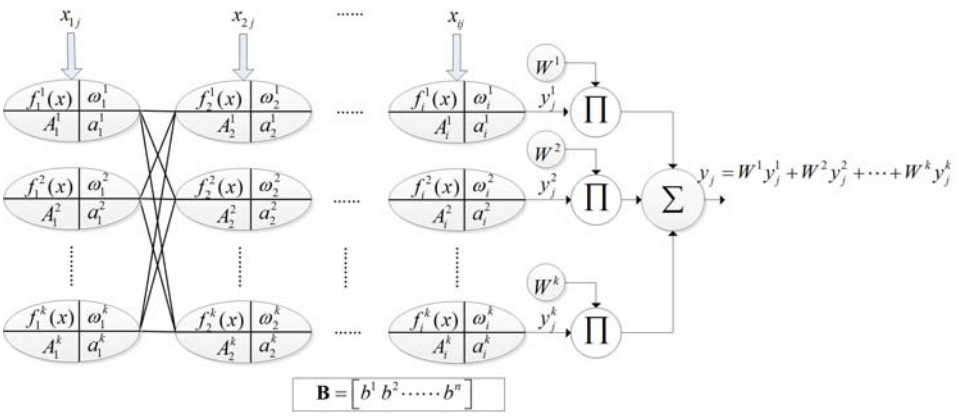

Fig. 2 *Structure of HRN.*

2.2 Analysis of a node

Two kinds of intervals exist: real interval (RI) and computing interval (CI). When initializing HRN, the number of RIs is confirmed, and the number of CIs is one greater than that of RIs. Therefore, if there are two RIs on certain dimension, then three CIs on the same dimension will be defined. The relationships between the intervals are shown in Fig. 3.

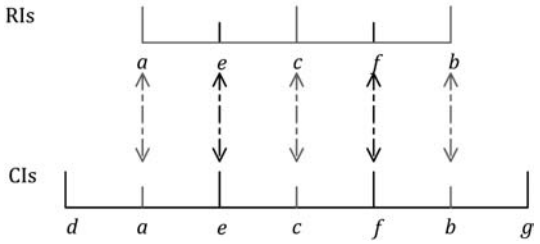

Fig. 3 *The different kind of intervals in the HRN: a is the minimal value of certain dimension of a sample, b is the maximal value, c is the middle value between a and b, e is the middle value between a and c, f is the middle value between c and b; a is also the middle value between d and e, b is also the middle value between f and g. The long vertical solid lines are the interval lines, and the arrow lines represent the corresponding values are equal. Therefore,* $[a, c]$ *and* $[c, b]$ *are RIs, whereas* $[d, e]$*,* $[e, f]$*, and* $[f, g]$ *are CIs.*

The way used to establish other CIs can be inferred even if there are more than 2 rules (RLs). The aim of establishing CIs is to guarantee that two or more different effective membership functions exist on one RI at least. The membership function $f_i^k(x)$ adopts the form of a univariate normal distribution model

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(\frac{-(x-\mu)^2}{2\sigma^2}), \tag{3}$$

where μ is the mean value of certain variable and σ is the standard deviation of certain variable. The background of central limit theorem theory suggests that there are many random variables in the objective reality, which are the results of comprehensive effects influenced by a large number of random factors independent of each other, and the effect of individual variable in the total influence is tiny. Those variables usually follow normal distribution, so the other probability distribution can utilize normal distribution as an approximation. Meanwhile, it has been proved that any kind of probability distribution can be approximated using a linear weighted sum of the finite number of normal distributions. Therefore, using the normal distribution to approximate the distribution of some variable in the input vector without knowing its distribution is reasonable. The membership function curves (MFCs) on CIs for certain dimension of the samples (3 CIs) are shown in Fig. 4.

As shown in Fig. 4, on a CI, three normal distribution models are summed through linear weighted sum to approximate the final membership degree. There-

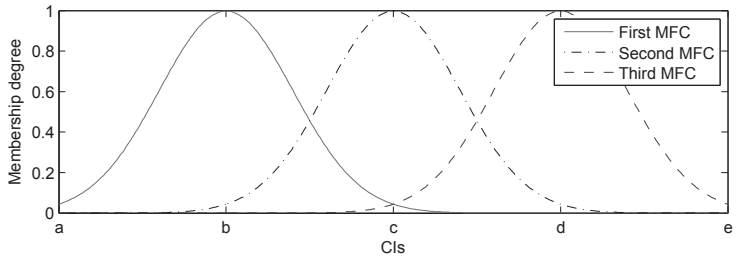

Fig. 4 *MFCs on CIs for certain dimension of the sample (3 CIs).*

fore, the final probability distribution model $f'(x)$ for that sample in a CI is obtained by

$$f'(x) = \omega_1 f_1(x) + \omega_2 f_2(x) + \omega_3 f_3(x), \tag{4}$$

where $f_1(x)$, $f_2(x)$, and $f_3(x)$ are the three normal distributions, and ω_1, ω_2, and ω_3 are the corresponding weights. Experience shows that the satisfactory result can be obtained if we use three or more normal distributions on the adjacent CIs to approximate an abnormal distribution on the current CI. Because the partitions based on the membership function are complete and the definition domain of the membership function has no boundary, the situation that there is no solution because of the variable out of definition domain does not occur.

Setnes and Roubos [23] pointed out that some methods obtaining the optimal fuzzy classification solution by modifying the intervals may generate a problem where the intervals are discontinuous after being learnt. There is a contrast between the original intervals and the modified intervals about MFCs shown in Fig. 5.

To make HRN have the capability to obtain the same optimal effects, the weight ω is introduced. The weight ω is used to recalculate the original membership degree as available membership degree (AMD) used to build a rule, which can optimize the CIs under the circumstance that the intervals do not need to be modified. For x_{ij}, its AMD is determined by either of

$$S_{ij}^k = \omega_i^k f_i^k\left(x_{ij}\right), \omega \in (0,1], \tag{5}$$

$$S_{ij}^k = f_i^k\left(\omega_i^k x_{ij}\right), \omega \in (0,1], \tag{6}$$

where S_{ij}^k is the AMD of x_{ij}calculated by $f_i^k(x)$ which is the membership function on A_i^k. In Eq. (5), ω is used to adjust the original membership degree directly. In Eq. (6), ω is used to adjust the original membership degree through multiplication by the corresponding x_{ij}. As a result, the performance of the model is improved because both of the two formulas are used to optimize AMD by adjusting one weight ω instead of two variables (μ and σ). The different effects of the above formulas are shown in Fig. 6 and Fig. 7, where the solid curve is the original membership degree calculated only by the membership function, and the dash curve is the AMD calculated by the membership function and the weight.

Analyzed from interval optimization, the weight ω is used to modify the original CI: if $S_{ij}^k > f_i^k\left(x_{ij}\right)$, then the result is similar to enlarging the CI, or the probability

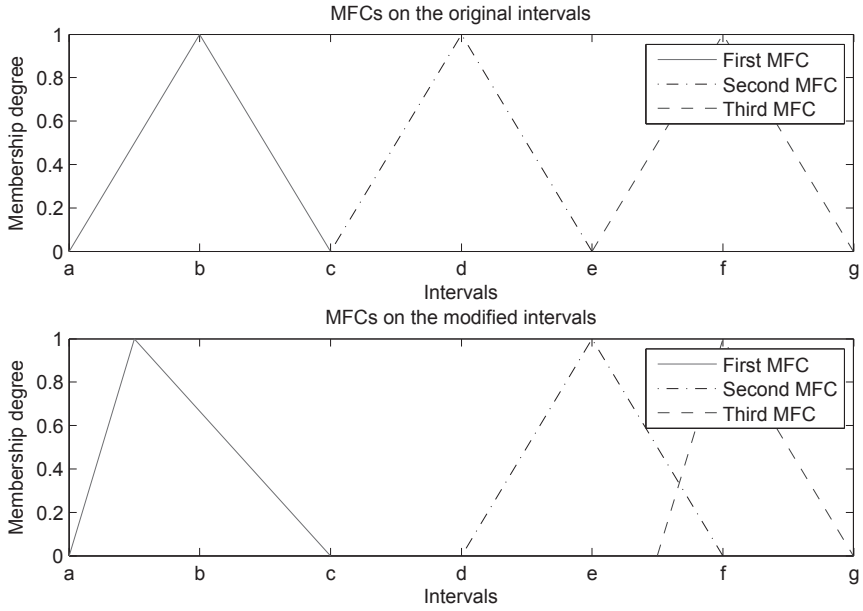

Fig. 5 *Contrast between the original intervals and the modified intervals.*

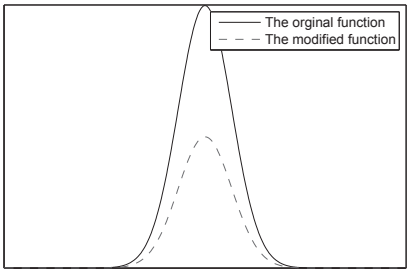

Fig. 6 *The effect of Eq. (5).*

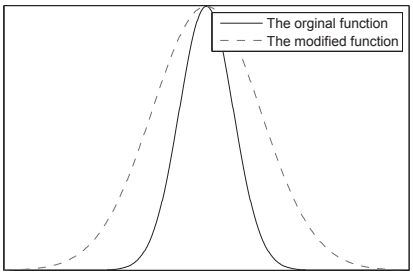

Fig. 7 *The effect of Eq. (6).*

that the point x_{ij} is located in the CI is enlarged; if $S_{ij}^k = f_i^k(x_{ij})$, then the result is similar to keeping the CI, or the probability that the point x_{ij} is located in the CI is not modified; if $S_{ij}^k < f_i^k(x_{ij})$, then the result is similar to shrinking the CI, or the probability that the point x_{ij} is located in the CI is reduced.

The pattern of the rule used in HRN is similar to that of the rule used in the typical TS fuzzy model. The rule in HRN is still based on local linearization, but it helps to obtain the global nonlinear result using the fuzzy reasoning method. Therefore, the implications of the real coefficient and the compensation matrix are similar to those in the typical fuzzy rules, but they are provided to enrich the selection pattern of the optimal rules for the samples. In fact, there may be no compensation values in a fundamental rule in HRN, but to obtain high performance, it is better if the rule has the same components as that in the typical TS fuzzy

model. And, the dimension of the compensation matrix usually is not less than the number of rules that is set before initializing the learning algorithm.

3. Learning algorithm of HRN

From the methodology point of view, many different kinds of methods can be used to approximate the global optimal solution in HRN, such as GA, ant colony optimization algorithm [7], particle swarm optimization algorithm [9, 17], etc. To simplify the solving logic, this research used GA which shows the global searching capability to approximate global optimal solution.

3.1 Chromosome coding

GA is used to search for a global optimal solution through optimizing the chromosomes in the population, and the optimal chromosome is a solution for the problem. The chromosome in HRN is coded by gray code [4] that is useful to search for the optimal solution in global, because the distance between two adjacent gray codes is 1.

The chromosome in HRN is composed of three parts: the weight matrix $\mathbf{W}$, the real coefficient matrix $\mathbf{A}$, and the compensation matrix $\mathbf{B}$. Each row in the matrix represents a variable of the corresponding kind, and the form of chromosome is shown as follows:

$$\mathbf{W} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}_{(i\times k,l)}, \quad \mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & \cdots & 1 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & 0 & \cdots & 1 & 0 \end{bmatrix}_{(i\times k,l)}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 1 & \cdots & 1 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 1 & \cdots & 0 & 0 \end{bmatrix}_{(n,l)}, \quad (7)$$

$$\text{chromosome}_c = (\mathbf{WAB})^{\mathrm{T}}.$$

Here, i is the number of the dimension for the sample, k is the number of CIs for the dimension, l is the number of gene digits for a coded value, n is the number of RLs, and c is the index of chromosome in the population.

3.2 DRSM

The typical TS fuzzy model generally does not show the rules-selected completeness, whose alternative rules are determined by the knowledge of the experts or some methods and that keep the same pattern for all the samples. Although, some methods can show a certain capability to select the rules, but cannot promise that the alternative rules are optimal for the samples. DRSM is a special method aiding the sample in selecting the optimal rule set from the whole rule-space consisting of different combinations by different node information, and DRSM can guarantee the rule diversity for certain sample. Given the dimension of the input vector of some sample is n, the number of CIs for each variable in the input vector is k, and then the number of alternative rules existing in HRN is k^n. Therefore, HRN shows rules-selected completeness, and makes the optimal rules can be selected from all the possible rules.

	Variable	MDOV of The Left Variable	Corresponding CIs
1	x_{1j}	$\left[\omega_1^1 f_1^1(x_{1j})\, \omega_1^2 f_1^2(x_{1j})\, \omega_1^3 f_1^3(x_{1j})\right]$	A_1^1, A_1^2, and A_1^3
2	x_{2j}	$\left[\omega_2^3 f_2^3(x_{2j})\, \omega_2^2 f_2^2(x_{2j})\, \omega_2^1 f_2^1(x_{2j})\right]$	A_2^3, A_2^2, and A_2^1
3	x_{3j}	$\left[\omega_3^3 f_3^3(x_{3j})\, \omega_3^1 f_3^1(x_{3j})\, \omega_3^2 f_3^2(x_{3j})\right]$	A_3^3, A_3^1, and A_3^2
4	x_{4j}	$\left[\omega_4^2 f_4^2(x_{4j})\, \omega_4^1 f_4^1(x_{4j})\, \omega_4^3 f_4^3(x_{4j})\right]$	A_4^2, A_4^1, and A_4^3

Tab. I *Supposed relationships.*

To build the optimal rule sets, membership degree ordered vector (MDOV) is proposed. MDOV is the vector consisting of membership degrees ordered by descent from different CIs of the variable in the input vector. Once MDOV for each variable is confirmed, the rule set of the sample can be also confirmed. Here is an example of a hypothetical HRN, with hypothetical nodes (interval symbol is omitted) matrix and compensation matrix, shown in Fig. 8.

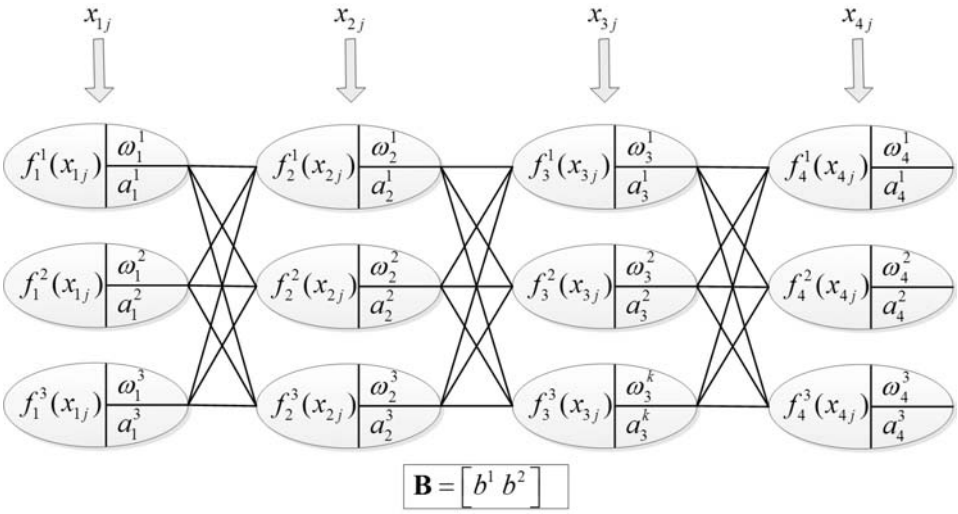

Fig. 8 *Hypothetical HRN. The dimension of the input vector of the sample is 4, the number of CIs is set to 3, and the number of RLs in the rule sets is set to 2.*

Then, the detail algorithm of DRSM for the j-th sample is described as follows:

Step 1: calculate the AMDs on the different CIs for each variable in the input vector and organize each variable's AMDs as each variable's MDOV. In the example, supposed that there are some relationships in Tab. I:

Step 2: Select the CIs and the corresponding real coefficient both contained in the nodes whose AMD is positioned at first in each MDOV (that means, the AMD is the maximal one) to build the first rule. So the first rule is shown in Fig. 9.

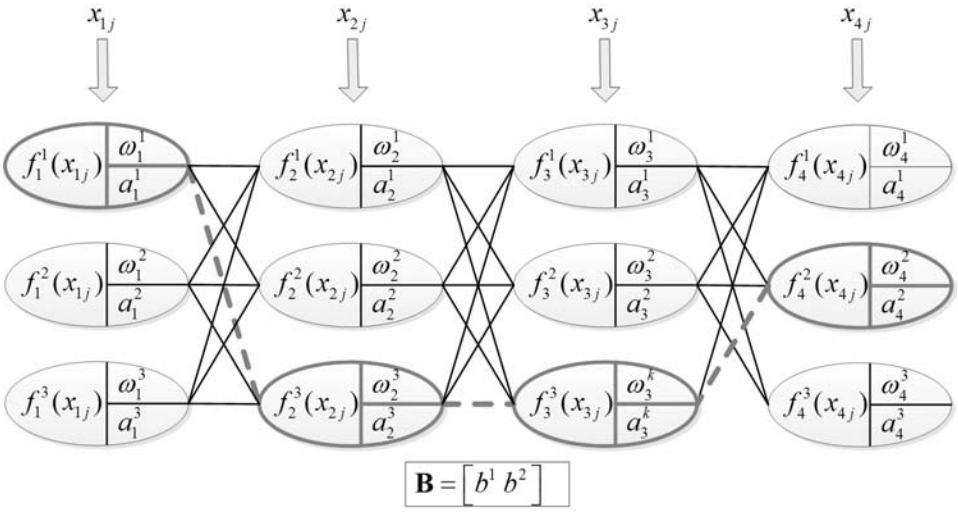

Fig. 9 *First rule. The nodes connected by the bold dash line are used to build the first rule.*

The first rule can be written as follows:

$$\begin{aligned} \mathrm{R}^1: \quad & \text{If } x_{1j} \text{ is } A_1^1 \text{ and } x_{2j} \text{ is } A_2^3 \text{ and } x_{3j} \text{ is } A_3^3 \text{ and } x_{4j} \text{ is } A_4^2 \\ & \text{then } y_j^1 = a_1^1\omega_1^1 f_1^1(x_{1j}) + a_2^3\omega_2^3 f_2^3(x_{2j}) + a_3^3\omega_3^3 f_3^3(x_{3j}) + a_4^2\omega_4^2 f_4^2(x_{4j}) + b^1. \end{aligned} \tag{8}$$

In the Eq. (8), the most effective nodes are selected because they are the first contributive for determining the output of the sample.

Step 3: Build the next optimal rule based on the previous rule. This step includes two stages. Given the order number of the next optimal rule is assumed to be t ($t > 1$). The first stage is to select both the CIs and the corresponding real coefficient contained in the nodes whose AMD is positioned at t-th in each MDOV to build the t-th possible rule. That is, the t-th contributed nodes are selected to build the t-th possible rule. In the example, if $t = 2$, then the second possible rule whose nodes linked by the bold dash line shown in Fig. 10 should be Eq. (9).

$$\begin{aligned} \mathrm{R}^{2,\mathrm{possible}}: \quad & \text{If } x_{1j} \text{ is } A_1^2 \text{ and } x_{2j} \text{ is } A_2^2 \text{ and } x_{3j} \text{ is } A_3^1 \text{ and } x_{4j} \text{ is } A_4^1 \\ & \text{then } y^2 = a_1^2\omega_1^2 f_1^2(x_{1j}) + a_2^2\omega_2^2 f_2^2(x_{2j}) + \\ & \qquad + a_3^1\omega_3^1 f_3^1(x_{3j}) + a_4^1\omega_4^1 f_4^1(x_{4j}) + b^2. \end{aligned} \tag{9}$$

In order to judge which node is more effective between the one positioned at t-th and the one positioned at $(t-1)$-th for certain variable, the rule similarity (RS) is proposed, which is then used as

$$A_i^m \vee A_i^n = \begin{cases} A_i^m & \text{if} \quad \dfrac{S_{ij}^m - S_{ij}^n}{S_{ij}^m} > \mathrm{RS}, \\ A_i^n & \text{if} \quad \dfrac{S_{ij}^m - S_{ij}^n}{S_{ij}^m} \leq \mathrm{RS}, \end{cases} \tag{10}$$

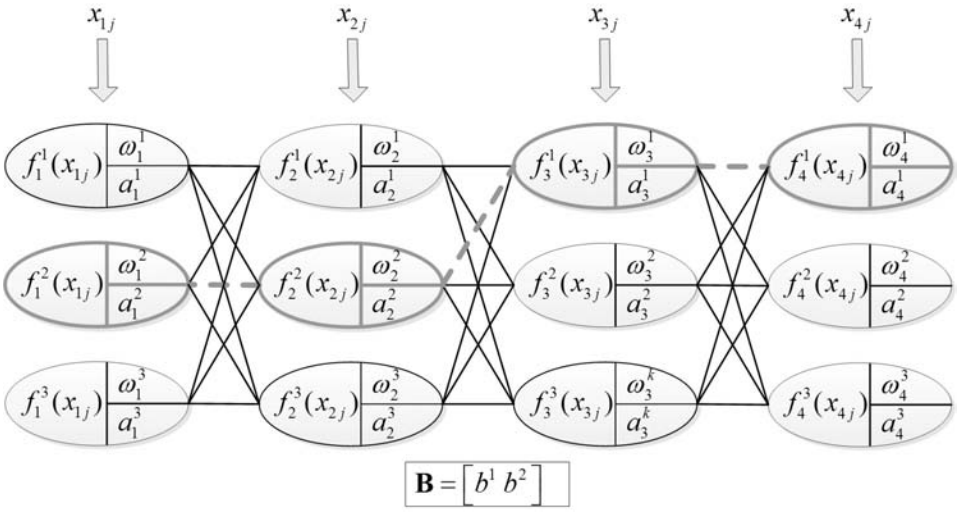

Fig. 10 *Building the next optimal rule (1).*

where $\vee$ is selection operator, A_i^m and A_i^n is the CIs of the i-th variable in the input vector of the$j$-thsample. Eq. (10) means that the selection operation (between the node positioned at t-th and the node positioned at $(t-1)$-th for each variable) is used to determine which one is more suitable. to build the next optimal rule before the next optimal rule is built. If the relationships exist in Eq. (11):

$$\begin{cases} (S_{1j}^1 - S_{1j}^2)/S_{1j}^1 \le \mathrm{RS} \\ (S_{2j}^3 - S_{2j}^2)/S_{2j}^3 > \mathrm{RS} \\ (S_{3j}^3 - S_{3j}^1)/S_{3j}^3 \le \mathrm{RS} \\ (S_{4j}^2 - S_{4j}^1)/S_{4j}^2 > \mathrm{RS} \end{cases}, \tag{11}$$

then, the second rule is revised to be Eq. (12), which is shown in Fig. 11.

$$\begin{aligned} &\mathrm{R}^2: \text{ If } x_{1j} \text{ is } A_1^2 \text{ and } x_{2j} \text{ is } A_2^3 \text{ and } x_{3j} \text{ is } A_3^1 \text{ and } x_{4j} \text{ is } A_4^2, \\ &\text{then } y_j^2 = a_1^2\omega_1^2 f_1^2(x_{1j}) + a_2^3\omega_2^3 f_2^3(x_{2j}) + a_3^1\omega_3^1 f_3^1(x_{3j}) + a_4^2\omega_4^2 f_4^2(x_{4j}) + b^2. \end{aligned} \tag{12}$$

The Step 3 can be repeated to build more rules and it can be synthesized as

$$\begin{cases} \mathrm{R}^1: y_j^1 = \sum_{i=1}^n |a_i^t\omega_i^t \ \ f_i^t(x_{ij})|_{t=1} + b^1, \\ \mathrm{R}^t: y_j^t = \sum_{i=1}^n |a_i^t\omega_i^t \ \ f_i^t(x_{ij})|_t \vee \left|a_i^{t-1}\omega_i^{t-1} \ \ f_i^{t-1}(x_{ij})\right|_{t-1} + b^t, t > 1. \end{cases} \tag{13}$$

where t is the position number in the MDOV for x_{ij} and $|a_i^t\omega_i^t \ \ f_i^t(x_{ij})|_t$ is the product by the corresponding values in the node whose AMD is positioned at t-th in the MDOV for x_{ij}.

Therefore, HRN can use DRSM to select the finite number (set by the number of RLs) of the optimal rules to obtain the output of certain sample; however, the TS fuzzy model must use all the predetermined rules for the same purpose. If all the rules have been confirmed by Eq. (13), the rule set for the j-th sample is established.

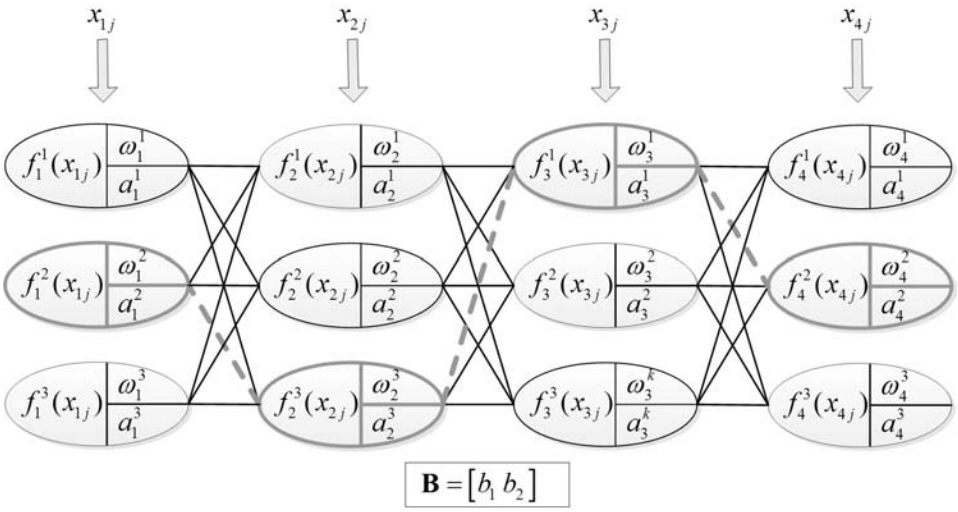

Fig. 11 *Building the next optimal rule (2).*

3.3 Fitness function

If the number of RLs is set to t, the output of the sample is

$$y_j = W^1 y_j^1 + W^2 y_j^2 + \cdots + W^t y_j^t. \tag{14}$$

That means the output of HRN is the same as that of the typical TS fuzzy model (see Eq. (4)), and the rule weight W is the product by corresponding AMDs of the rule, which can be described as

$$W^t = \prod S_k, S_k \subset \mathrm{R}^t, k = 1, 2, \cdots, n. \tag{15}$$

Here, W^t is the rule weight for the t-th rule, S_k is the corresponding AMD existing in the t-th rule and n is the dimension of the input vector of the sample.

Hence, least mean square or mean relative estimation error or root-mean-square error (RMSE) can be used as the fitness function based on Eq. (14).

3.4 Whole algorithm

If the fitness function uses RMSE, then the description of operating algorithm for HRN is as follows:

Step 1: Confirm the parameters, such as RIs, RLs, RS, the convergence condition (here is maximal RMSE), the maximum iterations (MI), and some parameters only for GA.

Step 2: Normalize all the samples, and set the current iteration to be 0.

Step 3: Initialize the chromosome shaped as the Eq. (7) in the population.

Step 4: Add the current iteration with 1.

Step 5: For each chromosome:

Step 5.1: Set current chromosome.
Step 5.2: For each sample:
Step 5.2.1: Set current sample.
Step 5.2.2: Obtain AMDs of current sample by Eq. (5) or (6).
Step 5.2.3: Establish MDOV of each variable of current sample.
Step 5.2.4: Build the optimal rule set for current sample by Eq. (13).
Step 5.2.5: Obtain the output of current sample by Eq. (15).
Step 5.2.6: Obtain the error between the output and the real value.
Step 5.3: Obtain RMSE of current chromosome.

Step 6: Obtain the best chromosome according to its RMSE. If its RMSE satisfies maximal RMSE, the algorithm stops, and recognize the best chromosome as optimal solution; otherwise, the algorithm continues.

Step 7: Judge whether MI is reached. If reached, the algorithm stops and returns the best chromosome; otherwise, the algorithm continues.

Step 8: Complete the selection, crossover, and mutation manipulation for chromosomes in the population to generate new population. And return to Step 4.

The algorithm flowchart is shown in Fig. 12 to illustrate the algorithm in details.

In each process of obtaining error of an sample, the time is mainly related to $(\mathrm{RIs}+1)\times D\times \mathrm{RLs}$, where D is the sample dimension. In each process of obtaining fitness of an chromosome, the time is proportional to $n\times(\mathrm{RIs}+1)\times D\times \mathrm{RLs}$, where n is the capacity of all the samples. For the population, C chromosomes exist, so the time is related to $n\times C\times(\mathrm{RIs}+1)\times D\times \mathrm{RLs}$. For the total of Iiterations, the entire learning time is $n\times C\times(\mathrm{RIs}+1)\times D\times \mathrm{RLs}\times I$. Because C, RIs, and RLs are set for the algorithm, and D is typically far less than n and I, the time complexity of the algorithm is linear, $T(n)=O(n)$ if $n\gg I$ or $I\gg n$, or at worst quadratic, $T(n)=O\left(n^2\right)$.

4. Experiments and analyses

HRN can be used to either classify or predict the unresolved samples. We will first study the fuzzy classification capability using Iris samples, on the basis of analyzing the influence of different parameters and the generalization ability, an experiment for one versus others was used to investigate the fuzzy classification capability. Second, we will study the prediction capability using the Mackey-Glass chaotic time series, both the prediction feasibility and the generalization ability were investigated based on the typical parameters. At last, to show the practicability, the exhausted gas temperature (EGT) of a particular aero engine was used in the experiment for prediction.

Two applications are programed for the experiments; the one shown in Fig. 13 is employed for fuzzy classification, whereas the one shown in Fig. 14 is employed for predication.

4.1 Experiment on fuzzy classification

A set of 150 Iris samples has been used in this experiment. Data preprocessing operations included: (i) take the four eigenvalue as input vector for one sample;

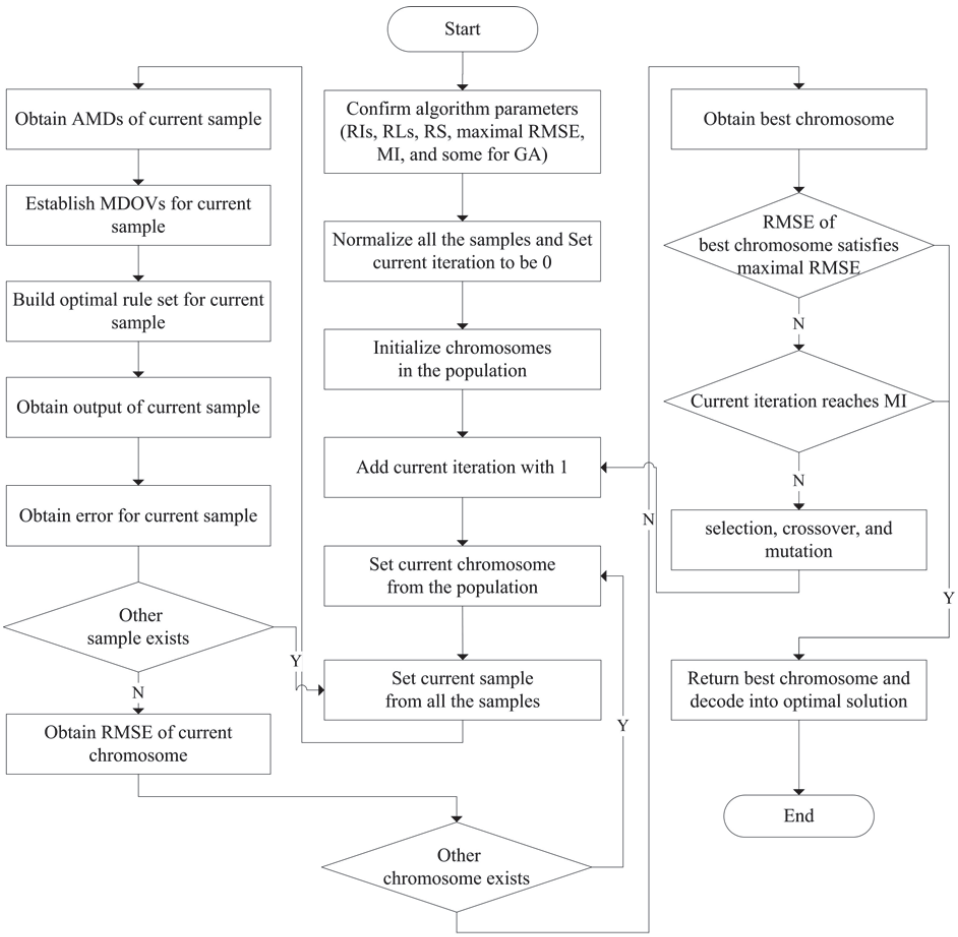

Fig. 12 *Algorithm flowchart.*

(ii) Iris setosa, Iris versicolor, and Iris virginica are respectively presented by 1, 2, and 3, and then gathered as output vector for the corresponding sample; (iii) all samples were normalized.

During the process of normalizing the samples, the fuzzy classification centers were generated by mapping the output vectors into the interval $(0, 1)$ evenly. Because three categories exist in Iris samples, the fuzzy classification centers are respectively 0.1667, 0.5, and 0.8333, and the corresponding intervals are $(0, 0.3334]$, $(0.3334, 0.6668)$, and $[0.6668, 1)$. When the obtained output falls into certain interval, the category of this sample can be confirmed.

4.1.1 Influence of different parameters

Randomly, 120 samples were selected as training samples and 30 samples as testing samples. The experiments were divided into four groups, each group contained

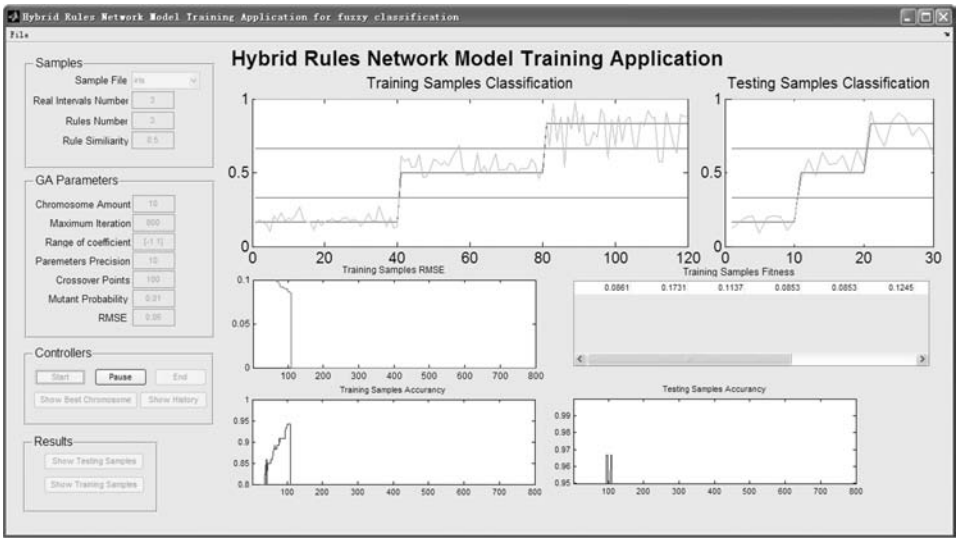

Fig. 13 *Application for fuzzy classification.*

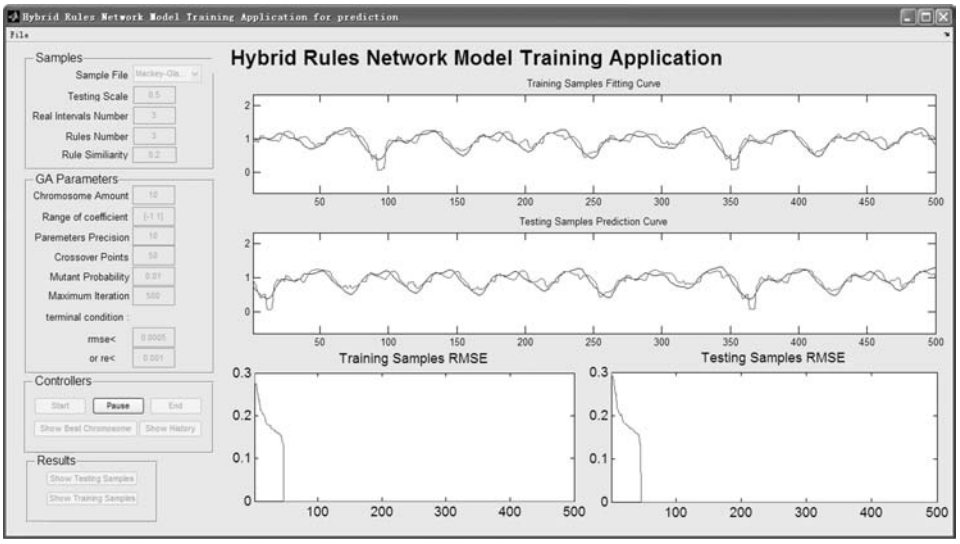

Fig. 14 *Application for predication.*

four fundamental experiments, and each experiment was repeated 10 times. The same samples including training samples and testing samples were used for all the experiments, and maximal RMSE was used as the convergence condition.

Group 1: Verify the effect of RI. Parameters setting: RLs = 3, RS = 0.5, and maximal RMSE < 0.06. The result is shown in Tab. II.

Conclusion: RI can affect the convergence rate of training samples, and the classification accuracy of testing samples is reduced if RIs were overestimated. For

RIs	Times of convergence	Average iterations	Average accuracy of training samples	RMSE of testing samples	Average accuracy of testing samples
2	6	386	97.0833%	0.05168	99.0%
3	3	367	96.3333%	0.04974	99.0%
4	6	440	97.0833%	0.05189	99.0%
5	6	384	97.1667%	0.05256	98.0%

Tab. II *Effects of different RIs*

RLs	Times of convergence	Average iterations	Average accuracy of training samples	RMSE of testing samples	Average accuracy of testing samples
1	3	494	95.5833%	0.045282	99.6%
2	4	282	96.6667%	0.043836	99.0%
3	3	367	96.3333%	0.049736	99.0%
4	8	339	97.0833%	0.050039	99.0%

Tab. III *Effects of different RLs*

fuzzy classification problem, the other experiments completed have also shown that better fuzzy classification results can be achieved in HRN if RIs is set to 2 or 3; the convergence rate can be boosted if certain better initial values for GA are set.

Group 2: Verify the effect of RLs. Parameters setting: RIs = 3, RS = 0.5, and maximal RMSE< 0.06. The result is shown in Tab. III.

Conclusion: the testing samples can deviate from the classification centers if RLs is overestimated, and increasing RLs is not guaranteed to be useful for improving the classification accuracy. For fuzzy classification, the other experiments completed have also shown that if RLs is set to about 2, the distance between testing samples and classification center would become closer.

Group 3: Verify the effect of RS. Parameters settings: RIs = 3, RLs = 3, and maximal RMSE < 0.06. The result is shown in Tab. IV.

Conclusion: If RSs is lower, HRN can reduce the distance between testing samples and classification center to a certain extent; if RSs is higher, HRN can reduce the accuracy of the test samples. When analyzed mathematically, if the

RSs	Times of convergence	Average iterations	Average accuracy of training samples	RMSE of testing samples	Average accuracy of testing samples
0.2	6	347	96.7%	0.046214	99.3%
0.4	6	388	96.7%	0.048271	99.7%
0.6	6	319	96.3%	0.049569	98.7%
0.8	4	378	96.5 %	0.052658	97.3%

Tab. IV *Effects of different RSs*

maximal RMSE	Times of convergence	Average iterations	Average accuracy of training samples	RMSE of testing samples	Average accuracy of testing samples
0.03	0	none	96.7%	0.05275	99.0%
0.05	0	none	96.8%	0.05342	98.7%
0.07	10	310.2	96.2%	0.06132	97.7%
0.09	10	63.5	92.5%	0.08217	95.3%

Tab. V *Effects of different maximal RMSE*

Algorithms	This paper	From [19]	From [11]	From [18]	From [22]
Accuracy	96.67%	96.00%	96.00%	95.33%	96.67%

Tab. VI *Results through different algorithms.*

difference of rules is large, then the diversity of rules can be fully reflected. It is also shown in the experiment that even if the distance between testing sample and classification center is reduced, the average accuracy of testing samples cannot be always improved.

Group 4: Verify the effect of the convergence condition, which was maximal RMSE of training samples. Parameters setting: RIs = 3, RLs = 0.5, and RS = 0.5. The result is shown in Tab. V.

Conclusion: The lower the maximal RMSE of training samples is, the higher the rate of convergence. If the maximal RMSE of training samples is increased, the RMSE of testing samples is also increased, while the average accuracy of testing samples can be reduced. Although the small maximal RMSE of training samples may cause HRN to not converge in the MI, better results can be still obtained.

4.1.2 Experiment on leave-one-out cross-validate

To obtain the more objective classification result, avoiding obtaining the better result due to using the better samples, the experiment on leave-one-out cross-validate was used. In this experiment: RIs = 2, RLs = 2, RS = 0.2, and maximal RMSE < 0.06. The application was modified in the aspect of graphic user interface shown in Fig. 15. To reduce the effect of initial values used for GA, the experiment was repeated many times, and the better result is shown in Fig. 16.

A comparison of results from different algorithms is presented in Tab. VI. We can see that the proposed HRN has the capability to resolve problems of fuzzy classification. Moreover, the results suggest that if nonlinear rule sets are used in HRN, the average accuracy of testing samples can be improved further more.

4.1.3 Analysis for robustness

To validate the robustness of HRN, the should be first combined with additional white noise, and then use a better parameters group. A sample of the original and

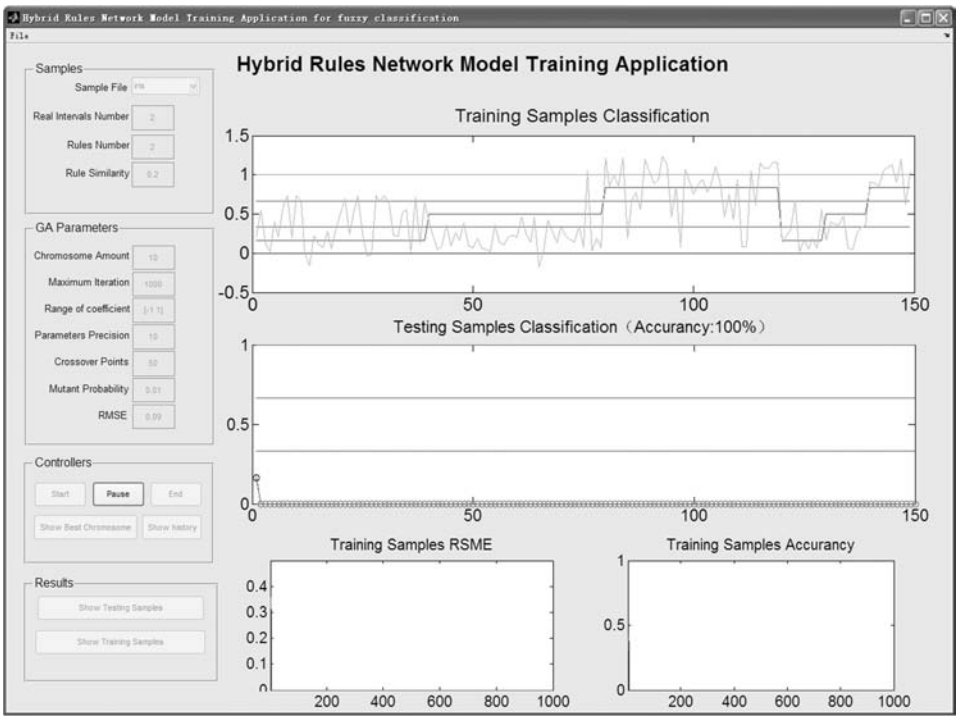

Fig. 15 *Modified application.*

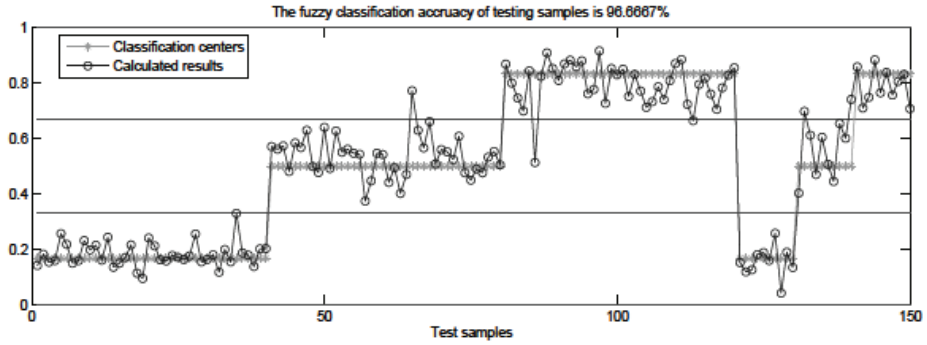

Fig. 16 *Result of Iris experiment of one versus others.*

noisy data is shown in Tab. VII, and in the learning process, the noisy data was used instead of original data.

Parameters setting: RIs = 2, RLs = 2, RS = 0.2, and maximal RMSE < 0.06, then the accuracy of test samples was 95.33%. Therefore, the generalization ability of HRN for fuzzy classification was proved to be satisfactory.

Original data	Noised data	Original data	Noised data	Original data	Noised data
5.700	5.732	3.500	3.433	1.500	1.546
5.300	5.287	3.100	3.125	1.300	1.278
4.900	4.856	2.900	2.873	0.400	0.399
4.600	4.633	2.800	2.760	0.200	0.203

Tab. VII *Noised data in the samples*

RIs	RLs	RSs	Convergence iteration	RMSE of training samples (average relative errors)	RMSE of testing samples (average relative errors)
2	2	0.2	3942	0.046882 (4.4%)	0.052772 (5.1%)
3	3	0.2	3007	0.034105 (3.2%)	0.036921 (3.4%)
4	4	0.2	3082	0.030739 (2.9%)	0.032018 (3.1%)

Tab. VIII *Experiment parameters and results*

4.2 Experiment on prediction

The Mackey-Glass chaotic time series was used to validate the prediction capability of HRN. The time series used in the experiment was generated by the differential delay equation

$$x(t+1) = 0.9x(t) + \frac{0.2x(t-17)}{1+x^{10}(t-17)}. \tag{16}$$

Among the points,$x(t-18)$, $x(t-12)$, $x(t-6)$, and $x(t)$ were used to construct the input vector, $x(t+6)$ was used to construct the output, and combine the input vector and output vector as one sample. The capacity of samples was 1000 and the first 500 samples were used as training samples while the remaining 500 samples were used as testing samples.

4.2.1 Fundamental experiment analysis

After normalizing the samples, MI was set to 4000. The result based on typical parameters is show in Tab. VIII.

Conclusion: for the experiment on the Mackey-Glass chaotic time series, the gained experience is different from that of fuzzy classification experiments. The other experiments completed have also shown that if RIs = 4, RLs = 4, and RS = 0.2, then the experiments about other kinds of chaotic time series can obtain the better results.

4.2.2 Experiments comparison

Parameters setting: RIs = 4, RLs = 4, and RS = 0.2. The result is shown in Fig. 17.

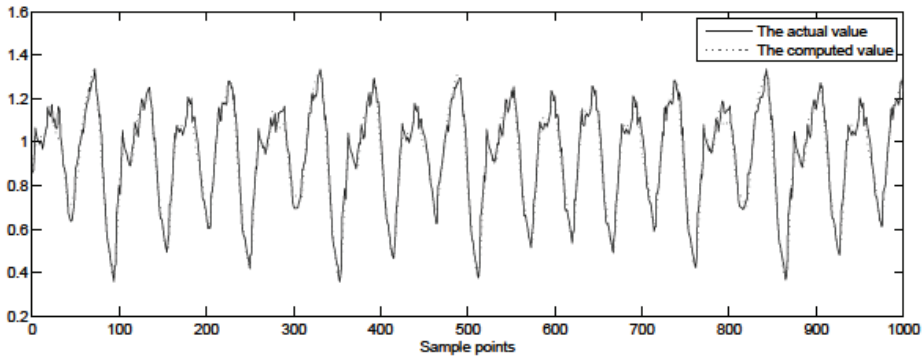

Fig. 17 *Result of the Mackey-Glass chaotic time series experiment.*

Algorithm	RMSE of training samples	RMSE of testing samples	Average relative errors
From [6]	0.00043	0.00041	-
From [20]	0.014	0.009	-
From [8]	0.024	0.025	-
This paper	0.031	0.032	3.1%

Tab. IX *Results through different algorithms. Note: "-" indicates that relevant studies did not provide the corresponding information.*

The results obtained from different algorithms were compared, and the result is shown in Tab. IX. We can see that HRN represents a reasonable capability in terms of resolving the chaotic time series problem.

4.2.3 Analysis for robustness

To validate the robustness capability of HRN, the samples should be first combined with additional white noise, and then use a better parameters group. In the learning process, the noisy data was used instead of original data. A sample of the original and noisy data is shown in Tab. X.

If RIs = 4, RLs = 4, and RS = 0.2, then the result was as follows: at the 4565

Original data	Noised data	Original data	Noised data	Original data	Noised data
1.081111	1.090000	0.752387	0.760000	0.869055	0.857000
0.982946	0.975000	0.520449	0.530000	0.376558	0.362000
1.001452	1.010000	0.628194	0.622000	1.224462	1.210000
0.762473	0.757000	0.522397	0.512000	0.856656	0.847000

Tab. X *Noised data in the samples.*

Time [s]	EGT [°C]	Time [s]	EGT [°C]	Time [s]	EGT [°C]	Time [s]	EGT [°C]
t(1)	533	t(6)	563	t(11)	548	t(16)	526
t(2)	538	t(7)	525	t(12)	556	t(17)	541
t(3)	538	t(8)	528	t(13)	521	t(18)	526
t(4)	536	t(9)	582	t(14)	543	t(19)	523
t(5)	537	t(10)	545	t(15)	546	t(20)	526

Tab. XI *First 20 original data points.*

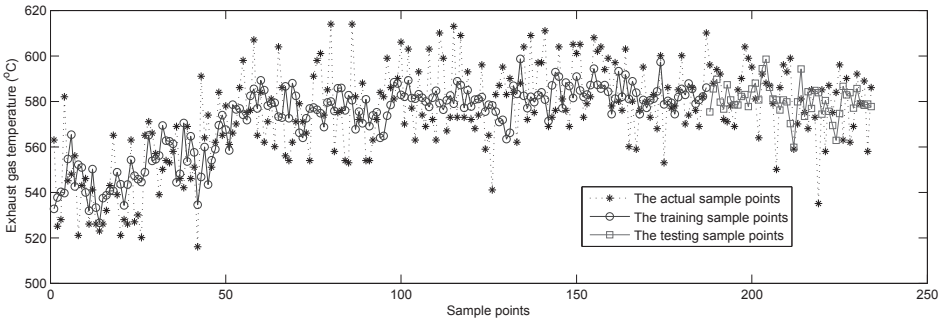

Fig. 18 *Result of EGT training.*

iterations, the RMSE was 0.036875 and average relative errors was 3.3982% for the training samples; and for the testing samples, the RMSE was 0.0392 and average relative errors was 3.5339%. Therefore, the better generalization capability can be still reflected even if the samples are noisy.

4.3 Experiments on practicability

Exhaust gas temperature, EGT, is an important parameter which is the characterization of aero engine health status and determines the availability of the engine, so there is a great practical application value in predicting the aircraft engine EGT in a future period of time accurately. The data in this experiment was the chaotic time series composed of EGT from a particular aero engine of China International Airlines Company. The first 20 original data points are shown in Tab. XI.

Among the data, $x\,(t-4)$, $x\,(t-3)$, $x\,(t-2)$, $x\,(t-1)$ and $x\,(t-1)$ were used to construct the input vector, $x\,(t+1)$ was used to construct the output, and combined the input vector and output as one sample. The whole dataset contained 234 samples and we took the first 192 samples as training samples, whereas the remaining 42 samples as testing samples. Using the appropriate parameters, the result is shown in Fig. 18.

Generally speaking, removing the noise for the data is implemented before training in this experiment. But, as known in Fig. 18, the data curve of the result is smoother than the original one, which means HRN combines the two operations.

Algorithm	RMSE of testing data	Average relative error of testing data
From [28]	-	1.8%
This paper	13.8910	2.0%
Standard BP neural network	20.3545	2.9%

Tab. XII *Experiment result and the results through other algorithms. Note: "-" indicates that relevant studies did not provide the corresponding information.*

Therefore HRN can reduce the time for removing the noise, and then improve on the performance of prediction. The results from different algorithms are shown in Tab. XII.

Therefore, after comparing with other algorithms, HRN has shown certain advantages over the other algorithms for predicting EGT of a particular aero engine of China International Airlines Company.

5. Conclusion remarks and open problems

In this paper we introduced a hybrid rule network (HRN), which is a network model consisting of nodes with links. DRSM is a mechanism proposed to make HRN behave the capability to search the optimal rule set from rule space dynamically, and to make HRN use the optimal rule set to obtain the sample's output. The learning algorithm proposed can make HRN able to approximate the global optimal solution. Experiments showed that HRN has good performance in fuzzy classification and prediction, and that HRN has stronger generalization ability.

At present, the research of HRN is at the initial stage, there are many aspects to be further researched. These are, for example: (i) how to avoid the redundant intervals dynamically; (ii) how to avoid the redundant rules dynamically; (iii) how to avoid the excessive homogenization or heterogenization for the rules dynamically; (iv) how to modify the range of real coefficient and compensation matrix to optimize the capability to approximate optimal solution; and (v) how to improve on the performance of HRN by using other algorithms.

Acknowledgement

The authors are grateful to the anonymous reviewers for their very helpful comments and constructive suggestions with regard to this paper. This paper is supported by National Natural Science Foundation of China (Grant No. 51075083), and also by the major project of National Defense Foundation of China.

References

[1] ANGELOV P.P., ZHOU X. Evolving Fuzzy-Rule-Based Classifiers From Data Streams. IEEE Transactions on Fuzzy Systems. 2008, 16(6), pp. 1462-1475, doi: 10.1109/TFUZZ.2008.925904.

[2] AZEEM M.F., HANMANDLU M., AHMAD N. Structure identification of generalized adaptive neuro-fuzzy inference systems. IEEE Transactions on Fuzzy Systems. 2003, 11(5), pp. 666-681, doi: 10.1109/TFUZZ.2003.817857.

[3] BUCKLEY J.J. Sugeno type controllers are universal controllers. Fuzzy Sets and Systems. 1993, 53, pp. 299-303, doi: 10.1016/0165-0114(93)90401-3.

[4] CARUANA R.A., DAVID SCHAFFER J., ESHELMAN L.J. Using multiple representations to improve inductive bias: gray and binary coding for genetic algorithms. In: A.M. SEGRE, ed. Proceedings of the Sixth International Workshop on Machine Learning, San Francisco, USA. CA: Morgan Kaufmann, 1989, pp. 375-378, doi: 10.1016/B978-1-55860-036-2.50095-3.

[5] CHIU S.L. Fuzzy model identification based on cluster estimation. Intelligent and Fuzzy Systems. 1994, 2, pp. 267-278, doi: 10.3233/IFS-1994-2306.

[6] CHOI J.-N., OH S.-K., PEDRYCZ W. Identification of fuzzy models using a successive tuning method with a variant identification ratio. Fuzzy Sets and Systems. 2008, 159(21), pp. 2873-2889, doi: 10.1016/j.fss.2007.12.031.

[7] DORIGO M., BIRATTARI M., STUTZLE T. Ant colony optimization. IEEE Computational Intelligence Magazine. 2006,1(4), pp. 28-39, doi: 10.1109/MCI.2006.329691.

[8] DUAN J.-C., CHUNG F.-L. Multilevel fuzzy relational systems: structure and identification. Soft Computing. 2002, 6, pp. 71-86, doi: 10.1007/s005000100144.

[9] EBERHART R.C., SHI Y. Evolutionary computation implementations. In: R.C. EBERHART AND Y. SHI, eds. Computational Intelligence. Burlington: Morgan Kaufmann, 2007, pp. 95-143, doi: 10.1016/B978-155860759-0/50004-4.

[10] GAO Y., ER M.J. NARMAX time series model prediction: feedforward and recurrent fuzzy neural network approaches. Fuzzy Sets and Systems. 2005, 150(2), pp. 331-350, doi: 10.1016/j.fss.2004.09.015.

[11] GOU B., HUANG X. The methods of multiclass classifiers based on SVM. Journal of Data Acquisition & Processing. 2006, 21(3), pp. 334-339.

[12] HOLLAND J.H. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Oxford, England: U Michigan Press, 1975.

[13] JANG J.-S.R. ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics. 1993, 23(3), pp. 665-685, doi: 10.1109/21.256541.

[14] JI R., YANG Y., ZHANG W. Incremental smooth support vector regression for Takagi-Sugeno fuzzy modeling. Neurocomputing. 2014, 123, pp. 281-291, doi: 10.1016/j.neucom.2013.07.017.

[15] JUANG C.-F., LIN C.-T. An online self-constructing neural fuzzy inference network and its applications. IEEE Transactions on Fuzzy Systems. 1998, 6(1), pp. 12-32, doi: 10.1109/91.660805.

[16] KALHOR A., ARAABI B.N., LUCAS C. A new systematic design for Habitually Linear Evolving TS Fuzzy Model. Expert Systems with Applications. 2012, 39(2), pp. 1725-1736, doi: 10.1016/j.eswa.2011.08.085.

[17] KENNEDY J., EBERHART R.C., SHI Y. The Particle Swarm. In: J. KENNEDY AND R.C.E. SHI, eds. Swarm Intelligence. San Francisco: Morgan Kaufmann, 2001, pp. 287-325, doi: 10.1016/B978-155860595-4/50007-3.

[18] LI J., ZHENG Y., SHEN S. A classification rule acquisition and reasoning algorithm based on the fuzzy regional distribution. Chinese J. Computers. 2008, 31(6), pp. 934-941.

[19] LIN L., DING G. A Multiple Classification Method Based on the Cloud Model. Neural Network World. 2010, 20(5), pp. 651-666.

[20] MAGUIRE L.P., et al. Predicting a chaotic time series using a fuzzy neural network. Information Sciences. 1998, 112, pp. 125-136, doi: 10.1016/S0020-0255(98)10026-9.

[21] MAMDANI E.H., ASSILIAN S. An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies. 1975, 7(1), pp. 1-15, doi: 10.1016/S0020-7373(75)80002-2.

[22] RAMANAN A., SUPPHARANGSAN S., NIRANJAN M. Unbalanced Decision Trees for Multi-class Classification. Second International Conference on Industrial and Information Systems (ICIIS 2007), Penadeniya, Sri Lanka, Ceylon. IEEE, 2007, pp. 291-294, doi: 10.1109/ICIINFS.2007.4579190.

[23] SETNES M., ROUBOS H. GA-fuzzy modeling and classification: Complexity and Performance. IEEE Transactions on Fuzzy Systems. 2000, 8(5), pp. 509-522, doi: 10.1109/91.873575.

[24] TAKAGI T., SUGENO M. Fuzzy Identification of Systems and Its Applications to Modeling and Control. IEEE Transactions on Systems, Man, and Cybernetics. 1985, 15, pp. 116-132, doi: 10.1109/TSMC.1985.6313399.

[25] WANG L., LANGARI R. Complex systems modeling via fuzzy logic. IEEE Transactions on Systems, Man, and Cybernetics, part B-cybernetics. 1996, 26(1), pp. 100-106, doi: 10.1109/3477.484441.

[26] WANG L.X. Adaptive Fuzzy Systems and Control: Design Stability Analysis. Upper Saddle River, NJ, USA: Prentice Hall Professional Technical Reference, 1994.

[27] WANG N., YANG Y. A fuzzy modeling method via Enhanced Objective Cluster Analysis for designing TSK model. Expert Systems with Applications. 2009, 36(10), pp. 12375-12382, doi: 10.1016/j.eswa.2009.04.048.

[28] ZHONG S., LEI D., DING G. Convolution Sum Discrete Process Neural Network and Its Application in Aeroengine Exhausted Gas Temperature Prediction. Acta Aeronautica et Astronautica Sinica. 2012, 33(3), pp. 438-445.

[29] ZHU B., et al. A GMDH-based fuzzy modeling approach for constructing TS model. Fuzzy Sets and Systems. 2012, 189(1), pp. 19-29, doi: 10.1016/j.fss.2011.08.004.