



Hierarchical fuzzy design by a multi-objective evolutionary hybrid approach

Yosra Jarraya¹ · Souhir Bouaziz¹ · Adel M. Alimi¹ · Ajith Abraham^{2,3}

Published online: 12 June 2019

© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

This paper presents a new tree hierarchical representation of type-2 fuzzy systems. The proposed system is called the type-2 hierarchical flexible beta fuzzy system (T2HFBFS) and is trained based on two-phase optimization mechanism. The first optimization step is a multi-objective structural learning phase. This phase is based on the multi-objective extended immune programming algorithm and aims to obtain an improved T2HFBFS structure with good interpretability-accuracy trade-off. The second optimization step is a parameter tuning phase. Using a hybrid evolutionary algorithm, this phase allows the adjustment of antecedent and consequent membership function parameters of the obtained T2HFBFS. By interleaving the two learning steps, an optimal and accurate hierarchical type-2 fuzzy system is derived with the least number of possible rules. The performance of the system is evaluated by conducting case studies for time series prediction problems and high-dimensional classification problems. Results prove that the T2HFBFS could attain superior performance than other existing approaches in terms of achieving high accuracy with a significant rule reduction.

Keywords Hierarchical design · Type-2 fuzzy systems · Beta basis function · Structure learning · Multi-objective optimization · Parameter tuning

1 Introduction

The modeling of fuzzy systems has gone through remarkable success over the last fifty years. They have proved their capacity in different application areas, such as pattern

recognition (Boutleux and Dubuisson 1996), classification (Jarraya et al. 2015; Jahromi and Moosavi 2011), control problems (Tanaka and Sano 1994; Singhala et al. 2014), data mining problems (Hüllermeier 2005) and time series forecasting (Jarraya et al. 2013, 2014).

In recent few years, type-2 fuzzy design (Zadeh 1975) has become a growing and an active research topic in the focus of several researchers. The main cause of the immigration from type-1 fuzzy systems to type-2 fuzzy systems is due essentially to the nature of the knowledge employed to produce fuzzy rules which are usually too uncertain and contain incomplete or inaccurate information. In fact, type-1 fuzzy sets are precise and unable to handle high levels of uncertainties. In this sense, several research studies have proved that type-2 fuzzy approaches have more potential than their type-1 counterpart in coping with uncertainties.

However, as the complexity and the dimensionality of the application problems increase, the number of fuzzy rules of a standard fuzzy system (type-1 or type-2) also increases which badly affects the interpretability of the resulting rules. That is why, standard fuzzy systems usually suffer from the “curse of dimensionality” or the

Communicated by V. Loia.

✉ Yosra Jarraya
yosra.jarraya@ieee.org

Souhir Bouaziz
souhir.bouaziz@ieee.org

Adel M. Alimi
adel.alimi@ieee.org

Ajith Abraham
ajith.abraham@ieee.org

¹ Research Groups in Intelligent Machines (REGIM-Lab), National School of Engineers (ENIS), University of Sfax, BP 1173, 3038 Sfax, Tunisia

² Machine Intelligence Research Labs (MIR Labs), Auburn, WA, USA

³ IT4Innovations, VSB-Technical University of Ostrava, Ostrava, Czech Republic

“combinatorial rule explosion” problem when they are applied to complex problems. As a solution, the hierarchical fuzzy modeling was proposed by Raju and Zhou (1993) as an effective alternative to solve this problem by arranging the inputs in a hierarchical architecture. In this case, instead of using a standard or a flat high-dimensional fuzzy system, a number of lower-dimensional sub-fuzzy models are linked in a hierarchical way. Consequently, the number of rules is reduced and the approximation abilities of the system are improved.

Since Raju's paper on hierarchical fuzzy systems (HFSs) (Raju and Zhou 1993), several works have appeared in that area applying machine learning and optimization techniques to help in the process of building type-1 HFSs. Indeed, the search for an optimal hierarchical structure is as important as the search for the optimal set of parameters. Therefore, hierarchical fuzzy modeling has been considered as a search problem and as an optimization task in both structure and parameter spaces. In this regard, several research works have been proposed in the literature to construct or to learn these systems. For example, the authors suggested in Shimojima et al. (1995) the use of the genetic algorithm (GA) for the optimization of the HFS' structure and the gradient descent algorithm to adjust the parameters. In Lin and Lee (1999), the authors proposed for the control low-speed problem, a method based on GA in order to optimize the parameters and the structure of five inputs hierarchical fuzzy controller. Moreover, an approach was introduced in Balazs et al. (2010) based on two structures evolving algorithms which are the genetic and the bacterial programming algorithms in order to build a multi-level rule-based system. Salgado proposed in Salgado (2008), a hierarchical collaborative structure (HCS) where three phases of structure building, parameter identification and data division were employed. In Cheong and Lai (2007), the authors applied the differential evolution (DE) algorithm for the automatic design of a hierarchical fuzzy logic controller. In addition, Chen et al. (2004) proposed a method that interleaves the ant programming (AP) and the particle swarm optimization (PSO) algorithms for, respectively, the structure learning and the rules' parameters tuning of a TSK HFS. Chen et al. ameliorated this version and proposed in Chen et al. (2007) a structure learning step using the probabilistic incremental program evolution (PIPE) algorithm and applied the evolutionary programming algorithm (EP) for adjusting the rules' parameters of the hierarchical system. In Fernández et al. (2009), the authors suggested a learning process of a hierarchical fuzzy rule base classification system (HFRBCS), where the knowledge base was created by means of a linguistic rule generation (LRG) method and the best rules were selected from the hierarchical rule base using the GA.

Nevertheless, all of these HFSs proposals employed type-1 fuzzy models in the hierarchical design and focused on improving only the system's accuracy using mono-objective hybrid optimization techniques. However, as far as we know, very few publications can be found exploiting multi-objective evolutionary algorithms (MOEAs) for evolving HFSs and/or addressing type-2 fuzzy systems in the hierarchical design (Hagras 2004; Benítez and Casillas 2013; Ojha et al. 2017). In fact, Hagras (2004) suggested a type-2 hierarchical fuzzy logic controller (type-2 HFCL) for autonomous mobile robots. In Benítez and Casillas (2013), a multi-objective genetic algorithm was proposed for the learning of an incremental HFS with the aim of minimizing two objectives, which are the error and the number of rules. In Ojha et al. (2017), the authors used a multi-objective genetic programming (MOGP) algorithm to train the structure of a proposed type-2 hierarchical fuzzy inference tree (T2HFIT), while the differential evolution (DE) algorithm was used to fit the parameters of the model. In this approach, both the accuracy and complexity of the system were used as objectives to reach.

In this work, a type-2 hierarchical flexible beta fuzzy system (T2HFBFS) is proposed. For the design process, the T2HFBFS is presented through a tree encoding method, and its membership functions (MFs) are modeled based on interval type-2 beta fuzzy sets. The use of type-2 beta fuzzy sets enables the system to handle uncertain information more efficiently than its type-1 counterpart. Regarding the optimization process, two main phases of structure learning and parameter optimization are introduced. The structure learning phase is performed based on the multi-objective extended immune programming (MOEIP) algorithm. This step aims to learn a population of T2HFBFS structures taking into account the optimization of both the accuracy and the interpretability objectives. For the tuning phase, the PSO-based update memory for improved harmony search algorithm (PSOUM-IHS) (Ammar et al. 2013) is applied to update the parameters of the antecedent and consequent parts of fuzzy rules. By interleaving the two optimization phases, a high-performance T2HFBFS is finally generated.

The rest of the paper is planned as follows: In Sect. 2, the type-2 hierarchical flexible beta fuzzy system is proposed. Section 3 introduces the initialization phase based on a clustering technique. Next, the multi-objective structure learning and the parameter optimization phases are explained in, respectively, Sects. 4 and 5. A global description about the hybrid evolving algorithm is then given in Sect. 6. Simulation studies are next depicted in Sect. 7. And finally, in Sect. 8, some conclusion remarks are given.

2 The type-2 hierarchical flexible beta fuzzy system

2.1 Type-1 beta membership function (T1 BMF)

The type-1 beta fuzzy set was proposed by Alimi (1997). In the one-dimensional case, the type-1 beta membership function (T1 BMF) is expressed by:

$$\beta(x; c, \sigma, p, q) = \begin{cases} \left[1 + \frac{(p+q)(x-c)}{\sigma p} \right]^p \times \left[1 - \frac{(p+q)(c-x)}{\sigma q} \right]^q & \text{if } x \in \left[c - \frac{\sigma p}{p+q}, c + \frac{\sigma p}{p+q} \right] \\ 0 & \text{elsewhere} \end{cases} \quad (1)$$

where c is the beta function center, σ is the width, and p and q present the form parameters, $p, q > 0$.

The beta function has a universal approximation characteristic and can approximate other functions like triangular or Gaussian functions (Alimi 2003). Indeed, in comparison with the Gaussian function which relies on the center and the width parameters, the beta function relies on two additional form parameters (p and q) which allow a greater flexibility in the modeling of fuzzy sets. Alimi (2003) demonstrated that the beta function has the capacity to approximate the Gaussian function and that the reverse is not true. Moreover, this function presents other advantages like its high flexibility and its ability to generate richer shapes (asymmetry, linearity, etc.) (Alimi 1997, 2000). Figure 1 presents some examples of type-1 beta MFs with different shapes.

2.2 Interval type-2 beta membership function (IT2 BMF)

A type-2 fuzzy set \tilde{A} (T2 FS) in the universe of discourse U is represented by a type-2 membership function $\mu_{\tilde{A}}$, expressed as follows (Mendel and John 2002):

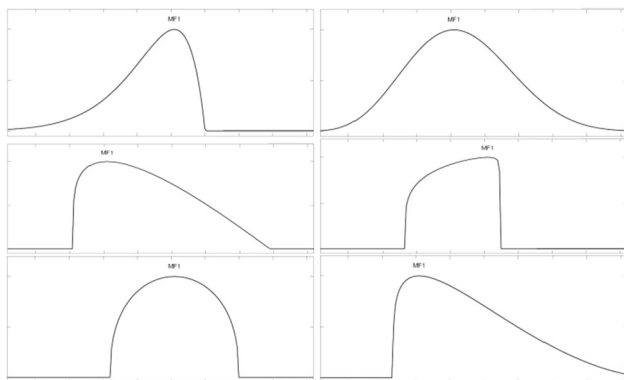


Fig. 1 T1 BMFs with different shapes

$$\tilde{A} = \int_{x \in U} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad (2)$$

where \int defines the union over all admissible x and u , J_x denotes an interval in $[0, 1]$ and $\mu_{\tilde{A}}(x, u)$ presents a type-1 fuzzy set known as the secondary set with $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. When all $\mu_{\tilde{A}}(x, u) = 1$, then \tilde{A} is considered as a special case of a type-2 fuzzy set called an interval type-2 fuzzy set (IT2 FS) and is expressed as follows:

$$\tilde{A} = \int_{x \in U} \int_{u \in J_x} 1 / (x, u) \quad (3)$$

where $J_x \subset [0, 1]$. Note that an IT2 FS is characterized by a bounded area FOU known as the footprint of uncertainty. This area is delimited by two type-1 fuzzy sets called upper MF (UMF), $\bar{\mu}_{\tilde{A}}(x)$, and lower MF (LMF), $\underline{\mu}_{\tilde{A}}(x)$, respectively.

In this study, instead of employing Gaussian or triangular interval type-2 MFs which are frequently used in the literature, we choose to adopt the beta function for its large flexibility as compared with the other functions. Therefore, the interval type-2 beta membership function (IT2 BMF) is proposed and employed in this work. The IT2 BMF is a beta primary MF with a fixed center c , and uncertain parameters σ, p and q . The IT2 BMF is expressed by:

$$\begin{cases} \beta(x; c, \sigma, p, q) = \left[1 + \frac{(p+q)(x-c)}{\sigma p} \right]^p \left[1 - \frac{(p+q)(c-x)}{\sigma q} \right]^q \\ \sigma \in [\sigma_L, \sigma_U], p \in [p_L, p_U] \text{ and } q \in [q_L, q_U] \end{cases} \quad (4)$$

where $\sigma_L, \sigma_U, p_L, p_U, q_L$ and q_U present positive real values. Figure 2 shows two examples of IT2 BMFs with uncertain σ, p and q . Upper and lower beta MFs are, respectively, expressed by the following equations:

$$\bar{\mu}_{\tilde{A}}(x) = \beta(x; c, \sigma_U, p_U, q_U) \quad (5)$$

$$\underline{\mu}_{\tilde{A}}(x) = \beta(x; c, \sigma_L, p_L, q_L) \quad (6)$$

2.3 Type-2 hierarchical flexible beta fuzzy system (T2HFBFS)

In general, building a hierarchical fuzzy model is considered as a difficult task. This is because we have to define the number of hierarchical levels, the rule base of each

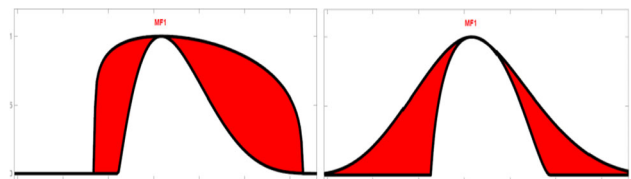


Fig. 2 IT2 BMFs with uncertain σ, p and q having different FOUs

module (sub-fuzzy model), the number of modules in intermediate levels, how to arrange the original input variables in the different levels and so on.

In this study, the design of an accurate type-2 hierarchical fuzzy system having a simple structure is considered as a search problem in both architecture and parameter spaces. For that, firstly, the multi-level aspect is illustrated by employing a tree-based encoding scheme. In fact, the tree representation can provide more adjustable and modifiable structures due to its natural and flexible hierarchical representation. In addition, the presented hierarchical tree design is proposed for type-2 beta fuzzy systems. In this case, the resulted system is termed the type-2 hierarchical flexible beta fuzzy system (T2HFBFS). The proposed system is characterized by the following node set S :

$$S = N \cup T = \left\{ \text{BFII}_{kj}^l / k \in \{2, \dots, K\}, j \in \{1, \dots, J\}, l \in \{1, \dots, (L - 1)\} \right\} \cup \{x_1, \dots, x_M\} \tag{7}$$

where:

- N is the non-terminal node set formed by a number of type-2 beta sub-fuzzy models BFII_{kj}^l . Each BFII_{kj}^l presents an interval A2-C1 TSK fuzzy model (Liang and Mendel 1999) having one output and a number of fuzzy rules. These rules rely on an antecedent part based on interval type-2 beta membership functions (IT2 BMFs) and a consequent part based on interval type-1 fuzzy sets.
- For a FII_{kj}^l , k defines the number of inputs, while K is the maximal degree of the tree. j is the index of the corresponding BFII_{kj}^l with k inputs, while J is the number of times in which BFII_{kj}^l occurs with k inputs. And l is the index of the level, while L presents the number of the tree levels;
- T is the terminal node set formed by the input variables x_1, x_2, \dots, x_M ;

The evaluation of a T2HFBFS is executed recursively from the lower level to the final one. Each level can contains one or several sub-fuzzy models. The inputs of a level can be the outputs of its lower level, or a combination of some original inputs with its lower-leveled outputs. For further clarification about the evaluation process of a T2HFBFS, Fig. 3 is considered as a simple example of a possible obtained T2HFBFS (Fig. 3b) from a tree structural representation (Fig. 3a). The evaluation process for this example is done as follows:

Firstly, the BFII_{22}^2 of the second level is evaluated. The fuzzy rules format of this system is as follows:

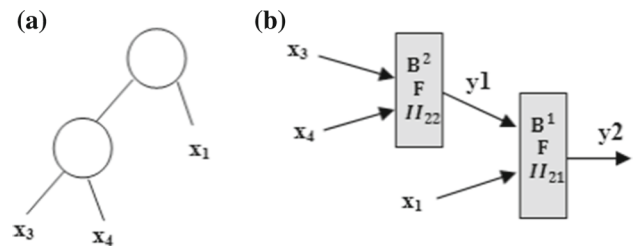


Fig. 3 a An example of a tree representation, b the corresponding T2HFBFS: $N = \{\text{BFII}_{21}^1, \text{BFII}_{22}^2\}$, $T = \{x_1, x_3, x_4\}$

$$R_i^{l=2} : \text{If } x_3 \text{ is } \tilde{A}_{1i}^2 \text{ and } x_4 \text{ is } \tilde{A}_{2i}^2 \text{ then} \\ Y_i^2 = C_{0i}^2 + C_{1i}^2 x_3 + C_{2i}^2 x_4 \tag{8}$$

where x_3 and x_4 are the original inputs and Y_i^2 is the output of rule i . \tilde{A}_{1i}^2 and \tilde{A}_{2i}^2 are, respectively, the antecedent fuzzy sets of variables x_3 and x_4 modeled by the IT2 BMFs. C_{0i}^2 , C_{1i}^2 and C_{2i}^2 are the consequent elements presented by interval type-1 fuzzy sets.

Next, the output of the whole T2HFBFS is evaluated by computing the root node output of BFII_{21}^1 . This module has two inputs which are $y1$ (the output of its lower level BFII_{22}^2) and x_1 . Rules of BFII_{21}^1 have the following format:

$$R_j^{l=1} : \text{If } y1 \text{ is } \tilde{A}_{1j}^1 \text{ and } x_1 \text{ is } \tilde{A}_{2j}^1 \text{ then} \\ Y_j^1 = C_{0j}^1 + C_{1j}^1 y1 + C_{2j}^1 x_1 \tag{9}$$

where Y_j^1 is the output of rule j . \tilde{A}_{1j}^1 and \tilde{A}_{2j}^1 are, respectively, the antecedent fuzzy sets modeled by the IT2 BMFs. C_{0j}^1 , C_{1j}^1 and C_{2j}^1 are the interval type-1 consequent elements. The final output of the whole system is given by $y2$ (the output of BFII_{21}^1). Note that, for each BFII_{kj}^l , its IT2 BMF parameters and rule consequents will be further optimized in the phase of parameter tuning.

3 Initialization step by a clustering method

The first step of optimization is a learning step of a set of T2HFBFS structures in order to get the best structure. But, it is important to note that optimizing the structures of a set of T2HFBFSs having totally random rules and parameters will not contribute to good results and will slow down the process of optimization. Therefore, a first step of T2HFBFS initialization is required before starting the optimization process. In this sense, the subtractive clustering (SC) algorithm is implemented in this work as a first step to automatically extract the initial rule bases of the hierarchy and the initial beta membership functions shapes.

In fact, the subtractive clustering algorithm proposed by Chiu (1994) is an unsupervised algorithm employed to divide a given data set into meaningful subgroups called

clusters. By applying this algorithm, the generated centers of clusters will correspond to the membership function centers, and each center of a cluster will be converted into a fuzzy rule. Based on this concept, an initialization mechanism using the SC algorithm is introduced in this study and is described as follows:

After generating a random population of trees, for each tree, terminal nodes of input variables are clustered in order to create the different BFII_{kj}^l modules of the hierarchy. These sub-fuzzy models are automatically generated with initial values of beta MF parameters and with rule bases extracted from data. In consequence, the resulting extracted rules are more adjusted to the input data than they are in fuzzy models generated randomly. Such initialization step by clustering terminal nodes of trees will speed up the whole evolutionary search process. Readers may refer to (Chiu 1994) to get all the details about the SC algorithm.

4 Multi-objective structure learning by MOEIP algorithm

4.1 Basic concepts of EIP for single optimization problem

In Musileket al. (2006), the authors proved that the immune programming (IP) algorithm is more effective than the genetic programming (GP) algorithm: Successful solutions are obtained using a smaller size of population and with less number of generations. In addition, the IP algorithm has a great ability to evolve structures of trees or programs and proves high flexibility to create more adjustable architectures. For that, we choose to employ in this work a modified version of the IP algorithm in the structure learning phase of T2HFBFSs. The employed algorithm is termed the extended immune programming (EIP) algorithm and is described by the following steps:

- a. *Initialization* An initial population of T2HFBFSs (antibodies) is randomly created with random structures (random number of levels and random nodes for each level). For each T2HFBFS, the IT2 BMFs parameters of BFII_{kj}^l modules and the fuzzy rules are initialized by the SC algorithm except for the beta form parameters (p and q) which are randomly generated.
 $EIP_Itr = 0$, EIP_Itr is the generation counter.
- b. *Evaluation* An antigen representing the problem to be addressed is introduced. The antigen is compared to all antibodies (NA antibodies), and their affinity fitness $\text{Fit}(i)$ with respect to the antigen is determined.
- c. *Cloning* in this step, an antibody Ab_i is selected from the actual population and a random number forming the cloning probability P_c is generated. If the Ab_i

affinity is higher than P_c , then Ab_i is cloned and introduced in the next population.

- d. *Mutation*: if the previous selected high-affinity antibody Ab_i has not performed a cloning step because of the stochastic nature of this step, so it is presented to hypermutation. As shown in Fig. 4, four mutation operators were used in this study:

Modifying one terminal node: choose a terminal node at random and replace it by another terminal node;

Modifying all terminal nodes: all terminal nodes of Ab_i are selected and replaced by other random terminal nodes;

Growing: choose a terminal node at random and replace it by a randomly created sub-fuzzy model.

Pruning: select randomly a non-terminal node (sub-fuzzy model) and replace it with another randomly generated terminal node.

The mutation operators are used based on the method of Chellapilla (1998). This method is described as follows: (i) generate a number N which presents a sample from a Poisson variable generated at random, (ii) choose N operators randomly from the previously proposed mutation operators and (iii) execute consecutively the N operators on the parents and obtain the offspring.

The mutation step is presented in various manners so that a population with a high genetic diversity is obtained. This diversity will help to overcome local optimum.

- e. *Replacement* if Ab_i is not selected for mutation or cloning steps, then we generate a new antibody for the next population (using a replacement probability Pr). Consequently, candidate solutions with low affinity fitness are implicitly replaced.
- f. *Iteration population* steps c–e presenting EIP operators are repeated until a full novel population is created.
- g. *Iteration algorithm* increment the generation counter after the creation of the new population, $EIP_Itr = EIP_Itr + 1$. The EIP is iteratively executed (steps b–f) until attaining stopping criteria.

This section introduces the EIP algorithm as a single optimization algorithm and the used fitness function reflects only the accuracy of the system. The next subsections will present an extended version of this algorithm showing how we can improve not only the system's accuracy but also the interpretability objective. In this case, the problem is called multi-objective.

4.2 Multi-objective optimization problem

A minimization multi-objective problem may be formulated as:

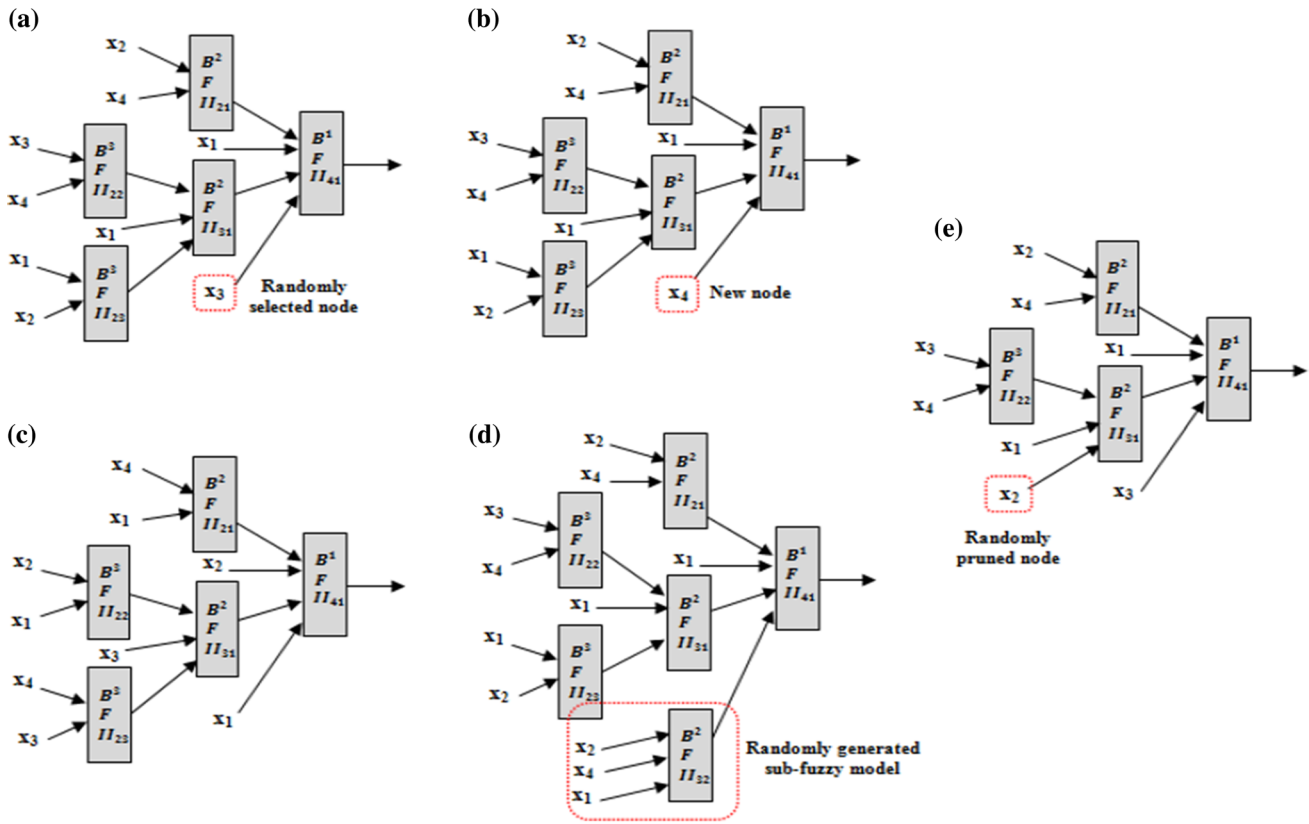


Fig. 4 Examples of the EIP mutation operators: **a** original T2HFBFS. **b** Modify one terminal node. **c** Modify all terminal nodes. **d** Growing. **e** Pruning

$$\text{Min } f(x) = [f_1(x), \dots, f_k(x)] \tag{10}$$

subject to:

$$g_j(x) \geq 0 \quad j = 1, \dots, p \tag{11}$$

$$h_j(x) = 0 \quad j = 1, \dots, q \tag{12}$$

where $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ is a vector of solutions defined on the space of decision variables. $g_j(x)$ and $h_j(x)$ are the functions that represent the constraints of the problem. k presents the number of objective functions, and p and q are, respectively, the number of equality and inequality constraints.

Unlike single-objective optimization problems where only one optimal solution is created, multi-objective optimization algorithms create a number of optimal solutions named Pareto optimal solutions or non-dominated solutions. The concept of dominance is presented as follows: a solution x dominates a solution y (expressed by $x \prec y$) if and only if x is greater than y in at least one objective function and x is not worse than y in any objective function. x is named Pareto optimal if x is not dominated by any other solution of the present population. In the objective space, the set of all Pareto optimal solutions is called the Pareto optimal front.

4.3 Objective functions

Multi-objective fuzzy design requires the consideration of different objective functions in the search process. In this study, both the accuracy and the interpretability metrics are treated in the multi-objective structure learning step. While the accuracy is measured by the root mean squared error (RMSE), the system's interpretability is defined by the number of used fuzzy rules:

$$\text{Objective 1 : RMSE} = \sqrt{\frac{1}{m} \sum_{j=1}^m (y_i^j - y_{\text{out}}^j)^2} \tag{13}$$

where m defines the number of samples, and y_i^j and y_{out}^j are, respectively, the actual and the calculated outputs.

$$\text{Objective2 : Interpretability} = \text{NR} \tag{14}$$

where NR is the total number of fuzzy rules.

4.4 Multi-objective extended immune programming algorithm (MOEIP)

In this section, a multi-objective version of the EIP algorithm is proposed and called the multi-objective extended immune programming (MOEIP) algorithm. This algorithm

is based on an elitist strategy and uses the EIP operators combined with a dominance concept to search for elite solutions. For that, an archive A formed by a number of non-dominated antibodies is used. This archive forms a secondary population of Pareto optimal solutions. In the MOEIP strategy, each antibody of the population is presented by a T2HFBFS, and as the search progresses, the dominance criterion with the EIP operators is applied to evolve the population toward an optimal Pareto front.

Moreover, in the evolutionary single optimization case, a child is selected over its parent if it has better fitness value. In the proposed MOEIP, the superiority is measured as a dominance relationship, and a child is selected over its parent only if this latter dominates the parent. Consequently, as the search progresses, the different solutions move closer to the Pareto front.

On the other hand, to generate a well-distributed Pareto front with good diversity, a pruning method using the crowding distance measure (applied in non-dominated sorting genetic algorithm II (NSGA II) (Deb et al. 2002)) is implemented. In fact, we calculate for each solution of the front a crowding distance value which presents the distance between this solution and its neighbors in the fitness function space. In each generation, the solutions of the archive A are sorted according to their values of crowding distance: Solutions with the highest crowding distance values (with best diversity) are maintained and the worst are discarded from the archive.

Suppose that x_i is a Pareto front solution. The crowding distance $cd(x_i)$ of x_i is calculated as follows:

- (i) The crowding distance of x_i is initialized: $cd(x_i) = 0$;
- (ii) For each objective function f_j do:
 - Sort the different non-dominated solutions along the objective function f_j ;
 - $cd(x_i) = cd(x_i) + f_j$ (the solution which precedes x_i in the ordered sequence)— f_j (the solution which follows x_i in the ordered sequence);

A pseudo-code of the MOEIP is introduced by Algorithm 1. To summarize, the basic ideas of this algorithm are:

- The structure learning of a population of T2HFBFSs has taken into consideration the optimization of both the accuracy and the interpretability.
- Guide the search process through an optimal Pareto front of non-dominated elite solutions.
- Keep a diverse set of spaced non-dominated solutions using the crowding distance measure.

- In the mutation step, the replacement of antibodies is done using dominating offspring.

It should be noted that the MOEIP structure learning step terminates when the maximum number of iterations is reached. As a result, a Pareto optimal front of T2HFBFS solutions is generated. In this case, the most suitable structure solution (having a good balance between the accuracy and the interpretability) is selected to undergo a second step of parameter optimization.

Algorithm 1: MOEIP Algorithm

Input: NA (size of the population)
 Max_Itr : (maximum iteration number)
 Max_Size : (the archive maximum size)

Output: A (an archive containing non-dominated solutions)

Begin

- a. Generation of an initial random population P_0 of antibodies (T2HFBFSs) with random structures and creation of an initial external empty archive $A_0 = \emptyset$.
 $MOEIP_Itr = 0$;
- b. Evaluation of P_0 .
- c. Applying dominance criterion on P_0 in order to store the initial Pareto optimal solutions in the archive A_0 .
- d. Pruning of A_0 based on the crowding distance value: If the archive size (number of Pareto optimal solutions) is greater than Max_Size , then the crowding distances of all individuals of the archive are calculated and sorted in a descending order. The first Max_Size antibodies are then chosen to update the archive.
- e. **While** ($MOEIP_Itr < Max_Itr$) **do**
 - Update of the population P_t : Apply EIP operators on $P_t \cup A_t$ to generate the new population P_{t+1} (section 4.1):
 - Replacement (P_r)
 - Cloning (P_c)
 - Mutation: if an offspring dominates the parent, then this latter is replaced by its offspring.
 - Update of the archive A_t : Applying the dominance criterion on $A_t \cup P_{t+1}$ to create the external archive A_{t+1} .
 - Pruning of A_{t+1} based on the crowding distance measure (as in step d).
 - $MOEIP_Itr = MOEIP_Itr + 1$.

End

- f. Return the final non-dominated solutions of A_t .

End

5 Parameter optimization using the hybrid PSOUM-IHS

The harmony search (HS) proposed by Gem et al. (2001) is a well-known evolutionary meta-heuristic music-inspired algorithm. The harmony search was inspired from musical jazz improvisation when a musician (decision variable) plays (create) a note (value) to reach a better state of harmony (global optimum). The HS explores the solution space to attain the optimum value. The main steps of the HS are briefly described as follows:

- Step 1 Formulate the problem and initialize the parameters.
- Step 2 Initialize at random the harmony memory.
- Step 3 Improvise a novel harmony.
- Step 4 Update of the harmony memory.
- Step 5 Repeat steps 3 and 4 until a stopping criterion is reached.

In 2007, Mandavi et al. (2007) suggested an optimized version of the original HS named the improved harmony search (IHS) algorithm. The authors suggested adjusting and evolving the parameters of the HS in the different iterations instead of using them with fixed values. Consequently, during the execution, the pitch adjustment rate (PAR) will increase linearly and the bandwidth (the Fret's width (FW)) will decrease exponentially according to the following expressions:

$$PAR = \frac{PAR_{max} - PAR_{min}}{MaxItr} * currentIteration + PAR_{min} \tag{15}$$

$$FW = FW_{max} * \exp(coef * currentIteration) \tag{16}$$

$$coef = \frac{\log\left(\frac{FW_{min}}{FW_{max}}\right)}{NI} \tag{17}$$

Although the HS was improved, it still has some weakness. In fact, most of the decision variables stored in the new harmony are taken from the elements of the harmony memory. In addition, it should be noted that in most of the time, the harmony memory is stable and does not present a wide variety of values to the improvisation. Thus, in general, the harmony search algorithm has a small probability to provide a good quality of new harmony vectors, and this may affect the convergence speed of the algorithm. To tackle this problem, Ammar et al. (2013) proposed to include the idea of the particle swarm optimization (PSO) velocity in the search process of IHS algorithm. Such hybridization will allow creating a wide variety of solutions in memory (with respect to their allowable ranges) and will help the convergence to the optimal solution. In fact, PSO mechanism can create iteratively a new

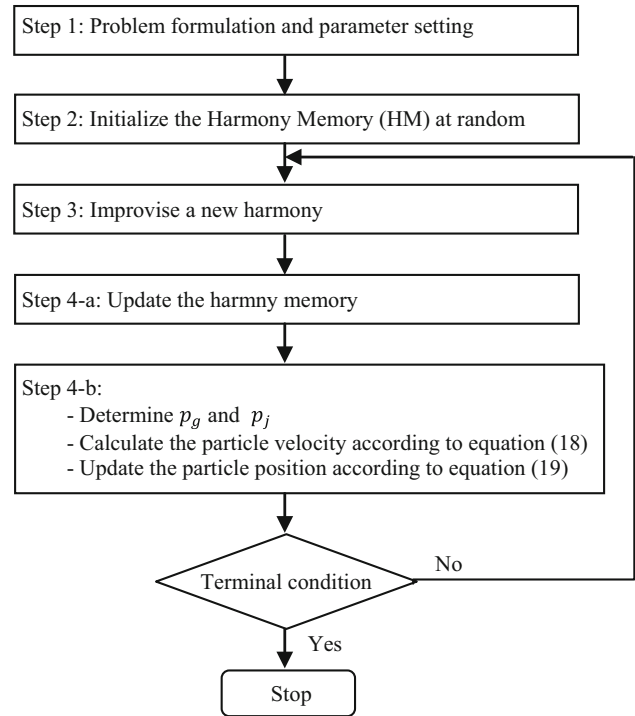


Fig. 5 Flowchart of PSOUM-IHS

population totally updated and closer to the optimum solution. Over the iterations, the particles in PSO converge together around an optimum through a combination of exploration and exploitation steps of the search space. The proposed hybrid algorithm is named PSO-based update memory for improved harmony search (PSOUM-IHS) (Ammar et al. 2013). Its flowchart is shown in Fig. 5. Here, the main idea focuses on exploiting the stochastic and dynamic aspects of particle velocities which aid to guide the search to the right areas of the work space. To realize this hybridization, the vectors of memory in IHS are considered as particles of the swarm, and the new values of memory for the new improvisation are considered as the novel positions reached by these particles. For each particle, the velocity v_j and the position x_j are calculated according to the following equations:

$$v_j(t + 1) = \Psi(t)v_j(t) + c_1\phi_1(p_j(t) - x_j(t)) + c_2\phi_2(p_g(t) - x_j(t)) \tag{18}$$

$$x_j(t + 1) = x_j(t) + (1 - \Psi(t))v_j(t + 1) \tag{19}$$

where:

c_1 and c_2 : acceleration/weighting factors.

ϕ_1 and ϕ_2 : random numbers in [0,1].

Ψ : inertia factor.

p_j : local best position (found by the th particle).

p_g : global best position (found by the swarm).

Readers can find more details concerning this hybrid algorithm in (Ammar et al. 2013).

In this study, the PSOUM-IHS is employed for the parameter adjustment phase of the obtained T2HFBFS. For that, the parameters of the best T2HFBFS (generated by the MOEIP structure learning phase) constitute an element of the memory to be evolved via the PSOUM-IHS. So, each element of the memory is represented by a $N_{Parm} \times Size$ matrix. N_{Parm} defines the parameter number, while $Size$ is the number of BFI_{kj}^l modules of the best T2HFBFS found. The parameters encoded in the matrix are the IT2 BMFs ($c, \sigma_L, \sigma_U, p_L, p_U, q_L, q_U$) and the linear weights of the consequent parts of rules. The used objective function in this phase is the RMSE (defined previously).

6 Hybrid algorithm to evolve the T2HFBFS

In general, the modeling of high-dimensional problems is usually preceded by a data preparation phase known as feature selection. Instead of working with all features of the original data set, feature selection algorithms allow the reduction in the feature space, which enhance the prediction performance of the system and speed up the search process. In this paper, a feature scoring technique called the statistical dependency (SD) algorithm (Pohjalainen et al. 2015) is exploited as a data preparation phase when the tested data set is a classification data set formed by a high input dimension. Next, the training process starts: The MOEIP and the PSOUM-IHS are alternately applied to find an optimum or a near-optimum T2HFBFS with good precision and a minimum number of rules. The following steps give a brief description of the hybrid evolving algorithm:

- (a) A data preparation step: The most useful input data are selected by the SD feature selection algorithm. This step is performed only when we are testing a high-dimensional classification problem.
- (b) Initialization step: An initial population of random trees is generated with random architectures. A population of T2HFBFSs is built by executing the subtractive clustering algorithm on the trees (Sect. 3).
 $Global_Gn = 0$; (the global generation number)
 $Global_Itr = 0$; (the global iteration number)
- (c) Multi-objective structure learning phase: The MOEIP algorithm is applied to evolve the T2HFBFS population toward a Pareto front of non-dominated solutions. Each solution of the Pareto front corresponds to a T2HFBFS tree. The used objective functions are defined by (13) and (14).
- (d) Go to step (e) in the case of attaining the maximum number of $MOEIP_Itr$.
 $Global_Itr = Global_Itr + MOEIP_Itr$;

Otherwise, go back to step (c);

- (e) The most suitable solution (T2HFBFS having a good trade-off between the accuracy and the interpretability) is selected to be the output of the multi-objective structure learning phase.
 - (f) Parameter optimization phase: The PSOUM-IHS is applied to evolve the parameters of the best T2HFBFS. These parameters formulate an element of the memory. The used objective function here is defined by (13).
 - (g) Go to step (h) in the case of attaining the maximum number of $Parm_Itr$, or in the case of not finding a better matrix of parameters after a fixed time.
 $Global_Itr = Global_Itr + Parm_Itr$;
- Otherwise return to step (f);
- (h) Stop in the case of generating a satisfactory T2HFBFS or reaching a maximum number of global iterations.
 Otherwise, $Global_Gn = Global_Gn + 1$ and go back to step (c).

7 Experimental results

The performance of the T2HFBFS is evaluated through six data sets of different nature. These data sets include four cases of time series forecasting problems and two cases of high-dimensional classification problems. For all experiments, the minimum and the maximum number of levels (depth of the trees) is, respectively, 2 and 4. Moreover, the minimum and the maximum number of nodes (degree of the used trees) is, respectively, 2 and 5. Table 1 gives the best-suited list of parameters used in the experiment.

Table 1 Parameter initialization

Parameters	Initial values
MOEIP	
Population size	30
Probability of cloning (P_c)	0.7
Probability of replacement (P_r)	0.5
PSOUM-IHS	
Population size	20
PAR_{min}	0.00001
PAR_{max}	1
$HMCR$	0.9
$c1$	0.2
$c2$	0.7

7.1 T2HFBFS for time series forecasting

For this kind of problems, four types of nonlinear benchmark prediction time series are tested, including the Mackey–Glass chaotic time series, the Box and Jenkins' gas furnace time series, the monthly NAO climatic index time series and the sunspot number time series.

For a meaningful comparison with other state-of-the-art learning systems, all the data sets are divided into the same number of training and testing patterns. The simulations are running 10 times and then results are averaged.

7.1.1 Predicting the Mackey–Glass Chaotic Time Series

The Mackey–Glass time series (MG) (Mackey and Glass 1977) is one of the widely known benchmark problems which have been intensively studied in several works. The chaotic system is the signal $x(t)$ produced by the numerical solution of the following time delay differential equation:

$$\frac{d(x(t))}{dt} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t) \quad (20)$$

The chaotic behavior of this series depends on the parameter τ . If $\tau > 16.8$, MG has a chaotic behavior. In our case, the parameters are initialized as follows: $\tau = 17$, $a = 0.2$, $b = 0.1$ and $c = 10$. These initial conditions are the same ones used by the comparison systems. Four past values are used to predict $x(t+6)$, which are $x(t)$, $x(t-6)$, $x(t-12)$ and $x(t-18)$. A total of 1000 observations were generated, where the first 500 samples are used for the training and the last 500 samples are used for the test.

After performing 13 global iterations, 2 global generations and 520 number of Function Evaluations FEs, a near-optimum T2HFBFS was obtained with 7 rules. The training and testing RMSE values of the generated T2HFBFS are equal to $9.2112e-16$ and $9.1012e-16$, respectively.

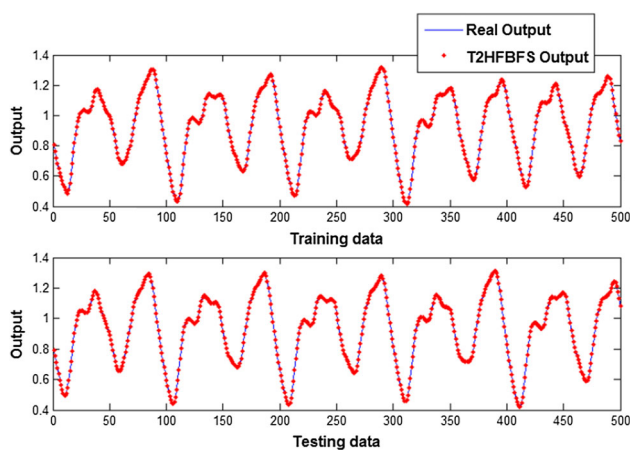


Fig. 6 Results of MG prediction for training and testing data

Figure 6 shows the prediction result of the T2HFBFS for training and testing data.

In Table 2, we make a comparison between the results of the proposed T2HFBFS and other state-of-the-art type-1/type-2 fuzzy systems and neural network approaches. As can be observed in this table, the proposed T2HFBFS shows better performance than the other works in terms of achieving high levels of accuracy using simpler fuzzy rule base.

7.1.2 Predicting the Box–Jenkins chaotic time series

The proposed T2HFBFS is applied to the Box and Jenkins gas furnace prediction problem. The data set was recorded from a combustion procedure of a methane–air mixture. This time series is a well-known real-world benchmark example frequently exploited for testing prediction methods. The used data set consists of 296 pairs of input–output measurements collected from a laboratory furnace (Box and Jenkins 1976). The first 200 data points were employed for training and the remaining 96 were employed for testing. $u(t)$ is the gas flow into the furnace considered as input, and $y(t)$ is the CO₂ concentration in outlet gas considered as output. In order to make meaningful comparisons with previous works, $y(t-1)$ and $u(t-4)$ are selected to be used as inputs to predict $y(t)$.

After performing 6 global iterations, 1 global generation and 240 number of FEs, the obtained RMSE values for the training data and the testing data are, respectively, 0.0044 and 0.0108. The optimum T2HFBFS is generated with 4 fuzzy rules. Figure 7 illustrates the prediction results including the actual and the predicted outputs for training and testing data. From these results, it is remarkable that the proposed system might attain good results after only few hundreds of FEs and using the minimum number of rules.

The proposed approach is compared in Table 3 with several learning techniques such as the genetic and memetic fuzzy design methods (Wadhawan et al. 2013), the belief rule-based (BRB) system (Chen et al. 2013), the adaptive neuro-fuzzy inference system (ANFIS) (Samanta 2011), the beta basis function neural network (BBFNN) system (Dhahri et al. 2013), the single multiplicative neuron trained with cooperative sub-swarms of PSO (SMN-COPSO) (Samanta 2011), the flexible neural tree (FNT) system (Chen et al. 2005) and the fuzzy granular evolving modeling (FBEM) system (Leite et al. 2011). Our system is also compared with the interval type-2 intuitionistic fuzzy logic system IT2 IFLS (Eyoh et al. 2018). As can be observed from the table, the T2HFBFS achieves better performance in terms of attaining the best training and testing errors with few number of fuzzy rules.

Table 2 Comparison results of Mackey–Glass time series

Approach	Training RMSE	Testing RMSE	Rules
SuPFuNIS (Paul and Kumar 2002)	3.7e−03	3.74e−03	10
NARMA (da Silva 2008)	6.3e−04	6.2e−04	–
Memetic-T2FS (León et al. 2015)	3.1e−03	–	–
TSK-SVR II (Uslan et al. 2014)	–	7.0e−03	32
GT2FLS-VSCTR (Almaraashi et al. 2016)	3.9e−02	3.9e−02	–
SA-IT2FLS (Almaraashi and John 2011)	9.0e−03	8.9e−03	16
IT2 IFLS (Eyoh et al. 2018)	–	4.0e−03	16
FBBFNT (Bouaziz et al. 2016)	9.9e−07	2.0e−06	–
T2HFBFS	9.2e−16	9.1 e−16	7

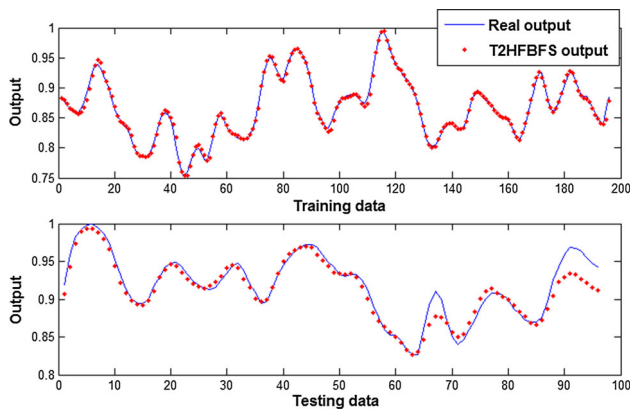


Fig. 7 Results of Box and Jenkins prediction for training and testing data

7.1.3 Predicting the monthly NAO index time series

The North Atlantic Oscillation (NAO) index (Hurrell and van Loon 1997) can be defined as the difference of sea level pressure between two stations located near the centers of Azores High and Icelandic Low. Positive values of the NAO index indicate a wet and warm winter in northern Canada and Greenland, in the Western Europe and in the Eastern US. Negative values of the NAO indicate a cold winter in these regions and storms are heading south

toward the Mediterranean Sea. This leads to increase rainfall and storm activity in North Africa and southern Europe.

To make a faithful comparison with other existing models, the used data set in this experiment is formed by 712 mean monthly NAO index values given by daily measures of the NAO index since January 1950 to April 2009. Six values are used to predict $y(t + 1)$, and the input–output data format is defined by: $[\text{inputs}; \text{output}] = [y(t - 5), y(t - 4), y(t - 3), y(t - 2), y(t - 1), y(t); y(t + 1)]$, where t defines an identifier of the month/year of measure and $y(t)$ presents the NAO monthly index. From the 712 data points, the first 500 data points are used for training and the remaining 212 are used for the test. The data set is available from: <http://www.cpc.ncep.noaa.gov/products/precip/CWlink/pna/nao.shtml>.

After 12 global iterations of learning, 2 global generations and 480 number of FEs, a near-optimum T2HFBFS was generated with 7 rules. The training and testing RMSEs of T2HFBFS are $1.6870e-07$ and $1.3021e-07$, respectively. Figure 8 shows the actual and the T2HFBFS predicted outputs. These results confirm again the capacity of the proposed system to achieve high levels of accuracy using a reduced number of rules. In fact, this is due to the multi-objective nature of the learning process which allows considering the two performance measures of accuracy and

Table 3 Comparison results of Box and Jenkins time series

Approach	Training RMSE	Testing RMSE	Rules
Fuzzy-GA (Wadhawan et al. 2013)	0.2779	–	8
BRB (Chen et al. 2013)	0.2939	0.4616	25
ANFIS (Samanta 2011)	0.0374	0.0640	–
Fuzzy-MA (Wadhawan et al. 2013)	0.0871	–	8
BBFNN (Dhahri et al. 2013)	0.2258	0.3876	–
SMN-COPSO (Samanta 2011)	0.2151	0.3416	–
FNT model (Chen et al. 2005)	0.0257	0.0264	–
FBeM (Leite et al. 2011)	0.0421	–	3
IT2 IFLS (Eyoh et al. 2018)	–	0.0249	4
T2HFBFS	0.0044	0.0108	4

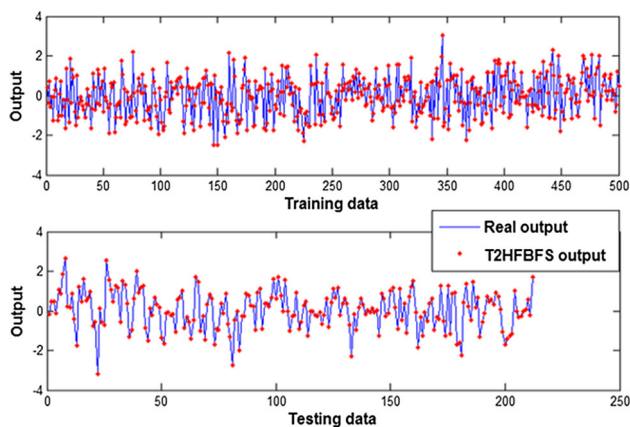


Fig. 8 Results of NAO index prediction for training and testing data

interpretability at the same time. Also, the proposed hierarchical fuzzy design contributes to the improvement in the system's approximation capacity by generating more accurate and interpretable solutions.

As given in Table 4, the performance of T2HFBFS is compared with that of the F-transforms (Di Martino et al. 2011), Wang–Mendel method (Wang and Mendel 1992) and the local linear wavelet neural network (LLWNN) system (Chen et al. 2006). RMSE testing values presented in this table indicate that the T2HFBFS gives considerably smaller error values than its competitors.

7.1.4 Predicting the sunspot number time series

It is a real-world non-stationary and highly complex time series. This data set presents the yearly average relative number of sunspots observed (Izeman et al. 1985). It is recorded for the years 1700–1979. To make a meaningful comparison with other existing studies, samples between 1700 and 1920 are exploited as training data. And for the test, two additional sets are used: the first set is from 1921 to 1955 and the second is from 1956 to 1979. The used inputs are $y(t - 4), y(t - 3), y(t - 2)$ and $y(t - 1)$, and the predicted output is $y(t)$.

After attaining 12 global iterations, 2 global generations and 480 number of FEs, an optimum T2HFBFS was generated with a training RMSE value equal to $2.8301e-016$ and with a number of fuzzy rules equal to 6. Regarding the

Table 4 Comparison results of NAO index time series

Approach	Testing RMSE
F-transforms (Di Martino et al. 2011)	$6.88e-01$
LLWNN (Chen et al. 2006)	$6.72e-01$
Wang–Mendel (Wang and Mendel 1992)	$2.86e-02$
T2HFBFS	$1.30e-07$

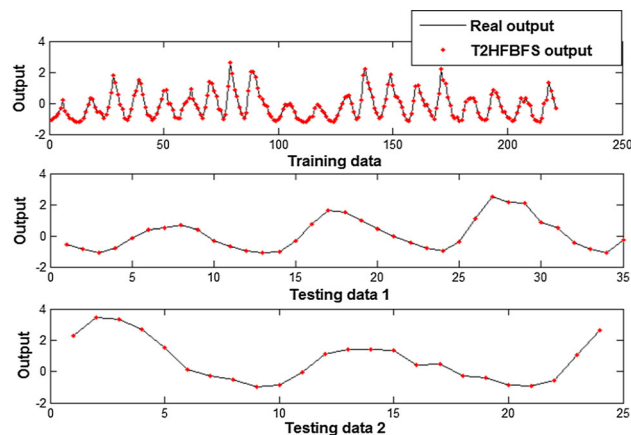


Fig. 9 Results of sunspot number prediction for training and testing data

testing part, the RMSE testing values for the first and the second testing sets are, respectively, $1.2741e-016$ and $1.6433e-016$. Figure 9 presents the actual and the T2HFBFS predicted outputs for training and testing data.

A comparison with existing studies is made in Table 5 such as the beta basis function neural network system trained by the artificial bee colony (BBFNN-ABC) (Dahri et al. 2012), the fuzzy wavelet neural network (FWNN) systems (Yilmaz and Oysal 2010), the recurrent fuzzy neural networks (RFNN) (Aliev et al. 2009) and the flexible beta basis function neural tree trained by a hybrid evolutionary algorithm (FBBFNT_EIP & HBFOA) (Bouaziz et al. 2013). As shown in the table, the T2HFBFS proves again its superiority against the other fuzzy and neural techniques by reaching lower training and testing errors.

7.2 T2HFBFS for high-dimensional biomedical classification problems

To more analyze the efficiency of the proposed system, the T2HFBFS is further applied for testing two examples of high-dimensional classification problems, which are the lung cancer and the prostate cancer data sets. These classification data sets were taken from the Kent Ridge Biomedical Data Set Repository (Available at <http://leo.ugr.es/elvira/DBCRepository/>). For this kind of problems having high input dimension, the SD feature selection algorithm is applied as an initial filtering step to select the most useful inputs. In addition, to more estimate the efficiency of the model, fivefold cross-validation method is implemented on the data in order to give a more honest assessment of the true classification accuracy. In general, k -fold cross-validation involves dividing randomly the data set into k folds, or groups, of approximately equal size. The first group of data is considered as a validation set, and the remaining

Table 5 Comparison results of sunspot number time series

Approach	Training RMSE	Testing 1 RMSE	Testing 2 RMSE
BBFNN-ABC (Dhahri et al. 2012)	0.0012	0.0018	0.0044
FWNN (Yilmaz and Oysal 2010)	0.0895	0.1093	0.1510
RFNN (Aliev et al. 2009)	–	0.074	0.210
FBBFNT_EIP& HBFOA (Bouaziz et al. 2013)	1.95e–10	4.15e–10	7.27e–10
T2HFBFS	2.83e–16	1.27e–16	1.64e–16

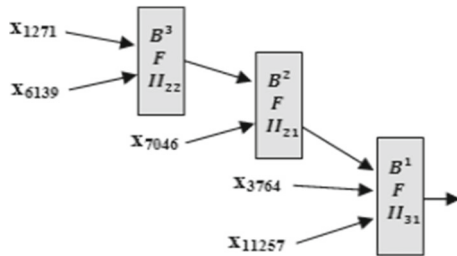


Fig. 10 Evolved T2HFBFS in the lung cancer case

$k - 1$ groups are treated as training sets. This process is repeated k times; each time, a different group of data is used as a validation set. The results are then given by computing the mean of the model skill scores.

7.2.1 Lung cancer data set

It is a classification data set utilized in the classification of adenocarcinoma (ADCA) and malignant pleural mesothelioma (MPM) of the lung. This data set is composed by 181 data points: Among them, 150 tissues are ADCA and the remaining 31 are MPM. Each data point is described by 12,533 features (genes).

After applying the SD feature selection algorithm on the data set, the number of the used features is reduced from 12,533 to 12 features. The classification results of fivefold cross-validation are, respectively, 98.87% for training data and 97.82% for the testing part. An optimum T2HFBFS with 6 fuzzy rules is generated after executing 5 global iterations, 1 global generation and 200 number of FEs. The evolved T2HFBFS is shown in Fig. 10.

As given in Table 6, the performance of the T2HFBFS is compared with other existing approaches, including a bootstrapping gene selection method (Pang et al. 2007), a linear discriminant analysis-based genetic algorithm (LDA-

GA) (Huerta et al. 2008), a multi-objective evolutionary algorithm based on interpretable fuzzy (MOEAIF) method (Wang and Palade 2011) and a fuzzy logic method combined with the ant colony optimization (ACO) algorithm and the principal components analysis (PCA) (Ayat and Rahi 2014). As can be seen from the table, the T2HFBFS outperformed all the other models and succeeds to achieve satisfactory classification performance with a small fuzzy rule base.

7.2.2 Prostate cancer data set

The prostate cancer database is used to more evaluate the classification performance of the system. This data set is employed for normal versus tumor classification. It is composed by 12,600 genes (features) and 136 data points (77 samples contain tumors and 59 do not contain tumors). The features introduce normalized gene expression values taken from the microarray image.

By applying the SD feature selection algorithm on the data, the number of features is reduced from 12,600 to 20 features. After performing 16 global iterations, 2 global generations and 640 number of FEs, an optimum T2HFBFS is generated with a rule base formed by 7 rules. The average classification rates on training and testing data are, respectively, 97.42% and 96.22%. The best-evolved T2HFBFS is shown in Fig. 11.

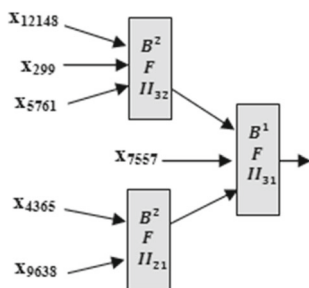
Table 7 makes some comparisons with other related works such as the granular support vector machines–recursive feature elimination (GSVM-RFE) algorithm (Tang et al. 2005), the transductive support vector machine (TSVM) method (Innocent and Kurian 2014) and the classification method based on association rule and information gain ratio on fuzzy rough set (AR-FRS-GR) theory (Innocent and Kurian 2014). As given in Table 7,

Table 6 Performance comparison in the case of lung cancer data set

Approach	Classification accuracy	Rules
Bootstrapping gene selection (Pang et al. 2007)	91.2%	–
MOEAIF (Wang and Palade 2011)	91.28%	3
LDA-GA (Huerta et al. 2008)	97.7% ~ 98.3%	–
Fuzy-PCA-ACO (Ayat and Rahi 2014)	98.21%	20
T2HFBFS	98.87%	6

Table 7 Performance comparison in the case of prostate cancer data set

Approach	Classification accuracy (%)
TSVM (Innocent and Kurian 2014)	92
AR-FRS-GR (Innocent and Kurian 2014)	94.3
GSVM-RFE (Tang et al. 2005)	90.18
T2HFBFS	97.42

**Fig. 11** Evolved T2HFBFS in the prostate cancer case

classification results prove again the superiority of the T2HFBFS against the other competing systems.

8 Conclusion

In this paper, the hierarchical design of type-2 fuzzy system is considered using a tree representation. The proposed T2HFBFS is formed by a set of low-dimensional rule bases arranged in a multi-level structure. The different sub-fuzzy models of the hierarchy are based on interval type-2 beta fuzzy sets in the premise, while the consequent part uses the TSK-type fuzzy reasoning. Regarding the learning process, two hybrid tasks of structure learning and parameter update are iteratively performed. The structure learning task relies on the MOEIP algorithm as a multi-objective algorithm in order to learn a population of T2HFBFSs. This step allows the generation of an optimal structure of T2HFBFS with a good balance of interpretability accuracy. The approximation ability of the obtained system is further enhanced by performing a second parameter tuning step using the PSOUM-IHS algorithm. This step allows the optimization of the IT2 BMF parameters and the consequent elements of rules. Simulations on time series forecasting problems and classification data sets show that the T2HFBFS outperforms other competing type-1/type-2 fuzzy approaches and neural network methods. In most of the cases, the proposed system is able to achieve higher levels of accuracy with a small rule base as compared to its competitors.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Aliev RA, Guirimov BG, Aliev RR (2009) Evolutionary algorithm-based learning of fuzzy neural networks, part 2: recurrent fuzzy neural networks. *Fuzzy Set Syst* 160(17):2553–2566
- Alimi AM (1997) Beta fuzzy basis functions for the design of universal robust neuro-fuzzy controllers. In: *Proceeding of Séminaire sur la Commande Robuste ses Applications: SCRA'97*, Nabeul, Tunisia, pp C1–C5
- Alimi AM (2000) The beta system: toward a change in our use of interval and general type-2 fuzzy logic systems. *Int J Manag* 15–19
- Alimi AM (2003) Beta neuro-fuzzy systems. *TASK Q J Spec Issue Neural Netw* 7(1):23–41
- Almaraashi M, John R (2011) Tuning of type-2 fuzzy systems by simulated annealing to predict time series. In: *Proceedings of the world congress on engineering, London, UK, vol 2*, pp 976–980
- Almaraashi M, John R, Hopgood A, Ahmadi S (2016) Learning of interval and general type-2 fuzzy logic systems using simulated annealing: theory and practice. *Inf Sci* 360:21–42
- Ammar M, Bouaziz S, Alimi AM, Abraham A (2013) Hybrid harmony search algorithm for global optimization. In: *Fifth world congress on nature and biologically inspired computing*, pp 69–75, Fargo, USA
- Ayat S, Rahi M (2014) Application of ant colony algorithm and principal components analysis in the diagnosis of lung cancer. *J Math Comput Sci (JMCS)* 13:343–352
- Balazs K, Botzheim J, Koczy LT (2010) Hierarchical fuzzy system modeling by genetic and bacterial programming approaches. In: *International conference on fuzzy systems*, pp 1–6. IEEE
- Benítez AD, Casillas J (2013) Multi-objective genetic learning of serial hierarchical fuzzy systems for large-scale problems. *Soft Comput* 17(1):165–194
- Bouaziz S, Alimi AM, Abraham A (2013) Evolving flexible beta basis function neural tree for nonlinear systems. In: *The 2013 international joint conference on neural networks (IJCNN)*, pp 1–8, Dallas, Texas
- Bouaziz S, Dhahri H, Alimi AM, Abraham A (2016) Evolving flexible beta basis function neural tree using extended genetic programming & hybrid Artificial Bee Colony. *Appl Soft Comput* 47:653–668
- Boutleux E, Dubuisson B (1996) Fuzzy pattern recognition to characterize evolutionary complex systems. Application to the French telephone network. In: *The fifth IEEE international conference on fuzzy systems fuzz-IEEE '96, vol 2*, pp 780–785
- Box GEP, Jenkins GM (1976) *Time series analysis forecasting and control*. Holden Day, San Francisco
- Chellapilla K (1998) Evolving computer programs without subtree crossover. *IEEE Trans Evol Comput* 1(3):209–216
- Chen Y, Dong J, Yang B (2004) Automatic design of hierarchical TS-FS model using ant programming and PSO algorithm. In: *International conference on artificial intelligence: methodology, systems, and applications*, pp 285–294. Springer
- Chen Y, Yang B, Dong J, Abraham A (2005) Time-series forecasting using flexible neural tree model. *Inf Sci* 174:219–235

- Chen Y, Yang B, Dong J (2006) Time-series prediction using a local linear wavelet neural network. *Neurocomputing* 69(4–6):449–456
- Chen Y, Yang B, Abraham A, Peng L (2007) Automatic design of hierarchical Takagi-Sugeno type fuzzy systems using evolutionary algorithms. *IEEE Trans Fuzzy Syst* 15(3):385–397
- Chen YW, Yang JB, Xu DL, Yang SL (2013) On the inference and approximation properties of belief rule based systems. *Inf Sci* 234:121–135
- Cheong F, Lai R (2007) Designing a hierarchical fuzzy logic controller using the differential evolution approach. *Appl Soft Comput* 7(2):481–491
- Chiu S (1994) Fuzzy model identification based on cluster estimation. *J Intell Fuzzy Syst* 2(3):267–278
- da Silva CG (2008) Time series forecasting with a non-linear model and the scatter search meta-heuristic. *Inf Sci* 178(16):3288–3299 **(Including Special Issue: Recent advances in granular computing)**
- Deb K, Agrawal S, Pratap A, Meyarivan T (2002) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
- Dahri H, Alimi AM, Abraham A (2012) Designing beta basis function neural network for optimization using Artificial Bee Colony (ABC). In: WCCI 2012 IEEE world congress on computational intelligence, pp. 1–7, Brisbane, Australia
- Dahri H, Alimi AM, Abraham A (2013) Hierarchical particle swarm optimization for the design of beta basis function neural network. In: Abraham A, Thampi S (eds) *Intelligent informatics*. Springer, Berlin, pp 193–205
- Di Martino F, Loia V, Sessa S (2011) Fuzzy transforms method in prediction data analysis. *Fuzzy Sets Syst* 180(1):146–163
- Eyoh I, John R, De Maere G, Kayacan E (2018) Hybrid learning for interval type-2 intuitionistic fuzzy logic systems as applied to identification and prediction problems. *IEEE Trans Fuzzy Syst* 26(5):2672–2685
- Fernández A, del Jesus MJ, Herrera F (2009) Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. *Int J Approx Reason* 50(3):561–577
- Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. *Simulation* 76(2):60–68
- Hagras H (2004) Hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Trans Fuzzy Syst* 12(4):524–539
- Huerta EB, Duval B, Hao JK (2008) Gene selection for microarray data by a LDA-based genetic algorithm. In: IAPR international conference on pattern recognition in bioinformatics, pp 250–261, Springer
- Hüllermeier E (2005) Fuzzy methods in machine learning and data mining: status and prospects. *Fuzzy Set Syst* 156(3):387–406
- Hurrell JW, van Loon H (1997) Decadal variations in climate associated with the North Atlantic Oscillation. *Clim Change* 36:301–326
- Innocent N, Kurian M (2014) Cancer prediction based on gene expression data through association rule based classification and fuzzy rough set attribute reduction on information gain ratio. *Int J Res Appl Sci Eng Technol* 2(4):42–46
- Izeman AJ (1985) J. R. Wolf, the Zurich sunspot relative numbers. *Math Intel* 7(1):27–33
- Jahromi MZ, Moosavi MR (2011) Designing cost-sensitive fuzzy classification systems using rule-weight. In: The first international conference on advances in information mining and management (IMMM 2011), pp 168–173
- Jarraya Y, Bouaziz S, Alimi AM, Abraham A (2013) Fuzzy modeling system based on hybrid evolutionary approach. In: 13th International conference on hybrid intelligent systems (HIS), Yasmine Hamamet, Tunisia
- Jarraya Y, Bouaziz S, Alimi AM, Abraham A (2014) Multi-agent evolutionary design of beta fuzzy systems. In: The 2014 IEEE international conference on fuzzy systems (FUZZ-IEEE 2014), Beijing, China
- Jarraya Y, Bouaziz S, Alimi AM, Abraham A (2015) Evolutionary multi-objective optimization for evolving hierarchical fuzzy system. In: 2015 IEEE congress on evolutionary computation (CEC 2015), Sendai, Japan
- Leite D, Gomide F, Ballini R, Costa P (2011) Fuzzy granular evolving modeling for time series prediction. In: 2011 IEEE international conference on fuzzy systems, Taipei, Taiwan
- León IC, Taylor PC (2015) Memetic type-2 fuzzy system learning for load forecasting. In: 2015 Conference of the international fuzzy systems association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15), Gijón, Spain
- Liang Q, Mendel JM (1999) An introduction to type-2 TSK fuzzy logic systems. In: *Proceedings IEEE international conference on fuzzy systems*, vol 3, pp 1534–1539. Seoul, South Korea
- Lin LC, Lee GY (1999) Hierarchical fuzzy control for C-axis of CNC tuning centers using genetic algorithms. *J Intell Robot Syst* 25(3):255–275
- Mackey MC, Glass L (1977) Oscillation and chaos in physiological control systems. *Science* 197(4300):287–289
- Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. *Appl Math Comput* 188(2):1567–1579
- Mendel J, John R (2002) Type-2 fuzzy sets made simple. *IEEE Trans Fuzzy Syst* 10(2):117–127
- Musilek P, Lau A, Reformat M, Wyard-scot L (2006) Immune programming. *Inf Sci* 176(8):972–1002
- Ojha V, Sinsel V, Abraham A (2017) Multiobjective programming for type-2 hierarchical fuzzy inference trees. *IEEE Trans Fuzzy Syst* 26(2):915–936
- Pang S, Havukkala I, Hu Y, Kasabov N (2007) Classification consistency analysis for bootstrapping gene selection. *Neural Comput Appl* 16:527–539
- Paul S, Kumar S (2002) Subsethood-product fuzzy neural inference system (SuPFuNIS). *IEEE Trans Neural Netw* 13(3):578–599
- Pohjalainen J, Räsänen O, Kadioglu S (2015) Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits. *Comput Speech Lang* 29:145–171
- Raju GVS, Zhou J (1993) Adaptive hierarchical fuzzy controller. *IEEE Trans Syst Man Cybern* 23(4):973–980
- Salgado P (2008) Rule generation for hierarchical collaborative fuzzy system. *Appl Math Model* 32:1159–1178
- Samanta B (2011) Prediction of chaotic time series using computational intelligence. *Expert Syst Appl* 38:11406–11411
- Shimozima K, Fukuda T, Hasegawa Y (1995) Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm. *Fuzzy Sets Syst* 71(3):295–309
- Singhala P, Shah DN, Patel B (2014) Temperature control using fuzzy logic. *Int J Instrum Control Syst (IJICS)* 4(1)
- Tanaka K, Sano M (1994) A robust stabilization problem of fuzzy control systems and its application to backing up control of a truck-trailer. *IEEE Trans Fuzzy Syst* 2:119–134
- Tang Y, Zhang YQ, Huang Z, Hu X (2005) Granular SVM-RFE gene selection algorithm for reliable prostate cancer classification on microarray expression data. In: *Proceedings of the 5th IEEE symposium on bioinformatics and bioengineering (BIBE'05)*, pp 290–293

- Uslan V, Seker H, John R (2014) A support vector-based interval type-2 fuzzy system. In: IEEE international conference on fuzzy systems, pp 2396–2401, Beijing, China
- Wadhawan S, Goel G, Kaushik S (2013) Data driven fuzzy modeling for Sugeno and Mamdani type fuzzy model using memetic algorithm. *Int J Inf Technol Comput Sci* 5(8):24–37
- Wang LX, Mendel JM (1992) Generating fuzzy rules by learning from examples. *IEEE Trans Syst Man Cybern* 22(6):1414–1427
- Wang Z, Palade V (2011) Building interpretable fuzzy models for high dimensional data analysis in cancer diagnosis. In: *BMC genomics*, vol 12, no 2, p S5, BioMed Central
- Yilmaz S, Oysal Y (2010) Fuzzy wavelet neural network models for prediction and identification of dynamical systems. *IEEE Trans Neural Netw* 21:1599–1609
- Zadeh LA (1975) The concept of a linguistic variable and its application to approximate reasoning. *Inf Sci* 8(3):199–249

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.