



# T2RFIS: type-2 regression-based fuzzy inference system

Krzysztof Wiktorowicz<sup>1</sup>

Received: 22 July 2022 / Accepted: 28 June 2023 / Published online: 20 July 2023

© The Author(s) 2023

## Abstract

This article discusses a novel type-2 fuzzy inference system with multiple variables in which no fuzzy rules are explicitly defined. By using a rule-free system, we avoid the serious disadvantage of rule-based systems, which are burdened with the curse of dimensionality. In the proposed system, Gaussian membership functions are used for its inputs, and linearly parameterized system functions are used to obtain its output. To obtain the system parameters, a genetic algorithm with multi-objective function is applied. In the presented method, the genetic algorithm is combined with a feature selection method and a regularized ridge regression. The objective functions consist of a pair in which one function is defined as the number of active features and the other as the validation error for regression models or the accuracy for classification models. In this way, the models are selected from the Pareto front considering some compromise between their quality and simplification. Compared to the author's previous work on the regression-based fuzzy inference system, a new inference scheme with type-2 fuzzy sets has been proposed, and the quality has been improved compared to the system based on type-1 fuzzy sets. Four experiments involving the approximation of a function, the prediction of fuel consumption, the classification of breast tissue, and the prediction of concrete compressive strength confirmed the efficacy of the presented method.

**Keywords** Type-2 fuzzy systems · Regression models · Multi-objective optimization · Feature selection

## 1 Introduction

One of the categories of fuzzy systems is systems based on type-2 fuzzy sets [101]. In these systems, which are an extension of systems with type-1 fuzzy sets, we are dealing with the uncertainty associated with defining the membership functions. In the last 20 years, there has been a significant increase in scientists' interest in systems with type-2 fuzzy sets. This is evidenced by the search result for the phrase 'type-2 fuzzy' on the Web of Science website, where we receive over 6000 records in mid-2023. The scope of application of systems with type-2 fuzzy sets is very wide. This is because these systems have a greater potential (by having more degrees of freedom) than classical systems with type-1 fuzzy sets. Here, we can cite

some sample articles from the last five years in the following areas (in alphabetical order): chaotic systems [25, 42, 56, 65], classification [9, 21, 28, 37, 61, 108], cognitive maps [1–3], communication systems [18, 19, 29, 44, 45, 74, 95, 97, 107], computing with words [41, 94], control systems [35, 63, 67, 71, 73, 76, 77, 99, 102], data clustering [13, 34, 54, 75, 96, 100, 105], decision making [16, 39, 46, 62, 70, 78, 84, 86, 87, 89], differential equations [6, 8, 58], fault detection [36, 64], image processing [31, 40, 48, 52, 80], jump systems [17, 47, 59, 69, 104, 106], linear programming [33, 38, 43], time series forecasting [51, 66, 68, 72].

Although the results for type-2 fuzzy systems are promising, the underlying problem is that the processing times for these systems are higher than for type-1 fuzzy systems. This is mainly due to the fact that these systems are more complex and have more parameters that require tuning. Therefore, the paper [55] proposes the following directions that scientists could explore (apart from type reduction, they also apply to type-1 fuzzy systems):

✉ Krzysztof Wiktorowicz  
kwiktor@prz.edu.pl

<sup>1</sup> Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, al. Powstancow Warszawy 12, 35–959 Rzeszow, Poland

- Model optimization—the choice of membership functions, rules, and operations is still a big open question,
- Type reduction—this time-consuming operation should still be improved,
- Defuzzification—despite the existence of many defuzzification methods, there is a need to improve their efficiency,
- Computational complexity—this issue can still be explored because of the curse of dimensionality,
- Hybridization—the search for new learning algorithms is desirable to improve the overall performance.

In addition, the authors point to the need for practical applications in areas such as hardware implementations and applications in medical diagnostics, in the area of Big Data, or in robotics. The literature review presented in the following cites some solutions that exist in the literature in the areas mentioned above.

### 1.1 Related work

It should be stressed that most of the solutions proposed in the literature are related to systems based on fuzzy rules. Such systems are limited in their use because of the curse of dimensionality associated with the exponential increase in the number of rules. To optimize fuzzy models, some attempts have been made to develop their structure without a rule base. The paper [60] focuses on the synthesis of a fuzzy logic control system using a new analytical approach without any rule base. Unfortunately, no way to select fuzzy system parameters is provided, especially when using observed data. In the paper [88], the authors present a new type of fuzzy inference system similar to that presented in [60]. Their system does not have a rule base, and the number of its parameters increases linearly. The authors do not provide a tuning method based on the observed data; it is only mentioned that a method similar to backpropagation is used. Another way to reduce the number of fuzzy rules is to create hierarchical systems, as described in [50], which proposes a type-2 fuzzy hierarchical system for high-dimensional data modeling.

The problems of type reduction and defuzzification in type-2 fuzzy systems are still the subject of research and have not yet been finally solved. This paragraph presents examples of papers devoted to these issues. The proposition of two type reduction algorithms along with a comprehensive study using various programming languages is presented in [10]. In the paper [14], the authors propose an inference and type reduction method for constrained interval type-2 fuzzy sets using the concept of switch indices. The paper [24] presents a technique, named stratic defuzzification, for discretized general type-2 fuzzy sets. This method is based on the transformation of a type-2

fuzzy set into a type-1 fuzzy set. Two algorithms, called binary algorithms, are proposed in [49] to calculate the centroid of interval type-2 fuzzy sets. A method to calculate the center of gravity of polygonal interval type-2 fuzzy sets is introduced in [57]. The authors' method can be applied both on discrete and continuous domains without the need of discretization. A non-iterative method to obtain the center of centroids for an interval type-2 fuzzy set is presented in [5]. This method is based on the weighted sum of the centroids for the lower and upper membership functions. The study [11] presents three types of sampling-based reduction algorithm for general type-2 fuzzy systems that are extended versions of the algorithms proposed in the literature for interval type-2 fuzzy systems. In the paper [23], the authors present a theoretical approach to the type reduction problem using the Chebyshev inequality. Through their method, they obtain the centroids and the bounds for type-1 and interval type-2 fuzzy numbers. The paper [93] proposes a type reduction method for general type-2 fuzzy systems using an  $\alpha$ -plane representation. In this representation, a series of  $\alpha$ -planes is applied to decompose a general type-2 fuzzy set. An experimental evaluation of various defuzzification algorithms, namely the Karnik-Mendel procedure, the Nie-Tan method, the  $q$  factor method, and the modified  $q$  factor method, can be found in [103].

Hybridization of type-2 fuzzy systems with various learning algorithms has been considered in the following sample papers. The paper [98] presents a hybrid approach to build a type-2 neural fuzzy system that incorporates particle swarm optimization and least-squares estimation. The authors of the paper [53] propose a hybrid mechanism for training type-2 fuzzy logic systems that uses a recursive square root filter to tune the type-1 consequent parameters and the steepest descent method to tune the interval type-2 antecedent parameters. In the paper [85], an evolving type-2 Mamdani neural fuzzy system is proposed. The work [15] introduces the extreme learning strategy to develop a fast training algorithm for the interval type-2 Takagi-Sugeno-Kang fuzzy logic systems. The paper [4] proposes a hybrid learning mechanism for type-2 fuzzy systems that uses the recursive orthogonal least-squares algorithm to tune the type-1 consequent parameters and the backpropagation algorithm to tune the type-2 antecedent parameters. The work [79] presents an adaptive neuro-fuzzy inference system (ANFIS) that uses interval Gaussian type-2 fuzzy sets in the antecedent part and Gaussian type-1 fuzzy sets in the consequent part. The structure of the proposed ANFIS2 is very similar to that of the traditional ANFIS, except for an extra layer for type reduction. In the paper [22], the authors propose a design of an interval type-2 fuzzy logic system based on the quantum-behaved particle swarm optimization algorithm. A trapezoidal type-2 fuzzy

inference system is proposed in [30]. To optimize this system, a tensor unfolding structure training method is applied. A method of variable selection and sorting to construct the type-2 Takagi-Sugeno-Kang fuzzy inference system is described in [90]. This method selects independent variables using Chi-square statistics. The paper [12] presents an application of interval type-2 fuzzy logic interval system to forecast the parameters of a permanent magnetic drive. The authors use backpropagation and recursive least-squares algorithms to optimize the fuzzy logic system. In the paper [7], a composite framework is presented that uses the deep learning technique for tuning interval type-2 fuzzy systems. All paper mentioned above use rule-based fuzzy inference systems.

## 1.2 Goals and contributions

According to the directions for the development of type-2 fuzzy systems mentioned above, the goals of the solution proposed in this article are as follows. In the area of model optimization, propose a system that does not use fuzzy inference rules, which allows one to avoid the explosion of their number along with the increase in the number of inputs and fuzzy sets. In the area of defuzzification, eliminate this operation from a fuzzy system. Regarding the type reduction, replace this operation with the weighted sum of regression matrices. In terms of computational complexity, propose a system in which the complexity increases linearly or at most squaredly, instead of the exponential increase as in the systems known in the literature. Finally, in the area of hybridization, the use of ridge regression and multi-criteria optimization with feature selection to train the proposed system. In view of these goals, the main contributions can be summarized as follows:

- proposal of a new type-2 fuzzy inference scheme without explicitly defined rules,
- using a hybrid method combining regularized regression and global optimization with feature selection to train this system,
- improving performance over the previously published type-1 fuzzy inference system.

Moreover, the proposed method is tested on four different fuzzy modeling problems, which involve the approximation of a one-variable function and the prediction of fuel consumption.

This paper is a continuation of the author's previous work [92], in which the type-1 RFIS model has been discussed.

## 1.3 Paper structure

The structure of this paper is as follows. Section 2 provides the description of a type-2 regression-based fuzzy inference system with linearly parameterized system functions. The regression and design matrices used to design the system are introduced in Sect. 3. A training method to calculate the system function parameters and an illustrative example are presented in Sect. 4. Section 5 presents a training method for type-2 fuzzy sets. In Sect. 6, the calculation of the objective functions is described. Section 7 contains the description of the experimental results. Finally, the conclusions are given in Sect. 8.

## 1.4 Problem statement

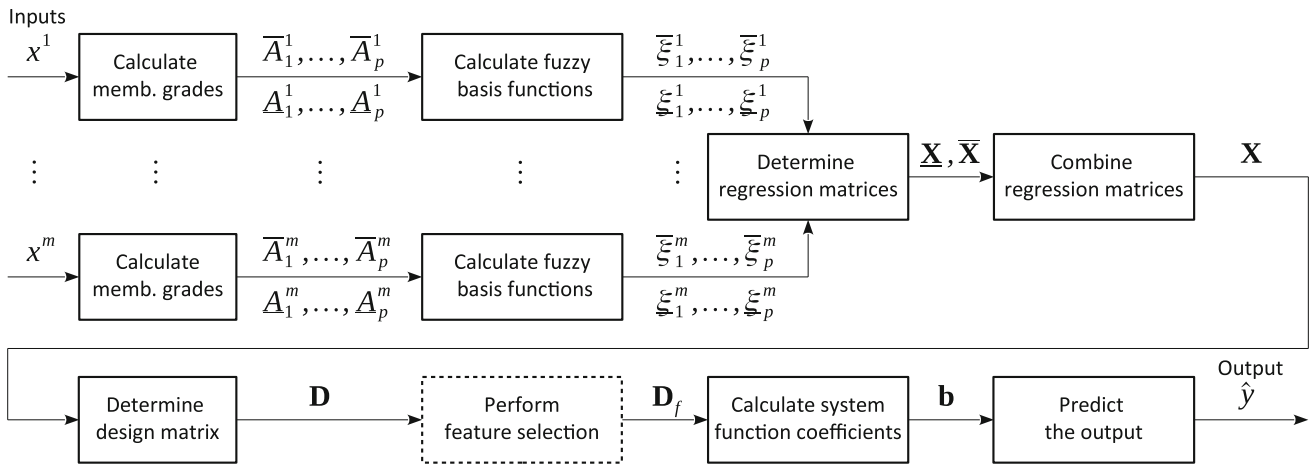
We consider the problem of training a type-2 regression-based fuzzy inference system (T2RFIS) with  $m$  inputs  $x^1, \dots, x^m$  and one output  $y$  (Fig. 1). The problem concerns the determination of type-2 fuzzy sets for the inputs of the system and the parameters of a system function used to obtain the output. To solve this problem, a hybrid training method is proposed, in which fuzzy sets are determined by a multi-objective genetic algorithm and the system function parameters by a regression method. Because we assume that the system function is linearly parameterized, this method is implemented here by regularized ridge regression. The models generated by the proposed method are simplified by applying a variable selection algorithm.

## 2 Type-2 fuzzy inference system

For the system inputs creating the vector  $\mathbf{x} = [x^1, \dots, x^m]$ , we define, for each input,  $p$  Gaussian fuzzy sets of type-2 (Fig. 2) described by the formulas

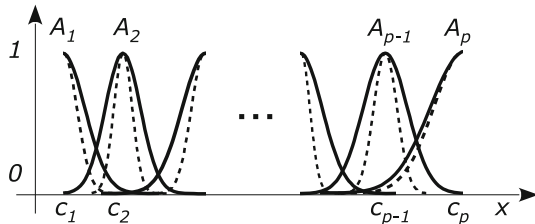
$$\begin{aligned} \underline{A}_j^1(x^1) &= g(x^1; c_j^1, \underline{\sigma}_j^1), & \overline{A}_j^1(x^1) &= g(x^1; c_j^1, \overline{\sigma}_j^1) \\ & \vdots & & \vdots \\ \underline{A}_j^m(x^m) &= g(x^m; c_j^m, \underline{\sigma}_j^m), & \overline{A}_j^m(x^m) &= g(x^m; c_j^m, \overline{\sigma}_j^m) \end{aligned} \quad (1)$$

where  $j = 1, 2, \dots, p$ ,  $x^k \in [c_1^k, c_p^k]$ ,  $k = 1, 2, \dots, m$ ,  $c_j^k$  is the center of a fuzzy membership function,  $\underline{\sigma}_j^k$  is the width of a lower fuzzy membership function,  $\overline{\sigma}_j^k$  is the width of an upper fuzzy membership function, and  $g(x; c, \sigma) = \exp(-0.5(x - c)^2/\sigma^2)$ . Using the defined fuzzy sets, we introduce a lower fuzzy basis function  $\underline{\xi}_j^k$  and an upper fuzzy basis function  $\overline{\xi}_j^k$  given by



**Fig. 1** Calculation scheme of the T2RFIS model. First, the membership grades (1) and the fuzzy basis functions (2) for the lower and upper membership functions are calculated, and then the lower (6) and upper (7) regression matrices are obtained. The regression matrices are combined to form one regression matrix (8), and the

design matrix (9) is determined. In the next steps, an optional feature selection is performed, resulting in the matrix  $\mathbf{D}_f$ , for which the parameters of the system function are calculated. In the end, the output of the system is predicted



**Fig. 2** Type-2 fuzzy sets for the inputs of the T2RFIS model

$$\underline{\xi}_j^k(x^k) = \frac{\underline{A}_j^k(x^k)}{\sum_{j=1}^p \underline{A}_j^k(x^k)}, \bar{\xi}_j^k(x^k) = \frac{\bar{A}_j^k(x^k)}{\sum_{j=1}^p \bar{A}_j^k(x^k)} \quad (2)$$

The lower fuzzy basis functions and upper fuzzy basis functions are written as vectors

$$\begin{aligned} \underline{\xi}(\mathbf{x}) &= [\underline{\xi}_1^1(x^1), \dots, \underline{\xi}_p^1(x^1), \dots, \underline{\xi}_1^m(x^m), \dots, \underline{\xi}_p^m(x^m)]_{1 \times p \cdot m} \\ \bar{\xi}(\mathbf{x}) &= [\bar{\xi}_1^1(x^1), \dots, \bar{\xi}_p^1(x^1), \dots, \bar{\xi}_1^m(x^m), \dots, \bar{\xi}_p^m(x^m)]_{1 \times p \cdot m} \end{aligned} \quad (3)$$

The output of the T2RFIS model is determined as

$$\hat{y} = \sum_{s=1}^t b_s d_s(\xi(\mathbf{x})) \quad (4)$$

where  $b_s$  is the unknown system function coefficient,  $t$  is the number of system function coefficients, and  $d_s(\xi(\mathbf{x}))$  is a linear or nonlinear expression dependent on the lower and upper fuzzy basis functions. The expressions  $d_s(\xi(\mathbf{x}))$  form a design matrix  $\mathbf{D}(\xi)$  as described in the next section. For the proposed T2RFIS model, the calculation scheme of the output  $\hat{y}$  is presented in Fig. 1.

### 3 Regression and design matrices

We assume that we know the pairs of observation data  $(\mathbf{x}_i, y_i)$ , where  $i = 1, 2, \dots, n$ . These data are written as the matrix  $\mathbf{X}_o$  and the vector  $\mathbf{y}$  of the form

$$\mathbf{X}_o = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}_{n \times m} = \begin{bmatrix} x_1^1 & \dots & x_1^m \\ x_2^1 & \dots & x_2^m \\ \vdots & & \vdots \\ x_n^1 & \dots & x_n^m \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{n \times 1} \quad (5)$$

Assuming these data, we introduce the following three regression matrices:

- the lower regression matrix

$$\begin{aligned} \underline{\mathbf{X}} &= \begin{bmatrix} \underline{\xi}(\mathbf{x}_1) \\ \underline{\xi}(\mathbf{x}_2) \\ \vdots \\ \underline{\xi}(\mathbf{x}_n) \end{bmatrix}_{n \times p \cdot m} \\ &= \begin{bmatrix} \underline{\xi}_1^1(x_1^1), \dots, \underline{\xi}_p^1(x_1^1), \dots, \underline{\xi}_1^m(x_1^m), \dots, \underline{\xi}_p^m(x_1^m) \\ \underline{\xi}_1^1(x_2^1), \dots, \underline{\xi}_p^1(x_2^1), \dots, \underline{\xi}_1^m(x_2^m), \dots, \underline{\xi}_p^m(x_2^m) \\ \vdots \\ \underline{\xi}_1^1(x_n^1), \dots, \underline{\xi}_p^1(x_n^1), \dots, \underline{\xi}_1^m(x_n^m), \dots, \underline{\xi}_p^m(x_n^m) \end{bmatrix} \end{aligned} \quad (6)$$

- the upper regression matrix

$$\begin{aligned} \bar{\mathbf{X}}_{n \times p \cdot m} &= \begin{bmatrix} \bar{\xi}(\mathbf{x}_1) \\ \bar{\xi}(\mathbf{x}_2) \\ \vdots \\ \bar{\xi}(\mathbf{x}_n) \end{bmatrix} \\ &= \begin{bmatrix} \bar{\xi}_1^1(x_1^1), \dots, \bar{\xi}_p^1(x_1^1), \dots, \bar{\xi}_1^m(x_1^m), \dots, \bar{\xi}_p^m(x_1^m) \\ \bar{\xi}_1^1(x_2^1), \dots, \bar{\xi}_p^1(x_2^1), \dots, \bar{\xi}_1^m(x_2^m), \dots, \bar{\xi}_p^m(x_2^m) \\ \vdots \\ \bar{\xi}_1^1(x_n^1), \dots, \bar{\xi}_p^1(x_n^1), \dots, \bar{\xi}_1^m(x_n^m), \dots, \bar{\xi}_p^m(x_n^m) \end{bmatrix} \end{aligned} \tag{7}$$

– the regression matrix

$$\begin{aligned} \mathbf{X}_{n \times p \cdot m} &= \frac{\underline{w}\mathbf{X} + \bar{w}\bar{\mathbf{X}}}{\underline{w} + \bar{w}} \\ &= \begin{bmatrix} \xi_1^1(x_1^1), \dots, \xi_p^1(x_1^1), \dots, \xi_1^m(x_1^m), \dots, \xi_p^m(x_1^m) \\ \xi_1^1(x_2^1), \dots, \xi_p^1(x_2^1), \dots, \xi_1^m(x_2^m), \dots, \xi_p^m(x_2^m) \\ \vdots \\ \xi_1^1(x_n^1), \dots, \xi_p^1(x_n^1), \dots, \xi_1^m(x_n^m), \dots, \xi_p^m(x_n^m) \end{bmatrix} \end{aligned} \tag{8}$$

where  $\xi_j^k = (\underline{w}\xi_j^k + \bar{w}\bar{\xi}_j^k)/(\underline{w} + \bar{w})$  and  $\underline{w}, \bar{w}$  are the weights of the lower and upper regression matrix, respectively. These weights are the variables subject to optimization, as described later in the article.

The elements of the lower and upper regression matrices are created from the elements of the vectors of the lower and upper fuzzy basis functions (3), respectively. The regression matrix  $\mathbf{X}$  is formed as a weighted arithmetic mean of the lower  $\mathbf{X}$  and upper  $\bar{\mathbf{X}}$  regression matrices. The elements of the described matrices are determined for all input data  $\mathbf{x}_i$ .

Moreover, we introduce the *design matrix*

$$\mathbf{D}_{n \times t} = \begin{bmatrix} d_1(\xi(\mathbf{x}_1)), \dots, d_t(\xi(\mathbf{x}_1)) \\ d_1(\xi(\mathbf{x}_2)), \dots, d_t(\xi(\mathbf{x}_2)) \\ \vdots \\ d_1(\xi(\mathbf{x}_n)), \dots, d_t(\xi(\mathbf{x}_n)) \end{bmatrix} \tag{9}$$

The matrix  $\mathbf{D}$ , calculated on the basis of the regression matrix (Fig. 1), consists of the expressions  $d_s(\xi(\mathbf{x}))$  of the system function used in formula (4). As in the case of regression matrices, the elements of the design matrix are determined for all input data  $\mathbf{x}_i$ . In the construction of this matrix, four types of regression functions are applied [83, 92]:

– ‘linear’—model contains a linear term for each predictor, for example

$$\hat{y} = b_1\xi_1 + b_2\xi_2 + b_3\xi_3 \tag{10}$$

– ‘purequadratic’—model contains linear and squared terms for each predictor, for example

$$\hat{y} = b_1\xi_1 + b_2\xi_2 + b_3\xi_3 + b_4\xi_1^2 + b_5\xi_2^2 + b_6\xi_3^2 \tag{11}$$

– ‘interactions’—model contains a linear term for each predictor and all products of pairs of distinct predictors, for example

$$\hat{y} = b_1\xi_1 + b_2\xi_2 + b_3\xi_3 + b_4\xi_1\xi_2 + b_5\xi_1\xi_3 + b_6\xi_2\xi_3 \tag{12}$$

– ‘quadratic’—model contains linear and squared terms for each predictor and all products of pairs of distinct predictors, for example

$$\begin{aligned} y &= b_1\xi_1 + b_2\xi_2 + b_3\xi_3 \\ &+ b_4\xi_1^2 + b_5\xi_2^2 + b_6\xi_3^2 \\ &+ b_7\xi_1\xi_2 + b_8\xi_1\xi_3 + b_9\xi_2\xi_3 \end{aligned} \tag{13}$$

The design matrix has the following sizes in subsequent examples:  $n \times 3, n \times 6, n \times 6,$  and  $n \times 9$ . The predictors of the lower and upper regression matrix form the columns of the design matrix.

### 4 Training system function parameters

Training the system function coefficients consists of calculating the elements of the vector  $\mathbf{b}$ . For this purpose, linear regression applied to the design matrix  $\mathbf{D}$  and the vector of output observations  $\mathbf{y}$  can be used. In this paper, a penalized least-squares method represented by ridge regression [26] is applied. In this method, the cost function  $J$  is given by

$$J = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \mathbf{b}^T \mathbf{b} \tag{14}$$

where the estimated output for the  $i$ th observation is calculated from

$$\hat{y}_i = \mathbf{D}(\mathbf{x}_i)\mathbf{b} \tag{15}$$

and  $\lambda > 0$  is a regularization parameter. For  $\lambda = 0$ , the ridge regression becomes an ordinary least-squares regression. The solution to the problem of minimizing the function  $J$  is given by

$$\mathbf{b} = (\mathbf{D}^T \mathbf{D} + \lambda \mathbf{I})^{-1} \mathbf{D}^T \mathbf{y} \tag{16}$$

where  $\mathbf{y} = [y_1, \dots, y_n]^T$  and  $\mathbf{I}$  is the identity matrix. Ridge regression, which has an additional parameter  $\lambda$ , offers the

important advantage of being a regularized regression, that is, it can be used for ill-conditioned problems. This can happen when the matrix  $\mathbf{D}^T\mathbf{D}$  is close to the singular matrix. Furthermore, this method is very fast because the vector  $\mathbf{b}$  is calculated directly from all the data in the matrix  $\mathbf{D}$  and the vector  $\mathbf{y}$ .

### 4.1 Illustrative example

In this example, the goal is to train the T2RFIS model in the regression problem for a very small amount of data [92]. The fuzzy system has one input ( $m = 1$ ) denoted by  $x$  and one output denoted by  $y$ . We assume two vectors of observations of the form  $\mathbf{x} = [1, 2, 3, 4]^T$  and  $\mathbf{y} = [6, 5, 7, 10]^T$ . For input  $x$ , we define  $p = 2$  fuzzy sets of type-2, where the lower and upper membership functions (1) are defined as (Fig. 3)

$$\underline{A}_1(x) = g(x, 1, 1.8), \underline{A}_2(x) = g(x, 4, 1.2) \tag{17}$$

$$\overline{A}_1(x) = g(x, 1, 2.6), \overline{A}_2(x) = g(x, 4, 1.8) \tag{18}$$

The parameters  $c$  and  $\sigma$  in the sets described in Equations (17) and (18) have been arbitrarily chosen. However, in the examples described in Section 7 ‘Experimental results,’ these parameters are calculated by a genetic algorithm.

For the defined fuzzy sets, the lower regression matrix (6) and the upper regression matrix (7) have the form

$$\underline{\mathbf{X}}_{4 \times 2} = \begin{bmatrix} \underline{\xi}_1(x_1) & \underline{\xi}_2(x_1) \\ \underline{\xi}_1(x_2) & \underline{\xi}_2(x_2) \\ \underline{\xi}_1(x_3) & \underline{\xi}_2(x_3) \\ \underline{\xi}_1(x_4) & \underline{\xi}_2(x_4) \end{bmatrix} = \begin{bmatrix} 0.8301 & 0.1699 \\ 0.6511 & 0.3489 \\ 0.4631 & 0.5369 \\ 0.3254 & 0.6746 \end{bmatrix} \tag{19}$$

$$\overline{\mathbf{X}}_{4 \times 2} = \begin{bmatrix} \overline{\xi}_1(x_1) & \overline{\xi}_2(x_1) \\ \overline{\xi}_1(x_2) & \overline{\xi}_2(x_2) \\ \overline{\xi}_1(x_3) & \overline{\xi}_2(x_3) \\ \overline{\xi}_1(x_4) & \overline{\xi}_2(x_4) \end{bmatrix} = \begin{bmatrix} 0.8004 & 0.1996 \\ 0.6326 & 0.3674 \\ 0.4647 & 0.5353 \\ 0.3395 & 0.6605 \end{bmatrix} \tag{20}$$

where the lower fuzzy basis functions and the upper fuzzy basis functions (8) are given by

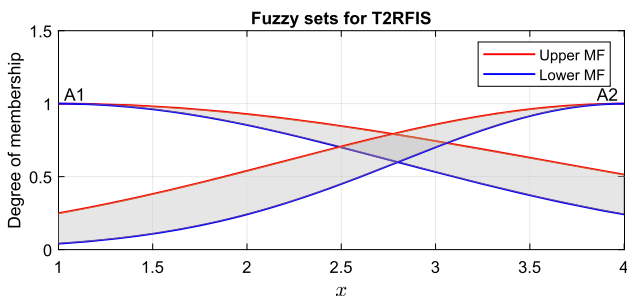


Fig. 3 Fuzzy sets for the input  $x$  of the T2RFIS model

$$\underline{\xi}_1(x) = \frac{\underline{A}_1(x)}{\underline{A}_1(x) + \underline{A}_2(x)}, \underline{\xi}_2(x) = \frac{\underline{A}_2(x)}{\underline{A}_1(x) + \underline{A}_2(x)} \tag{21}$$

$$\overline{\xi}_1(x) = \frac{\overline{A}_1(x)}{\overline{A}_1(x) + \overline{A}_2(x)}, \overline{\xi}_2(x) = \frac{\overline{A}_2(x)}{\overline{A}_1(x) + \overline{A}_2(x)} \tag{22}$$

For the weights  $\underline{w} = \overline{w} = 1$  in (8), the regression matrix  $\mathbf{X}$  has the form of

$$\mathbf{X}_{4 \times 2} = \begin{bmatrix} \underline{\xi}_1(x_1) & \underline{\xi}_2(x_1) \\ \underline{\xi}_1(x_2) & \underline{\xi}_2(x_2) \\ \underline{\xi}_1(x_3) & \underline{\xi}_2(x_3) \\ \underline{\xi}_1(x_4) & \underline{\xi}_2(x_4) \end{bmatrix} = \begin{bmatrix} 0.8153 & 0.1847 \\ 0.6419 & 0.3581 \\ 0.4639 & 0.5361 \\ 0.3324 & 0.6676 \end{bmatrix} \tag{23}$$

Assuming that the design matrix (9) is of type ‘interactions’, we obtain

$$\begin{aligned} \mathbf{D}_{4 \times 3} &= \begin{bmatrix} d_1(\xi(x_1)) & d_2(\xi(x_1)) & d_3(\xi(x_1)) \\ d_1(\xi(x_2)) & d_2(\xi(x_2)) & d_3(\xi(x_2)) \\ d_1(\xi(x_3)) & d_2(\xi(x_3)) & d_3(\xi(x_3)) \\ d_1(\xi(x_4)) & d_2(\xi(x_4)) & d_3(\xi(x_4)) \end{bmatrix} \\ &= \begin{bmatrix} \underline{\xi}_1(x_1) & \underline{\xi}_2(x_1) & \underline{\xi}_1(x_1)\underline{\xi}_2(x_1) \\ \underline{\xi}_1(x_2) & \underline{\xi}_2(x_2) & \underline{\xi}_1(x_2)\underline{\xi}_2(x_2) \\ \underline{\xi}_1(x_3) & \underline{\xi}_2(x_3) & \underline{\xi}_1(x_3)\underline{\xi}_2(x_3) \\ \underline{\xi}_1(x_4) & \underline{\xi}_2(x_4) & \underline{\xi}_1(x_4)\underline{\xi}_2(x_4) \end{bmatrix} \\ &= \begin{bmatrix} 0.8153 & 0.1847 & 0.1506 \\ 0.6419 & 0.3581 & 0.2299 \\ 0.4639 & 0.5361 & 0.2487 \\ 0.3324 & 0.6676 & 0.2219 \end{bmatrix} \end{aligned} \tag{24}$$

Applying the ridge regression (16) with the regularization parameter  $\lambda = 0$ , we obtain the vector of parameters

$$\mathbf{b} = [9.827, 24.72, -43.86]^T \tag{25}$$

which means that the output of the T2RFIS model is described by the function

$$\hat{y} = 9.827\underline{\xi}_1 + 24.72\underline{\xi}_2 - 43.86\underline{\xi}_1\underline{\xi}_2 \tag{26}$$

**Table 1** Results for the T1RFIS and T2RFIS models in the illustrative example;  $y_i$  is the training data,  $\hat{y}_i$  is the estimated value, and  $r_i = y_i - \hat{y}_i$  is the residual (error)

$i$	$y_i$	T1RFIS [92]		T2RFIS	
		$\hat{y}_i$	$r_i$	$\hat{y}_i$	$r_i$
1	6	5.942	0.0584	5.974	0.0262
2	5	5.155	-0.1549	5.081	-0.0807
3	7	6.835	0.1653	6.906	0.0937
4	10	10.06	-0.0633	10.04	-0.0392

The list of estimated values  $\hat{y}_i$  and the residuals (errors)  $r_i = y_i - \hat{y}_i$  for the T1RFIS and T2RFIS models are presented in Table 1. For the models obtained, the square root of the mean square error defined as  $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n r_i^2}$  is equal to 0.1432 and 0.0662, respectively. The approximation of the observation data using the T1RFIS and T2RFIS models is shown in Fig. 4.

## 5 Training fuzzy sets

### 5.1 Multi-objective genetic algorithm

The genetic algorithm (GA) [27, 82, 91] is a global optimization method inspired by the biological process of evolution. In each generation of GA, individuals are selected from the current population as parents and used to obtain children for the next generation. The population evolves in subsequent generations toward the optimal solution. To create the next generation from the current population, GA uses three main types of rules: selection, crossover, and mutation. In addition to these rules, dominance, rank, and crowding distance are also used in the multi-objective GA (MGA) algorithm [82]. A more detailed description of these terms is given in [92].

### 5.2 Structure of an individual

The structure of an individual for the multi-objective GA algorithm is presented in Fig. 5. This individual consists of  $3pm + 4$  variables, where  $p$  is the number of fuzzy sets defined for  $m$  inputs of the T2RFIS model. The variables constituting the elements of the individual are the centers  $c_1, \dots, c_{p-m}$  of the membership functions, the widths  $\underline{\sigma}_1, \dots, \underline{\sigma}_{p-m}$  of the upper membership functions, the distances  $\delta_1, \dots, \delta_{p-m}$  between the lower and upper membership functions, the parameter  $q_1$  for a feature selection

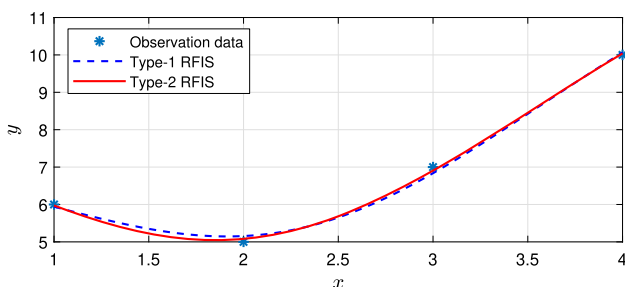


Fig. 4 Approximation of observation data using the T1RFIS and T2RFIS models

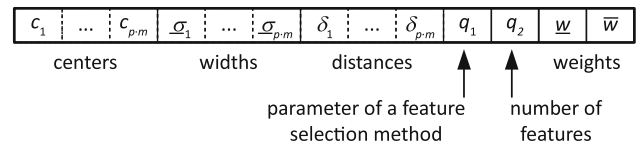


Fig. 5 Structure of an individual for the multi-objective genetic algorithm and T2RFIS models;  $c_1, \dots, c_{p-m}$  are the centers of membership functions,  $\underline{\sigma}_1, \dots, \underline{\sigma}_{p-m}$  are the widths of lower membership functions,  $\delta_1, \dots, \delta_{p-m}$  are the distances between lower and upper membership functions,  $q_1$  is the parameter of a feature selection method,  $q_2$  is the number of selected features, and  $\underline{w}, \bar{w}$  are the weights of regression matrices

method, the number  $q_2$  of selected features and the weights  $\underline{w}, \bar{w}$  of regression matrices.

## 6 Determining objective functions

### 6.1 Feature selection

Feature selection is a process by which a subset of features is selected to build the model. The main rationale for using the feature selection technique is that the data can contain certain features that are redundant or irrelevant, and therefore, can be deleted without losing a significant amount of information. The use of feature selection can improve prediction performance, reduce training time, and avoid the curse of dimensionality. In the proposed approach, feature selection is used to select predictors (columns) from the design matrix. This matrix can contain multiple columns, which can lead to complex models with consequent performance degradation. In the previous work of the author [92], the following feature selection methods are considered [83]: F-test, ReliefF, NCA (neighborhood component analysis), and Lasso (least absolute shrinkage and selection operator). Since a feature selection method is called inside the objective function, it should be fast. Based on the work [92], in which the calculation time of the feature selection algorithms mentioned above is analyzed, it can be concluded that the F-test method turned out to be the fastest. For this reason, it is selected for use in the experiments presented in Sect. 7.

### 6.2 Objective functions

In the proposed approach, we use multi-criteria optimization, thus we have to define objective functions that are minimized during optimization. It is proposed to use two objective functions, one responsible for the complexity of the model ( $f_1$ ) and the other responsible for its accuracy ( $f_2$ ). The values of these functions are determined as follows:

$$\begin{cases} f_1 = N \\ f_2 = \sqrt{\frac{1}{V} \sum_{k=1}^V (y_k - \hat{y}_k)^2} \end{cases} \quad (27)$$

where  $f_1$  is specified by the number of active features ( $N$ ) in the design matrix, and  $f_2$  is specified by the root of the mean squared validation error (RMSE). The number  $N$  denotes the number of predictors of the design matrix after applying a feature selection method. In the formula describing the function  $f_2$ ,  $V$  is the number of observations in the validation set,  $y_k$  is the  $k$ th validation observation, and  $\hat{y}_k$  is the estimate for the  $k$ th validation observation. The value of this estimate is given by formula (15).

In addition to the prediction of real value for regression models, we will also consider the prediction of classes for classification models. In this case, the pair of objective functions are as follows:

$$\begin{cases} f_1 = N \\ f_2 = \frac{n_{mis}}{n_{all}} \end{cases} \quad (28)$$

where  $f_1$  denotes, as for regression models, the number of selected features,  $f_2$  is the confusion error,  $n_{mis}$  is the number of misclassified validation records, and  $n_{all}$  is the number of all validation records.

### 6.3 Calculation scheme

Figure 6 shows the calculation scheme of the objective functions for the T2RFIS models. In the first step, an individual is decoded, giving the parameters of type-2 fuzzy sets, that is, the centers  $c_j^k$ , upper widths  $\bar{\sigma}_j^k$ , and

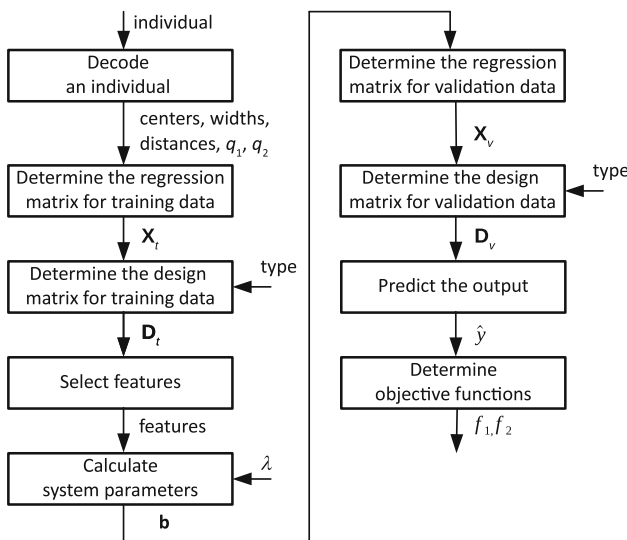


Fig. 6 Calculation scheme of the objective functions for the T2RFIS model

distances  $\delta_j^k$ . Additionally, the parameter  $q_1$  for a feature selection method and the parameter  $q_2$  for the number of features are obtained. On the basis of the lower widths and distances, the upper widths are determined from the relationship

$$\bar{\sigma}_j^k = \underline{\sigma}_j^k + \delta_j^k \quad (29)$$

In the next two steps, the regression matrix  $\mathbf{X}_t$  and the design matrix  $\mathbf{D}_t$  of a given type are calculated for the training data. The design matrix is then truncated using a feature selection method, and the parameters of the system function  $\mathbf{b}$  are obtained by the ridge regression with the given value of the parameter  $\lambda$ . Before predicting the output, the regression matrix  $\mathbf{X}_v$  and the design matrix  $\mathbf{D}_v$  are determined for the validation data. On their basis, the system output  $\hat{y}$  is predicted, and after that, the objective functions  $f_1$  and  $f_2$  are determined.

## 7 Experimental results

In this section, the following four experiments utilizing the proposed method are presented:

- approximation of a one-variable function,
- prediction of fuel consumption,
- classification of breast tissue,
- prediction of concrete compressive strength.

In these experiments, the proposed method is compared with the well-known ANFIS model [32] and the RFIS model with type-1 fuzzy sets [92]. The ANFIS model is realized with constant rule consequents (ANFIS ‘constant’) and with linear ones (ANFIS ‘linear’). Moreover, in the first experiment, the result of an approximation with a polynomial model is presented. The T1RFIS and T2RFIS models are applied with the design matrix of four types: ‘linear’, ‘purequadratic’, ‘interactions’, and ‘quadratic’ as presented in Sect. 3. The number of bins in the F-test method is bounded in the range from 1 to 20. The number of active predictors is limited from one to the number of all features. The MGA algorithm is used to train the T1RFIS and T2RFIS models. Training includes ten trials, for which the results are averaged. The final model is chosen as the model whose result is closest to the average value.

### 7.1 Experiment 1

In this experiment, we consider the following one-variable function

$$y = 5 \sin(4\pi x) \exp(-|4x|) \quad (30)$$



where  $x \in [-1, 1]$  (Fig. 7). The task is to train the T2RFIS approximator of the given function assuming an accuracy not greater than the threshold value of the validation error  $RMSE_t = 0.075$ . The number of observation data is 121, which are divided into training and validation sets consisting of 60% and 40% randomly selected observations, respectively. For the input  $x$  of the ANFIS, T1RFIS, and T2RFIS models, the number of fuzzy sets is five. In training the T1RFIS and T2RFIS models, the centers are bounded by  $c_{min} = -1$ ,  $c_{max} = 1$ . The distances of the membership functions in the training of the T2RFIS models are bounded by  $\delta_{min} = 0$ , and  $\delta_{max} = 0.5$ . For the T2RFIS models, the widths of membership functions are bounded by  $\sigma_{min} = 0.05$ ,  $\sigma_{max} = 0.8$ , and for the T1RFIS models by  $\sigma_{min} = 0.05$ ,  $\sigma_{max} + \delta_{max} = 1.3$ . The T1RFIS and T2RFIS models are generated with the regularization parameter  $\lambda = 1e-5$  of the ridge regression. The population size for the MGA algorithm is  $5n$ , where  $n$  is the number of variables in the individual. The number of iterations is 100, which gives  $500n$  evaluations of the objective function.

### 7.1.1 Polynomial approximation

Figure 8 shows the validation error of a polynomial approximator depending on the polynomial degree (changing from 1 to 25). As we can see, the smallest validation error is 1.070 for a polynomial degree equal to 18 (Table 2). Thus, the polynomial approximator does not achieve the prescribed accuracy  $RMSE_t = 0.075$  in any case. This approximator has the shortest training time of all the models considered.

### 7.1.2 ANFIS models

In training the ANFIS models, the number of epochs was 300. Of all the epochs, the value for which the validation error reaches the minimum is selected to obtain the best model. For the two considered cases, that is, ANFIS ('constant') and ANFIS ('linear'), the results are presented in Table 2. In the first case, the minimum RMSE

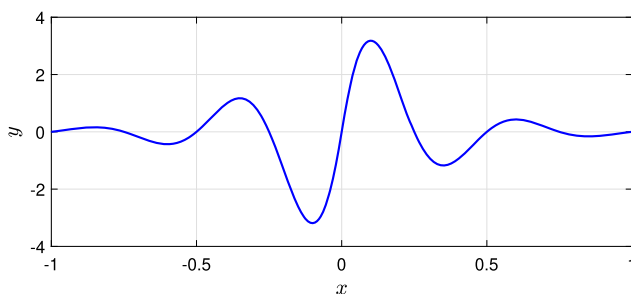


Fig. 7 Experiment 1: The target function being approximated

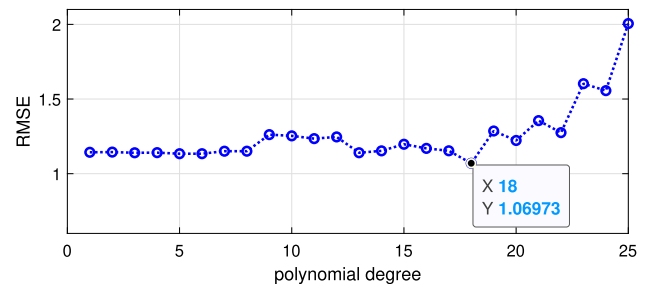


Fig. 8 Experiment 1: Validation error depending on the polynomial degree for the polynomial approximator

Table 2 Performance comparison for Experiment 1; RMSE is the validation error,  $N$  is the number of selected features from the design matrix,  $N_{all}$  is the number of all features in the design matrix, and  $T_t$  is the training time; n/a stands for 'not applicable'

Method	RMSE	$N(N_{all})$	$T_t$ [s]
Polynomial model	1.070	n/a	0.045
ANFIS models [32]			
ANFIS ('constant')	0.5192	n/a	1.898
ANFIS ('linear')	0.1042	n/a	1.921
T1RFIS models [92]			
F-test + 'linear'	0.4037	4(5)	8.515
F-test + 'purequadratic'	0.1685	8(10)	9.409
F-test + 'interactions'	0.0793	13(15)	10.87
F-test + 'quadratic'	0.0685	11(20)	11.87
T2RFIS models			
F-test + 'linear'	0.3624	4(5)	12.17
F-test + 'purequadratic'	0.1135	7(10)	14.00
F-test + 'interactions'	0.0709	11(15)	14.85
F-test + 'quadratic'	<b>0.0462</b>	14(20)	16.24

The best RMSE result is marked in bold

is 0.5192 for the number of epochs equal to 983, while in the second case—0.1042 for the number of epochs equal to 1000. As we can see, the threshold value  $RMSE_t = 0.075$  is not reached in any case.

### 7.1.3 T1RFIS models

The results for the T1RFIS models are presented in Table 2, where it is seen that only the model F-test + 'quadratic' achieves the specified accuracy. The best T1RFIS model, chosen from the Pareto front, is F-test + 'quadratic', for which the RMSE is 0.0685. For this model, 11 of 20 features are selected from the design matrix. The training times for the T1RFIS models are longer than the training times for the ANFIS models. The use of feature selection reduces the number of features in the range of 13% ( $1 - 13/15$ ) to 45% ( $1 - 11/20$ ).

### 7.1.4 T2RFIS models

Based on the results of the T2RFIS models in Table 2, we can see that the models F-test + ‘interactions’ and F-test + ‘quadratic’ have the RMSE less than the desired accuracy. Among them, the best is the model F-test + ‘quadratic’ with RMSE = 0.0462, and this is the best result for all considered models. This model, selected from the Pareto front depicted in Fig. 9, is described as follows:

- lower membership functions for the input  $x$  (Fig. 10):

$$\underline{A}_1 = g(x; -0.0458, 0.1150) \tag{31}$$

$$\underline{A}_2 = g(x; -0.0387, 0.2103) \tag{32}$$

$$\underline{A}_3 = g(x; 0.2081, 0.1591) \tag{33}$$

$$\underline{A}_4 = g(x; -0.7946, 0.5638) \tag{34}$$

$$\underline{A}_5 = g(x; 0.8179, 0.5265) \tag{35}$$

- upper membership functions for the input  $x$  (Fig. 10):

$$\bar{A}_1 = g(x; -0.0458, 0.2717) \tag{36}$$

$$\bar{A}_2 = g(x; -0.0387, 0.6341) \tag{37}$$

$$\bar{A}_3 = g(x; 0.2081, 0.6014) \tag{38}$$

$$\bar{A}_4 = g(x; -0.7946, 0.9740) \tag{39}$$

$$\bar{A}_5 = g(x; 0.8179, 0.8032) \tag{40}$$

- system function to determine the output:

$$\begin{aligned} \hat{y} = & b_1 \xi_1 + b_2 \xi_2 + b_4 \xi_4 + b_5 \xi_5 \\ & + b_6 \xi_1 \xi_2 + b_8 \xi_1 \xi_4 + b_{10} \xi_2 \xi_3 + b_{11} \xi_2 \xi_4 \\ & + b_{12} \xi_2 \xi_5 + b_{14} \xi_3 \xi_5 \\ & + b_{16} \xi_1^2 + b_{17} \xi_2^2 + b_{19} \xi_4^2 + b_{20} \xi_5^2 \end{aligned} \tag{41}$$

where  $b_1 = 170.4, b_2 = -19.57, b_4 = -62.50, b_5 = 84.91, b_6 = -75.96, b_8 = -226.9, b_{10} = -55.16, b_{11} = 156.6, b_{12} = -30.54, b_{14} = -122.7,$

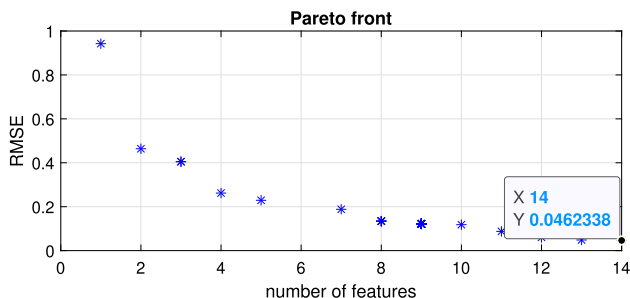


Fig. 9 Experiment 1: Pareto front for the T2RFIS model; the highlighted point indicates the selected solution

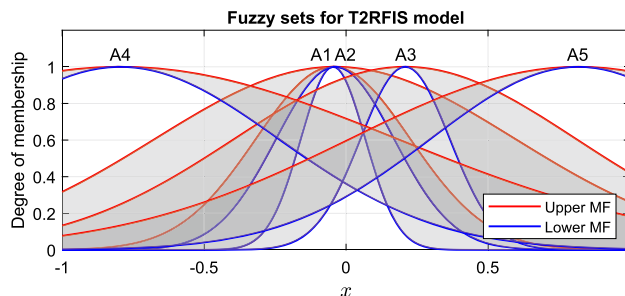


Fig. 10 Experiment 1: Fuzzy sets for input  $x$  of the T2RFIS model

$$\begin{aligned} b_{16} = -351.4, \quad b_{17} = -14.49, \quad b_{19} = 56.68, \\ b_{20} = -82.63. \end{aligned}$$

As shown in Table 2, 14 of 20 characteristics of the design matrix are applied in the final model. This model has the longest training time of more than 16 s. The approximation of the function by the obtained T2RFIS model and the ANFIS (‘linear’) model is shown in Fig. 11. It is seen in Table 2 that that the training times of the T2RFIS models are longer than the training times of the T1RFIS models. The reduction in the number of features ranges from 20% (1 – 4/5) to 30% (1 – 7/10).

### 7.2 Experiment 2

In this experiment, the objective is to predict automobile fuel consumption in miles per gallon (MPG) [20, 81]. The data set consists of 392 pairs  $[x, y]$  of input–output observations. This set is divided into 196 pairs of training data (odd-indexed samples) and 196 pairs of validation data (even-indexed samples) (Fig. 12). The following six automobile attributes of various makes and models are used as the model inputs: the number of cylinders ( $x^1$ ), displacement ( $x^2$ ), horsepower ( $x^3$ ), weight ( $x^4$ ), acceleration ( $x^5$ ), and model year ( $x^6$ ). The seventh attribute, which is MPG, is used as the model output ( $y$ ). We assume that the RMSE cannot exceed a maximum value  $RMSE_t = 2.6$ . The number of fuzzy sets for the ANFIS, T1RFIS and T2RFIS models is three. The centers of the membership functions

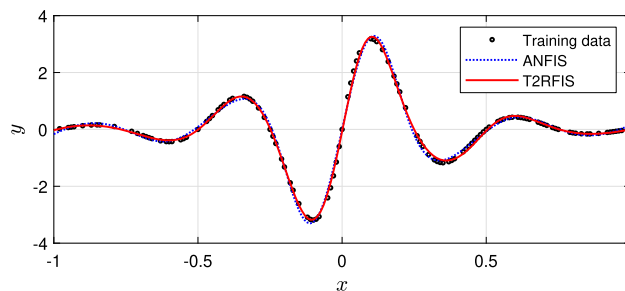


Fig. 11 Experiment 1: Approximation of the function (30) by the ANFIS and T2RFIS models

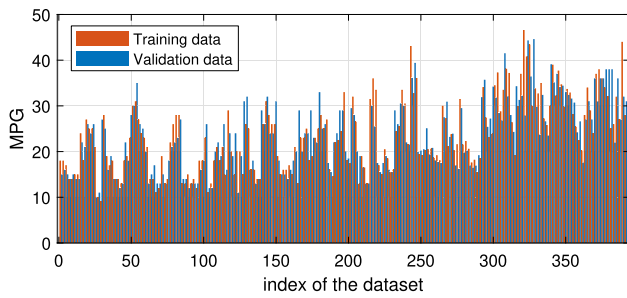


Fig. 12 Experiment 2: Training and validation data

are bounded by  $\mathbf{c}_{min} = [3, 68, 46, 1613, 8, 70]$  and  $\mathbf{c}_{max} = [8, 455, 230, 5140, 24.8, 82]$ . The distances of the membership functions in the training of the T2RFIS models are bounded by  $\delta_{min} = [0, 0, 0, 0, 0, 0]$  and  $\delta_{max} = [1.051, 81.35, 38.68, 741.4, 3.532, 2.523]$ . For the T2RFIS models, the widths of the membership functions are bounded by  $\sigma_{min} = [0.2123, 16.43, 7.814, 149.8, 0.7134, 0.5096]$ ,  $\sigma_{max} = [2.123, 164.3, 78.14, 1498, 7.134, 5.096]$ , and for the T1RFIS models by  $\sigma_{min} = [0.2123, 16.43, 7.814, 149.8, 0.7134, 0.5096]$ ,  $\sigma_{max} + \delta_{max} = [3.174, 245.7, 116.8, 2239, 10.67, 7.618]$ . The regularization parameter for ridge regression is  $\lambda = 1e-03$ . The population size for the MGA algorithm is  $n$ , where  $n$  is the number of variables in the individual. The number of iterations is 50, which gives  $50n$  objective function evaluations.

7.2.1 ANFIS models

The ANFIS models were trained with a number of epochs equal to 200. The results for two cases, namely, ANFIS ('constant') and ANFIS ('linear'), are presented in Table 3. In the first case, the minimum RMSE is 10.53 for the number of epochs equal to 48, and in the second case, the minimum RMSE is 88.39 for the number of epochs equal to 29. In any case, the threshold  $RMSE_t = 2.6$  is not reached. For both cases, the number of fuzzy rules is 729.

7.2.2 T1RFIS models

The results of the T1RFIS models are presented in Table 3. All T1RFIS models achieve the specified accuracy. The smallest RMSE is for the model F-test + 'interactions', and is equal to 2.517. The training times for the T1RFIS models are shorter than the training times for the

Table 3 Performance comparison for Experiment 2; RMSE is the validation error,  $N$  is the number of selected features from the design matrix,  $N_{all}$  is the number of all features in the design matrix, and  $T_t$  is the training time; n/a stands for 'not applicable'

Method	RMSE	$N(N_{all})$	$T_t$ [s]
ANFIS models [32]			
ANFIS ('constant')	10.53	n/a	184.5
ANFIS ('linear')	88.39	n/a	5268
T1RFIS models [92]			
F-test + 'linear'	2.572	13(18)	6.988
F-test + 'purequadratic'	2.552	29(36)	7.762
F-test + 'interactions'	2.517	15(171)	13.35
F-test + 'quadratic'	2.531	38(189)	14.96
T2RFIS models			
F-test + 'linear'	2.557	15(18)	10.80
F-test + 'purequadratic'	2.544	29(36)	12.06
F-test + 'interactions'	<b>2.462</b>	43(171)	19.76
F-test + 'quadratic'	2.492	114(189)	24.76

The best RMSE result is marked in bold

ANFIS models. The use of the feature selection method reduces the number of features in the range of 19% ( $1 - 29/36$ ) to 91% ( $1 - 15/171$ ).

7.2.3 T2RFIS models

Table 3 presents the results for the T2RFIS models. Similarly to the T1RFIS models, all models achieve the specified accuracy. Among them, the best is the model F-test + 'interactions' with  $RMSE = 2.462$ , and this is the best result for all considered models. This model is selected from the Pareto front depicted in Fig. 13. As we can see in Table 3, 43 of 171 features of the design matrix are applied in the final model. The training times of the T2RFIS models are shorter than the training times of the ANFIS models, but longer than those for the T1RFIS models. Figures 14, 15, and 16 show the fuzzy sets for the inputs  $x^1 - x^6$ . Figure 17 shows the real value of MPG, the

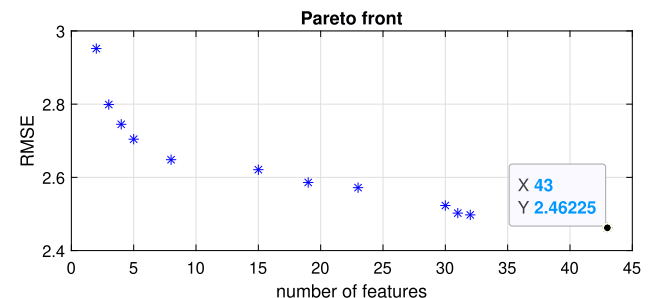
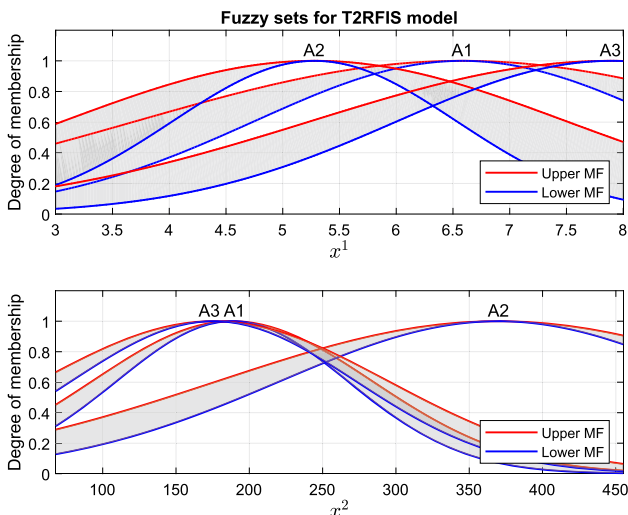
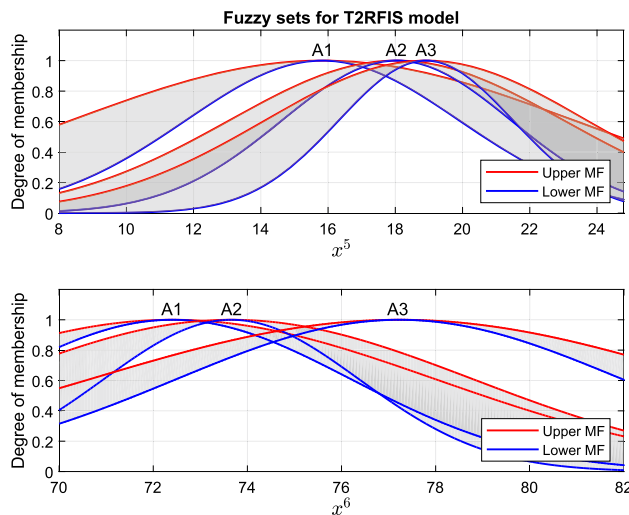


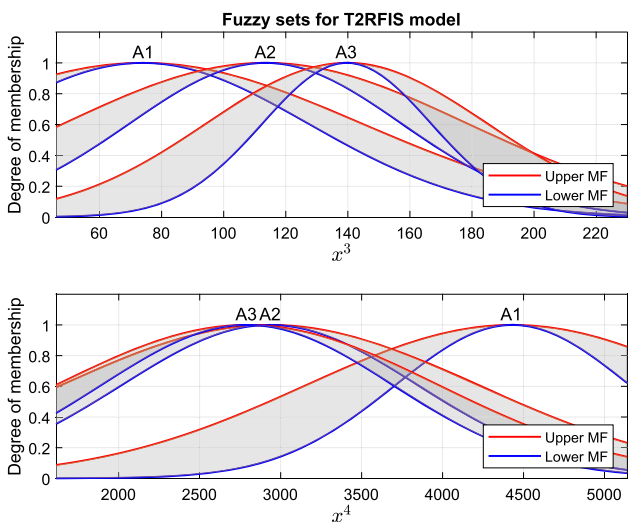
Fig. 13 Experiment 2: Pareto front for the T2RFIS model; the highlighted point indicates the selected solution



**Fig. 14** Experiment 2: Fuzzy sets for inputs  $x^1$  and  $x^2$  of the T2RFIS model



**Fig. 16** Experiment 2: Fuzzy sets for inputs  $x^5$  and  $x^6$  of the T2RFIS model

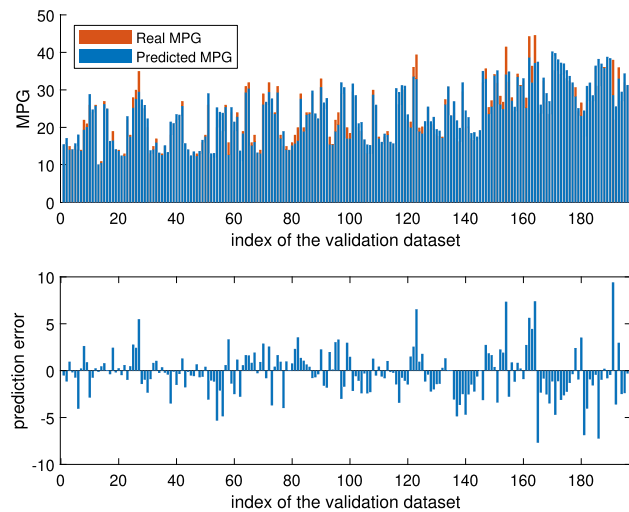


**Fig. 15** Experiment 2: Fuzzy sets for inputs  $x^3$  and  $x^4$  of the T2RFIS model

predicted value, and the error for the obtained model. The reduction in the number of features ranges from 17% ( $1 - 15/18$ ) to 75% ( $1 - 43/171$ ).

### 7.3 Experiment 3

This experiment deals with breast disease class prediction based on electrical impedance measurements of freshly excised tissue samples [20]. The data set contains 106 observations in the form of input–output pairs  $[x, y]$ . The observations are randomly divided into two sets, a training set containing 74 records and a validation set containing 32 records. The elements of the input vector  $x$  are the following nine variables: impedance (ohm) at zero frequency



**Fig. 17** Experiment 2: Comparison of the real and predicted values for the T2RFIS model

( $x^1$ ), phase angle at 500 KHz ( $x^2$ ), high-frequency slope of phase angle ( $x^3$ ), impedance distance between spectral ends ( $x^4$ ), area under spectrum ( $x^5$ ), area normalized by impedance distance ( $x^6$ ), maximum of the spectrum ( $x^7$ ), distance between the impedance and the real part of the maximum frequency point ( $x^8$ ), and length of the spectral curve ( $x^9$ ). The output variable  $y$  represents six classes: carcinoma, fibro-adenoma, mastopathy, glandular, connective, and adipose. The number of fuzzy sets for the ANFIS, T1RFIS and T2RFIS models is two. The centers of Gaussian fuzzy sets are bounded by the minimum and maximum values of the input variables given by  $c_{min} = [103.0, 0.0124, -0.0663, 19.65, 70.43, 1.596, 7.969, -9.258, 125.0]$ ,  $c_{max} = [2800, 0.3583, 0.4677, 1063, 174, 500, 164.1, 436.1, 977.6, 2897]$ . The distances of the

membership functions for the T2RFIS models are bounded by  $\delta_{min} = [0, 0, 0, 0, 0, 0, 0, 0, 0]$  and  $\delta_{max} = [916.3, 0.1175, 0.1814, 354.6, 59250, 55.20, 145.5, 335.2, 941.6]$ . For the T2RFIS models, the widths  $\sigma$  are bounded by  $\sigma_{min} = [229.1, 0.0294, 0.0454, 88.65, 14810, 13.80, 36.36, 83.81, 235.4], \sigma_{max} = [2291, 0.2938, 0.4536, 886.5, 148, 100, 138.0, 363.6, 838.1, 2354]$ , and for the T1RFIS models by  $\sigma_{min}, \sigma_{max} + \delta_{max} = [3207, 0.4113, 0.6350, 1241, 207, 400, 193.2, 509.1, 1173, 3296]$ . The regularization parameter for ridge regression is  $\lambda = 1e-06$ . The population size for the MGA algorithm is  $n$ , where  $n$  is the number of variables in the individual. The number of iterations is 50, which gives  $50n$  objective function evaluations. Because the considered models generate the real value  $y$ , this value must be converted to a class number. This paper uses the solution presented in [92]. First, the value of  $y$  is limited to the range [1, 6], and then rounded to an integer from the set  $\{1, 2, \dots, 6\}$ . In this way, we get a predicated class for our problem.

### 7.3.1 ANFIS models

The ANFIS models were trained with a number of epochs equal to 100. The results for the ANFIS with constant consequents ('constant') and for linear consequents ('linear') are presented in Table 4. In the first case, the classification accuracy (ACC) is 53.13% for the number of epochs equal to 21, and in the second case, the ACC is 43.75% for the number of epochs equal to 95. For both cases, the number of fuzzy rules is 512.

### 7.3.2 T1RFIS models

The class predictions for the T1RFIS models are presented in Table 4. It can be seen that the best accuracy of ACC = 78.13% is achieved by two models, namely F-test + 'interactions' and F-test + 'quadratic'. This result is 25% better than the ANFIS model with constant consequents and about 34% better than the same model with linear consequents. As in Experiment 2, the training times for the T1RFIS models are much shorter than the training times for the ANFIS models. The use of the feature selection method made it possible to reduce the number of features in the range of 28% (1 – 13/18) to 97% (1 – 6/189).

### 7.3.3 T2RFIS models

Table 4 presents the classification results for the T2RFIS models. Among them, the best result equal to 81.25% achieved the model F-test + 'quadratic' (this is the best result for all considered models). This model is chosen from the Pareto front shown in Fig. 18. As we can see in

Table 4, 29 of 189 features of the design matrix are applied in the final model. The training times of the T2RFIS models are shorter than those of the ANFIS models, but longer than those of the T1RFIS models. The reduction in the number of features for the considered models ranges from 28% (1 – 13/18) to 85% (1 – 29/189).

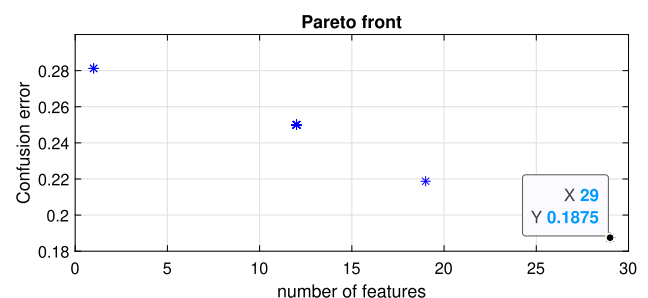
## 7.4 Experiment 4

This experiment is about prediction of concrete compressive strength based on eight components used for concrete preparation [20]. The data set consists of 1030 records in the form of input–output pairs  $[x, y]$  representing the relationship between the output variable and the input variables. The data set is randomly divided into two sets, a training set containing 721 records and a validation set containing 309 records. The elements of the input vector  $x$

**Table 4** Performance comparison for Experiment 3; ACC is the classification accuracy,  $N$  is the number of selected features from the design matrix,  $N_{all}$  is the number of all features in the design matrix, and  $T_t$  is the training time; n/a stands for 'not applicable'

Method	ACC [%]	$N(N_{all})$	$T_t$ [s]
ANFIS models [32]			
ANFIS ('constant')	53.13	n/a	75.43
ANFIS ('linear')	43.75	n/a	2594
T1RFIS models [92]			
F-test + 'linear'	62.50	13(18)	4.860
F-test + 'purequadratic'	71.88	20(36)	6.440
F-test + 'interactions'	78.13	21(171)	10.66
F-test + 'quadratic'	78.13	6(189)	13.09
T2RFIS models			
F-test + 'linear'	62.50	13(18)	7.687
F-test + 'purequadratic'	75.00	13(36)	8.848
F-test + 'interactions'	78.13	45(171)	15.75
F-test + 'quadratic'	<b>81.25</b>	29(189)	16.43

The best ACC result is marked in bold



**Fig. 18** Experiment 3: Pareto front for the T2RFIS model; the highlighted point indicates the selected solution

are the following variables: cement ( $x^1$ ), blast furnace slag ( $x^2$ ), fly ash ( $x^3$ ), water ( $x^4$ ), superplasticizer ( $x^5$ ), coarse aggregate ( $x^6$ ), fine aggregate ( $x^7$ ), and age ( $x^8$ ). The number of fuzzy sets for fuzzy models is two. The centers of Gaussian fuzzy sets are bounded using the minimum and maximum values of the input variables  $\mathbf{c}_{min} = [102.0, 0, 0, 121.8, 0, 801.0, 594.0, 1.000]$ ,  $\mathbf{c}_{max} = [540.0, 359.4, 200.1, 247.0, 32.20, 1145, 992.6, 365.0]$ . The distances of the membership functions for the T2RFIS models are bounded by  $\delta_{min} = [0, 0, 0, 0, 0, 0, 0, 0]$  and  $\delta_{max} = [167.4, 137.4, 76.48, 47.87, 12.31, 131.5, 152.3, 139.1]$ . For the T2RFIS models, the widths  $\sigma$  are bounded by  $\sigma_{min} = [37.20, 30.52, 16.99, 10.64, 2.735, 29.22, 33.85, 30.92]$  and  $\sigma_{max} = [372.0, 305.2, 169.9, 106.38, 27.35, 292.2, 338.6, 309.2]$ . For the T1RFIS models, the bounds are  $\sigma_{min}$  and  $\sigma_{max} + \delta_{max} = [539.4, 442.6, 246.4, 154.2, 39.65, 423.6, 490.9, 448.3]$ . The regularization parameter for ridge regression is  $\lambda = 1e-03$ . As in previous experiments, the population size for the MGA algorithm is  $n$ , where  $n$  is the number of variables in the individual. The number of iterations is 50, which gives  $50n$  objective function evaluations.

7.4.1 ANFIS models

The results for two ANFIS models, namely ANFIS ('constant') and ANFIS ('linear'), are presented in Table 5. These models were trained with a number of epochs equal to 100. For the first model, the minimum RMSE is 8.909 and for the second model, the minimum RMSE is 171.7. For both models, the number of fuzzy inference rules is 256. The table shows a very long training time of 12,580 s for the model with linear consequents.

7.4.2 T1RFIS models

The concrete strength predictions for the T1RFIS models are presented in Table 5. As we can see, the best accuracy of  $RMSE = 6.814$  is achieved by the model F-test + 'quadratic'. This result is better by 2.095 than for the ANFIS model with constant consequents and by 164.9 than for the ANFIS model with linear consequents. As in Experiments 2 and 3, the training times for the T1RFIS models are much shorter than the training times for the ANFIS models. The use of the feature selection method made it possible to reduce the number of features in the range of 47% ( $1 - 17/32$ ) to 72% ( $1 - 43/152$ ).

7.4.3 T2RFIS models

In Table 5, the concrete strength predictions are presented for the T2RFIS models. Among these models, the best

**Table 5** Performance comparison for Experiment 4; RMSE is the validation error,  $N$  is the number of selected features from the design matrix,  $N_{all}$  is the number of all features in the design matrix, and  $T_t$  is the training time; n/a stands for 'not applicable'

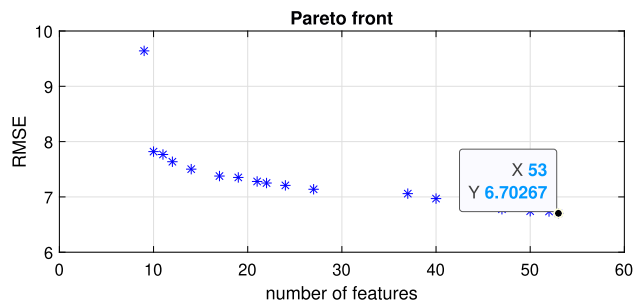
Method	RMSE	$N(N_{all})$	$T_t$ [s]
ANFIS models [32]			
ANFIS ('constant')	8.909	n/a	358.9
ANFIS ('linear')	171.7	n/a	12580
T1RFIS models [92]			
F-test + 'linear'	8.733	8(16)	6.810
F-test + 'purequadratic'	7.283	17(32)	9.793
F-test + 'interactions'	6.896	37(136)	17.16
F-test + 'quadratic'	6.814	43(152)	18.03
T2RFIS models			
F-test + 'linear'	8.752	10(16)	10.99
F-test + 'purequadratic'	7.032	22(32)	13.37
F-test + 'interactions'	6.773	63(136)	25.72
F-test + 'quadratic'	<b>6.703</b>	53(152)	27.65

The best RMSE result is marked in bold

result equal to 6.703 achieved the model F-test + 'quadratic' (this is the best result for all considered models). This model is chosen from the Pareto front presented in Fig. 19. As we can see in Table 5, 53 of 152 predictors of the design matrix are applied in this model. The training times of the T2RFIS models are shorter than those of the ANFIS models, but longer than those of the T1RFIS models. The reduction in the number of features ranges from 31% ( $1 - 22/32$ ) to 65% ( $1 - 53/152$ ).

7.5 Remarks on computational complexity

An important issue is the complexity of the model, for which we can observe that for the T2RFIS models it is the same as for the T1RFIS models. This is because the number of predictors in the regression matrices in both cases is  $n \times pm$  [92], so the number of predictors in the design matrices is the same (Table 6). It is seen that for a



**Fig. 19** Experiment 4: Pareto front for the T2RFIS model; the highlighted point indicates the selected solution

**Table 6** Structure complexity for the ANFIS and RFIS models;  $p$  is the number of fuzzy sets, and  $m$  is the number of inputs

Method	Complexity
ANFIS models [32]	
ANFIS ('constant')	$p^m$
ANFIS ('linear')	$p^m$
T1RFIS [92] and T2RFIS models	
RFIS + 'linear'	$pm$
RFIS + 'purequadratic'	$2pm$
RFIS + 'interactions'	$pm(pm + 1)/2$
RFIS + 'quadratic'	$pm(pm + 3)/2$

given number of fuzzy sets, the complexity is linearly related to the number of inputs for models of type 'linear' and 'purequadratic', and squarely for the models interactions and 'quadratic'. In comparison, the complexity of the ANFIS models expressed in terms of the number of fuzzy rules depends exponentially on the number of inputs (Table 6), so in their case the curse of dimensionality arises.

The complexity of the model affects the training time, which can be clearly seen in the results for the experiments presented above. With the exception of the first experiment, where we deal with the approximation of a function of one variable, in the remaining experiments the training times for the ANFIS model are much longer than for the T1RFIS and T2RFIS models. In Experiments 2, 3 and 4 for the ANFIS model, we have 729 ( $3^6$ ), 512 ( $2^9$ ) and 256 ( $2^8$ ) inference rules, respectively. A large number of rules causes the training times of this model to be long. For example, in Experiment 4, the training times for the ANFIS models are 358.9 s and 12,580 s, while for the T1RFIS and T2RFIS models the time ranges from 6.81 s to 27.65 s. Moreover, it can be seen that in this example, as well as in the others, the training time for the ANFIS models with linear consequents is much longer than for the models with constant consequents. This analysis shows the low efficiency of the ANFIS model for more complex (multidimensional) problems, which is not visible in the T1RFIS and T2RFIS models.

## 8 Conclusions

A novel multi-variable fuzzy inference system with type-2 fuzzy sets has been proposed. This system does not have explicitly defined fuzzy inference rules. It consists of Gaussian fuzzy sets of type-2 defined for the inputs and linearly parameterized system functions for determining

the output. The system training is performed using observation data in the form of input/output pairs. The fuzzy sets are determined by a multi-objective genetic algorithm that uses a feature selection method, and the system function parameters are obtained by ridge regression. Calculating the output is simple and fast; it requires only the multiplication of the design matrix and the vector of parameters ( $\hat{\mathbf{y}} = \mathbf{D}\mathbf{b}$ ).

The experiments carried out confirmed the usefulness of the proposed method. On the basis of these experiments, it can be seen that the proposed method can improve the results obtained by the ANFIS and T1RFIS models. Future work will be devoted to the use of the proposed method for multidimensional data, applications of other algorithms for training fuzzy sets, and the development of this method for use with nonlinearly parameterized system functions.

**Funding** No funding was received for the conduct of this study.

**Data availability** Experiments were carried out in publicly available data sets.

**Code availability** The MATLAB code is available at: <https://www.mathworks.com/matlabcentral/fileexchange/107205-t2rfis-type-2-regression-based-fuzzy-inference-system>

## Declarations

**Conflict of interest** Not applicable.

**Consent to participate** Not applicable.

**Consent for publication** Not applicable.

**Ethical approval** Not applicable.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Al Farsi A, Petrovic D, Doctor F (2023) A non-iterative reasoning algorithm for fuzzy cognitive maps based on type 2 fuzzy sets. Inf Sci 622:319–336. <https://doi.org/10.1016/j.ins.2022.11.152>

2. Amirkhani A, Shirzadeh M, Kumbasar T (2020) Interval type-2 fuzzy cognitive map-based flight control system for quadcopters. *Int J Fuzzy Syst* 22(8, SI):2504–2520. <https://doi.org/10.1007/s40815-020-00940-8>
3. Amirkhani A, Shirzadeh M, Kumbasar T, Mashadi B (2022) A framework for designing cognitive trajectory controllers using genetically evolved interval type-2 fuzzy cognitive maps. *Int J Intell Syst* 37(1):305–335. <https://doi.org/10.1002/int.22626>
4. de los Angeles Hernandez M, Melin P, Mendez GM, Castillo O, Lopez-Juarez I (2015) A hybrid learning method composed by the orthogonal least-squares and the back-propagation learning algorithms for interval a2-c1 type-1 non-singleton type-2 tsk fuzzy logic systems. *Soft Comput* 19(3), 661–678. <https://doi.org/10.1007/s00500-014-1287-8>
5. Arman H (2022) A simple noniterative method to accurately calculate the centroid of an interval type-2 fuzzy set. *Int J Intell Syst* 37(12):12057–12084. <https://doi.org/10.1002/int.23076>
6. Bandyopadhyay A, Kar S (2019) System of type-2 fuzzy differential equations and its applications. *Neural Comput Appl* 31(9, SI):5563–5593. <https://doi.org/10.1007/s00521-018-3380-x>
7. Beke A, Kumbasar T (2023) More than accuracy: a composite learning framework for interval type-2 fuzzy logic systems. *IEEE Trans Fuzzy Syst* 31(3):734–744. <https://doi.org/10.1109/TFUZZ.2022.3188920>
8. Biswas S, Moi S, Pal(Sarkar) S (2022) Study of interval type-2 fuzzy singular integro-differential equation by using collocation method in weighted space. *New Math Nat Comput* 18(01):113–145. <https://doi.org/10.1142/S1793005722500077>
9. Carvajal O, Melin P, Miramontes I, Prado-Arechiga G (2021) Optimal design of a general type-2 fuzzy classifier for the pulse level and its hardware implementation. *Eng Appl Artif Intell* 97. <https://doi.org/10.1016/j.engappai.2020.104069>
10. Chen C, Wu D, Garibaldi JM, John IR, Twycross J, Mendel JM (2021) A comprehensive study of the efficiency of type-reduction algorithms. *IEEE Trans Fuzzy Syst* 29(6):1556–1566. <https://doi.org/10.1109/TFUZZ.2020.2981002>
11. Chen Y (2022) Design of sampling-based noniterative algorithms for centroid type-reduction of general type-2 fuzzy logic systems. *Complex Intell Syst* 8(5, SI):4385–4402. <https://doi.org/10.1007/s40747-022-00789-4>
12. Chen Y, Li C, Yang J (2023) Design and application of nagarardini structure-based interval type-2 fuzzy logic systems optimized with the combination of backpropagation algorithms and recursive least square algorithms. *Exp Syst Appl* 211:96. <https://doi.org/10.1016/j.eswa.2022.118596>
13. Cherif S, Baklouti N, Hagrass H, Alimi AM (2022) Novel intuitionistic-based interval type-2 fuzzy similarity measures with application to clustering. *IEEE Trans Fuzzy Syst* 30(5):1260–1271. <https://doi.org/10.1109/TFUZZ.2021.3057697>
14. D'Alterio P, Garibaldi JM, John IR (2021) Wagner C A fast inference and type-reduction process for constrained interval type-2 fuzzy systems. *IEEE Trans Fuzzy Syst* 29(11):3323–3333. <https://doi.org/10.1109/TFUZZ.2020.3018379>
15. Deng Z, Choi KS, Cao L, Wang S (2014) T2fela: type-2 fuzzy extreme learning algorithm for fast training of interval type-2 tsk fuzzy logic system. *IEEE Trans Neural Netw Learn Syst* 25(4):664–676. <https://doi.org/10.1109/TNNLS.2013.2280171>
16. Deveci M, Cali U, Kucuksari S, Erdogan N (2020) Interval type-2 fuzzy sets based multi-criteria decision-making model for offshore wind farm development in ireland. *Energy* 198. <https://doi.org/10.1016/j.energy.2020.117317>
17. Dong H, Zhou S (2022) Extended dissipativity and dynamical output feedback control for interval type-2 singular semi-markovian jump fuzzy systems. *Int J Syst Sci* 53(9):1906–1924. <https://doi.org/10.1080/00207721.2022.2031337>
18. Dong Y, Song Y, Wei G (2022) Efficient model-predictive control for nonlinear systems in interval type-2 t-s fuzzy form under round-robin protocol. *IEEE Trans Fuzzy Syst* 30(1):63–74. <https://doi.org/10.1109/TFUZZ.2020.3031394>
19. Du Z, Kao Y, Zhao X (2021) An input delay approach to interval type-2 fuzzy exponential stabilization for nonlinear unreliable networked sampled-data control systems. *IEEE Trans Syst Man Cybern Syst* 51(6):3488–3497. <https://doi.org/10.1109/TSMC.2019.2930473>
20. Dua D, Graff C (2023) UCI machine learning repository. <http://archive.ics.uci.edu/ml>
21. Erdem D, Kumbasar T (2021) Enhancing the learning of interval type-2 fuzzy classifiers with knowledge distillation. In: *IEEE CIS INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS 2021 (FUZZ-IEEE)*, IEEE International Conference on Fuzzy Systems. IEEE Comput Intell Soc; IEEE. <https://doi.org/10.1109/FUZZ45933.2021.9494471>. IEEE CIS International Conference on Fuzzy Systems (FUZZ-IEEE), ELECTR NETWORK, JUL 11-14, 2021
22. Fan Qf (2018) Wang T, Chen Y, Zhan Zf Design and application of interval type-2 tsk fuzzy logic system based on qpso algorithm. *Int J Fuzzy Syst* 20(3):835–846. <https://doi.org/10.1007/s40815-017-0357-3>
23. Figueroa-Garcia JC, Roman-Flores H, Chalco-Cano Y (2022) Type-reduction of interval type-2 fuzzy numbers via the chebyshev inequality. *Fuzzy Sets Syst* 435(SI):164–180. <https://doi.org/10.1016/j.fss.2021.04.014>
24. Greenfield S, Chiclana F (2021) The stratic defuzzifier for discretised general type-2 fuzzy sets. *Inf Sci* 551:83–99. <https://doi.org/10.1016/j.ins.2020.10.062>
25. Han M, Zhong K, Qiu T, Han B (2019) Interval type-2 fuzzy neural networks for chaotic time series prediction: a concise overview. *IEEE Trans Cybern* 49(7):2720–2731. <https://doi.org/10.1109/TCYB.2018.2834356>
26. Hoerl A, Kennard R (1970) Ridge regression - biased estimation for nonorthogonal problems. *Technometrics* 12(1):55–000. <https://doi.org/10.1080/00401706.1970.10488634>
27. Holland JH (1992) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press
28. Hosseinpour M, Ghaemi S, Khanmohammadi S, Daneshvar S (2022) A hybrid high-order type-2 fcm improved random forest classification method for breast cancer risk assessment. *Appl Math Comput* 424. <https://doi.org/10.1016/j.amc.2022.127038>
29. Hu S, Yue D, Dou C, Xie X, Ma Y, Ding L (2022) Attack-resilient event-triggered fuzzy interval type-2 filter design for networked nonlinear systems under sporadic denial-of-service jamming attacks. *IEEE Trans Fuzzy Syst* 30(1):190–204. <https://doi.org/10.1109/TFUZZ.2020.3033851>
30. Huang S, Zhao G, Weng Z, Ma S (2022) Trapezoidal type-2 fuzzy inference system with tensor unfolding structure learning method. *Neurocomputing* 473:54–67. <https://doi.org/10.1016/j.neucom.2021.12.011>
31. Huang YP, Singh P, Kuo WL, Chu HC (2021) A type-2 fuzzy clustering and quantum optimization approach for crops image segmentation. *Int J Fuzzy Syst* 23(3):615–629. <https://doi.org/10.1007/s40815-020-01009-2>
32. Jang J (1993) ANFIS - adaptive-network-based fuzzy inference system. *IEEE Trans Syst Man Cybern* 23(3):665–685. <https://doi.org/10.1109/21.256541>
33. Javanmard M, Nehi HM (2019) A solving method for fuzzy linear programming problem with interval type-2 fuzzy numbers. *Int J Fuzzy Syst* 21(3):882–891. <https://doi.org/10.1007/s40815-018-0591-3>



34. Jiang Z, Wang Z, Kim EH (2023) Noise-robust fuzzy classifier designed with the aid of type-2 fuzzy clustering and enhanced learning. *IEEE Access* 11:8108–8118. <https://doi.org/10.1109/ACCESS.2023.3238798>
35. Jin K, Zhang X (2023) Output feedback stabilization of type 2 fuzzy singular fractional-order systems with mismatched membership functions. *Soft Comput* 27(8):4917–4929. <https://doi.org/10.1007/s00500-022-07553-3>
36. Jung S, Kim M, Kim J, Kim S (2021) Fault detection method based on auto-associative kernel regression and interval type-2 fuzzy logic system for multivariate process. In: *IEEE CIS INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS 2021 (FUZZ-IEEE)*, IEEE International Conference on Fuzzy Systems. IEEE Computat Intelligence Soc; IEEE. <https://doi.org/10.1109/FUZZA5933.2021.9494486>. IEEE CIS International Conference on Fuzzy Systems (FUZZ-IEEE), ELECTR NETWORK, JUL 11–14, 2021
37. Kalhori MRN, FazelZarandi MH (2021) A new interval type-2 fuzzy reasoning method for classification systems based on normal forms of a possibility-based fuzzy measure. *Inf Sci* 581:567–586. <https://doi.org/10.1016/j.ins.2021.09.060>
38. Kundu P, Majumder S, Kar S, Maiti M (2019) A method to solve linear programming problem with interval type-2 fuzzy parameters. *Fuzzy Opt Dec Making* 18(1):103–130. <https://doi.org/10.1007/s10700-018-9287-2>
39. Lathamaheswari M, Nagarajan D, Kavikumar J, Broumi S (2021) Interval type-2 fuzzy aggregation operator in decision making and its application. *Complex Intell Syst* 7(3):1695–1708. <https://doi.org/10.1007/s40747-021-00287-z>
40. Leon-Garza H, Hagrass H, Pena-Rios A, Conway A, Owusu G (2022) A type-2 fuzzy system-based approach for image data fusion to create building information models. *Inf Fusion* 88:115–125. <https://doi.org/10.1016/j.inffus.2022.07.007>
41. Li H, Dai X, Zhou L, Wu Q (2023) Encoding words into interval type-2 fuzzy sets: the retained region approach. *Inf Sci* 629:760–777. <https://doi.org/10.1016/j.ins.2023.02.022>
42. Li JF, Jahanshahi H, Kacar S, Chu YM, Gomez-Aguilar JF, Alotaibi ND, Alharbi KH (2021) On the variable-order fractional memristor oscillator: Data security applications and synchronization using a type-2 fuzzy disturbance observer-based robust control. *CHAOS SOLITONS & FRACTALS* 145. <https://doi.org/10.1016/j.chaos.2021.110681>
43. Li X, Ye B, Liu X (2022) The solution for type-2 fuzzy linear programming model based on the nearest interval approximation. *J Intell Fuzzy Syst* 42(3):2275–2285. <https://doi.org/10.3233/JIFS-211568>
44. Li Z, Yan H, Zhang H, Lam HK, Wang M (2021) Aperiodic sampled-data-based control for interval type-2 fuzzy systems via refined adaptive event-triggered communication scheme. *IEEE Trans Fuzzy Syst* 29(2):310–321. <https://doi.org/10.1109/TFUZZ.2020.3016033>
45. Lian Z, Shi P, Lim CC (2021) Hybrid-triggered interval type-2 fuzzy control for networked systems under attacks. *Inf Sci* 567:332–347. <https://doi.org/10.1016/j.ins.2021.03.050>
46. Liu HC, Shi H, Li Z, Duan CY (2022) An integrated behavior decision-making approach for large group quality function deployment. *Inf Sci* 582:334–348. <https://doi.org/10.1016/j.ins.2021.09.020>
47. Liu J, Ran G, Huang Y, Han C, Yu Y, Sun C (2022) Adaptive event-triggered finite-time dissipative filtering for interval type-2 fuzzy markov jump systems with asynchronous modes. *IEEE Trans Cybern* 52(9):9709–9721. <https://doi.org/10.1109/TCYB.2021.3053627>
48. Liu Q, Li X, Yang J (2022) Optimum codesign for image denoising between type-2 fuzzy identifier and matrix completion denoiser. *IEEE Trans Fuzzy Syst* 30(1):287–292. <https://doi.org/10.1109/TFUZZ.2020.3030498>
49. Liu X, Lin Y, Wan SP (2021) New efficient algorithms for the centroid of an interval type-2 fuzzy set. *Inf Sci* 570:468–486. <https://doi.org/10.1016/j.ins.2021.04.032>
50. Liu Z, Chen CLP, Zhang Y (2012) Li Hx Type-2 hierarchical fuzzy system for high-dimensional data-based modeling with uncertainties. *Soft Comput* 16(11):1945–1957. <https://doi.org/10.1007/s00500-012-0867-8>
51. Lu YN, Bai YL, Tang LH, Wan WD, Ma YJ (2021) Secondary factor induced wind speed time-series prediction using self-adaptive interval type-2 fuzzy sets with error correction. *Energy Rep* 7:7030–7047. <https://doi.org/10.1016/j.egy.2021.09.150>
52. Mai DS, Ngo LT, Trinh LH, Hagrass H (2021) A hybrid interval type-2 semi-supervised possibilistic fuzzy c-means clustering and particle swarm optimization for satellite image analysis. *Inf Sci* 548:398–422. <https://doi.org/10.1016/j.ins.2020.10.003>
53. Mendez GM, de los Angeles Hernandez M (2013) Hybrid learning mechanism for interval a2-c1 type-2 non-singleton type-2 takagi-sugeno-kang fuzzy logic systems. *Inf Sci* 220, 149–169. <https://doi.org/10.1016/j.ins.2012.01.024>
54. Mishro PK, Agrawal S, Panda R, Abraham A (2021) A novel type-2 fuzzy c-means clustering for brain mr image segmentation. *IEEE TRANSACTIONS ON CYBERNETICS* 51(8):3901–3912. <https://doi.org/10.1109/TCYB.2020.2994235>
55. Mittal K, Jain A, Vaisla KS, Castillo O, Kacprzyk J (2020) A comprehensive review on type 2 fuzzy logic applications: Past, present and future. *Eng Appl Artif Intell* 95. <https://doi.org/10.1016/j.engappai.2020.103916>
56. Moradi Zirkohi M, Shoja-Majidabad S (2022) Chaos synchronization using an improved type-2 fuzzy wavelet neural network with application to secure communication. *J Vib Control* 28(15–16):2074–2090. <https://doi.org/10.1177/10775463211005903>
57. Naimi M, Tahayori H, Sadeghian A (2021) A fast and accurate method for calculating the center of gravity of polygonal interval type-2 fuzzy sets. *IEEE Trans Fuzzy Syst* 29(6):1472–1483. <https://doi.org/10.1109/TFUZZ.2020.2979133>
58. Najariyan M, Qiu L (2022) Interval type-2 fuzzy differential equations and stability. *IEEE Trans Fuzzy Syst* 30(8):2915–2929. <https://doi.org/10.1109/TFUZZ.2021.3097810>
59. Nguyen TB, Kim SH (2020) Dissipative control of interval type-2 nonhomogeneous markovian jump fuzzy systems with incomplete transition descriptions. *Nonlinear Dyn* 100(2):1289–1308. <https://doi.org/10.1007/s11071-020-05564-z>
60. Novakovic B (1999) Fuzzy logic control synthesis without any rule base. *IEEE Trans Syst Man Cybern Part B-Cybern* 29(3):459–466. <https://doi.org/10.1109/3477.764883>
61. Ontiveros-Robles E, Melin P (2020) Toward a development of general type-2 fuzzy classifiers applied in diagnosis problems through embedded type-1 fuzzy classifiers. *Soft Comput* 24(1, SI):83–99. <https://doi.org/10.1007/s00500-019-04157-2>
62. Pan X, Wang Y, He S, Chin KS (2022) A dynamic programming algorithm based clustering model and its application to interval type-2 fuzzy large-scale group decision-making problem. *IEEE Trans Fuzzy Syst* 30(1):108–120. <https://doi.org/10.1109/TFUZZ.2020.3032794>
63. Pan Y, Wu Y, Lam HK (2022) Security-based fuzzy control for nonlinear networked control systems with dos attacks via a resilient event-triggered scheme. *IEEE Trans Fuzzy Syst* 30(10):4359–4368. <https://doi.org/10.1109/TFUZZ.2022.3148875>
64. Pan Y, Yang GH (2021) Event-driven fault detection for discrete-time interval type-2 fuzzy systems. *IEEE Trans Syst Man*

- Cybern-Syst 51(8):4959–4968. <https://doi.org/10.1109/TSMC.2019.2945063>
65. Pham DH, Lin CM, Giap VN, Huynh TT, Cho HY (2022) Wavelet interval type-2 takagi-kang-sugeno hybrid controller for time-series prediction and chaotic synchronization. *IEEE Access* 10:104313–104327. <https://doi.org/10.1109/ACCESS.2022.3210260>
  66. Qazani MRC, Asadi H, Al-Ashmori M, Mohamed S, Lim CP, Nahavandi S (2021) Time series prediction of driving motion scenarios using fuzzy neural networks. In: 2021 IEEE INTERNATIONAL CONFERENCE ON MECHATRONICS (ICM). IEEE. <https://doi.org/10.1109/ICM46511.2021.9385693>. IEEE International Conference on Mechatronics (ICM), ELECTRONETWORK, MAR 07-09, 2021
  67. Qin P, Zhao T, Dian S (2023) Interval type-2 fuzzy neural network-based adaptive compensation control for omnidirectional mobile robot. *Neural Comput Appl* 35(16, SI):11653–11667. <https://doi.org/10.1007/s00521-023-08309-2>
  68. Ramirez M, Melin P (2023) A new interval type-2 fuzzy aggregation approach for combining multiple neural networks in clustering and prediction of time series. *Int J Fuzzy Syst* 25(3):1077–1104. <https://doi.org/10.1007/s40815-022-01426-5>
  69. Ran G, Li C, Sakthivel R, Han C, Wang B, Liu J (2022) Adaptive event-triggered asynchronous control for interval type-2 fuzzy markov jump systems with cyberattacks. *IEEE Trans Control Netw Syst* 9(1):88–99. <https://doi.org/10.1109/TCNS.2022.3141025>
  70. Runkler TA (2022) Pareto interval type-2 fuzzy decision making for labeled objects. In: 2022 IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (FUZZ-IEEE), IEEE International Fuzzy Systems Conference Proceedings. IEEE; Int Neural Network Soc; IEEE Computat Intelligence Soc; Evolutionary Programming Soc; IET; Univ Padova, Dept Math Tullio Levi Civita; European Space Agency; expert.ai; Elsevier; Springer Nature; Google; Baker & Hughes; NVIDIA. <https://doi.org/10.1109/FUZZ-IEEE55066.2022.9882586>. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) / IEEE World Congress on Computational Intelligence (IEEE WCCI) / International Joint Conference on Neural Networks (IJCNN) / IEEE Congress on Evolutionary Computation (IEEE CEC), Padua, ITALY, JUL 18-23, 2022
  71. Sabahi K, Zhang C, Kausar N, Mohammadzadeh A, Pamucar D, Mosavi AH (2023) Input-output scaling factors tuning of type-2 fuzzy pid controller using multi-objective optimization technique. *Aims Math* 8(4):7917–7932. <https://doi.org/10.3934/math.2023399>
  72. Safari A, Hosseini R, Mazinani M (2022) A type-2 fuzzy time series model for pattern similarity analysis: A case study on air quality forecasting. *IEEE Intell Syst* 37(2):92–102. <https://doi.org/10.1109/MIS.2021.3095727>
  73. Sahin I, Ulu C (2023) Altitude control of a quadcopter using interval type-2 fuzzy controller with dynamic footprint of uncertainty. *ISA Trans* 134:86–94. <https://doi.org/10.1016/j.isatra.2022.08.020>
  74. Sakthivel R, Kwon OM, Park MJ, Sakthivel R (2023) Event-triggered finite-time dissipative filtering for interval type-2 fuzzy complex dynamical networks with cyber attacks. *IEEE Trans Syst Man Cybernetics-Syst* 53(5):3042–3053. <https://doi.org/10.1109/TSMC.2022.3221641>
  75. Salehi F, Keyvanpour MR, Sharifi A (2021) GT2-CFC: General type-2 collaborative fuzzy clustering method. *Inf Sci* 578:297–322. <https://doi.org/10.1016/j.ins.2021.07.037>
  76. Sambas A, Mohammadzadeh A, Vaidyanathan S, Ayob AFM, Aziz A, Mohamed MA, Sulaiman IM (2023) Nawi MAA Investigation of chaotic behavior and adaptive type-2 fuzzy controller approach for permanent magnet synchronous generator (pmsg) wind turbine system. *Aims Math* 8(3):5670–5686. <https://doi.org/10.3934/math.2023285>
  77. Sun J, Zhang H, Wang Y, Sun S (2022) Fault-tolerant control for stochastic switched it2 fuzzy uncertain time-delayed nonlinear systems. *IEEE Trans Cybern* 52(2):1335–1346. <https://doi.org/10.1109/TCYB.2020.2997348>
  78. Tang G, Long J, Gu X, Chiclana F, Liu P, Wang F (2022) Interval type-2 fuzzy programming method for risky multicriteria decision-making with heterogeneous relationship. *Inf Sci* 584:184–211. <https://doi.org/10.1016/j.ins.2021.10.044>
  79. Tavoosi J, Suratgar AA, Menhaj MB (2016) Stable anfis2 for nonlinear system identification. *Neurocomputing* 182:235–246. <https://doi.org/10.1016/j.neucom.2015.12.030>
  80. Tavoosi J, Zhang C, Mohammadzadeh A, Mobayen S, Mosavi AH (2021) Medical image interpolation using recurrent type-2 fuzzy neural network. *Front Neuroinf* 15. <https://doi.org/10.3389/fninf.2021.667375>
  81. The MathWorks Inc (2022) Fuzzy Logic Toolbox User's Guide. Natick, Massachusetts, United States
  82. The MathWorks Inc (2022) Global Optimization Toolbox User's Guide. Natick, Massachusetts, United States
  83. The MathWorks Inc (2022) Statistics and Machine Learning Toolbox User's Guide. Natick, Massachusetts, United States
  84. Zp Tian, Rx Nie (2019) Wang Jq Social network analysis-based consensus-supporting framework for large-scale group decision-making with incomplete interval type-2 fuzzy information. *Inf Sci* 502:446–471. <https://doi.org/10.1016/j.ins.2019.06.053>
  85. Tung SW, Quek C, Guan C (2013) et2fis: an evolving type-2 neural fuzzy inference system. *Inf Sci* 220:124–148. <https://doi.org/10.1016/j.ins.2012.02.031>
  86. Wang H, Pan X, He S (2019) A new interval type-2 fuzzy vikor method for multi-attribute decision making. *Int J Fuzzy Syst* 21(1):145–156. <https://doi.org/10.1007/s40815-018-0527-y>
  87. Wang H, Pan X, Yan J, Yao J, He S (2020) A projection-based regret theory method for multi-attribute decision making under interval type-2 fuzzy sets environment. *Inf Sci* 512:108–122. <https://doi.org/10.1016/j.ins.2019.09.041>
  88. Wang S, Chung KFL, Lu J, Han B (2004) Hu D Fuzzy inference systems with no any rule base and linearly parameter growth. *J Control Theor Appl* 2(2):185–192. <https://doi.org/10.1007/s11768-004-0067-x>
  89. Wang T, Li H, Qian Y, Huang B, Zhou X (2022) A regret-based three-way decision model under interval type-2 fuzzy environment. *IEEE Trans Fuzzy Syst* 30(1):175–189. <https://doi.org/10.1109/TFUZZ.2020.3033448>
  90. Wei XJ, Zhang DQ, Huang SJ (2022) A variable selection method for a hierarchical interval type-2 tsk fuzzy inference system \*. *Fuzzy Sets Syst* 438:46–61. <https://doi.org/10.1016/j.fss.2021.09.017>
  91. Whitley D (1994) A genetic algorithm tutorial. *Statist Comput* 4(2):65–85
  92. Wiktorowicz K (2022) RFIS: regression-based fuzzy inference system. *Neural Comput Appl* 34(14, SI):12175–12196. <https://doi.org/10.1007/s00521-022-07105-8>
  93. Wu L, Qian F, Wang L, Ma X (2022) An improved type-reduction algorithm for general type-2 fuzzy sets. *Inf Sci* 593:99–120. <https://doi.org/10.1016/j.ins.2022.01.078>
  94. Yadav S, Tiwari SP, Kumari M, Yadav VK (2022) An interval type-2 fuzzy model of computing with words via interval type-2 fuzzy finite rough automata with application in covid-19 deduction. *New Math Natural Comput* 18(01):61–101. <https://doi.org/10.1142/S1793005722500053>
  95. Yang H, Wang X, Zhong S, Shu L (2022) Observer-based asynchronous event-triggered control for interval type-2 fuzzy systems with cyber-attacks. *Inf Sci* 606:805–818. <https://doi.org/10.1016/j.ins.2022.05.087>

96. Yang X, Yu F, Pedrycz W (2021) Typical characteristic-based type-2 fuzzy c-means algorithm. *IEEE Trans Fuzzy Syst* 29(5):1173–1187. <https://doi.org/10.1109/TFUZZ.2020.2969907>
97. Yang Y, Niu Y, Lam HK (2022) Sliding mode control for networked interval type-2 fuzzy systems via random multiaccess protocols. *IEEE Trans Fuzzy Syst* 30(11):5005–5018. <https://doi.org/10.1109/TFUZZ.2022.3165379>
98. Yeh CY, Jeng WH, Lee SJ (2011) Data-based system modeling using a type-2 fuzzy neural network with a hybrid learning algorithm. *IEEE Trans Neural Netw* 22(12, 2):2296–2309. <https://doi.org/10.1109/TNN.2011.2170095>
99. You Z, Yan H, Zhang H, Zeng L, Wang M (2023) Further stability criteria for sampled-data-based interval type-2 fuzzy systems via a refined two-side looped-functional method. *IEEE Trans Fuzzy Syst* 31(1):265–277. <https://doi.org/10.1109/TFUZZ.2022.3185711>
100. Yuste-Delgado AJ, Cuevas-Martinez JC, Trivino-Cabrera A (2022) Statistical normalization for a guided clustering type-2 fuzzy system for wsn. *IEEE Sens J* 22(6):6187–6195. <https://doi.org/10.1109/JSEN.2022.3150066>
101. Zadeh L (1975) The concept of a linguistic variable and its application to approximate reasoning-I. *Inf Sci* 8(3):199–249. [https://doi.org/10.1016/0020-0255\(75\)90036-5](https://doi.org/10.1016/0020-0255(75)90036-5)
102. Zandieh F, Ghannadpour SF (2023) A comprehensive risk assessment view on interval type-2 fuzzy controller for a time-dependent hazmat routing problem. *Eur J Oper Res* 305(2):685–707. <https://doi.org/10.1016/j.ejor.2022.06.007>
103. Zhan T, Li WT, Fan BJ, Liu S (2023) Experimental evaluation on defuzzification of tsf-type-based interval type-2 fuzzy inference systems. *Int J Control Auto Syst* 21(4):1338–1348. <https://doi.org/10.1007/s12555-021-0370-z>
104. Zhang L, Lam HK, Sun Y, Liang H (2020) Fault detection for fuzzy semi-markov jump systems based on interval type-2 fuzzy approach. *IEEE Trans Fuzzy Syst* 28(10):2375–2388. <https://doi.org/10.1109/TFUZZ.2019.2936333>
105. Zhang T, Ma F, Yue D, Peng C, O'Hare GMP (2020) Interval type-2 fuzzy local enhancement based rough k-means clustering considering imbalanced clusters. *IEEE Trans Fuzzy Syst* 28(9):1925–1939. <https://doi.org/10.1109/TFUZZ.2019.2924402>
106. Zhang X, Wang H, Stojanovic V, Cheng P, He S, Luan X, Liu F (2022) Asynchronous fault detection for interval type-2 fuzzy nonhomogeneous higher level markov jump systems with uncertain transition probabilities. *IEEE Trans Fuzzy Syst* 30(7):2487–2499. <https://doi.org/10.1109/TFUZZ.2021.3086224>
107. Zhang Z, Niu Y, Cao Z, Song J (2021) Security sliding mode control of interval type-2 fuzzy systems subject to cyber attacks: the stochastic communication protocol case. *IEEE Trans Fuzzy Syst* 29(2):240–251. <https://doi.org/10.1109/TFUZZ.2020.2972785>
108. Zhao J, Liu Y, Wang L, Wang W (2020) A generalized heterogeneous type-2 fuzzy classifier and its industrial application. *IEEE Trans Fuzzy Syst* 28(10):2287–2301. <https://doi.org/10.1109/TFUZZ.2019.2930492>

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