



Type-1 and type-2 radial basis function neural networks Mandami system to evaluate quality features

Pascual Noradino Montes Dorantes^{1,2} · Gerardo Maximiliano Méndez³ · Marco Aurelio Jiménez Gómez⁴ · Adriana Mexicano Santoyo²

Received: 31 August 2021 / Accepted: 12 January 2022 / Published online: 7 February 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

This paper presents a methodology that uses the central composite design and the radial basis function neural networks in type-1 or in interval type-2 model to generate a network that evaluates quality features in an industrial image processing. The methodology includes a couple of radial basis functions as Huygen's tractrix and triangular membership functions as complementary contributions that have not been reported in literature as radial basis functions. The advantage of using this proposal is that the training is not required to get an accurate result, also the generation of the IT2 RBFNN fuzzy rule base for evaluating quality characteristics is simplified by using the central composite design method and statistical indicators extracted from the product specification data. Experimental results show an error reduction of 90% when the interval type-2 Mandami Radial basis function neural network was compared against its type-1 counterpart using the Gaussian membership functions onto a radial basis function network. On the other hand, the implementation of the Huygen's tractrix, found a reduction error of 50% in comparison to the Gaussian function.

Keywords Radial basis function network · RBFNN · T1 RBFNN · Interval type-2 · IT2 · IT2 RBFNN · Image processing · Quality assurance · Fuzzy neural network

✉ Pascual Noradino Montes Dorantes
pascualresearch@gmail.com; cualmontes@hotmail.com

Gerardo Maximiliano Méndez
gmm_paper@yahoo.com.mx

Marco Aurelio Jiménez Gómez
m_jimenez81@yahoo.com.mx

Adriana Mexicano Santoyo
mexicanao@gmail.com

- ¹ División de Estudios de Posgrado E Investigación, Universidad Autónoma del Noreste, Blvd. José Musa de León Y General Medardo de La Peña S/N, Col. Los Pinos, C.P. 25100 Saltillo, Coahuila, Mexico
- ² División de Estudios de Posgrado E Investigación, Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Victoria, Boulevard Emilio Portes Gil #1301 Pte. A.P. 175, CP. 87010 Ciudad Victoria, Tamps., Mexico
- ³ Posgrado en Ingeniería Mecatrónica, Tecnológico Nacional de México, Instituto Tecnológico de Nuevo León, Av. Eloy Cavazos, 2001 Guadalupe Nuevo León, Mexico
- ⁴ División de Estudios de Posgrado E Investigación, Tecnológico Nacional de México, Instituto Tecnológico de Saltillo, Blvd. V. Carranza No. 2400, Col. Tecnológico, C.P. 25280 Saltillo, México

1 Introduction

While the Interval type-2 (IT2) systems arise as an alternative to manage uncertainties present in all industrial processes, the type-1 (T1) models such as T1 singleton fuzzy logic or their equivalent radial basis function neural network (RBFNN), also called T1 RBFNN, both cannot manage uncertainties [1]. T1 models require several cycles, epochs or iterations of training, and adjustment to get an acceptable result or an adequate level of precision. For example in [2], 1300 epochs of training were required to get an adequate result. Therefore, as mentioned by [3], the main problem to obtain better precision output relies on the modeling of the system. Most researchers base their modeling by intuition and as a thumb rule because there are not specific criteria for determining how many rules are needed to model this kind of systems.

Specialized literature shows a couple of techniques to model intelligent systems [4–11], but all of them are applied to T1 models. Only a pair of proposals was found that uses the IT2 [2, 11]. In [4, 5] has been proposed a model that uses linguistic labels as fuzzy sets, but in [5] the center label

is missing to create the universe of discourse (UOD) and with them the fuzzy rule base. In [4] the model is based in the Gaussian data distribution, but they do not consider the mean which means that the model is restricted to have odd input variables, while in [5] the model is restricted to even input variables.

In [6] a method that uses control charts to model the rule base was developed, but it was restricted to use a genetic algorithm in order to generate the membership functions (MF), the support or width of the fuzzy sets is non-uniform, and finally a combination of trapezoidal and triangular MFs is produced to get the UOD. In [7] a similar method presented in [6] was used, but it only uses triangular MFs and presents the presence of blank spaces in the UOD. In [8] five linguistic labels have been used which are restricted to a constant rule base that is inflexible and unadaptable with blank spaces in the UOD.

The statistical properties of the Gaussian distribution were used in [3] and [9] to model the fuzzy rule base. The central composite design (CCD) of the design of experiments was used in [10, 11] to obtain a simplified and compact rule base to assemble the UOD. In [10] a T1 RBFNN that needs to be trained was proposed. Finally, in [11] the technique of CCD was used to model an IT2 rule base that considers uncertainties.

In the classic approach of the RBFNN or T1 RBFNN, only two types of membership functions can be used, the logistic function, see Fig. 1, or the Gaussian function, see Fig. 2, that are defined as receptive fields or fuzzifiers [12] (which were presented by [3]).

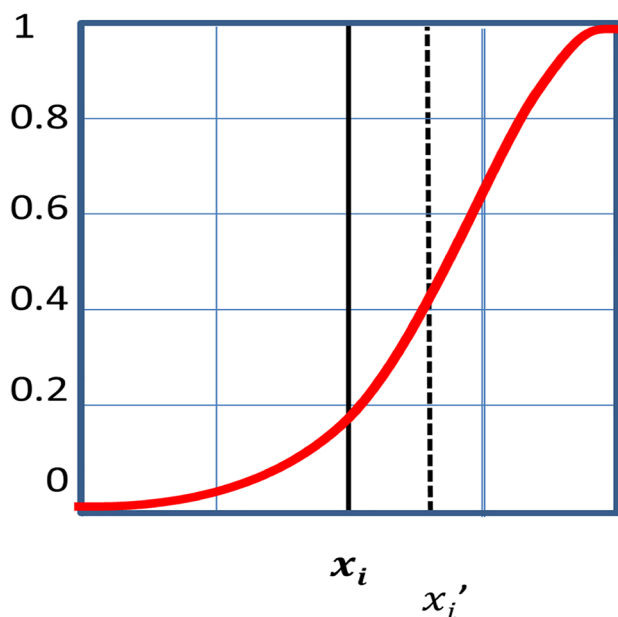


Fig. 1 Logistic function

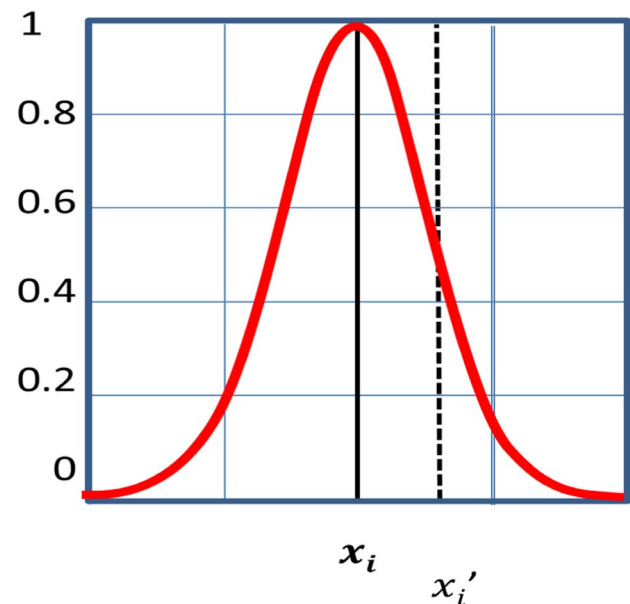


Fig. 2 Gaussian function

1.1 Related works

Using the phrase of Mendel [1], the IT2 RBFNN technology is going through its infancy phase because it is a new technique developed in 2015 by Rubio-Solis and Panoutsos in [2], therefore, there are very few related works. Between the theoretical proposals for modeling and assembling IT2 RBFNN network, it can be mentioned the framework presented in [2] to model an IT2 network for the first time, where the necessity of expert knowledge to assemble the IT2 is mentioned. Nonetheless, in [1] is stated that “Rules may be provided by experts or can be extracted from numerical data”, which is demonstrated in [3, 4, 9–11].

Theoretical basis about IT2 was developed since 2015 in [2, 10, 13–16]; however, your application has barely been shown in works as [13–29]; this fact shows that this is an emerging and current technology. Among the works found in the current literature appears: a general IT2 network presented in [13], which is restricted to only one type of membership function, as a Gaussian function. A recurrent self-evolving RBFNN is presented in [14] to adapt the neurons in the hidden layer of the network in a non-linear dynamical system. In [15] a model to convert the IT2 Mamdani model into an IT2 Takagi–Sugeno model via genetic algorithms was proposed, Baklouti et al. [16] applied an IT2 RBFNN to type Takagi–Sugeno–Kang (TSK) in order to evaluate time series. From [17] could be obtained the basis of interval type-2 fuzzy logic systems that serves as basis to generate the IT2 RBFNN.

In [18], the use of an IT2 RBFNN to avoid the numerous arithmetic operations required by the interval type-2 fuzzy

models, which reduce their complexities and their type of reduction process was proposed. In [16], a theoretical method called Beta Basis Function Interval type-2 fuzzy neural network (BIT2FNN) in a TSK model to manage the uncertainties and noise was proposed. In [19], a classifier using IT2 RBFNN to recognize alphabets and manage the noise present in the upcoming signals was presented. The classifier uses the IT2 RBFNN to avoid the problem of the non-linearities and noise present in time series with a recurrent interval type-2 fuzzy neural network or RIT2IFNN with a TSK model [20]. In [21], a multilayer type-2 extreme learning machine method for classifying and recognizing walking was presented. In [22], a method to forecast and adjust the neural network using backpropagation algorithm based on general type-2 model was developed. In [23], the IT2 radial basis function network was used in a rail manufacturing process to classify. In [24], the IT2 RBFNN was used in a rotary steerable system and in directional drilling for oil and gas exploration in prospecting. In [25], a classifier with IT2 RBFNN based on clustering was developed, it was used to deal with the uncertainties present in the data and adjust the connection of the weights. In [26], a recurrent IT2 RBFNN was used as an adaptive network for a micro-electro-mechanical system of a gyroscope to control the non-singular sliding control. In [27], the IT-2 RBFNN was used to overcome the control of ammonia flow in a selective catalytic reduction, the IT2 RBFNN aided to modeling uncertainties and constrain the predictive control. In [28], the IT2 RBFNN was used to identify and online prediction of the glucose level on diabetes patients. In [29], the IT2 fuzzy radial basis function neural network (IT2 FRBFNN) was used to sliding mode controls of non-linear systems to approximate the sliding. Finally, in [30], a general type-2 radial basis function neural network (GT2 RBFNN) was developed to manage the trajectory of a remote operated underwater vehicle (ROV).

1.2 Contributions

The main contribution of this paper is the presentation of a novel method for modeling IT2 RBFNN using the CCD technique that is part of a methodology used in the design of experiments to model a compact and simple fuzzy rule base. Other contribution is the implementation of an application of the IT2 RBFNN in a real industrial process using image processing to evaluate quality features, this combination of processes is not documented in the current literature. Additionally, the implementation of two membership functions from fuzzy logic: the Huygen's tractrix or unnamed function (Fig. 3) and the triangular function (Fig. 4), called in this form in [17]. Both used for the first time as RBF in this work since, according to our knowledge, there is no evidence that they have been used in the literature as RBF.

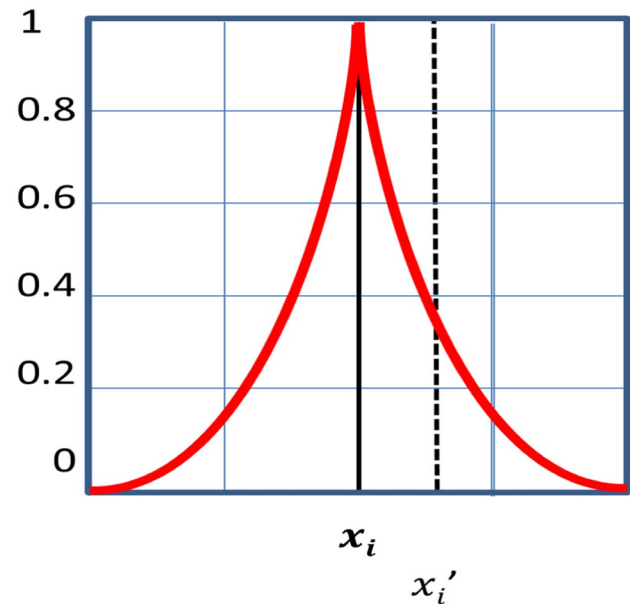


Fig. 3 Huygen's tractrix function

2 Theoretical foundations

2.1 T1 Radial basis function neural networks

This kind of neural network is a type of interpolation focused on getting approximations based on a distance between a pattern and a sample [12] and is graphically depicted in Fig. 5. The T1 RBFNN is similar to the K-nearest neighbor's (KNN) classifier.

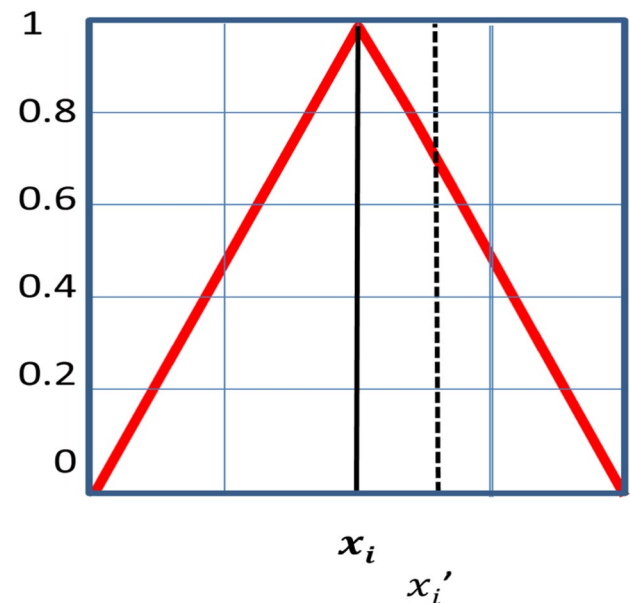


Fig. 4 Triangular function

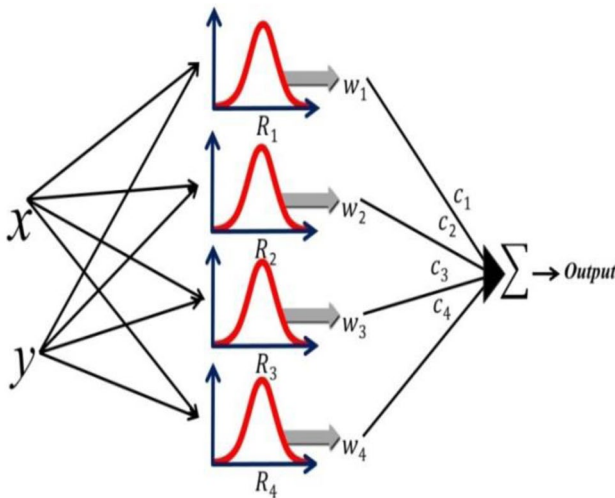


Fig. 5 Two input case topology of a T1 RBFNN

The T1 RBFNN operates with assigned weights in the inputs, each of them is placed in a multidimensional vector given by Eq. (1) and is based on a Gaussian distribution function Eq. (2) or by a logistic distribution function Eq. (3). The Gaussian distribution function is equivalent to a membership function in fuzzy models. The output of the T1 RBFNN model could be calculated in many forms Eqs. (4)–(7).

$$w_i = R_i(x) = R_i\left(\|x - u_x\| \frac{u_i}{\sigma_x}\right) \forall x, u \in U \tag{1}$$

where u_x is the mean in the radial basis function, x is the input of the function, and σ_x is the spread of the function.

$$R(x) = e^{-\frac{\|x-u_x\|^2}{2\sigma_x^2}} \forall x, u \in U \tag{2}$$

$$R(x) = \frac{1}{1 + \frac{e^{\|x-u_i\|^2}}{\sigma_i^2}} \forall x, u \in U \tag{3}$$

$$d(x) = \sum_{i=1}^H c_i w_i \forall c, R \in x \tag{4}$$

$$d(x) = \sum_{i=1}^H c_i R_i \forall c, R \in x \tag{5}$$

$$d(x) = \frac{\sum_{i=1}^H c_i w_i}{\sum_{i=1}^H w_i} \tag{6}$$

$$d(x) = \frac{c_i R_i(x)}{R_i(x)} \forall c, R \in x \tag{7}$$

where c_i is the output in the dataset for the prediction and the final approximation is obtained by Eqs. (4, 5, 6 or 7) which can be reinterpreted as a center of gravity defuzzifier in fuzzy model Eq. (8).

$$y_c(x) = \frac{\sum_1^n y_i \mu_B(y_i)}{\sum_1^n \mu_B(y_i)} \tag{8}$$

2.2 IT2 fuzzy logic system

The IT2 arises from the union of two T1 fuzzy sets represented by Eq. (9) to generate an interval. An IT2 fuzzy set is given by Eq. (10) and is characterized by \tilde{A} with a membership function $\mu_{\tilde{A}}(x, u)$,

$$A = \{(x, \mu_A(x)) | \forall x \in X\} \tag{9}$$

where A represents the set and μ_A is the grade of membership of some x' in A .

$$\tilde{A} = \{(x, u, \mu_{\tilde{A}}(x, u)) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \tag{10}$$

The classic approach for the IT2 models is the Gaussian fuzzifier defined on Eq. (11), showing a difference against the T1 membership function that presents two means,

$$\mu_{\tilde{A}k}^i(x_k) = \tilde{F}_k^i = \exp\left[-\frac{1}{2} \left[\frac{x_k - m_k^i}{\sigma_k^i}\right]^2\right] \tag{11}$$

where $m_k^i \in [m_{k1}^i, m_{k2}^i]$ is the uncertain mean, $k = 1, 2, \dots, p$ (p is the number of inputs) and $i = 1, 2, \dots, M$ (the number of M rules), and σ_k^i is the standard deviation.

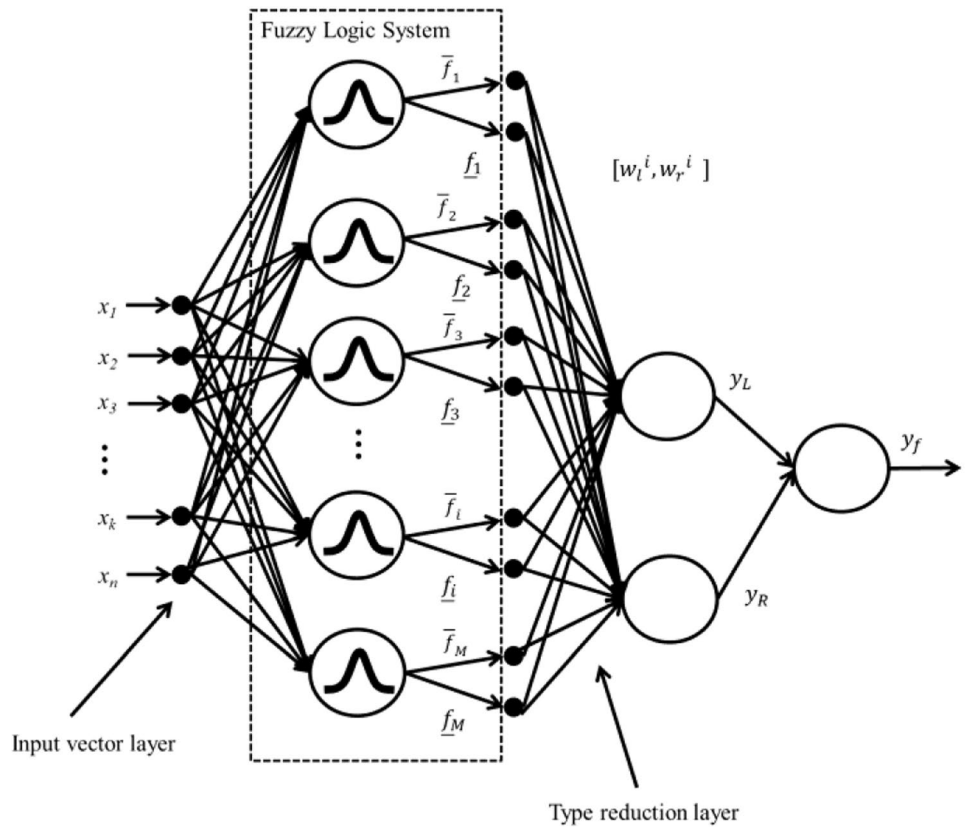
2.3 IT2 Radial basis function neural networks

IT2 RBFNN has been presented for the first time in 2015 by [2], since then it has evolved into new models such as the ones presented by Mandami [2], Takagi–Sugeno [14], and Takagi–Sugeno-Kang model [15].

The basis of this model is the extension of the model shown by [12] to obtain the left and right functions (Fig. 6). Basically the similitudes of the T1 fuzzy model are equivalent to the ones in T1 RBFNN, which are well known since 1993 in [16–18].

The IT2 RBFNN algorithm is generated with the adjustment of T1 RBFNN. First, Eq. (2) is replaced by Eq. (11). Second, Eqs. (4–7) are converted to Eqs. (12–15) to get the lower output and similarly Eqs. (16–19) to get the upper output.

Fig. 6 IT2 RBFNN topology, extracted from [2]



$$\underline{d}(x) = \sum_{i=1}^H c_i \underline{R}_i(x) \forall c, R \in x$$

$$\overline{d}(x) = \frac{\bar{c}_i \bar{R}_i(x)}{\bar{R}_i(x)} \forall c, R \in x \tag{12}$$

$$\underline{d}(x) = \sum_{i=1}^H c_i \underline{w}_i \forall c, R \in x$$

(13) where c_i is the output in the database for the prediction and the low and high are obtained by applying Eqs. (20–21). The final approximation is obtained by Eq. (22) which can be reinterpreted as Eq. (23).

$$\underline{d}(x) = \frac{\sum_{i=1}^H c_i \underline{w}_i}{\sum_{i=1}^H \underline{w}_i}$$

$$d(x_i) = \frac{\overline{d}(x_i) + \underline{d}(x_i)}{2} \tag{14}$$

$$d(x_i) = \frac{\overline{d}(x_i) + \underline{d}(x_i)}{2} \tag{20}$$

$$\underline{d}(x) = \frac{c_i R_i(x)}{\underline{R}_i(x)} \forall c, R \in x$$

$$d(x_i) = \frac{\left(\frac{\bar{c}_i \bar{R}_i(x_i)}{\bar{R}_i(x_i)} \right) + \left(\frac{c_i R_i(x_i)}{\underline{R}_i(x_i)} \right)}{2} \tag{15}$$

$$d(x_i) = \frac{\left(\frac{\bar{c}_i \bar{R}_i(x_i)}{\bar{R}_i(x_i)} \right) + \left(\frac{c_i R_i(x_i)}{\underline{R}_i(x_i)} \right)}{2} \tag{21}$$

$$\overline{d}(x) = \sum_{i=1}^H \bar{c}_i \bar{R}_i(x) \forall c, R \in x$$

$$\overline{d}(x) = \sum_{i=1}^H \bar{c}_i \bar{w}_i \forall c, R \in x \tag{16}$$

$$\overline{d}(x) = \sum_{i=1}^n d(x_i) \tag{17}$$

$$d(x) = \frac{\sum_{i=1}^n d(x_i)}{2} \tag{22}$$

$$\overline{d}(x) = \sum_{i=1}^H \bar{c}_i \bar{w}_i \forall c, R \in x$$

$$\overline{d}(x) = \frac{\sum_{i=1}^H \bar{c}_i \bar{w}_i}{\sum_{i=1}^H \bar{w}_i} \tag{18}$$

$$d(x) = \frac{\left(\frac{\bar{c}_i \bar{R}_i(x)}{\bar{R}_i(x)} \right) + \left(\frac{c_i R_i(x)}{\underline{R}_i(x)} \right)}{2} \tag{23}$$

$$\overline{d}(x) = \frac{\sum_{i=1}^H \bar{c}_i \bar{w}_i}{\sum_{i=1}^H \bar{w}_i}$$

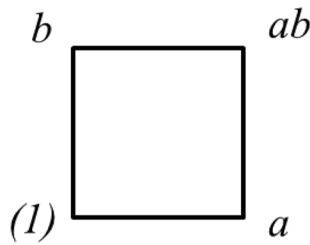


Fig. 7 2^k CCD model

A. Central composite design

The factorial design or CCD is a technique used to analyze the factors and their possible correlation. These factors present limits to define the universe, and their levels are called low and high. In order to establish the model, a combinatorial via permutations is needed to define a pattern where one of the variables must change, meanwhile the rest remain constant. The simplest model is called 2^k (Fig. 7).

In Fig. 7, each sing at a corner represents different levels of variables. For (1), both variables are on low level, for (a) the first variable is on high level, and the second one is on the low level, for (b) the first variable is on the low level and the second on the high level, and for (ab) both variables are on the high level.

3 Revisited equivalence of the T1 FLS and T1 RBFNN

In [2], a series of restrictions have been established to this equivalence as follows:

- The number of perceptive fields in the hidden layer is equal to the number of fuzzy rules.

First, a definition for the receptive field is required.

3.1 Definition 1: a receptive field is a neuron in the network that represents a mathematical operation such as Eq. (2) or Eq. (3).

But this restriction needs to be enhanced, and Fig. 5 needs to be redrawn since the receptive field on this figure only represents a part of the fuzzy rule, e.g., in T1 model a fuzzy rule is defined by Eq. (24), every variable in the T1 RBFNN requires a receptive field represented mathematically by Eq. (2) or Eq. (3), then the number of receptive fields (rf) required is equal to the number of variables (V) multiplied by the number of rules (r), given by Eq. (25). The enhanced graphical representation is depicted in Fig. 8,

$$Rule_i : IF \times 1 \text{ is } a \text{ and } \times 2 \text{ is } b \text{ then } y \text{ is } G \tag{24}$$

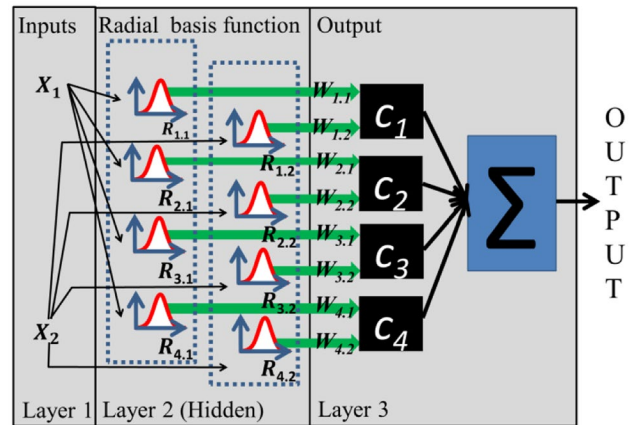


Fig. 8 T1 RBFNN of T1 topology, two input case expanded

where: $\times 1, \times 2$ are the inputs, and G is the output of the rule.

$$rf = V * r \tag{25}$$

The IT2 RBFNN topology of [2] shown in Fig. 6 needs to be redrawn (Fig. 9) because Fig. 6 has additional receptive fields; this condition requires additional calculus and training to tuning precise output, also require additional receptive fields to achieve a precise output. Then, the quantity

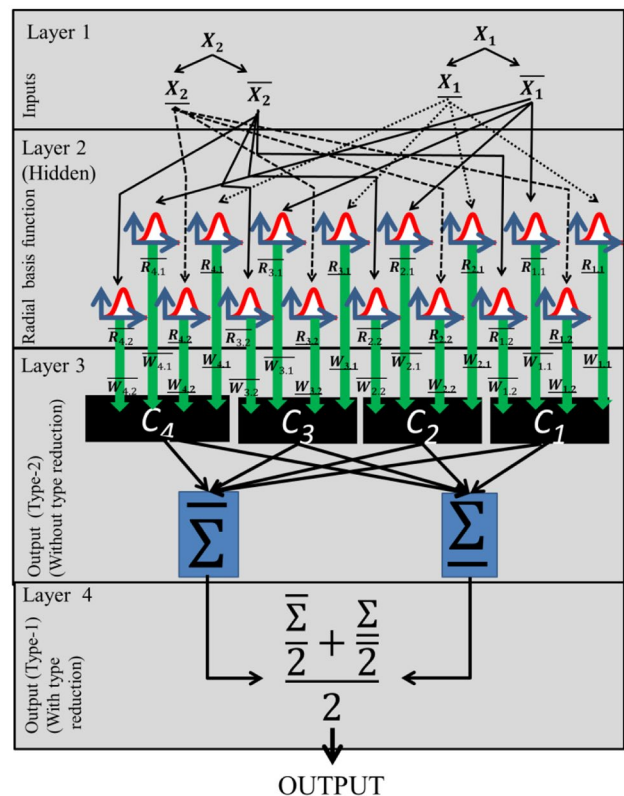


Fig. 9 IT2 RBFNN topology, two input case expanded

of receptive fields is given by Eq. (26), and the topology is reorganized with the addition of the new receptive fields,

$$rf = 2(V \times r) \tag{26}$$

where: rf are the receptive fields, V is the number of variables and, r is the number of rules.

- The membership function within each rule is chosen as Gaussian.

But this fact requires testing because in [12] is mentioned that the logistic function is an alternative model of RBF that needs to be tested. Therefore, a definition of membership function is required in the neural network.

3.2 Definition 2: The membership function for neural networks is a function that defines the weight for the input variable.

- The T-norm operator used to compute each rule’s firing strength is the multiplication.

The T-norm has been defined in [31] as the intersection of two sets. But the intersection in the case of T1 RBFNN is not used in the receptive fields as occurs with the fuzzy rules before fuzzification in the FLS.

4 Proposal

A. Assemble and calculation of IT2 RBFNN parameters based on the CCD

In CCD and [11], the IT2 CCD is obtained and used to generate the fuzzy rule base that serves as input for the IT2 RBFNN.

The initial calculi of the rule base or the antecedents for the IT2 RBFNN are given by Eq. (27), as were proposed in [11], where 2^k is the matrix that conforms the CCD model, and every pair in the matrix is formed by the possible combinations of the lower and upper limits of control for the variables. These parameters or receptive fields for the rules or the rf are obtained from the process control specifications, and their equivalences to the CCD 2^k model are presented in Table 1.

Table 1 Equivalences of CCD states and quality control limits

CCD symbolic representation	Treatment	Quality control limits
ab = 1	A_{low}, B_{low}	<i>LCLa LCLb</i>
Ab = a	A_{high}, B_{low}	<i>LCLa UCLB</i>
aB = b	A_{low}, B_{High}	<i>UCLA LCLb</i>
AB = ab	A_{high}, B_{High}	<i>UCLA UCLB</i>

$$2^k = \begin{bmatrix} LCLa & LCLb \\ LCLa & UCLB \\ UCLA & LCLb \\ UCLA & UCLB \end{bmatrix} \tag{27}$$

From Eq. (27), the inputs are obtained for the receptive fields on the IT2 RBFNN. The input for the first receptive field in the T1 RBFNN is *LCLa*, and for calculating the receptive fields for the first variable in IT2 RBFNN, additional calculations are needed. Firstly, it is required the spread of the data specifications, which is given by Eq. (28),

$$\sigma_{x_i} = \sum_{i=1}^n \left(\frac{x_i - \bar{x}_i}{n} \right) \tag{28}$$

The IT2 matrix is calculated using the matrix presented in Eq. (27) and σ_{x_i} from Eq. (28). The lower limit of interval represented by \underline{L} is given by Eq. (29), and the upper interval limit \bar{R} is given by Eq. (30), and their respective solutions \bar{y} upper and \underline{y} lower are obtained by interpolation, which are given by Eqs. (31) and (32).

Equation (27) is converted into Eq. (33) to get data in the receptive field of the IT2 RBFNN,

$$\underline{L} = x_i - \sigma_{x_i} \tag{29}$$

$$\bar{R} = x_i + \sigma_{x_i} \tag{30}$$

$$\underline{y} = y_i - \sigma_{y_i} \tag{31}$$

$$\bar{y} = y_i + \sigma_{y_i} \tag{32}$$

$$Rf = \begin{bmatrix} \underline{a} & \bar{a} & \underline{b} & \bar{b} & y_1 & \bar{y}_1 \\ \underline{a} & \bar{a} & \underline{B} & \bar{B} & y_b & \bar{y}_b \\ \underline{A} & \bar{A} & \underline{b} & \bar{b} & y_a & \bar{y}_a \\ \underline{A} & \bar{A} & \underline{B} & \bar{B} & - & \bar{y}_{ab} \\ - & - & - & - & y_{ab} & \bar{y}_{ab} \end{bmatrix} \tag{33}$$

For $i = 1, a, b, ab$

where: Rf represents the receptive field universe, \underline{a} represents the lower limit of the IT2 for the variable a , and this variable (a) is the first receptive field or RF1 in the RBFNN. The same case is for the rest of the elements in the matrix Rf . The interval for the output response is obtained from $\underline{Y}_i, \bar{Y}_i$. Those values represent the lower and upper output or response for a specific receptive field of the inputs.

B. Enhancement and changes to the classic IT2 RBFNN

First, the RBF was changed from logistic to a Gaussian function form, see eq. (3).

Table 2 Input–output data for tests

Sample	X1	X2	Goal
1	139	208	0.6
2	141	214	3.6
3	135	218	1.2
4	142	218	5.4
5	142	210	3.0
6	144	210	4.2
7	141	209	2.1
8	145	203	2.7
9	137	205	−1.5
10	143	218	6.0
11	141	206	1.2
12	146	211	5.7
13	147	211	6.3
14	143	208	3.0
15	135	208	−1.8
16	145	207	3.9
17	144	213	5.1
18	132	200	−6.0
19	135	210	−1.2

Second, the use of the CCD IT2 from [11] is adapted to model the IT2 RBFNN using Eqs. (25)–(31).

C. Improvements

The use of the Huygen’s tractrix (Fig. 3) given by Eq. (34) and the triangular radial basis function (Fig. 4) given by Eq. (35), as RBFs.

$$\mu_{X_i} = \frac{1}{1 + \left| \frac{(x_i - x'_i)}{c} \right|^2} \tag{34}$$

where: x_i represents the mean of the variable, x'_i represents the input, and c represents the spread of the set.

$$\mu_{X_i} = \text{MAX}((\text{MIN}(\frac{x_i - A}{B - A}, \frac{C - x_i}{C - B}), 0)) \tag{35}$$

where: x_i represents the mean of the variable, A and C represent the lower and upper limits, respectively, and B represents the mean of the set.

D. Input–output data

For generating the T1 RBFNN and the IT2 RBFNN, 19 data pairs were used, see Table 2.

E. Conformation of the T1 RBFNN and IT2 RBFNN architectures

The basic architecture for the T1 RBFNN is assembled by three layers defined in the following way: input layer, hidden layer, and output layer, the conformation and the number of neurons needed on every layer can be seen in Table 3.

The architecture for the IT2 RBFNN is assembled by three layers defined in the following way: input layer, hidden layer, and output layer, the conformation and the number of neurons needed on every layer can be seen in Table 4. In this case, a four layer was added for the type reduction and the output.

5 Results

The results of this proposal are organized as follows. First, the test for the logistic function as radial basis or receptive field in IT2 RBFNN without type reduction (Fig. 10) and IT2 RBFNN with type reduction (Fig. 11) is presented. Second, the test for the Gaussian RBF as receptive field in IT2 RBFNN without type reduction (Fig. 12) and IT2 RBFNN with type reduction (Fig. 13) is shown. Third, the test of the Huygen’s tractrix without type reduction (Fig. 14) and with type reduction (Fig. 15) is presented. Fourth, the tests of the triangular RBF without type reduction (Fig. 16) and with type reduction (Fig. 17) are presented.

To calculate the accuracy and the enhancement of the proposal the mean squared error was used to document the variations given by Eq. (36),

$$\text{MSE} = \frac{\sum_{i=1}^n (\hat{Y}_i - Y_i)^2}{n} \tag{36}$$

where: \hat{Y}_i is the goal or expected value, Y_i is the obtained value by the model, and n is the total of samples tested. The values obtained in the experiments are shown in Table 5.

The use of the logistic RBF, as is shown in Table 5 and Fig. 10, demonstrates that this function produces a big error rate with an MSE value of 5.144 for T1 RBFNN and 9.278

Table 3 Description of T1 RBFNN architecture

Layer	Description	Number of neurons	Description
1	Input layer	2	One neuron for every variable
2	Hidden layer (weights layer)	8	Two neurons for every rule, one per every variable
3	Output layer	9	Two neurons for every variable, each neuron contains the weight of input and their respective output Remaining neuron produces overall output

Table 4 Description of IT2 RBFNN architecture

Layer	Description	Number of neurons	Description
1	Input layer	2	One neuron for every variable
2	Hidden layer (weights layer)	16	Four neurons for every rule, one for low value and one for high value per every variable
3	Type-2 Output layer	16	Four neurons for every variable, each neuron contains the weight of input and their respective output in low and high value
4	Type reduction layer and type-1 output	3	Two neurons produce the low and high value (The low and high value constitutes type reduction with two neurons) and the overall output is produced with the remaining neuron

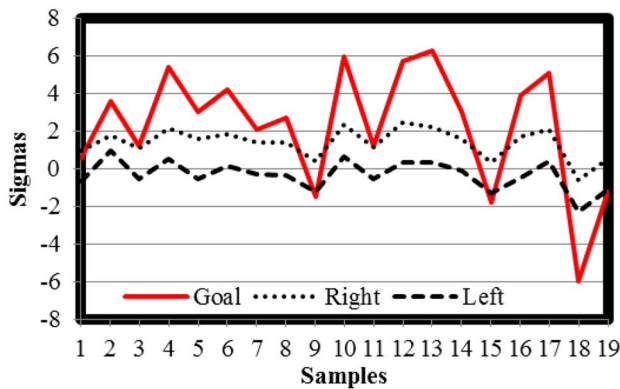


Fig. 10 IT2 RBFNN approximations with logistic RBFs without type reduction

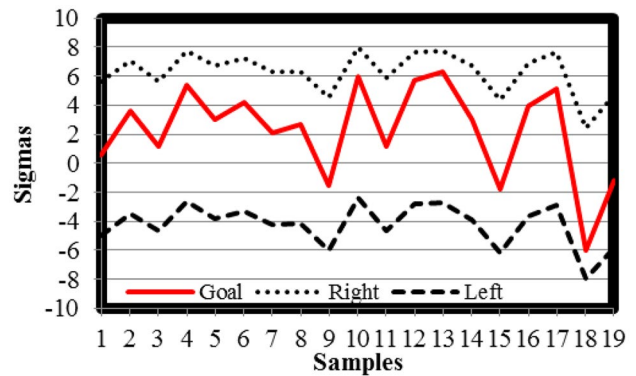


Fig. 12 IT2 RBFNN approximations with Gaussian RBFs without type reduction

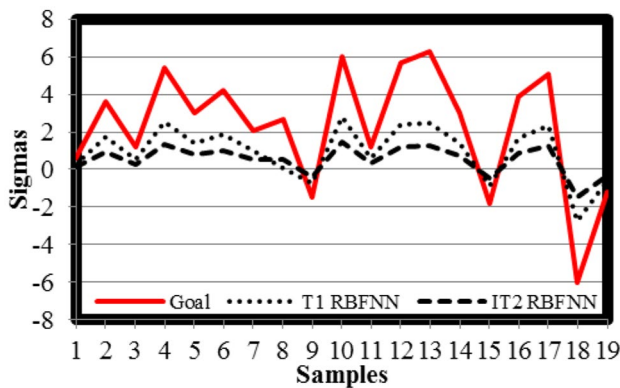


Fig. 11 IT2 RBFNN approximations with logistic RBFs, with type reduction

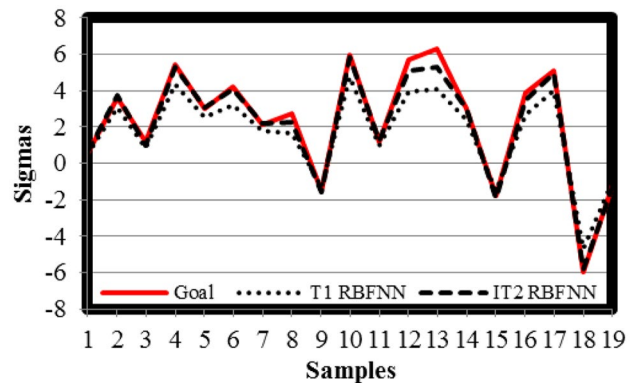


Fig. 13 IT2 RBFNN approximations with Gaussian RBFs, with type reduction

for IT2 RBFNN that increases the error using the IT2 model (see Fig. 11).

The goal values appear inside the interval (Fig. 12) and with the type reduction the values obtained as outputs of IT2 RBFNN are equal or too close to the expected values of prediction (Fig. 13). The classic Gaussian RBF provides good

results for the IT2 model (Fig. 13), when it is compared to their counterpart of T1, in comparison to the logistic RBFNN (Fig. 11), in both types, the classic Gaussian RBF reduced de error in a proportion of 5.22 times for T1 and 47 times for IT2.

Huygen’s tractrix used as RBF shows a particular case, the obtained results appear inside the low and high values of

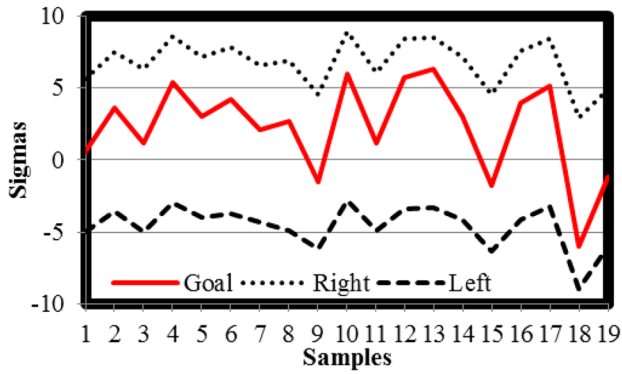


Fig. 14 IT2 RBFNN approximations with Huygen's tractrix RBFs without type reduction

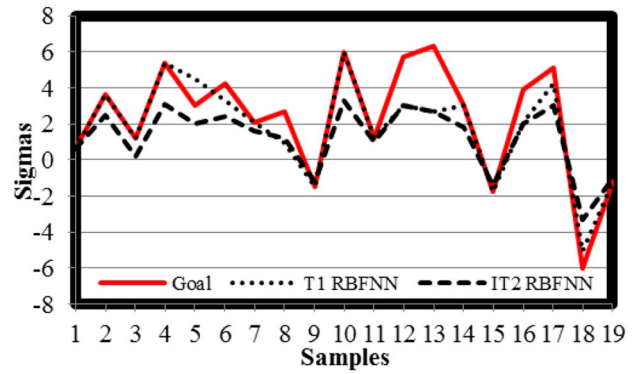


Fig. 17 IT2 RBFNN approximations with triangular RBFs, with type reduction

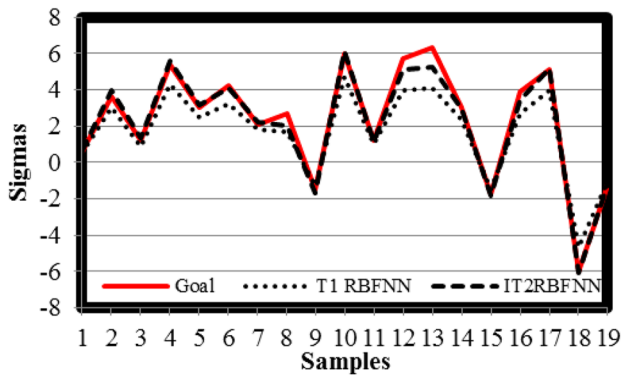


Fig. 15 IT2 RBFNN approximations with Huygen's tractrix RBFs, with type reduction

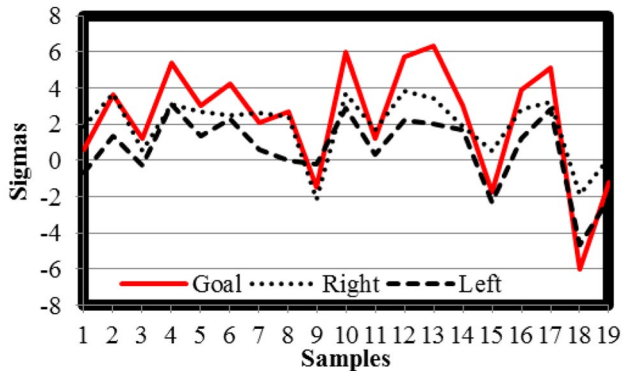


Fig. 16 IT2 RBFNN approximations with triangular RBFs, without type reduction

Table 5 MSE obtained in the experiments

Model	MSE
T1 RBFNN (Logistic RBF)	5.144
T1 RBFNN (Gaussian RBF)	0.9853
T1 RBFNN (Huygen's tractrix RBF)	0.5511
T1 RBFNN (Triangular RBF)	1.745
IT2 RBFNN (Logistic RBF)	9.278
IT2 RBFNN (Gaussian RBF)	0.1974
IT2 RBFNN (Huygen's tractrix RBF)	0.1071
IT2 RBFNN (Triangular RBF)	3.335

the type-2 interval, similar to the results obtained using the Gaussian RBF (Fig. 12), as shown in (Fig. 14). This function never has been used in literature, but their results are excellent due to a reduction near to the 50% when contrasted with the Gaussian RBF in both types (type-1 and type-2 RBFNN), as is shown in (Fig. 15).

The triangular RBF presents an interval of predictions above the expected value (Fig. 16), it can be said that this phenomenon occurs because the IT2 RBFNN system has a non-linear behavior in contrast to the triangular RBF prediction values. Therefore, the error is increased in a linear way. The error rate achieved by the IT2 RBFNN system represents the third part of the obtained by the Logistic RBF and the double of the Gaussian RBF (see Table 5). The predictions are so similar in both models type-1 and type-2 (Fig. 17), nonetheless, T1 RBFNN achieves better results.

6 Conclusion

The use of CCD provides a better method to assemble the fuzzy rule base in a simplified and compact manner with the advantage of a compact base with a few rules that provide precise approximations as outputs.

A simplified and compact rule base reduces the computational time expend in the calculations.

The use of the logistic RBF turned out not to be suitable for a network without training, and this is part of future work. The logistic function does not produce better results by their shape that is non-symmetrical, and these are future research for the application of this class of systems.

The adapted model of Fig. 9 provides an accurate result without training.

The classic Gaussian RBF provides good results for the IT2 model when it is compared to their counterpart of T1, achieving a prediction five times better.

The most important enhancement is the use of a RBFNN which does not need training and provides accurate results.

The results have shown a significant enhancement with the application of the Huygen's tractrix as RBF, obtaining a 50% in error reduction in comparison to the Gaussian T1 RBFNN and IT2 RBFNN.

Author contribution All the authors contributed significantly to the work in accordance with the order provided.

Code availability Not applicable.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Conflict of interest The authors declare no competing interests.

References

- Mendel JM (2017) Uncertain rule-based fuzzy systems. Springer, Introduction and new directions
- Rubio-Solis A, Panoutsos G (2015) Interval type-2 radial basis function neural network: a modeling framework. *IEEE Trans Fuzzy Syst* 23(2):457–473
- Dorantes PNM, Nieto González J, Méndez GM (2014) Fault detection systems via a novel hybrid methodology for fuzzy logic systems based on individual base inference and statistical process control, *Latin Am Trans IEEE (Revista IEEE America Latina)*, 12(4):706–712.
- Macvicar-Whelan PJ (1978) Fuzzy sets, the concept of height, and the hedge very. *IEEE Trans Syst Man Cybern* 8(6):507–511
- Dongale TD, Kulkarni TG, Kadam PA, Muldholkar RR (2012) Simplified method for compiling rule base matrix". *Int J Soft Comput Eng (IJSCE)* 2(1).
- Zarandi FMH, Alaeddini A, Turksen IB (2008) A hybrid fuzzy adaptative sampling-run rules for the Shewart control charts. *Inf Sci* 178:1152–1170
- Serturk S, Erginel N (2009) Development of fuzzy X-R and X-S control charts using α -cuts. *Inf Sci* 179:1542–1551
- Gülbay M, Kahraman C (2007) An alternative approach to fuzzy control charts: Direct fuzzy approach. *Inf Sci* 177:1463–1480
- Dorantes PNM, Mexicano Santoyo A, Méndez GM (2018) Modeling type-1 singleton fuzzy logic systems using statistical parameters in foundry temperature control application. *Smart and Sustainable Manufacturing systems ASTM* 2(1):180–203
- Dorantes PNM, Méndez GM (2015) Non-iterative radial basis function neural networks to quality control via image processing. *IEEE Lat Am Trans* 13(10):3457–3451
- Méndez GM, Dorantes PNM, Mexicano SA (2019) Interval type-2 fuzzy logic systems optimized by central composite design to create a simplified fuzzy rule base in image processing for quality control application. *Int J Adv Manuf Technol* 102(9):3757–3766
- Jang RS, Sun C, Mizutani E (1997) Neuro-fuzzy and soft computing. A computational approach to learning and machine intelligence, Upper Saddle River, NJ: Prentice-Hall.
- Rubio-Solis A, Melin P, Martínez-Hernández U, Panoutsos G (2018) General type-2 radial basis function neural network: a data-driven fuzzy model. *IEEE Trans Fuzzy Syst* 27(2):333–347
- Tavoosi J, Suratgar AA, Menhaj MB (2016) Nonlinear system identification based on a self-organizing type-2 fuzzy RBFN. *Eng Appl Artif Intell* 54:26–38
- Ngo PD, Shin YC (2016) Modeling of unstructured uncertainties and robust controlling of nonlinear dynamic systems based on type-2 fuzzy basis function networks. *Eng Appl Artif Intell* 53:74–85
- Baklouti N, Abraham A, Alimi AM (2018) A Beta basis function Interval Type-2 Fuzzy Neural Network for time series applications. *Eng Appl Artif Intell* 71:259–274
- Mendel JM (2001) Uncertain rule-based fuzzy logic systems: introduction and new directions. Prentice-Hall, Upper Saddle River, NJ
- Interval type-2 fuzzy radial basis function neuron model for neural networks. In 2018 International Conference on Fuzzy Theory and Its Applications iFUZZY (367–368). KIIS.
- Lall S, Saha A, Konar A, Laha M, Ralescu AL, kumar Mallik K, Nagar AK (2016) EEG-based mind driven type writer by fuzzy radial basis function neural classifier. In 2016 Int Joint Conf Neu Net (IJCNN) (pp. 1076–1082). IEEE.
- Luo C, Tan C, Wang X, Zheng Y (2019) An evolving recurrent interval type-2 intuitionistic fuzzy neural network for online learning and time series prediction. *Appl Soft Comput* 78:150–163.
- Rubio-Solis A, Panoutsos G, Beltran-Perez C, Martínez-Hernández U (2020) A multilayer interval type-2 fuzzy extreme learning machine for the recognition of walking activities and gait events using wearable sensors. *Neurocomputing* 389:42–55
- Pal SS, Kar S (2019) A hybridized forecasting method based on weight adjustment of neural network using generalized type-2 fuzzy set. *Int J Fuzzy Syst* 21(1):308–320
- Rubio-Solis A, Baraka A, Panoutsos G, Thornton S (2018) Data-driven interval type-2 fuzzy modelling for the classification of imbalanced data. In *Practical issues of intelligent innovations* (37–51). Springer, Cham.
- Zhang C, Zou W, Cheng N, Gao J (2018) Trajectory tracking control for rotary steerable systems using interval type-2 fuzzy logic and reinforcement learning. *J Franklin Inst* 355(2):803–826
- Kim WD, Oh SK, Seo K (2016) Design of interval type-2 fcm-based neural networks. In 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th Int Sympo Adv Intel Sys (ISIS) (759–763). IEEE.
- Zirkohi MM (2021) Adaptive interval type-2 fuzzy recurrent RBFNN control design using ellipsoidal membership functions with application to MEMS gyroscope. *ISA transactions*.

27. Wang M, Wang Y, Chen G (2021) Interval type-2 fuzzy neural network based constrained GPC for NH₃ flow in SCR de-NO_x process. *Neural Comput & Applic* 33:16057–16078. <https://doi.org/10.1007/s00521-021-06227->
28. Qasem SN, Mohammadzadeh A (2021) A deep learned type-2 fuzzy neural network: singular value decomposition approach. *Appl Soft Comput* 105, 107244.
29. Tavoosi J (2020) Sliding mode control of a class of nonlinear systems based on recurrent type-2 fuzzy RBFN. *International Journal of Mechatronics and Automation* 7(2):72–80
30. Rubio-Solis A, Salgado-Jimenez T, Garcia-Valdovinos LG, Nava-Balanzar L, Hernandez-Hernandez RA, Martinez-Hernandez U (2020) An evolutionary general type-2 fuzzy neural network applied to trajectory planning in remotely operated underwater vehicles. In 2020 IEEE Int Conf Fuzz Sys (FUZZ-IEEE) (1–8). IEEE.
31. Wang LX (1996) *A course in fuzzy systems and control*, Prentice Hall. 1st Edition.

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