Article:

Baraka, A. and Panoutsos, G. (2019) Long-term learning for type-2 neural-fuzzy systems. Fuzzy Sets and Systems, 368. pp. 59-81. ISSN: 0165-0114

https://doi.org/10.1016/j.fss.2018.12.014

Long-term Learning for Type-2 Neural-Fuzzy Systems

Ali Baraka and George Panoutsos
Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, S1 3JD, UK
Email: {a.baraka, g.panoutsos}@sheffield.ac.uk

Abstract—The development of a new long-term learning framework for interval-valued neural-fuzzy systems is presented for the first time in this article. The need for such a framework is twofold: to address continuous batch learning of data sets, and to take advantage the extra degree of freedom that type-2 Fuzzy Logic systems offer for better model predictive ability. The presented long-term learning framework uses principles of granular computing (GrC) to capture information/knowledge from raw data in the form of interval-valued sets in order to build a computational mechanism that has the ability to adapt to new information in an additive and long-term learning fashion. The latter, is to accommodate new input-output mappings and new classes of data without significantly disturbing existing input-output mappings, therefore maintaining existing performance while creating and integrating new knowledge (rules). This is achieved via an iterative algorithmic process, which involves a two-step operation: iterative rule-base growth (capturing new knowledge) and iterative rule-base pruning (removing redundant knowledge) for type-2 rules. The two-step operation helps create a growing, but sustainable model structure. The performance of the proposed system is demonstrated using a number of well-known non-linear benchmark functions as well as a highly nonlinear multivariate real industrial case study. Simulation results show that the performance of the original model structure is maintained and it is comparable to the updated model's performance following the incremental learning routine. The study is concluded by evaluating the performance of the proposed framework in frequent and consecutive model updates where the balance between model accuracy and complexity is further assessed.

Keywords — Radial-Basis-Function Neural Fuzzy (RBF-NF) System, Interval-Valued Fuzzy Logic System, Granular Computing (GrC), Long-term Learning, Incremental Learning, Similarity Measures for type-2 Fuzzy Sets.

1. Introduction

Soft-computing methods and systems have been applied in the past to solve many real-world problems with great success [1]. However, the complexities and continuously changing characteristics associated with some real-world problems, such as chaotic time series prediction [2, 3] and adaptive decision-making modelling and control systems [4] require computational frameworks that are able to adapt incrementally in an online manner [5], learn

* Corresponding authors: {a.baraka, g.panoutsos}@sheffield.ac.uk

and generalise from data automatically, and dynamically evolve and change their structure to accommodate new knowledge [6, 7]. In the literature, a number of adaptive neural fuzzy systems have been developed to deal with such problems, for example online learning [8], incremental learning [9, 10], lifelong learning [11], and knowledge-based learning neural networks [12]. Recent research in online learning concentrates on adaptive learning rates to follow time-varying distributions [5, 12, 13]. Incremental learning is the process of repeatedly training a model with new data without destroying the old prototype patterns [14], while lifelong learning also termed long-term or "continuous learning" addresses learning through the entire lifespan of a system [15]. In [7], a dynamic evolving neural fuzzy system for the prediction of time-series data was proposed; this approach requires to normalise all data prior to training, which infers that all the data points must be present prior to any training. Other dynamic neuro-fuzzy system approaches include recurrent fuzzy neural networks [16], a self-constructing neural fuzzy inference network [17], and a dynamic parsimonious fuzzy neural network [6]. However, the aforementioned methods never prune the rules once generated, regardless of their relevance. Therefore, a large number of redundant rules maybe generated each time new data (information) is available. Other alternative approaches for online learning which adapt the feature of rule creation and pruning mechanisms exist, such as the sequential adaptive fuzzy inference system [2], and the most recent sequential probabilistic learning for adaptive fuzzy inference system [5]. In on-line learning, only one data pattern is provided at a time and then discarded after the learning process has been completed. The online methodology is not very demanding on computing resources and at the same time it fits well with dynamically changing environments.

All of the adaptive learning methods in soft-computing that have been discussed so far are in the field of type-1 fuzzy logic systems (T1-FLSs). In recent years, research on type-2 fuzzy logic systems (T2-FLSs) have attracted significant attention [18, 19]. This is due to their ability to capture uncertainty, hence their tolerance to impression associated with the input data. T2-FLSs are the extensions of T1-FLSs, where the membership functions (MFs) associated with the fuzzy rules are type-2 fuzzy sets (T2-FSs) [19-21]. A T2-FS has a membership function (MF) that is itself a fuzzy set in [0, 1], unlike a normal fuzzy set (T1-FS) where the membership degree has a crisp number in [0, 1] as in [22]. T2-FLSs take the advantage of the extra degree of freedom from the type-2 footprint of uncertainty (FOU) to better handle uncertainties associated with the meaning of words [23]. In addition, T2-FLSs use large number of T1-FSs which are embedded within the FOUs of the T2-FSs [24]. The studies reported in [25, 26] confirm that due to the additional degree of freedom from the FOU of the T2-FSs, T2-FLSs are more accurate than their T1-FLSs counterparts. Usually, the T2-FLS is more computationally expensive than that of its T1-FLS counterpart mainly because of the complexity associated with the type-reduction stage to reduce the T2-FS to a type reduced T1-FS. Because of the great computational complexity involved in processing T2-FLS especially during the type reduction stage, the interval type-2 fuzzy logic system (IT2-FLS) or interval-valued fuzzy logic system (IV-FLS) is the most widely used type of T2-FLSs. Here, the secondary membership function is an interval i.e. the secondary grades are all equal to unity. Using this type of fuzzy set considerably reduces the type reduction stage. In this article, the IV-FLS modelling is used to simplify the overall computational effort [19-21]. Many methods have been proposed to design of interval-valued fuzzy logic systems [27, 28].

The online learning methodology has also been recently investigated under the framework of type-2 fuzzy logic systems [29]. A self-evolving interval type-2 fuzzy neural network is proposed in [30], which learns its structure and the corresponding parameters in an online manner. In [30], the antecedent and consequent parameters of the network are optimised via a gradient descent algorithm and rule-ordered Kalman filter algorithm respectively; its performance was validated for time-varying systems. In [31], the authors proposed a mutually recurrent interval type-2 neural fuzzy system for the identification of non-linear and time-varying systems. The proposed structure also used the gradient descent and rule-ordered Kalman filter algorithm for parameters tuning [31]. In more recent studies [32-34], online leaning methods have been proposed. In [32] the TSK-Type-Based self-evolving compensatory interval type-2 fuzzy neural network improves the system's robustness in noisy environments, and a Mamdani-type interval type-2 neural fuzzy chip with on-chip incremental learning ability [33] utilises a simplified type-reduction operation into an interval Type-2 NFSs to reduce the computational cost. In [34], Lin et al. proposed an interval type-2 NFS for online system identification and feature elimination. The proposed structure possesses a self-organising property that can automatically generate fuzzy rule and optimise its structure via a gradient descent based approach.

The aforementioned adaptive learning methodologies that have been proposed so far incrementally evolve their structure and optimise the parameters. The online adaptive learning ability makes it feasible for learning data streams that are generated from non-stationary environments, for example in processes where time-series data are generated (i.e., evolving data) [7]. However, in some industrial/manufacturing applications such as Charpy impact energy test for mechanical properties of heat-treated steels [9] and friction stir welding [35], obtaining batch data is a slow and expensive process and one of the practical challenges of such processes is the new small/medium (compared to the original data size) size of batch points. In this case, the batch learning paradigm is often assumed, where the model structure uses all training examples simultaneously and allowed to use them as often as desired [36]. Developing efficient data-driven computational models require significant effort and the training process is highly dependent on expert knowledge [25, 26]. Repeating the whole modelling process is often a laborious and non-automated process as well as time-consuming.

There is no guarantee that the new model will retain a good performance comparable to the original model [35]. Therefore, this motivates the need for the development of a system that has the ability to learn from an initial batch of data (with the help of an appropriate training algorithm) and periodically adapt to new data when these are available. An additional need is to include the capability to interact with a changing environment in a continuous fashion and also to have an open structure; this entails to dynamically expand the system's structure to accommodate new data/information – without significantly disturbing the initial model structure. A rule pruning mechanism would also be needed, in order to remove/prune redundant rules that have limited contribution to the system's performance [35].

To the best of our knowledge, there is no existing work on batch incremental learning in the field of type-2 fuzzy logic systems. In this research article, for the first time a new long-term learning structure is proposed, where the initial model uses principles of granular computing to capture information/knowledge from raw data in the form of interval-valued sets in order to build a computational mechanism that has the ability to adapt to new information in an additive and long-term learning fashion. The latter, is to accommodate new input-output mappings and new classes of data without significantly disturbing existing input-output mappings, therefore maintaining existing performance while creating and integrating new knowledge (rules). This is achieved via an iterative algorithmic process, which involves a two-step operation: iterative rule-base growth (capturing new knowledge) and iterative rule-base pruning (removing redundant knowledge) for type-2 rules. The main contribution of our proposed framework relies in its ability to provide a reliable model updating procedure that resulted in a dynamically expandable framework without ignoring any previously obtained knowledge. The proposed architecture has satisfied the requirements needed for incremental learning as it can handle the short and long-term change in the input conditions in a lifelong learning mode by incrementally updating its structure to accommodate the change in the process input data space. The proposed long-term learning system has also the ability to improve its structure periodically by removing redundant rules. In this research work, a number of key characteristics of incremental (long-term) learning are adopted as follows [9, 37]:

1. An initial data set is trained to construct the initial/original model.
2. Each time, a new batch of points is sequentially – batch by batch – made available to the model.
3. The model dynamically expands its structure to accommodate the uncovered data by the original model without disturbing the original structure.
4. There is the ability to 'memorise' the knowledge acquired from the original model.

5. There is the ability to improve the system's structure over time by pruning rules that evolved to be redundant, which allows the model to be used in a long-term learning mode.

The rest of the article is organised as follows: the systematic modelling framework of interval-valued radial basis function neural fuzzy (IV-RBF-NF) model and the proposed long-term learning mechanism are presented in Section 2. Simulation results are presented in Section 3, which include three non-linear benchmark functions and one real application in civil engineering. Finally, concluding remarks are provided in Section 4.

2. Long-term Learning Framework

This section introduces the structure of the proposed long-term learning framework, which is based on an interval-valued fuzzy logic system, realised with a radial-basis-function neural network. The core structure of the proposed framework is similar to the modelling frameworks that presented in [25, 26], which is based on an interval-valued radial basis function neural network. However, in the context of this article the development of a long-term learning framework is considered rather than only a modelling framework. The overall structure of the proposed long-term learning is shown in Fig. 1. This includes a novelty detection algorithm to determine and classify the input data based a predefined threshold into novel and partially novel data. The proposed long-term learning framework assumes that the new input data that are presented to the model are valid for learning and clean from noise. The structure can dynamically generate new rules when new data available. After the new rules are created and optimised, the optimised new rules are added to the structure of the original model (incremental update). Thereafter, the redundant rules of the incrementally updated model are pruned to improve the overall interpretability and to prevent the potentially unsustainable rule-base growth. This also serves to further optimise the inference mechanism of the IV-FLS. Finally, the pruned incrementally updated model is fine-tuned via a constrained parametric optimisation algorithm.

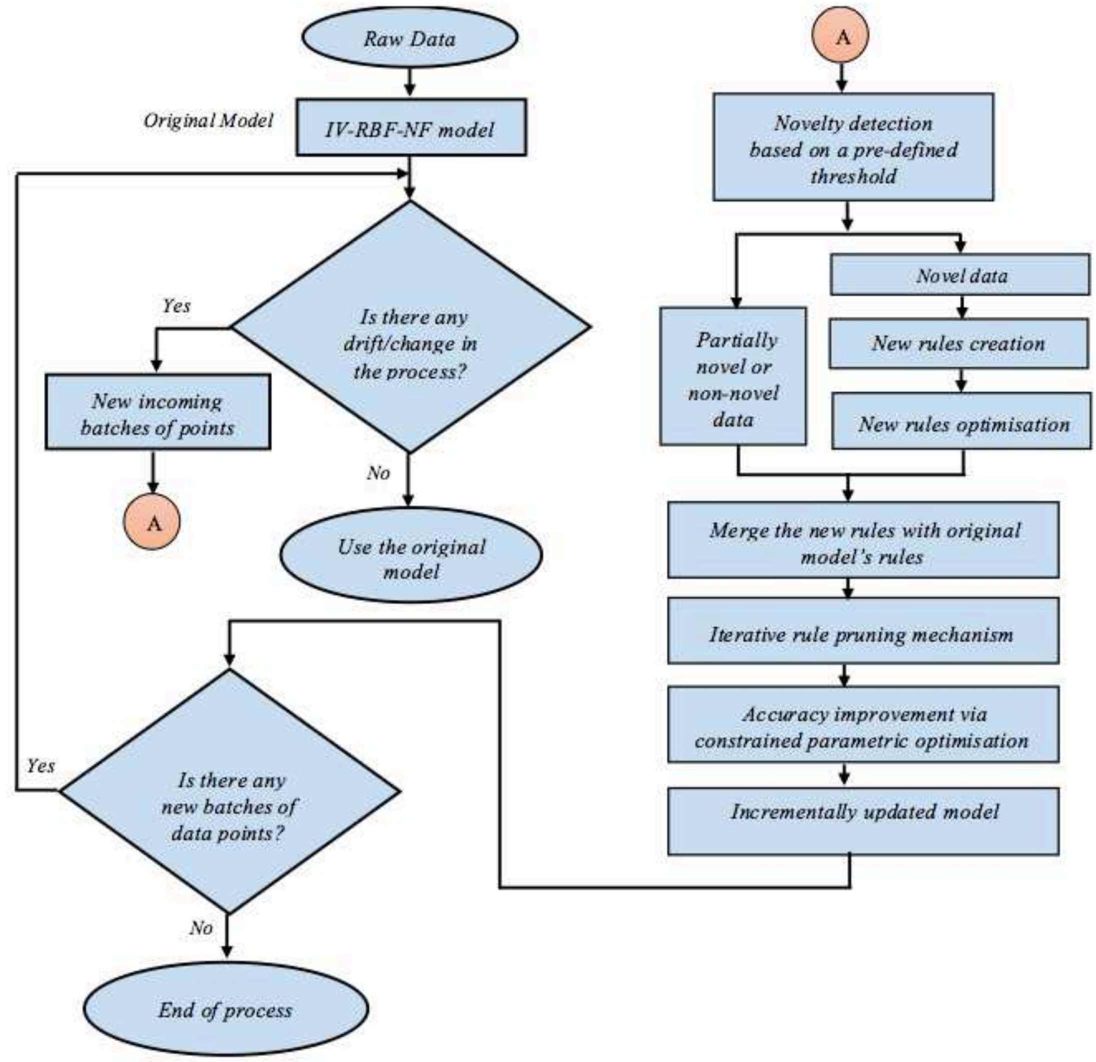


Fig. 1. The structure of the long-term learning framework.

The proposed long-term learning framework is provided as an algorithmic procedure as follows:

Algorithm 1 Long-term Learning

Input: Training and Testing Data

Step 1: The first batch of data points is available.
Step 2: Generate the initial interval-valued fuzzy rules:
- **Step 2-1:** Determine the antecedent parameters from the data granulation process described in Section 2.
- **Step 2-2:** Initialise the values of the consequent parameters.
- **Step 2-3:** Use the adaptive back-propagation algorithm to optimise the initial IV-RBF-NF structure.
- **Step 2-4:** Measure the performance of the model using *RMSE*, *VAF* %, and $\boldsymbol{R^2}$.

Step 3: while a new batch of points is available do
Step 4: Pass the new batch of points into the novelty detection algorithm using a pre-defined threshold.
- **Step 4-1:** Construct new rules to accommodate the novel data set.
- **Step 4-1-1:** Generate the initial new rules using the data granulation process described in

Section 2.

Step 4-1-2: Optimise the initial structure of the rules by using the adaptive back-propagation algorithm.

Step 5: Merge the original model with the generated rules to expand the structure of the original model.

Step 6: Apply the pruning mechanism described in Section 2 to manage the redundant rules and improve the structure of the updated model.

Step 7: Optimise the pruned structure with constraints and measure its performance using $\boldsymbol{RMSE}$, $\boldsymbol{VAF}$ %, and $\boldsymbol{R^2}$ on both old data and the new data set.

Step 8: If new incoming batches of points are available go to **Step 4**; otherwise stop.

Output: Incrementally updated IV-RBF-NF model

A detailed description of the overall diagram is provided in the following subsections.

2.1. Interval-Valued Radial Basis Function Neural Fuzzy (IV-RBF-NF) Model

The interval-valued radial basis function neural fuzzy (IV-RBF-NF) model proposed in this study has a similar structure to the structure used in [25, 26] (as shown in Fig. 2). The structure of the proposed IV-RBF-NF model different from one of the first and most popular hybrid neural fuzzy techniques of adaptive neuro fuzzy system (ANFIS) which was first introduced by Jang [38]. The main distinction is that the structure of ANFIS consists of five layers; the antecedent part of each fuzzy rule is of the Mamdani-type model which is made up of type-1 fuzzy sets and the consequent is of the Sugeno-type model which is made up of linear equations (deterministic). It also combines the recursive least-square estimation and the steepest descent optimisation algorithms for optimising both antecedent and consequent parameters respectively. While the proposed IV-RBF-NF model is a RBF neural network, which consists of six layers. The antecedent and consequent of each fuzzy rule are both of the Mamdani-type model (interval-valued fuzzy sets). An interval-valued fuzzy set differs from type-1 fuzzy set (normal fuzzy set) in which it has a membership function that itself a type-1 fuzzy set. Unlike type-1 fuzzy set where the membership function includes a crisp number in [0, 1]. This characteristic gives the interval-valued fuzzy set an extra degree of freedom to better handle uncertainties associated in the input space [25, 26]. The initial structure of the proposed model is estimated via an iterative data granulation algorithm, which is described in detail in Section 2.2. The interval-valued MFs – FOUs – are generated based on the heuristic approach described in [39], and the model is further parametrically optimised via an adaptive back propagation of the error approach [40]. The consequent part of each fuzzy rule is of the Mamdani type model, each of which has the following linguistic **IF-THEN** form:

$$\boldsymbol{Rule_i}\text{: } \boldsymbol{IF}\ x_1\ is\ \tilde{A}_1^i\ \boldsymbol{AND}, \dots, \boldsymbol{AND}\ x_n\ is\ \tilde{A}_n^i, \boldsymbol{THEN}\ y\ is\ \tilde{B}^i \quad (1)$$

where $x_{j=1}, \dots, x_n$, are the input vectors, $\tilde{A}_n^i$, $j = 1, \dots, n$ and $i = 1, \dots, M$ and $\tilde{B}^i$ are the interval-valued fuzzy sets, M is the number of rules, and n is the number of system's inputs. The mathematical description of the IV-RBF-NF model is provided below:

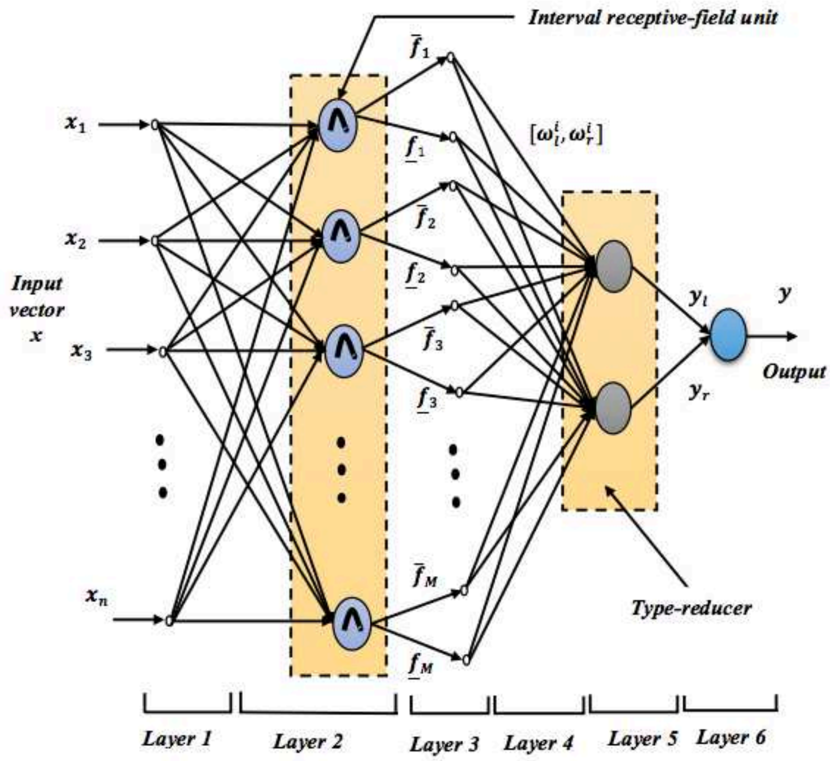


Fig. 2. IV-RBF-NF model general structure.

1) Layer 1 (Input Layer):

This layer only transmits the current input values to the next layer directly without performing any computation. Each node in this layer represents one crisp variable from the multidimensional input data $\vec{x} = [x_{j=1}, \dots, x_n] \in R^n$, where n is the number of input variable.

2) Layer 2 (Fuzzification Layer):

Each node in this layer uses an interval-valued MF to perform the fuzzification process in order to produce the upper and lower intervals $[\underline{\mu}_{\tilde{A}_j^i}\ ,\ \overline{\mu}_{\tilde{A}_j^i}]$. With the choice of a Gaussian primary MF having fixed mean m_j^i and uncertain standard deviation that the value in the interval $\sigma_j^i \in [\sigma_{j1}^i, \sigma_{j2}^i]$ can be stated as:

$$\tilde{A}_j^i(x_j) = exp\left[-\frac{1}{2}\left(\frac{x_j - m_j^i}{\sigma_j^i}\right)^2\right] \equiv N(m_j^i, \sigma_j^i; x_j),\ \ \sigma_j^i \in [\sigma_{j1}^i, \sigma_{j2}^i] \quad (2)$$

where $\tilde{A}_j^i(x_j)$ is the ith fuzzy set in input variable x_j. It is clear that the IV-FS is bounded by the upper MF $\overline{\mu}_{\tilde{A}_j^i}$ and lower MF $\underline{\mu}_{\tilde{A}_j^i}$, and the area in between is called the footprint of uncertainty (FOU). The upper membership function (UMF) is $\overline{\mu}_{\tilde{A}_j^i} = N(m_j^i, \sigma_{j1}^i; x_j)$ and the lower membership function (LMF) is $\underline{\mu}_{\tilde{A}_j^i} = N(m_j^i, \sigma_{j2}^i; x_j)$. Each node in this layer corresponds to a linguistic variable (e.g. fast, very fast, etc.) and the output of each node can be represented as an interval in $[\underline{\mu}_{\tilde{A}_j^i}\ ,\ \overline{\mu}_{\tilde{A}_j^i}]$.

3) Layer 3 (Rules Firing Layer):

This layer performs the join ⊔ and meet ⊓ operations, which are new concepts introduced in type-2 fuzzy logic theory that are used instead of intersection and union operators in type-1 fuzzy logic theory [41]. The output of this layer is an interval type-1 fuzzy set (i.e. rule node firing strength). The node rule firing strength F^i is calculated by using an algebraic product operation $F^i = \left[\underline{f}^i(x_j), \overline{f}^i(x_j)\right]$, where $\underline{f}^i(x_j)$ and $\overline{f}\ (x_j)$ can be written, where * denotes the meet operation under product t-norm:

$$\overline{f}^i = \overline{\mu}_{\tilde{A}_1^i}(x_1) * \dots * \overline{\mu}_{\tilde{A}_n^i}(x_n) = \prod_{j=1}^{n} \overline{\mu}_{\tilde{A}_j^i} \tag{3}$$

$$\underline{f}^i = \underline{\mu}_{\tilde{A}_1^i}(x_1) * \dots * \underline{\mu}_{\tilde{A}_n^i}(x_n) = \prod_{j=1}^{n} \underline{\mu}_{\tilde{A}_j^i} \tag{4}$$

4) Layer 4 (Compensatory Firing Layer):

Each node in this layer has its corresponding firing strength, which is calculated from layer 3. The layer defines the consequents of the rule nodes and the links between this layer and the next layer consist of interval weighing factors $[\omega_l^i, \omega_r^i]$, which will decide outputs of this network.

5) Layer 5 (Type-reducer Layer):

This layer generates a T1-FS output, which is then converted to a numeric output through the defuziffication layer. This T1-FS is also an interval set $[\omega_l^i, \omega_r^i]$, which is determined by its two end points (i.e, left end-point l, and right end-point r). The centroid of the interval-valued fuzzy set $\tilde{B}$ which is $\mathrm{G}_{\tilde{B}}(\mathrm{y})$, for the case of centre-of-sets (COS) type-reduction method can be represented by the union of the centroids all the embedded interval-valued FSs of $\tilde{B}$, i.e., [21]

$$\mathrm{G}_{\tilde{B}}(\mathrm{y}) = \int_{\theta_1 \in J_{y_1}} \dots \int_{\theta_N \in J_{y_M}} 1 \Big/ \frac{\sum_{i=1}^{M} y_i \theta_i}{\sum_{i=1}^{M} \theta_i} = \left[\omega_l^i, \omega_r^i\right] \tag{5}$$

where θ_i is the primary membership for y_i and each θ_i belongs to some interval in [0, 1].

The extended output is computed as follows:

$$Y_{COS} = [y_l^i, y_r^i] = \int_{\omega^1 \in [\omega_l^1, \omega_r^1]} \dots \int_{\omega^M \in [\omega_l^M, \omega_r^M]} \int_{f^1 \in \left[\underline{f}^1, \overline{f}^1\right]} \dots \int_{f^M \in \left[\underline{f}^M, \overline{f}^M\right]} 1 \Big/ \frac{\sum_{i=1}^{M} f^i \omega^i}{\sum_{i=1}^{M} f^i} \tag{6}$$

Each node in this layer calculates this interval output. The outputs y_l and y_r can be computed by $y_l = \frac{\sum_{i=1}^{M} f_l^i \omega_l^i}{\sum_{i=1}^{M} f_l^i}$ and $y_r = \frac{\sum_{i=1}^{M} f_r^i \omega_r^i}{\sum_{i=1}^{M} f_r^i}$. According to [21, 42], the interval type-reduced sets y_l and y_r can be expressed as:

$$y_l = \frac{\sum_{i=1}^{L} \overline{f}^i \omega_l^i + \sum_{i=L+1}^{M} \underline{f}^i \omega_l^i}{\sum_{i=1}^{L} \overline{f}^i + \sum_{i=L+1}^{M} \underline{f}^i}, \quad y_r = \frac{\sum_{i=1}^{R} \underline{f}^i \omega_r^i + \sum_{i=L+1}^{M} \overline{f}^i \omega_r^i}{\sum_{i=1}^{R} \underline{f}^i + \sum_{i=R+1}^{M} \overline{f}^i} \tag{7}$$

where M is the number of rules in the rule base of the IV-RBF-NF model, i is the index of the rules, and $[\omega_l^i, \omega_r^i]$ represents the centroid interval of the consequent type-2 FS of the ith rule. ω_l^i, ω_r^i are also called the weighting factors of the consequent part of the IV-RBF-NF model [43]. The values of L and R can be obtained from the iterative Karnik-Mendel type-reduction method [21, 42].

6) Layer 6 (Defuzzification Layer):

Once y_l and y_r are obtained by using the KarnikMendel iterative type-reduction approach [21, 42], the type-reduced set can be defuzzified to compute the output values of the system (crisp). For an interval type-reduced set, the defuzzified output y. In this layer, the defuzzified output is then computed by the average of y_l and y_r.

2.2. Initial Structure Identification of the IV-RBF-NF Model

Extracting meaningful knowledge out of numerical data is a critical step in the process of developing efficient data-driven computational intelligence models [44]. The iterative human-like information granulation algorithm of Granular Computing (GrC) described in detail in [36, 45] is used to group similar input-output mappings based on their features and characteristics such as similarity, indistinguishability, coherency, proximity or functionality [46]. Compared to other clustering methods, GrC has shown its efficiency and simplicity in extracting information out of raw data. In GrC, the information granules grow from the raw data rather than generated from the algorithm. Additionally, it offers transparency through the information collected during the information granulation process. These advantages make the GrC algorithm ideal for combing it with IV-RBF-NF model. To achieve the information granulation, the iterative GrC employs a criterion measure that calculates a compatibility index based on granular similarity [45]. In essence, the information granulation is a two stage iterative process that can be carried out as follows: First, find the most two compatible information granules and merge them together to form a new information granule containing both original granules and then repeat the process of finding the most two compact information granules and forming a new information granules until a satisfactory level of data granulation is achieved. The compatibility measure $C_{(A,B)}$ between two information granules A and B can be defined as a function of distance between two information granules and the information density of the obtained information granule as resulted from merging the two information granules [45]:

$$C_{(A,B)} = d_{MAX} - d_{(A,B)} . e^{-\alpha\left(\frac{Q_{(A,B)}/Q_{MAX}}{L_{(A,B)}/L_{MAX}}\right)} \tag{8}$$

where d_{MAX} is the maximum distance in the input data set; and $d_{(A,B)}$ is weighted the average multi-dimensional Euclidean distance between two information granules A and B calculated by $d_{(A,B)} = \frac{\sum_{j=1}^{n} \omega_j (max(A_j^+ - B_j^+) - min\,(A_j^- - B_j^-))}{n}$; A_j^-, A_j^+ are the lower and upper boundaries for information granule A in dimension j; ω_j is the dimensional weighting importance factor for dimension j and n is the total number of dimensions; α the weight in the interval [0, 1] that is used in order to balance the requirements between distance and compactness; Q_{MAX} represents the total number of granules in the input data set; the term $Q_{(A,B)}$ is the number of granule of the resulting information granule; L_{MAX} is the maximum length of an information granule in the input data set; $L_{(A,B)}$ is the length of the resulting information granule in multi-dimension such that $\mathrm{L_{(A,B)}} = \sum_{\mathrm{j=1}}^{\mathrm{n}}(\max\,(\mathrm{A_j^+} - \mathrm{B_j^+}) - \min\,(\mathrm{A_j^-} - \mathrm{B_j^-}))$. To illustrate the concept of iterative data granulation, Fig. 3 shows granulated data (5 granules) obtained from 2-dimensioanl synthetic data consisting of 200 points after a number of iterations. In iterative granular computing, the data are granulated/clustered across each dimension in the input space simultaneously to capture the dynamics of the process.

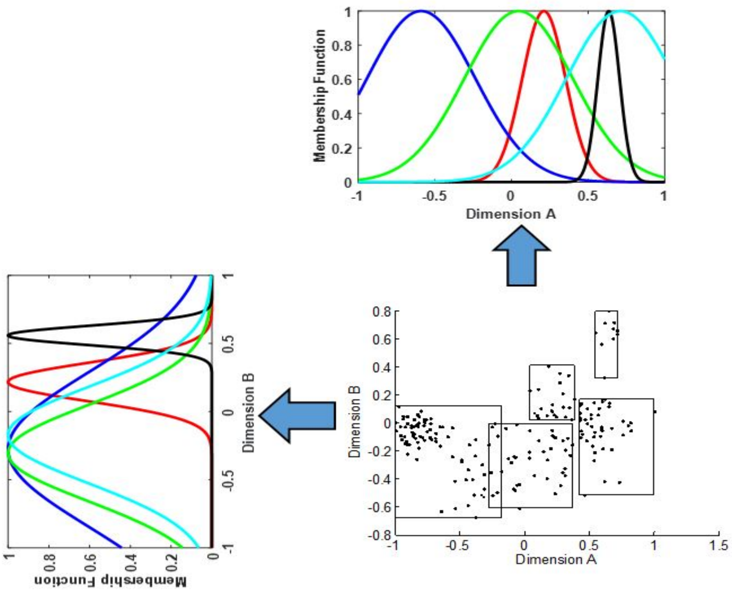


Fig. 3. Initial fuzzy model creation.

The granulated input space determines the number of linguistic rules extracted from the input raw data as well as the number of FSs on the universe of discourse of each input variable. In this approach, the relationship between an information granule in multi-dimension and a fuzzy rule is one-to-one relationship as shown in Fig. 3. Geometrically, one information granule corresponds directly to one fuzzy linguistic rule; the centres of the MFs m_j^i are defined by calculating the average hyper-box limits of each information granule $m_j^i = \frac{1}{2}\left(max\left(A_j^+ - B_j^+\right) - min\left(A_j^- - B_j^-\right)\right)$.

Other parameters relating to the MFs (spread parameters $\sigma_j^i \in [\sigma_{j1}^i, \sigma_{j2}^i]$) to automatically generate IV-MFs from training data can be defined using three common methods. Common methods are based on histograms, heuristics, and interval type-2 fuzzy C-means (IT2-FCM) [39, 47]. The histogram method uses a suitable parameterised function chosen to model the smoothed histograms of sample data. The heuristic method simply generates the IV-MF using heuristics type-1 fuzzy membership function (T1-MF) and a scaling factor. The IT2-FCM method is the derived formulas of the IV-s similar to the well-known Fuzzy C-means clustering (FCM) algorithm. Compared to other two methods, the computational load of the heuristic method is much lower as it simply uses an appropriate type-1 fuzzy membership function (e.g., Gaussian membership function) to represent the distribution of the pattern data [39, 47]. Therefore, in this study, the heuristic method is used in which the parameters of the type-1 fuzzy membership (i.e., centre and width) are determined from the granulated data as shown in Fig. 3. The location of each Gaussian function's centre across each dimension is calculated by $\mathrm{m}_\mathrm{j}^\mathrm{i} = \frac{1}{2}\left(\max\left(\mathrm{A}_\mathrm{j}^+ - \mathrm{B}_\mathrm{j}^+\right) - \min\left(\mathrm{A}_\mathrm{j}^- - \mathrm{B}_\mathrm{j}^-\right)\right)$ and its corresponding width parameter (standard deviation) can be estimated by $\sigma_j^i = \frac{d_{max}}{\sqrt{2M}}$ which is fully described in [36], where M is the number of centres and d_{max} is the maximum distance between any two selected centres. Once T1-MF that is suitable from the granulated data of a given pattern set, the parameters of LMF and UMF of the IV-MF can be obtained by scaling the T1-MFs by a factor K_r between 0 and 1. The factor K_r controls the interval between the LMF and UMF of the FOU in which the width parameters vary as follows: $\sigma_{j1}^i = \frac{1}{K_r}\sigma_j^i$ and $\sigma_{j2}^i = K_r\sigma_j^i$ where $K_r \in [0.3, 1]$ as in [39, 47]. In the heuristic method the scaling factor is constrained in the range $K_r \in [0.3, 1]$ this because the smaller the K_r, the larger the FOU, which implies the greater uncertainty in the IV-MFs. The FOU adds an extra degree of freedom in the IV-MF to account for uncertainty.

2.3. Parametric Optimisation of the IV-RBF-NF Model

Once the initial structure of IV-RBF-NF model is defined from the iterative information granulation algorithm and its initial parameters are estimated, there are several learning algorithms that can be employed to parametrically optimise such a system based on evolutionally algorithms [48] or gradient descent theory [40]. Probably the most prevalent algorithm is the gradient descent based optimisation of this neural-fuzzy modelling structure using the back-propagation algorithm (BP) of the error, introduced by Werbos [49] and further enhanced by Rumelhart et al. [50]. The BP is a gradient descent based algorithm in the weighted-space of a cost (objective) function normally equivalent to the mean square error (MSE). The final form of the BP algorithm is not only simple but also efficient in terms of the number calculations required to update it. Optimising the IV-RBF-NF model requires the determination of the initial IV-RBF parameters (centres and widths) and the output weights, which is a critical step towards achieving high accuracy performance. It also requires a careful selection of the learning rate, for instance a large learning rate may cause the problem of oscillation. To overcome this problem, a small learning rate is recommended. However, the BP algorithm is a non-linear optimisation problem, which in general requires more computational effort (slow convergence) and one of its drawbacks is the possibility of leading the objective function to get trapped into a local minimum. Hence compromising the algorithm's performance. One way to overcome this is to introduce a momentum term and a learning rate term to the algorithm. By monitoring the convergence rate of the objective function, the momentum and learning terms can be adapted/modified to help the algorithm to jump out of local minimum when needed. The adaptation mechanism, along with the BP (adaptive BP of the error) will be used in this research work to parametrically optimise the proposed neural-fuzzy structure due its efficiency [40] and to establish the best parametric optimisation performance, a number of systematic simulation runs will be carried to find the best initial IV-RBF parameters and learning rate and momentum factor for BP algorithm.

2.4. Novelty Detection and the Creation of New Fuzzy Rules

When a new batch of points is available to the model, it passes through a novelty detection algorithm before it is fed to the incremental learning process. Novelty detection is the process of identifying the new or partially new or unknown data points that the model is not aware of during training. Novelty detection is one of the fundamental requirements of a good identification system since sometimes the new batch of data contains information about objects that were not known at the time of training the model. Several statistical and neural networks based approaches can be used to estimate whether a test sample comes from the same samples or not [51, 52]. In this research work, a simple novelty detection approach is used based on Euclidean distance. This approach is considered as a distance-based metric where a point is regarded a novelty if its distance to a k-nn neighbour exceeds the predefined threshold [51, 52]. The novelty detection algorithm checks point by point the new data in the batch in terms of their Euclidean distance to the existing data clusters (rules). Thus, the new data are then split into two data sets (namely novel and partially/non-novel data) based on a predefined threshold. The novel data consist of data that belong to a different data space – new input space – as compared to the old/original process data. The partially/non-novel data consist of data that are close to or belong to the input data space of the original data (i.e. mostly covered by the input space of the original data). The two datasets are treated differently by the incremental learning process. The partially/ non-novel data are fed to the existing model and if the model's performance on these data is satisfactory, the existing model is not further tuned/optimised. Otherwise, the existing model is fine-tuned without significantly disturbing the existing structure (constrained optimisation) to improve the performance on the partially/non-novel data. Since the input space of the non-novel/partially data is mainly covered by the original model (by one or more rules), there is no need to generate a new rule but only fine-tune the previously developed (existing) model. The novel data are utilised to generate new rules to cover the input space of the novel data, using the same GrC-IV-RBF modelling approach. The new rules are optimised and then merged with the rest of the IV-RBF-NF model rules to form a new IV-RBF-NF model (as shown in Section 2).

The so far presented modelling framework does not remove any rules (following model integration) regardless of their importance. This leads to a growing number of redundant rules after each incremental update. To resolve this issue, and create a sustainable long-term learning system, a rule pruning mechanism is proposed in order to remove the redundant rules, while maintaining good modelling performance.

2.5. Iterative Rule Pruning Mechanism for the IV-RBF-NF Model

The rule-base structure of the IV-RBF-NF model is updated by introducing new rules to accommodate the new data, thus creating a model with larger rule-base. As a result of the incremental updating process, the updated model often contains redundant rules. In addition, the iterative rule-base growth after each model update contradicts the main requirement of long-term learning to allow a sustainable model structure. In this light, an iterative rule-base pruning approach is used to minimise/reduce the number of fuzzy sets in the universe of discourse of each input variable and eliminate possible redundant rules after each incremental updating routine. However, to the best of the authors' knowledge, no rule-pruning methods have been proposed in the field of type-2 fuzzy systems. The research to date has tended to focus to type-1 fuzzy logic systems, therefore a new rule-pruning method as an extension to the rule-pruning mechanism for type-1 fuzzy logic system [53] is introduced in this study. The proposed iterative rule-pruning mechanism can be achieved via a four-step procedure, which includes: (1) removing redundant fuzzy sets, (2) merging similar fuzzy sets, (3) removing redundant fuzzy rules, and (4) merging similar fuzzy rules. These four operations are controlled by thresholds $Th_1 - Th_4$. A detailed description of each step of the iterative rule-pruning algorithm is provided below:

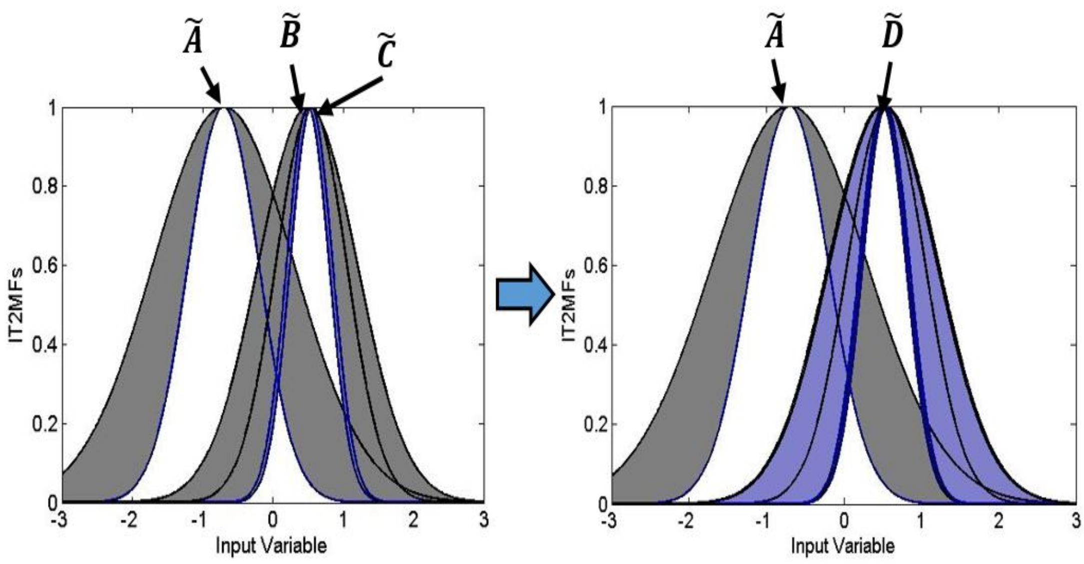
(1) Merging Similar Fuzzy Sets: According to [54], when the rule-base is acquired from process data, it may consist of redundant/superfluous information in the form of similarity between fuzzy sets. A rule-based system with many similar fuzzy sets becomes superfluous, unnecessarily complex and computational expensive [55]. Since the linguistic interpretability of such a model lies in the idea of assigning qualitatively meaningful variables to fuzzy sets. However, it is difficult to assign qualitatively meaningful linguistic variables to highly similar

fuzzy sets. Similarity between FSs can be defined as the degree to which the FSs are equal. For instance, Fig. 4(a) depicts three interval-valued fuzzy sets $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$. $\tilde{A}$ and $\tilde{C}$ are distinguishable (high degree of overlap), while $\tilde{B}$ and $\tilde{C}$ are indistinguishable (high degree of overlap). To measure the degree of overlap between two fuzzy sets, a similarity measure is generally used. Although a quite extensive research has been carried out in the area of type-1 fuzzy sets similarity measures [56, 57], only a few number of similarity measures for T2-FSs have appeared to date [58]. For two IV-FSs $\tilde{A}$ and $\tilde{B}$, calculation of their similarity degree $S(\tilde{A},\tilde{B})$ is much more complex than of their type-1 fuzzy sets counterparts, particularly for those with primary Gaussian MFs. In this study, the Jaccard's similarity measure [58], which is an extension of Jaccard's similarity measure for T1-FSs is used to measure the similarity between two type-2 FSs [58]. The use of Jaccard's similarity measure is motivated by: 1) the value $f(\tilde{A}\cap\tilde{B})/f(\tilde{A}\cup\tilde{B})$ can be computed directly instead of $f(\tilde{A}\cap\tilde{B}')/f(\tilde{A}\cup\tilde{B}')$ without having to align $\tilde{A}$ and $\tilde{B}$ and compute their centroids as in other existing T2-FSs similarity measures [58], which reduced computational cost; 2) the Jaccard's similarity measure satisfies reflexivity, symmetry, transitivity, and overlapping properties [58].The Jaccard's similarity measure between $\tilde{A}$ and $\tilde{B}$ is defined as $S_J(\tilde{A},\tilde{B})=\frac{f(\tilde{A}\cap\tilde{B})}{f(\tilde{A}\cup\tilde{B})}$, where f is a function satisfying $f(\tilde{A}\cup\tilde{B})=f(\tilde{A})+f(\tilde{B})$ for disjoint $\tilde{A}$ and $\tilde{B}$. For simplicity, the function f is chosen as the cardinality (Q). Then the Jaccard's similarity measure can be written as $S_J(\tilde{A},\tilde{B})=\frac{Q(\tilde{A}\cap\tilde{B})}{Q(\tilde{A}\cup\tilde{B})}$ and further as

$$S_J(\tilde{A},\tilde{B})=\frac{\int_X \min\left(\bar{\mu}_{\tilde{A}}(x),\bar{\mu}_{\tilde{B}}(x)\right)dx+\int_X \min\left(\underline{\mu}_{\tilde{A}}(x),\underline{\mu}_{\tilde{B}}(x)\right)dx}{\int_X \max\left(\bar{\mu}_{\tilde{A}}(x),\bar{\mu}_{\tilde{B}}(x)\right)dx+\int_X \max\left(\underline{\mu}_{\tilde{A}}(x),\underline{\mu}_{\tilde{B}}(x)\right)dx} \quad (9)$$

The similarity measure can be used to quantify/estimate the degree of similarity between IV-FSs in the rule base. If the similarity value $S_J(\tilde{A},\tilde{B})$ is larger than a predefined threshold $S_{J_{Th}}$ then $\tilde{A}$ and $\tilde{B}$ are considered as being highly overlapped fuzzy sets. Therefore, $\tilde{A}$ and $\tilde{B}$ can be merged to form a new FS that is representative of the merged FSs. The choice of a suitable threshold $S_{J_{Th}}$ in the process of merging two similar fuzzy sets can be conducted via a heuristic method and it an application dependent, where $S_{J_{Th}}\in[0,1]$, and this range represent the degree of similarity between two fuzzy sets. Where 0 represents no similarity and 1 represents 100% similarity. The lower the value of the threshold, the more fuzzy sets are merged. Thus, the new rules after the merging process will retain most of the redundant information resulting in generating a more complex fuzzy inference system (FIS) structure. On the contrary, the higher the value of the threshold, the less fuzzy sets are merged. Thus, the new rules after the merging process will not retain most of the redundant information resulting in a less complex fuzzy inference system (FIS) structure. To illustrate the concept,

Table 1 shows the similarity matrix representation for the IV-FSs in Fig. 4(a) and calculated by Eq. 9. A threshold Th_1 for merging similar FSs is then defined, where $Th_1 \in [0,1]$. If $S_J(\widetilde{A}, \widetilde{B}) > Th_1$, i.e., the FSs $\widetilde{B}$ and $\tilde{C}$ are highly overlapped, then these two FSs should be merged into one new FSs. With $Th_1 = 0.75$ and from Table 1, $S_J(\tilde{B}, \tilde{C}) > Th_1$, then the FSs $\tilde{B}$ and $\tilde{C}$ are merged to form a new FS $\widetilde{D}$ without losing any information from both $\tilde{B}$ and $\tilde{C}$. The resulting FSs in the same universe of discourse after the merging process is shown in Fig. 4(b). In general, there are three possible methods for merging highly overlapped fuzzy sets: (1) replace $\widetilde{B}$ by C; (2) replace $\tilde{C}$ by $\tilde{B}$ (3) replace both $\widetilde{B}$ and $\tilde{C}$ by a new fuzzy set $\widetilde{D}$. In this study, the third method is used, where the newly merged IV-FS ($\widetilde{D}$) is formed by the combination of the FOUs of both $\widetilde{B}$ and $\tilde{C}$ in order not to lose any information from the merging process.



(a) (b)

Fig. 4. Example of three IV-FSs: (a) IV-FSs before the merging process. (b) Resulting IV-FSs after the merging process.

Table 1. Similarity matrix for the three IV-FSs in Fig. 4 (a) when the Jaccard similarity matrix is used.

IV-FS	$\widetilde{A}$	$\widetilde{B}$	$\widetilde{C}$
$\widetilde{A}$	1.0000	0.2277	0.1791
$\widetilde{B}$	0.2277	1.0000	0.7645
$\widetilde{C}$	0.1791	0.7645	1.0000

(2) Removing Redundant Fuzzy Sets: A fuzzy set in the antecedent part of linguistic rule is said to be redundant if it has a MF $\mu_{\tilde{A}}(x) \approx 1, \forall x \in X$, it is similar to the universal set $U(\mu_U(x) = 1)$ and can be removed [54]. The similarity of a FS $\widetilde{A}$ to the universal set U is calculated by $S_J(\tilde{A}, U)$. If the $S_J(\tilde{A}, U)$ is greater than a predefined threshold Th_2, then this FS is considered as a redundant FS. As a result, the corresponding FS should be removed.

(3) Merging Similar Fuzzy Rules: Two fuzzy rules are said to be similar enough for merging if only the antecedent of the rules are equal and the consequents do not [53]. Two fuzzy rules with different consequents but very similar antecedent parts usually indicates conflicting rules [53]. Therefore, conflicting rules are either merged together to form a new rule or one of them is removed. To evaluate the similarity degree between two linguistic fuzzy rules, the similarity measure of every FS pair has to be calculated [53]. For the ith fuzzy rule $Rule_i$, the corresponding IV-FSs are $\tilde{A}_1^i, \tilde{A}_j^i, \dots, \tilde{A}_n^i$. In similar fashion, the corresponding antecedent parts of the kth fuzzy rule $Rule_k$, the are $\tilde{B}_1^k, \tilde{B}_j^k, \dots, \tilde{B}_n^k$. Therefore, the similarity measure can be expressed as follows [59]:

$$\boldsymbol{S_J\left(Rule_j, Rule_k\right) = \prod_{j=1}^{n} S_J\left(\tilde{A}_j^i, \tilde{B}_j^k\right)} \qquad \mathbf{(10)}$$

where $S_J\left(\tilde{A}_j^i, \tilde{B}_j^k\right)$ is the Jaccard similarity measure of two IV-FSs $\tilde{A}_j^i$ and $\tilde{B}_j^k$ and it is defined in Eq. 9. If $S_J(Rule_i, Rule_k)$ is greater than a predefined threshold Th_3, then the FS pairs of these two fuzzy rules are similar. Therefore, these two rules are also considered to be similar and then merged into a new rule $Rule_{new}$. The antecedent and consequent parts of the new rule $Rule_{new}$ are obtained via the merging operation of similar fuzzy sets.

(4) Removing Redundant Fuzzy Rules: If the membership value of an IV-FS is always near zero over its entire universe of discourse, i.e., $\mu_{\tilde{A}}(x) \approx 0, \forall x \in X$, its corresponding rule is considered redundant [54]. Since this redundant rule will almost never be fired, which means its output is always near zero. A threshed Th_4 is also defined to determine whether the rule is redundant or not. If $\mu_{\tilde{A}}(x) < Th_4$, then the corresponding fuzzy rule is deemed redundant. Therefore, the redundant rule should be removed [54]. In general, Th_4 is defined in the range [0, 0.01].

The rule-pruning algorithm depicted as shown above is an iterative algorithmic process where at each iteration, the similarity measure between all pairs of IV-FSs for each input variable is calculated. The pairs of IV-FSs having the highest similarity value $S_J\left(\tilde{A}, \tilde{B}\right) > Th_1$ are merged to form a new IV-FS. Then the rule-base of the IV-RBF-NF model is updated by substituting this new IV-FS for the IV-FS merged to form it. The process of calculating the similarity measure on the updated rule-base structures continues until there are no more IV-FSs for which $S_J\left(\tilde{A}, \tilde{B}\right) \geq Th_1$. Thereafter, the IV-FSs that have similarity $S_J\left(\tilde{A}, \tilde{B}\right) \geq Th_2$ to the universal set U are removed. Finally, the similarity measure between all pairs of linguistic rules for entire rule-base is computed. The pairs of fuzzy rules having the highest similarity value $S_J\left(\tilde{A}, \tilde{B}\right) > Th_3$ are merged to form a new IV-FS. Repeat the process of merging similar fuzzy rules until there are no more rules for which $S_J\left(\tilde{A}, \tilde{B}\right) \geq Th_3$. Then the redundant rules are removed based a predefined threshold Th_4. The rule-pruning algorithm is summarised as follows:

Algorithm 1 Iterative rule pruning algorithm

Input: Given a linguistic rule-base $R = \{Rule_i\}_{i=1}^{M}$, where $Rule_i$ is the ith rule, choose thresholds $Th_1 - Th_4$

$$Rule_i: IF\ x_1\ is\ \tilde{A}_1^i\ and, \dots, and\ x_n\ is\ \tilde{A}_n^i, THEN\ y\ is\ \tilde{B}^i$$

Step 1: Calculate the similarity matrix in R, $S_J\left(\tilde{A}_j^i, \tilde{B}_j^k\right)$, $i = 1, \dots, M$, $j = 1, \dots, n$, $k = 1, \dots, M$. Select two most similar IV-FSs $S_{Jmax}\left(\tilde{A}_j^i, \tilde{B}_j^k\right) = max_{i \neq k}\left\{S_{Jmax}\left(\tilde{A}_j^i, \tilde{B}_j^k\right)\right\}$.

Step 2: If $S_{Jmax}\left(\tilde{A}_j^i, \tilde{B}_j^k\right) \geq Th_1$, then merge the two IV-FS to form a new fuzzy set.

Continue until: no more IV-FSs have similarity measure such that $S_J\left(\tilde{A}_j^i, \tilde{B}_j^k\right) \geq Th_1, i \neq k$.

Step 3: Calculate similarity measure of a FS $\widetilde{A}_j^i$ to the universal set $S_J\left(\tilde{A}_j^i, U\right)$. If the similarity value $S_J\left(\tilde{A}_j^i, U\right) \geq Th_2$, then the $\tilde{A}_j^i$ is considered to be a redundant fuzzy set and should be removed from the antecedent of $Rule_i$.

Step 4: Calculate the similarity matrix between the rules in R,

$$S_J(Rule_i, Rule_k) = \prod_{j=1}^{n} S_J\left(\tilde{A}_j^i, \tilde{B}_j^k\right)$$

where $S_J\left(\tilde{A}_j^i, \tilde{B}_j^k\right)$ is the Jaccard similarity measure of two IV-FSs $\tilde{A}_j^i$ and $\tilde{B}_j^k$. If $S_J(Rule_i, Rule_k) \geq Th_3$, then the IV-FS pairs of these two fuzzy rules are similar and merged into a new rule $Rule_{new}$.

Continue until: no more rules have similarity measure such that $S_J(Rule_i, Rule_k) \geq Th_3, i \neq k$.

Step 5: Remove the redundant fuzzy rules if $\mu_{\tilde{A}_j^i}(x) \leq Th_4, \forall x \in X$.

Output: Pruned IV-RBF-NF model

The obtained rule-based system is improved in its structure, including the variation of the fuzzy sets and fuzzy rules, considering the simplicity and interpretability issues. Pruning the redundant rules results in distinguishable fuzzy sets and thus simplified fuzzy rules. It also controls the growth in the number of rules over time.

2.6. Accuracy Improvement via Constrained Optimisation

After iterative rule-base pruning process is applied to the incrementally updated model, the obtained model is structurally simpler and more interpretable. However, the model is less accurate than the originally created model. Accuracy and interpretability are two contradictory and conflicting modelling requirements, as improving interpretability of rule-based systems – pruning the redundant rules – generally degrades the overall performance of the model and vice versa. To further improve the accuracy of the pruned model, a parametric optimisation method is used based on the adaptive BP algorithm. A good trade-off between model interpretability and accuracy requires imposing constraints on parameter adjustment.

3. Simulation Results

In order to demonstrate the functionality of the proposed long-term learning framework, simulation experiments have been carried out using well-known benchmark functions and real industrial data obtained from the UCI Machine Learning Data Repository. The first three simulation examples validate the performance of the proposed structure via a non-linear function identification for three different functions; in each example the complexity of the function has been increased. The fourth simulation example uses a multi-modal complex function to confirm the effectiveness of the proposed structure in case where more frequent model updates are required (iterative incremental learning). In this simulation example, two scenarios are created: in the first scenario weak/relaxed constraints are imposed to the framework, therefore the proposed structure is geared towards accuracy, whereas in the second scenario strong constraints are imposed. Thus, the proposed framework is geared towards interpretability. Finally, the proposed framework is applied to the prediction of compressive of high-performance concrete in the concrete construction industry. Since there is no existing work on batch long-term learning in the field of type-2 neuro-fuzzy systems. This is the first attempt to develop such a framework and there are no existing long-term frameworks to compare against. However, despite the absence of directly relevant work to compare against, to confirm the effectiveness of the proposed modelling framework alone (without the long-term learning) a comparison to existing popular neuro-fuzzy systems is included. Therefore, for performing the comparison, an alternative neuro-fuzzy system is trained on the batch data, without the incremental learning functionality (as this does not currently exist in the literature). This 'replicates' what one would do when more data become available on the same case-study e.g. train again, from scratch, using the whole combined dataset, which is common practice in industrial case studies. To maintain a level of fairness in this comparison, the number of rules between the different models is kept the same. Therefore, what this comparison has shown is that the incremental learning functionality performs well, as opposed to the non-incremental one, one similar class (type) of neural-based models. This comparison also provides useful conclusions between the type-1 fuzzy logic system and its type-2 fuzzy logic system counterpart in terms of their general structure and performance.

3.1. Example 1: Uni-modal Non-linear System Identification

The system to be identified is represented by the following equation:

$$f(x_1, x_2) = (2x_1^2 + x_2^2) \tag{11}$$

The non-linear static system is taken from [60]. One hundred data points were generated randomly from $-0.5 \leq x_1, x_2 \leq 0.5$ and the corresponding output data were obtained from Eq. 11. The data set has been randomly divided into 75 (75%) data points to train the model and 25 (25%) data points to test the prediction performance of the final model. The training raw data are granulated into 5 information granules (optimal number of information granules in this case) via the iterative data granulation process as shown in Section 2. The extracted optimal information granules are then mapped into linguistic type-2 fuzzy logic rules to elicit the initial structure of the IV-FLS rule-base. Once the initial structure of the IV-FLS rule-base (5 fuzzy rules) is obtained, the initial IV-RBF-NF structure is optimised via the adaptive-BP algorithm. After structure identification and parametric optimisation, a 5-rule-based model was produced. For comparison, a 5-rule ANFIS model was trained on the same data. Table 2 shows the RMSE, VAF %, and R^2 for ANFIS and IV-RBF-NF system during the training phase. It can be concluded that in general the IV-RBF-NF model outperforms the ANFIS model. As it was expected, the interval-valued fuzzy sets have the ability to capture more uncertainty, hence outperforms the type-1 fuzzy sets.

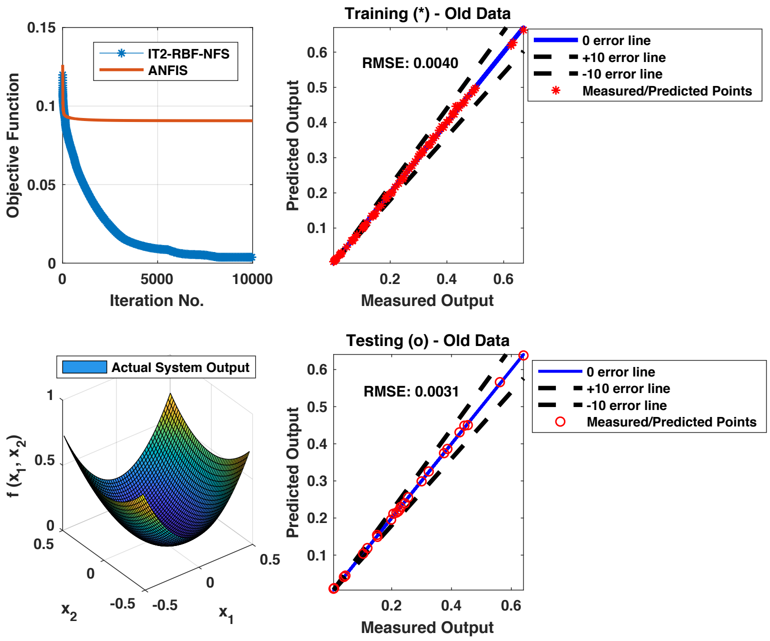


Fig. 5. Regression line for non-linear system approximation for training and testing.

To test the incremental learning function, a hundred random data points in the range between $-0.65 \leq x_1, x_2 \leq 0.65$ (i.e. generate new data that are not covered by the old/original data) was generated. When the new data are available to the developed model, they pass through the novelty detection algorithm before they are fed to the incremental learning process. The performance of the model is tested on the new data (both partially novel and totally new data), as it was expected without any further tuning, the performance of the model on the new data is far worse than its performance on the partially novel data since the input space of the novel data is entirely new. The performance indices based on RMSE for 5-rule IV-RBF-NF model on the partially novel and totally novel are 0.01 and 0.08 respectively.

It is clear that the model does not perform well on the novel data. This is because the model has not seen the novel data during the initial training of the model. Subsequently, the novel data set is used to generate new rules to cover the input space of the novel data set, using the same GrC-IV-RBF-NF modelling framework. The new data set produced 4 rules are combined with the existing 5 rules to construct a new IV-FLS with 9 rules. Following the iterative rule-pruning mechanism, two rules were merged based on merging thresholds $S_{J_{Th}}$ of 0.8 in this case. The simplified 8 FLS rules are then fined-tuned to improve its accuracy. The performance of the updated model is tested on the old and original data as well as the new data (both training and testing). The incremental updating algorithmic process provides a reliable model updating procedure that results in an open structure (i.e., dynamically expandable structure) without neglecting any previously gained knowledge. Since there is no an incremental version for ANFIS, for the purpose of comparison the ANFIS with same number of rules as the IV-RBF-NF model was trained on the initial data and on the new data with the same number of rules after the incremental update. Therefore, a 8-rule ANFIS model was trained on the old and original data and the results are reported in Table 2. It is shown from Table 2 that the proposed framework outperforms the ANFIS model and it is able to model/learn from an initial process data and incrementally updates its structure when needed and at the same time improves its structure by removing the redundant rules. The performance of the updated model on the original data set is maintained, and its performance on the new data after the incremental update is comparable to the original performance.

Table 2. Performance of the original model and updated model for non-linear uni-modal function approximation in Example 1

Model		Number of Rules	Performance Indices					
			Training Data			Testing Data		
			RMSE	VAF (%)	R^2	RMSE	VAF (%)	R^2
ANFIS	Initial Model	5	0.0907	87.16	0.9122	0.2012	86.20	0.9013
	Updated Model	8	0.1389	83.09	0.9084	0.1900	82.53	0.8631
IV-RBF-NF Model	Initial Model	5	0.0040	98.99	0.9899	0.0031	96.88	0.9770
	Updated Model	8	0.0089	96.88	0.9770	0.0093	96.16	0.9652

3.2. Example 2: Multi-modal Non-linear System Identification

This example employs the proposed structure to model a well-known complex multi-modal benchmark function [61]. The multi-modal function is generated from the following equation:

$$f(x_1, x_2) = \sin\left(\pi\sqrt{(x_1^2 + x_2^2)}\right) \tag{12}$$

One thousand data points were generated randomly from $-0.5 \leq x_1, x_2 \leq 0.5$ and the corresponding output data were obtained from Eq. 12. As in the previous case, the data set has been divided into 750 (75%) data points to train the model and 250 (25%) data points to test its prediction performance. The training raw data are granulated into 8 information granules (optimal number of information granules) via the iterative information granulation approach. The granulated data are then mapped into linguistic type-2 fuzzy rules to elicit the initial structure of fuzzy rule-base. After the initial structure of IV-FLS (13 fuzzy rules) is obtained, the initial IV-RBF-NF structure is optimised via the adaptive BP approach. After structure identification and parametric optimisation, a 13-rule model was produced and the rule-base. For comparison, Fig. 6 and Table 3 show the simulation results and RMSE, VAF %, and R^2 for ANFIS and IV-RBF-NF system. It can be concluded that in general the IV-RBF-NF outperforms the ANFIS model.

To test the generalisation ability of the proposed incremental learning structure in a more complex system identification problem, a synthetic new data set that contains both novel and partially novel data in the range $-0.65 \leq x_1, x_2 \leq 0.65$ was generated. After the new data are passed though the incremental update process, 11 new rules are generated from the data granulation process. The updated model (20 rule-based system) is then pruned and fine-tuned to construct the final updated model (20 rules), when the rule pruning thresholds $S_{J_{Th}}$ were set to 0.80. For the purpose of comparison, Table 3 also shows the performance of the original model (rule-based system) and the updated model (20 rule-based system). As it can be seen, the incremental learning structure has an adaptive behaviour by incrementally updating itself to accommodate the unseen input data set. The incrementally updated model is able to retain a good prediction performance without ignoring any previously gained knowledge.

Table 3. Performance of the original model and updated model for non-linear function approximation in Example 2

Model		Number of Rules	Performance Indices					
			Training Data			Testing Data		
			RMSE	VAF (%)	R^2	RMSE	VAF (%)	R^2
ANFIS	Initial Model	13	0.3218	65.32	0.6641	0.4490	64.85	0.6512
	Updated Model	20	0.5923	64.10	0.6478	0.8123	63.24	0.6357
IV-RBF-NF Model	Initial Model	13	0.0090	97.73	0.9811	0.0079	98.88	0.9813
	Updated Model	20	0.0079	98.78	0.9807	0.0080	98.95	0.9901

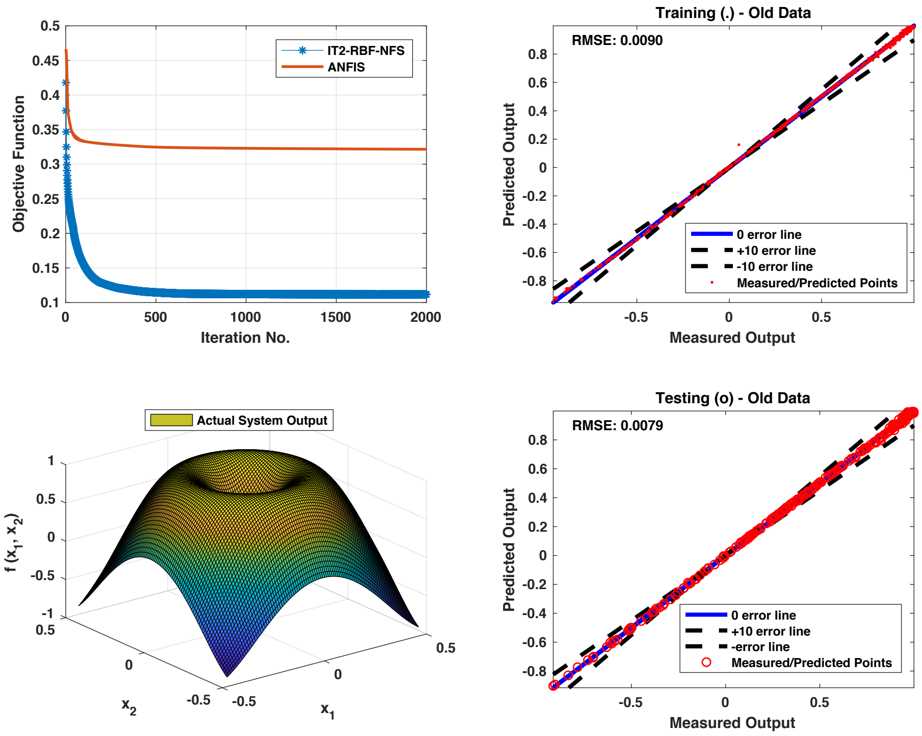


Fig. 6. Regression line for multi-modal function approximation for training and testing.

3.3. Example 3: Multi-modal Butterfly System Identification

In this example, double-input and single-output static complex multi-modal butterfly function is chosen to be a target system for the proposed incremental learning strategy. The function is taken from [62] and represented as

$$f(x_1, x_2) = (x_1^2 - x_2^2)\frac{\sin(x_1^2 + x_2^2)}{(x_1^2 + x_2^2)}, \qquad -0.5 \le x_1, x_2 \le 0.5 \tag{13}$$

For which 1000 data points are generated. The same modelling procedures were adopted. Fig. 7 shows the results of the obtained model for the butterfly function approximation. In this simulation, the same incremental learning process was adopted by generating a new data set in the range of $-0.65 \le x_1, x_2 \le 0.65$ to make the function more complex. Table 4 shows the performance of the original model (15 rule-based system) on the new data and the updated model (24 rule-based system) respectively. From the performance indices reported in Table 4, it can be clearly seen that, the incremental learning structure has a dynamic behaviour by updating itself to accommodate the unseen input data for complex multi-modal function. The obtained updated model is able to maintain a good prediction performance (as compared to the original model) without ignoring any previously gained knowledge.

Table 4. Performance of the original model and updated model for non-linear function approximation in Example 3

Model		Number of Rules	Performance Indices					
			Training Data			Testing Data		
			RMSE	VAF (%)	R^2	RMSE	VAF (%)	R^2
ANFIS	Initial Model	15	0.2973	78.45	0.7789	0.3241	76.73	0.7429
	Updated Model	24	0.2318	79.05	0.7871	0.6910	73.49	0.7316
IV-RBF-NF Model	Initial Model	15	0.0529	96.41	0.9755	0.0529	96.43	0.9786
	Updated Model	24	0.0463	97.03	0.9788	0.0473	96.89	0.9856

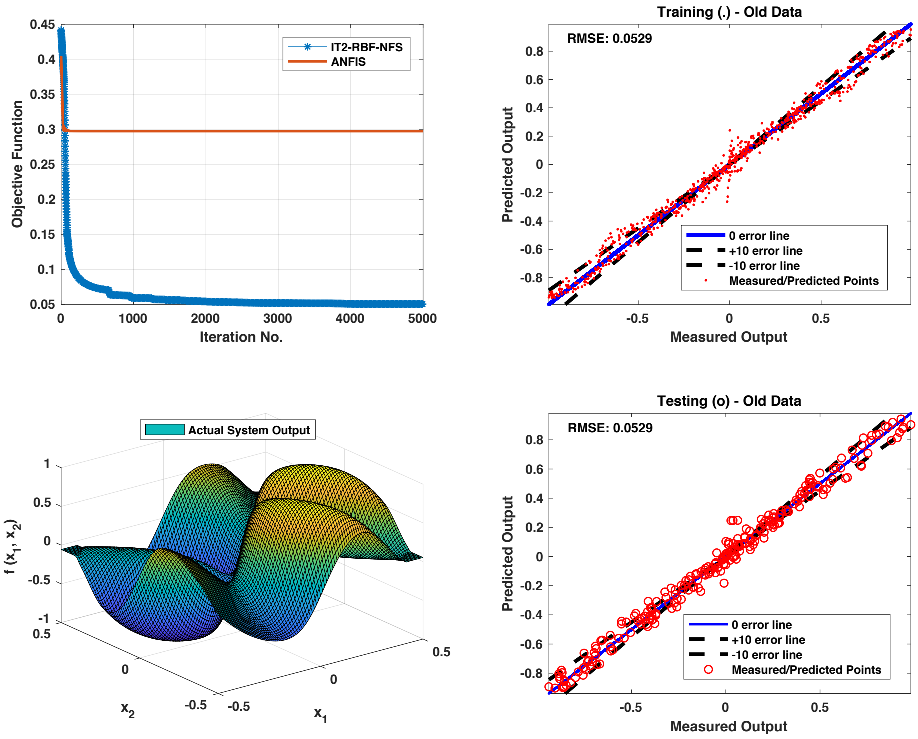


Fig. 7. Regression line for multi-modal butterfly function approximation for training and testing.

3.4. Example 4: Iterative/Frequent Incremental Updates Performance

In this example, it is shown how the proposed incremental learning structure performs in consecutive incremental updates. A hypothetical case study of 10 steps (2% incremental step in this study) for the non-linear complex function in Example 2 has been created. In the incremental update for 10 consecutive steps, the modelling structure goes through a series of iterative rule-base growing and pruning steps, to accommodate in each step the new data while maintain satisfactory performance. In the following case relating to iterative incremental updates where weak/relaxed constraints are imposed on the iterative rule pruning algorithm (0.6 in this case) and constrained parametric optimisation (0.6 in this case). Consequently, a large number of redundant rules is generated after the final incremental step. In other words, the long-term learning framework is geared towards accuracy. The performance of the iterative incremental updates is summarised in Table 5. As shown, the number of rules grows but not by a substantial increment due the pruning mechanism applied to the incrementally updated architecture at each step. The incremental learning architecture is able to maintain a good performance without ignoring any previously gained knowledge. From Table 5, it is concluded that the proposed incremental structure produces good accuracy after 10 incremental steps. Although the incrementally updated model after 10 steps with nearly three times rule base size (38 rule-based system) of that of the core model (13 rule-based system), the updated structure is able to learn more accurately over time. In summary, the results show that the incremental learning structure achieves good balance between model accuracy (by inserting new rules) and complexity (by removing redundant rules), while yielding sustainable and reliable incremental update architecture that can be adapted incrementally in a lifelong learning mode (iterative rule-base growing and pruning).

Table 5. Performance of the updated model during 10 iterative incremental learning updates (with weak/relaxed constraints) for multi-modal function approximation in Example 2

Step No.	Number of Rules		Performance Index				No. of Added Rules
			Training RMSE		Testing RMSE		
	Before the update	After the update	Before the update	After the update	Before the update	After the update	
1	13 (Core Model)	15	0.0090 (Core Model)	0.0099	0.0079 (Core Model)	0.0093	3
2	15	17	0.0099	0.0063	0.0093	0.0062	2
3	17	19	0.0063	0.0058	0.0062	0.0058	2
4	19	21	0.0058	0.0083	0.0058	0.0081	2
5	21	24	0.0083	0.0068	0.0081	0.0061	3
6	24	27	0.0068	0.0108	0.0061	0.0107	3
7	27	30	0.0108	0.0061	0.0107	0.0065	3
8	30	32	0.0061	0.0055	0.0065	0.0054	2
9	32	35	0.0055	0.0072	0.0054	0.0057	3
10	35	38 (Final updated Model)	0.0072	0.0053 (Final updated Model)	0.0057	0.0053 (Final updated Model)	3

In the following case, strong constraints are imposed on the iterative rule-pruning algorithm (0.9 in this case) and constrained parametric optimisation (0.9 in this case). On one hand, imposing strong constraints deteriorates the performance of the long-term learning over time (i.e., after a number of incremental steps), and on the other hand it preserves a low number of rules over time with good level of interpretability. Table 6 summarises the performance index based on RMSE for the 10 incremental steps. It is evident that as a result the strong constraints after 10 steps, the rule-base size of the updated model (18 rule-based system) is not big as compared to the rule-base of the original model (13 rule-based system), but the final performance of the updated structure is not as good as the final updated model in case where weak/relaxed constraints are imposed.

Table 6. Performance of the updated model during 10 iterative incremental learning updates (with strong constraints) for multi-modal function approximation in Example 2

Step No.	Number of Rules		Performance Index Training RMSE		Performance Index Testing RMSE		No. of Added Rules
	Before the update	After the update	Before the update	After the update	Before the update	After the update	
1	13 (Core Model)	14	0.0090 (Core Model)	0.0114	0.0079 (Core Model)	0.0125	1
2	14	14	0.0114	0.0147	0.0125	0.0180	0
3	14	14	0.0147	0.0201	0.0180	0.0234	0
4	14	15	0.0201	0.0289	0.0234	0.0270	1
5	15	16	0.0289	0.0356	0.0270	0.0319	1
6	16	16	0.0356	0.0468	0.0319	0.0491	0
7	16	17	0.0468	0.0542	0.0491	0.0580	1
8	17	18	0.0542	0.0604	0.0530	0.0510	1
9	18	18	0.0604	0.0693	0.0510	0.6810	0
10	18	18 (Final updated Model)	0.0691	0.0720 (Final updated Model)	0.6810	0.0786 (Final updated Model)	0

3.5. Example 5: Compressive Strength of Concrete Data Prediction

The objective of this simulation example is to employ the proposed framework in a multidimensional highly nonlinear data in civil engineering [63]. The proposed framework is applied to the prediction of compressive of high-performance concrete in the concrete construction industry. The data used in this example were obtained from the UCI Data Repository and are described in [63]. According to [63], the high-performance concrete is a highly complex material and the effects of the proportions of each variable on the concrete mix are difficult to model. The compressive strength of concrete is a multiple-input single-output (MISO) process and its data set consists of 1030 data vectors of various input features. Eight numeric features namely: Cement ($\mathrm{kg/m^3}$), Fly ash ($\mathrm{kg/m^3}$), Blast furnace slag ($\mathrm{kg/m^3}$), Water ($\mathrm{kg/m^3}$), Superplasticizer ($\mathrm{kg/m^3}$), Coarse aggregate ($\mathrm{kg/m^3}$), Fine aggregate ($\mathrm{kg/m^3}$), and Age of testing (days) are used to predict the compressive strength of concrete. For the purpose of modelling, the data set (1030 instances) has been split as follows:

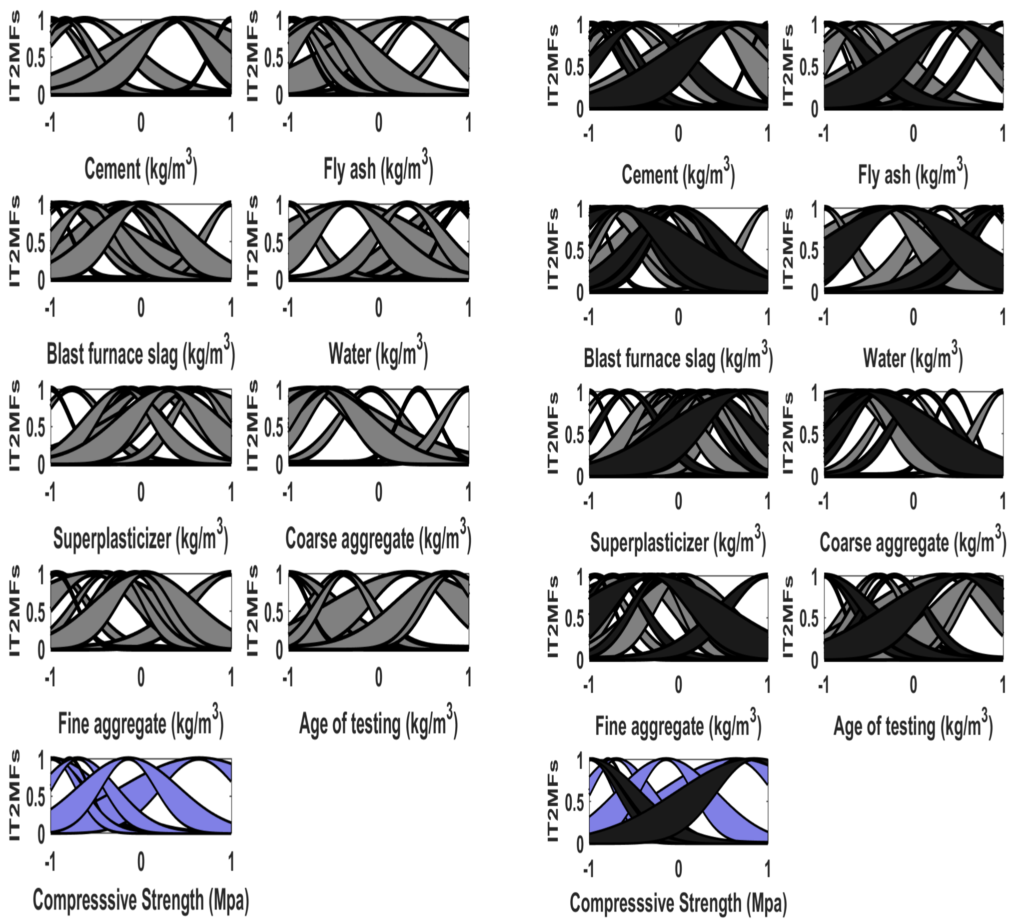
a) Initial data set (721 data points) which has been divided into 541 (75%) data points to train the model and 180 (25%) data points to test the generalisation capability of the trained model; b) New data set (309 data points), which has been used to test incremental learning performance.

After a number of systematic simulations (increased/reduced number of rules) it was established that the modelling performance was acceptable between the range of 4 to 9 rules. The model having 4 rules achieved less accuracy performance compared to the model having 9 rules. However, the IV-RBF-NF model having 9 linguistic rules achieved good accuracy but not a substantial performance improvement compared the model with 8 rules. Hence, the 8-rule model shown in Fig. 8(a) was chosen as the best overall model, in terms of balancing interpretability and good performance. In general, which is also what the literature confirms [26], models with more linguistic rules (hence more parameters to be optimised) often lead to better accuracy due their ability to capture more complex information (with the risk of overtraining). However, this leads to lack of interpretabiltiy and simplicity, while models with fewer linguistic rules and parameters result in simpler models and easier to interpret, albeit with lower performance in terms of accuracy. The rule-base in a linguistic format is shown in Fig. 8(a) can be used to describe the non-linear relationships between the input features and compressive strength. In order to obtain the linguistic interpretation to the rules in Fig. 8(a), expert knowledge (civil engineer) can be used to set the crisp limits of each linguistic label (i.e., membership function) within a normalised input-output space [-1,1]. Here is one example out of the resulting 8 rules:

Rule 1: ***IF*** *Cement is 'medium-low'* ***and*** *Fly ash is 'medium-low to medium'* ***and*** *…****and*** *Age of testing 'medium-high to more-or-less high'* ***THEN*** *Compressive strength of concrete is 'medium-high*

The above IF-THEN statement agrees with expert knowledge regarding the influence of the input features to the measured compressive strength of concrete. It is obvious from Fig. 8(a), the type-2 fuzzy sets are visually indeed complex to interpret, using the figures alone. This is because of their obvious geometrical features that simply need a more complex graph to capture. However, one must not undermine that linguistic interpretability in the system still exists, in the form of being able to directly extract language-based statements.

To test the performance of the incremental learning framework, the new data set (309 data points) is divided into two subsets (training and testing). The new training set is then presented to the incremental learning structure, 4 new rules are obtained from the GrC algorithm to cover the new dataset and trained via the same algorithmic procedure as the initial IV-RBF model. Subsequently, the newly generated fuzzy rules, which are shown in a shaded black colour in Fig. 8(b) are combined with the rest of the fuzzy rules. Following the rule-pruning process, two rules are removed based on a pre-defined similarity threshold; the resulting rule-base is further optimised (fine-tuned). The resulting model (10-rule base shown in Fig. 9) is tested for its performance on the initial dataset as well as new dataset (training and testing). The prediction performance achieved by the original model and incrementally updated model for ANFIS and IV-RBF-NF system on training and testing is shown in Table 7. As illustrated in the rule-base plots, the incremental structure is able to maintain a good performance by adding new knowledge in the form of new rules and at the same time removing redundant knowledge. Similar behaviour is observed for the testing data set for old and new data sets. The model is able to predict correctly the new – unseen – input patterns.



(a) (b)

Fig. 8. The rule-base of IV-RBF-NF model constructed by using granular computing and adaptive BP algorithm: (a) 8 fuzzy rules for the initial model; (b) 12 fuzzy rules after the incremental update process and before rule-pruning.

Table 7. Performance of the original model and updated model for compressive strength of concrete data prediction in Example 5

Model		Number of Rules	Performance Indices					
			Training Data			Testing Data		
			RMSE	VAF (%)	R^2	RMSE	VAF (%)	R^2
ANFIS	Initial Model	8	20.50	84.73	0.8023	27.01	78.26	0.7913
	Updated Model	10	25.23	81.90	0.7984	42.35	59.09	0.6046
IV-RBF-NF Model	Initial Model	8	9.15	91.57	0.8756	10.29	89.13	0.8915
	Updated Model	10	9.59	90.61	0.8801	11.28	88.04	0.8614

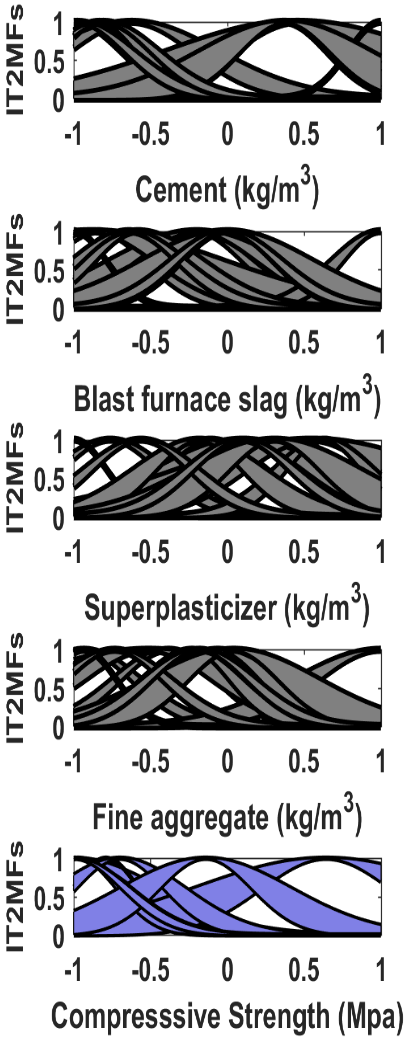


Fig. 9. Incrementally updated rule-base (10 fuzzy rules after interpretability improvement and constrained optimisation).

In general, from the above simulation examples when dealing with long-term learning in a lifelong learning mode there are two conflicting requirements: overall system accuracy and interpretability, as improving the global system accuracy of the final model (after a number of updates) generally degrades overall system interpretability of final model, and vice versa. Hence one challenging problem is how to select the thresholds in the iterative rule-pruning algorithm to remove the redundant rules as a result of the incremental update process. A careful selection of these thresholds determines the number of redundant rules to be removed. Thus, improving the overall interpretability of the updated model. Small thresholds result in a large number of redundant rules to be removed, and then more interpretable fuzzy model but less global model accuracy. In addition, the constraints on the parametric optimisation after the incremental updating algorithm play an important role on the overall system accuracy. Imposing strong constraints leads to less accurate models but a more interpretable fuzzy rule-base.

4. Conclusion

In this article, a new long-term learning framework that is based on type-2 fuzzy systems is presented. The framework is based on interval-valued radial basis function neural fuzzy (IV-RBF-NF) modelling structure. The main advantages of the incremental learning structure is the ability to provide a reliable model updating procedure that results in a dynamically expandable structure without ignoring any previously obtained knowledge. The proposed architecture satisfies the requirements needed for incremental learning as it is able to handle the short and long-term change in the input conditions in a lifelong learning mode by incrementally updating its structure to accommodate the change in the process input data space. The incremental updating algorithm also incorporates a rule-base pruning mechanism to prune the redundant rules without compromising the overall predictive accuracy. The long-term learning framework could be used in industrial processes such as friction stir welding and laser welding process, where for example the process changes and there is a need to update the underlying process model.

In the first three simulation studies, the efficiency of the proposed structure was tested against three case studies, which include a non-linear uni-modal function and two more complex non-linear multi-modal functions as well as real industrial data. In each case study it is demonstrated the generalisation capability of the proposed structure by increasing the complexity of the function. Results show that the ability of the proposed methodology in updating itself to accommodate the unseen input data set and maintain a good performance without significantly disturbing any previously gained knowledge. In the fourth simulation study, the sustainability (consecutive updates) of the proposed incremental methodology was tested against a multi-modal complex function where more frequent model updates are simulated while maintaining good system accuracy. In the final simulation study, the proposed framework was applied to real industrial data obtained from the UCI Data Repository, demonstrating good performance.

The proposed long-term learning framework assumes that the new input data that are presented to the model are valid for learning and cleaned from noise. Therefore, further research could be conducted on testing the robustness of the proposed framework in the presence of uncertainty, in the form of added noise, missing data and outliers.

While the proposed framework proves to work rather well in the presented case-studies, it relies on expert knowledge, and heuristic estimation of various algorithmic parameters and thresholds. It would be useful in the future to investigate the creation of an autonomous learning and updating system that does not rely on human input.

References

[1] M. Chandrasekaran, M. Muralidhar, C. M. Krishna *et al.*, “Application of soft computing techniques in machining performance prediction and optimization: a literature review,” *The International Journal of Advanced Manufacturing Technology,* vol. 46, no. 5-8, pp. 445-464, 2010.

[2] H.-J. Rong, N. Sundararajan, G.-B. Huang *et al.*, “Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction,” *Fuzzy sets and systems,* vol. 157, no. 9, pp. 1260-1275, 2006.

[3] F. M. Pouzols, A. Lendasse, and A. B. Barros, “Autoregressive time series prediction by means of fuzzy inference systems using nonparametric residual variance estimation,” *Fuzzy Sets and Systems,* vol. 161, no. 4, pp. 471-497, 2010.

[4] J. S. Albus, “A new approach to manipulator control: The cerebellar model articulation controller (CMAC),” *Journal of Dynamic Systems, Measurement, and Control,* vol. 97, no. 3, pp. 220-227, 1975.

[5] R. J. Oentaryo, M. J. Er, S. Linn *et al.*, “Online probabilistic learning for fuzzy inference system,” *Expert Systems with Applications,* vol. 41, no. 11, pp. 5082-5096, 2014.

[6] M. Pratama, M. J. Er, X. Li *et al.*, “Data driven modeling based on dynamic parsimonious fuzzy neural network,” *Neurocomputing,* vol. 110, pp. 18-28, 2013.

[7] N. K. Kasabov, and Q. Song, “DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction,” *Fuzzy Systems, IEEE Transactions on,* vol. 10, no. 2, pp. 144-154, 2002.

[8] C.-J. Lin, and Y.-J. Xu, “A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications,” *Fuzzy Sets and Systems,* vol. 157, no. 8, pp. 1036-1056, 2006.

[9] G. Panoutsos, and M. Mahfouf, "An incremental learning structure using granular computing and model fusion with application to materials processing," *Intelligent Techniques and Tools for Novel System Architectures*, pp. 139-153: Springer, 2008.

[10] X. Deng, and X. Wang, “Incremental learning of dynamic fuzzy neural networks for accurate system modeling,” *Fuzzy Sets and Systems,* vol. 160, no. 7, pp. 972-987, 2009.

[11] D. Saad, *On-line learning in neural networks*: Cambridge University Press, 2009.

[12] N. K. Kasabov, “Evolving connectionist systems for adaptive learning and knowledge discovery: Trends and directions,” *Knowledge-Based Systems,* vol. 80, pp. 24-33, 2015.

[13] K. Ganapathy, V. Vaidehi, and J. B. Chandrasekar, “Optimum steepest descent higher level learning radial basis function network,” *Expert Systems with Applications,* vol. 42, no. 21, pp. 8064-8077, 2015.
[14] N. Kasabov, “Evolving Connectionist and Fuzzy Connectionist Systems–Theory and Applications for Adaptive On-line Intelligent Systems,” *Neuro-Fuzzy Techniques for Intelligent Information Processing, Kasabov N. and Kozma R.,(Eds.), Physica Verlag*, 1999.
[15] F. H. Hamker, “Life-long learning cell structures—continuously learning without catastrophic interference,” *Neural Networks,* vol. 14, no. 4, pp. 551-573, 2001.
[16] R. A. Aliev, B. Guirimov, B. Fazlollahi *et al.*, “Evolutionary algorithm-based learning of fuzzy neural networks. Part 2: Recurrent fuzzy neural networks,” *Fuzzy Sets and Systems,* vol. 160, no. 17, pp. 2553-2566, 2009.
[17] C.-F. Juang, and C.-T. Lin, “An online self-constructing neural fuzzy inference network and its applications,” *Fuzzy Systems, IEEE Transactions on,* vol. 6, no. 1, pp. 12-32, 1998.
[18] R. Martínez-Soto, O. Castillo, and L. T. Aguilar, “Type-1 and Type-2 fuzzy logic controller design using a Hybrid PSO–GA optimization method,” *Information Sciences,* vol. 285, pp. 35-49, 2014.
[19] J. M. Mendel, and R. I. B. John, “Type-2 fuzzy sets made simple,” *Fuzzy Systems, IEEE Transactions on,* vol. 10, no. 2, pp. 117-127, 2002.
[20] S. Greenfield, F. Chiclana, S. Coupland *et al.*, “The collapsing method of defuzzification for discretised interval type-2 fuzzy sets,” *Information Sciences,* vol. 179, no. 13, pp. 2055-2069, 2009.
[21] J. M. Mendel, *Uncertain rule-based fuzzy logic system: introduction and new directions*, Upper Saddle River, NJ: Prentice-Hall, 2001.
[22] J. M. Mendel, “Type-2 fuzzy sets and systems: an overview,” *Computational Intelligence Magazine, IEEE,* vol. 2, no. 1, pp. 20-29, 2007.
[23] F. Doctor, H. Hagras, and V. Callaghan, “A type-2 fuzzy embedded agent to realise ambient intelligence in ubiquitous computing environments,” *Information Sciences,* vol. 171, no. 4, pp. 309-334, 2005.
[24] H. Hagras, and C. Wagner, “Introduction to interval type-2 fuzzy logic controllers—towards better uncertainty handling in real world applications,” *IEEE Systems, Man and Cybernetics eNewsletter,* vol. 27, 2009.
[25] A. Rubio-Solis, and G. Panoutsos, “Interval Type-2 Radial Basis Function Neural Network: A Modeling Framework,” *Fuzzy Systems, IEEE Transactions on,* vol. 23, no. 2, pp. 457-473, 2015.
[26] A. Baraka, G. Panoutsos, and S. Cater, “A real-time quality monitoring framework for steel friction stir welding using computational intelligence,” *Journal of Manufacturing Processes,* vol. 20, pp. 137-148, 2015.
[27] C.-F. Juang, R.-B. Huang, and W.-Y. Cheng, “An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems,” *IEEE Transactions on Fuzzy Systems,* vol. 18, no. 4, pp. 686-699, 2010.
[28] J. Zeng, and Z.-Q. Liu, “Type-2 fuzzy hidden Markov models and their application to speech recognition,” *IEEE Transactions on Fuzzy Systems,* vol. 14, no. 3, pp. 454-467, 2006.
[29] C.-F. Juang, Y.-Y. Lin, and R.-B. Huang, “Dynamic system modeling using a recurrent interval-valued fuzzy neural network and its hardware implementation,” *Fuzzy Sets and Systems,* vol. 179, no. 1, pp. 83-99, 2011.
[30] C.-F. Juang, and Y.-W. Tsao, “A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning,” *Fuzzy Systems, IEEE Transactions on,* vol. 16, no. 6, pp. 1411-1424, 2008.
[31] Y.-Y. Lin, J.-Y. Chang, N. R. Pal *et al.*, “A mutually recurrent interval type-2 neural fuzzy system (MRIT2NFS) with self-evolving structure and parameters,” *Fuzzy Systems, IEEE Transactions on,* vol. 21, no. 3, pp. 492-509, 2013.

[32] Y.-Y. Lin, J.-Y. Chang, and C.-T. Lin, "A TSK-type-based self-evolving compensatory interval type-2 fuzzy neural network (TSCIT2FNN) and its applications," *Industrial Electronics, IEEE Transactions on,* vol. 61, no. 1, pp. 447-459, 2014.

[33] C.-F. Juang, and C.-Y. Chen, "An interval type-2 neural fuzzy chip with on-chip incremental learning ability for time-varying data sequence prediction and system control," *Neural Networks and Learning Systems, IEEE Transactions on,* vol. 25, no. 1, pp. 216-228, 2014.

[34] C.-T. Lin, N. R. Pal, S.-L. Wu *et al.*, "An Interval Type-2 Neural Fuzzy System for Online System Identification and Feature Elimination," *IEEE transactions on neural networks and learning systems,* vol. 26, no. 7, pp. 1442-1455, 2015.

[35] A. Baraka, G. Panoutsos, and S. Cater, "Perpetual Learning Framework based on Type-2 Fuzzy Logic System for a Complex Manufacturing Process," *IFAC-PapersOnLine,* vol. 49, no. 20, pp. 143-148, 2016.

[36] A. Baraka, G. Panoutsos, M. Mahfouf *et al.*, "A Shannon Entropy-Based Conflict Measure For Enhancing Granular Computing-Based Information Processing," in 2014 IEEE International Conference on Granular Computing (GrC), Noboribetsu, Japan, 2014, pp. 13 - 18.

[37] N. K. Kasabov, "On-line learning, reasoning, rule extraction and aggregation in locally optimized evolving fuzzy neural networks," *Neurocomputing,* vol. 41, no. 1, pp. 25-45, 2001.

[38] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *Systems, Man and Cybernetics, IEEE Transactions on,* vol. 23, no. 3, pp. 665-685, 1993.

[39] J. Zeng, L. Xie, and Z.-Q. Liu, "Type-2 fuzzy Gaussian mixture models," *Pattern Recognition,* vol. 41, no. 12, pp. 3636-3643, 2008.

[40] J. M. Mendel, "Computing derivatives in interval type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems,* vol. 12, no. 1, pp. 84-98, 2004.

[41] H. Wu, and J. M. Mendel, "Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems,* vol. 10, no. 5, pp. 622-639, 2002.

[42] Q. Liang, and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *Fuzzy Systems, IEEE Transactions on,* vol. 8, no. 5, pp. 535-550, 2000.

[43] C.-H. Wang, C.-S. Cheng, and T.-T. Lee, "Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 34, no. 3, pp. 1462-1477, 2004.

[44] R. P. Paiva, and A. Dourado, "Interpretability and learning in neuro-fuzzy systems," *Fuzzy sets and systems,* vol. 147, no. 1, pp. 17-38, 2004.

[45] G. Panoutsos, and M. Mahfouf, "A neural-fuzzy modelling framework based on granular computing: Concepts and applications," *Fuzzy Sets and Systems,* vol. 161, no. 21, pp. 2808-2830, 2010.

[46] L. A. Zadeh, "Toward a perception-based theory of probabilistic reasoning with imprecise probabilities," *Journal of statistical planning and inference,* vol. 105, no. 1, pp. 233-264, 2002.

[47] B.-I. Choi, and F. C.-H. Rhee, "Interval type-2 fuzzy membership function generation methods for pattern recognition," *Information Sciences,* vol. 179, no. 13, pp. 2102-2122, 2009.

[48] Y. Y. Yang, M. Mahfouf, and G. Panoutsos, "Development of a parsimonious GA–NN ensemble model with a case study for Charpy impact energy prediction," *Advances in Engineering Software,* vol. 42, no. 7, pp. 435-443, 2011.

[49] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," *Doctoral Dissertation, Applied Mathematics, Harvard University, MA*, 1974.

[50] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning internal representations by error propagation*, DTIC Document, 1985.

[51] M. Markou, and S. Singh, “Novelty detection: a review—part 1: statistical approaches,” *Signal processing,* vol. 83, no. 12, pp. 2481-2497, 2003.

[52] M. Markou, and S. Singh, “Novelty detection: a review—part 2:: neural network based approaches,” *Signal processing,* vol. 83, no. 12, pp. 2499-2521, 2003.

[53] Q. Zhang, “Nature-Inspired Multi-Objective Optimisation and Transparent Knowledge Discovery via Hierarchical FuzzyModelling,” Department of Automatic Control and Systems Engineering, The University of Sheffield, 2008.

[54] M. Setnes, R. Babuška, U. Kaymak *et al.*, “Similarity measures in fuzzy rule base simplification,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 28, no. 3, pp. 376-386, 1998.

[55] H. Wang, S. Kwong, Y. Jin *et al.*, “Agent-based evolutionary approach for interpretable rule-based knowledge extraction,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 35, no. 2, pp. 143-155, 2005.

[56] D. Dubois, H. Prade, and L. Ughetto, “Checking the coherence and redundancy of fuzzy knowledge bases,” *Fuzzy Systems, IEEE Transactions on,* vol. 5, no. 3, pp. 398-417, 1997.

[57] G. Deng, Y. Jiang, and J. Fu, “Monotonic similarity measures between fuzzy sets and their relationship with entropy and inclusion measure,” *Fuzzy sets and systems,* vol. 287, pp. 97-118, 2016.

[58] D. Wu, and J. M. Mendel, “A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets,” *Information Sciences,* vol. 179, no. 8, pp. 1169-1192, 2009.

[59] A. Baraka, “Soft-Computing and Human-Centric Approaches for Modelling Complex Manufacturing Systems,” University of Sheffield, 2017.

[60] M. Jamil, and X.-S. Yang, “A literature survey of benchmark functions for global optimisation problems,” *International Journal of Mathematical Modelling and Numerical Optimisation,* vol. 4, no. 2, pp. 150-194, 2013.

[61] V. Cherkassky, D. Gehring, and F. Mulier, “Comparison of adaptive methods for function estimation from samples,” *Neural Networks, IEEE Transactions on,* vol. 7, no. 4, pp. 969-984, 1996.

[62] X.-S. Yang, Z. Cui, R. Xiao *et al.*, *Swarm intelligence and bio-inspired computation: theory and applications*: Newnes, 2013.

[63] I.-C. Yeh, “Modeling of strength of high-performance concrete using artificial neural networks,” *Cement and Concrete research,* vol. 28, no. 12, pp. 1797-1808, 1998.