RIT2FIS : a recurrent interval type 2 Fuzzy Inference System and its rule base estimation

Samanta, Subhrajit; Hartanto, Andre; Pratama, Mahardhika; Sundaram, Suresh; Srikanth, Narasimalu

2019

Samanta, S., Hartanto, A., Pratama, M., Sundaram, S., & Srikanth, N. (2019). RIT2FIS : a recurrent interval type 2 Fuzzy Inference System and its rule base estimation. International Joint Conference on Neural Networks (IJCNN 2019).

<https://hdl.handle.net/10356/106766>

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Downloaded on 26 Apr 2023 08:26:05 SGT

RIT2FIS: A Recurrent Interval Type 2 Fuzzy Inference System and its Rule Base Estimation

Subhrajit Samanta	Andre Hartanto	Mahardhika Pratama	Suresh Sundaram	Narasimalu Srikanth
<i>ERI@N-IGS</i>	<i>SCSE</i>	<i>SCSE</i>	<i>Aerospace Engineering</i>	<i>ERI@N</i>
<i>NTU</i>	<i>NTU</i>	<i>NTU</i>	<i>IISC</i>	<i>NTU</i>
Singapore	Singapore	Singapore	Bengaluru, India	Singapore
sama0021@e.ntu.edu.sg	C150007@e.ntu.edu.sg	MPRATAMA@ntu.edu.sg	sureshsundaram@iisc.ac.in	nsrikanth@ntu.edu.sg

Abstract—Two of the major challenges associated with time series modelling are handling uncertainty present in the data and tracing its dynamical behaviour. A Recurrent Interval Type 2 Fuzzy Inference System or RIT2FIS is presented in this paper. RIT2FIS adopts an interval type 2 fuzzy inference mechanism for superior handling of uncertainty. The memory neurons employed in its hidden and output layer, retain the temporal information, making RIT2FIS highly proficient in tracing system dynamics at a granular level. RIT2FIS also benefits from incorporating a k-means algorithm inspired approach to cluster the data in an unsupervised manner. An ‘Elbow Method’ is utilized next to determine the optimal clustering which is then employed as the optimal fuzzy rule base for RIT2FIS, eliminating the necessity of expert knowledge for fuzzy initiation. The antecedent and consequent parameters of RIT2FIS are updated using a gradient descent based backpropagation through time algorithm where the learning is made self-regulatory to avoid over-fitting and ensure generalization. Performance of RIT2FIS is evaluated against popular neuro-fuzzy methods on different benchmark and real-world time series problems which distinctly indicates an improved accuracy and a parsimonious rule base.

Index Terms—Recurrent Neural Fuzzy Network, type 2 Fuzzy Inference System, Fuzzy Rule Base Estimation, k-Means Clustering, Time Series Forecasting, Dynamic System Identification

I. INTRODUCTION

A. Background

Time series modelling and forecasting have been an active research topic for a long time where the goal is to estimate single or multiple time steps ahead output. Such models use historical data containing exogenous inputs and past outputs to model the internal dynamic characteristics and perform the prediction task. Time series forecasting has various applications across different domains, to name a few are energy demand prediction [1], weather forecasting [2], financial prediction [3] etc.

The time series problem is inherent with the uncertainty and impreciseness associated with external factors such as noise, inaccurate measurement, faulty sensors, etc, thus hindering to achieve decent predictive performance. On the other hand, there has been a plethora of researches to bring the uncertainty handling capability of fuzzy inference systems (FIS) and learning capability of artificial neural networks together, in the form of neuro-fuzzy systems [4]–[9] which offer some

advantage of the time-series analysis. However, most of these researches are mainly built upon the type 1 fuzzy inference systems (FIS) claimed to be less efficient than the type-2 fuzzy system in the context of uncertainty handling [10].

The type 2 fuzzy systems [10], [11] incorporate a secondary membership function on top of the primary membership function. This modification, however, comes at a cost of increased computational complexity which leads to the proposal of interval type 2 fuzzy systems as an alternative [12]–[15]. The interval type-2 fuzzy system simplifies the type-2 fuzzy system with a unity secondary membership but introduces the interval-valued fuzzy parameters to retain its traits against uncertainty.

B. Principle Aspects and Major Contributions

In this paper, a Recurrent Interval type 2 Fuzzy Inference System, namely RIT2FIS is proposed. The novelty of RIT2FIS lies in the proposal of a spatio-temporal functionality embedded into its type 2 fuzzy framework at a cellular level. The noteworthy features and benefits of RIT2FIS are outlined,

1) *Network Architecture*: An interval Type 2 fuzzy inference system is proposed here which is essentially a six-layered network structure where the first 3 layers form the antecedent and the last 3 constitute the consequent part. Memory type neurons are incorporated in the last two layers **to retain both spatial and temporal information of long pasts** in order to trace system dynamics. The crisp inputs are fuzzified and inferred using the type-2 fuzzy inference engine followed by the type reduction mechanism to convert the type-2 fuzzy variable back to the type-1 fuzzy variable. Layer 4 utilizes a normalization technique to assure the partition of unity property. In the final layer, the weighted summation operation is performed to get the final prediction.

2) *Rule Base Estimation*: In any fuzzy inference system (FIS) one of the major challenges is to determine the optimal rule base (i.e. Consisting **If x is A Then y is B** type fuzzy rules). A k-means algorithm inspired approach is used to cluster the data first in an unsupervised manner without any manual intervention. Next, an elbow method is employed **to determine the number of clusters** constructing the IF part of a fuzzy rule.

3) *Parameter Tuning*: The popular gradient descent based backpropagation through time [16] algorithm or GDBP is

utilized to tune the antecedent (i.e. center and spreads of fuzzy rules) and consequent parameters (i.e. output weights etc.). The learning process is made self-regulatory which in the beginning learns from fewer high error samples and gradually trains from more and more samples. **Self-regulation helps the network in generalization** as it exempts redundant samples to be involved in the training process. The advantages of RIT2FIS is studied both numerically and theoretically where it is found to be superior in two aspects: accuracy and number of rules.

Rest of the paper starts with a literature review followed by a short study on interval type 2 FIS, problem formulation, design aspects of RIT2FIS, learning method and is concluded with experimental results. A cursory summary of the author's most closely related works is also presented to demonstrate the originality of the proposed work in this paper.

II. LITERATURE SURVEY

A brief survey of the relevant literature is presented here. In [17] artificial neurons were first proposed from the motivation of biological neurons. [18] first used a fuzzy inference system (FIS) and its linguistic reasoning for dynamic system identification. Since there have been numerous works to fuse fuzzy logic and neural networks. ANFIS [4] was one of the earlier attempts in this domain. eTS [5] introduces the concept of statistical potential associated with each data point to decide how the structure evolves. SAFIS [6] utilizes the influence of a rule to grow/prune the rules in the network. In McFIS [7] spherical potential is used to update the structure. PANFIS [8] and GENEFIS [9] employ multi-dimensional membership function and a statistical contribution approach to evolve rules.

A recurrent structure is of high significance when it comes to modeling time series hence there has been a multitude of researches to incorporate this property into the neuro-Fuzzy framework. The methods found in literature can be broadly categorized into local and global recurrence. Examples of **local recurrence** are DFNN [19] (self-feedback in the membership and output layer), RFNN [20] and WRFNN [21] (self-feedback is allowed in membership layer only), DRFNN [22] (self-feedback in the rule layer) etc. In **global recurrence** feedbacks are not restricted only to a single layer such as in RSONFIN [23] and TRFN [24] firing strength of a rule becomes another input for the membership layer. Apart from the aforementioned techniques, there is an alternative approach, in MNN [25] memory neurons are presented which use self-feedback to retain past memory.

Tackling uncertainty and vagueness coming from erroneous measurements or noise is a challenge which type-1 fuzzy systems might fail to overcome sometimes. This issue is addressed with the use of type-2 fuzzy set [10] which handles uncertainty using a secondary membership function in addition to the primary membership functions of type 1 fuzzy. This often leads to more computational cost and higher run-time. Hence interval type-2 fuzzy set [11] is introduced which sets the secondary membership as unity. Many interval type-2 neuro-fuzzy inference systems have been developed since. In [12] and [13] fuzzy clustering is used to build the network

and the parameters are optimized by backpropagation. PBL-McIT2FIS [14] proposes a projection based learning to tune its consequent structure and uses meta-cognition to estimate the rule base. In [15] an incremental extreme machine learning method based IT2NFIS is proposed.

In [16] Backpropagation Through Time was introduced where the network unfolds in time and propagates the error back through the unfolded structure to update the parameters.

III. INTERVAL TYPE 2 FUZZY SYSTEM

If UOD_x is the universe of discourse (or domain) of the real crisp values x then corresponding fuzzy set X such that membership function (MF) $\phi_X(x) : UOD_x \rightarrow [0, 1]$, can be defined as,

$$X = \int_{x \in UOD_x} \phi_X(x) \quad (1)$$

where $\int$ represents the collection of all point $x \in UOD_x$ with associated membership value $\phi_X(x)$. In type 2 FIS, even the membership grades are converted into fuzzy domain with the help of a secondary variable u with domain $UOD_u \in [0, 1]$. The secondary membership grade is $\phi_{\tilde{X}}(x, u)$. Hence the type 2 fuzzy set $\tilde{X}$ is,

$$\tilde{X} = \int_{x \in UOD_x} \int_{u \in UOD_u} \phi_{\tilde{X}}(x, u) \quad (2)$$

From Fig.2 one can observe that the membership degree of type 2 FIS is bounded by the upper MF and lower MF (i.e. In Fig. 2 the type 2 membership grade corresponding to the crisp value 3.6 can be anything between 0.25 to 0.95).

Figure 1: type 2 fuzzy Inference with uncertain means

Remark 1: type 2 FIS is claimed more robust than type-1 FIS in handling uncertainty due to the existence of footprint of uncertainty formed by the upper and lower membership function. This trait, however, comes at the cost of computational complexity and network parameters.

IV. PROBLEM FORMULATION

Time series forecasting can be modeled as a function approximation problem where τ time steps ahead output is estimated based on its past output $y(k)$ and exogenous inputs $\mathbf{u}(k)$ (where $\mathbf{u}(k) = [u_1(k), u_2(k), \dots, u_P(k)]^T \in \mathbb{R}^P$) as,

$$\hat{y}(k+\tau) = h(y(k), \dots, y(k-n_y); \mathbf{u}(k), \dots, \mathbf{u}(k-n_u)) \quad (3)$$

here $h(\cdot)$ is the arbitrary nonlinear function representing the model and n_y, n_u are the required number of lags for the time series prediction.

V. RECURRENT INTERVAL TYPE 2 FUZZY INFERENCE SYSTEM (RIT2FIS)

In RIT2FIS memory neurons [25] are adopted into the type 2 fuzzy framework in the final two layers to capture spatio-temporal characteristics of the dataset.

A. Spatio-Temporal functionality of Memory Neurons

Memory neurons provide one static output which accounts for the spatial functionality of the fed inputs and one dynamic output which represents the temporal characteristics as shown in the equations below. Let's assume the input to the memory neuron is $\mathbf{d}^I(k) \in \mathbb{R}^p$, then the **spatial output** $d_O(k)$ and **temporal output** $d_O^m(k)$ can be stated as,

$$d_O(k) = f(\mathbf{d}^I(k)) \quad (4)$$

$$d_O^m(k) = \alpha d_O(k-1) + (1-\alpha)d_O^m(k-1) \quad (5)$$

where $f(\cdot)$ can be any nonlinear function depending on the spatial functionality of the memory neuron and α is the memory neuron parameter ($0 \leq \alpha \leq 1$). From Eqn. 5 it can be observed that the temporal output retains the effect of all past instances with exponentially decreasing co-efficient.

B. RIT2FIS Architecture

The proposed interval type 2 neuro-fuzzy architecture is given in Figure 2. It comprises of six layers i.e. Input layer, Membership layer, Firing layer, type reduction layer, Normalization layer and the output layer. RIT2FIS puts memory neurons in use both in defuzzification and output layers (shown by solid squares in figure 2). Details of the layer wise functionality are provided here,

Input Layer: This layer simply transfers the input knowledge to the next level. The network accepts $\hat{P}$ inputs (i.e. this can be current and previous instances of exogenous inputs and past outputs), hence the number of nodes is $\hat{P}$ in this layer. The input vector is represented as $\mathbf{v}(k) = [\mathbf{u}(k), y(k-1)]^T$ hence the corresponding outputs from layer 1 are simply,

$$x_i(k) = \mathbf{v}_i(k), \quad i = 1, 2, \dots, \hat{P} \quad (6)$$

Fuzzification Layer: An interval type 2 fuzzy inference engine is incorporated here to handle the uncertainty in real world time series data deftly. Gaussian membership function is employed to obtain the fuzzified values. Now with a type 1 membership function, the fuzzified value for $x_i(k)$, for r^{th} rule is,

$$\begin{aligned} \phi_{ri}(k) &= \exp \left[-\frac{1}{2} \left[\frac{(x_i(k) - \mu_{ri})}{\sigma_{ri}} \right]^2 \right] \\ &= \phi(\mu_{ri}, \sigma_{ri}, x_i(k)) \end{aligned} \quad (7)$$

but in type 2 fuzzy engines even the membership functions are assumed to have uncertainty, hence two type 1 membership functions with centers at μ_{ri}^{lef} and μ_{ri}^{rig} will be acting together to form the type 2 engine. Our experimental study reveals that the interval type-2 fuzzy system outperforms the type-1 fuzzy systems in accuracy but it comes with cost of increased

Figure 2: Network architecture of RIT2FIS

execution time. Here, $\mu_{ri} \in [\mu_{ri}^{lef}, \mu_{ri}^{rig}]$ can be dubbed as the left and right boundaries of the r^{th} rule center with σ_{ri} as the width. Hence the footprint of uncertainty for this rule can be demonstrated with an interval between the upper and lower membership functions as $\Phi_{ri}(k) = [\phi_{ri}^{lo}(k), \phi_{ri}^{up}(k)]$, where

$$\phi_{ri}^{up}(k) = \begin{cases} \phi(\mu_{ri}^{lef}, \sigma_{ri}, x_i(k)), & x_i(k) < \mu_{ri}^{lef} \\ 1, & \mu_{ri}^{lef} \leq x_i(k) \leq \mu_{ri}^{rig} \\ \phi(\mu_{ri}^{rig}, \sigma_{ri}, x_i(k)), & x_i(k) > \mu_{ri}^{rig} \end{cases} \quad (8)$$

$$\phi_{ri}^{lo}(k) = \begin{cases} \phi(\mu_{ri}^{rig}, \sigma_{ri}, x_i(k)), & x_i(k) \leq \frac{\mu_{ri}^{lef} + \mu_{ri}^{rig}}{2} \\ \phi(\mu_{ri}^{lef}, \sigma_{ri}, x_i(k)), & x_i(k) > \frac{\mu_{ri}^{lef} + \mu_{ri}^{rig}}{2} \end{cases} \quad (9)$$

From Eqn. 8 one can observe that the upper membership function is bounded by 1 hence this is regarded as an Interval type 2 FIS. The rule base size determination (i.e. number of rules) and rule centers initialization (i.e. $\mu_{ri}^{lef}, \mu_{ri}^{rig}$) is presented in detail later in the paper.

Firing Layer: In this layer the firing strengths are computed for each fuzzy rules. RIT2FIS has two firing strengths (upper and lower) corresponding to the upper and lower membership values from previous layer. The rule aggregation is achieved with a t-norm operation, realized by multiplying all the fuzzified values. So the firing strength corresponding to the r^{th} rule can be given by $[F_r^{lo}(k), F_r^{up}(k)]$ where,

$$\begin{aligned} F_r^{lo}(k) &= \prod_{i=1}^{\hat{P}} (\phi_{ri}^{lo}(k)) \\ F_r^{up}(k) &= \prod_{i=1}^{\hat{P}} (\phi_{ri}^{up}(k)) \end{aligned} \quad (10)$$

type Reduction Layer: This layer performs a type reduction procedure to obtain type 1 fuzzy set from the type 2 sets. The technique used here is a variant of Nie-Tan's [26] weighted sum method (of lower and upper firing strengths) with type reduction co-efficient γ (set at 0.5) as given below,

$$F_r(k) = \gamma F_r^{lo}(k) + (1-\gamma) F_r^{up}(k) \quad (11)$$

Defuzzification Layer : This layer executes the defuzzification task to convert the type 1 fuzzy sets into crisp values. This layer has R number of memory neurons corresponding to the fuzzy rules. For the r^{th} memory neuron in this layer (layer designated by ' R ') the inputs are $[F_1(k), F_2(k), \dots, F_R(k)]^T$. Hence the corresponding **spatial** and **temporal** mappings can be defined using eqn. 12 and eqn. 13 as below,

$$\bar{F}_r(k) = f_r^R(F_1(k), \dots, F_R(k)) = \frac{F_r(k)}{\sum_{r=1}^R F_r(k)} \quad (12)$$

$$\bar{F}_r^m(k) = \alpha_r^R \bar{F}_r(k-1) + (1 - \alpha_r^R) \bar{F}_r^m(k-1) \quad (13)$$

where α_r^R is the memory neuron parameter ($0 \leq \alpha_r^R \leq 1$).

Output Layer: The output layer (layer designated by ' F ') contains a single memory neuron. Its is fed $[\bar{F}_1(k), \dots, \bar{F}_R(k), \bar{F}_1^m(k), \dots, \bar{F}_R^m(k)]^T$ as input, so the **spatial** and **temporal** outputs are,

$$\hat{y}(k) = \sum_{r=1}^R w_r \cdot \bar{F}_r(k) + \sum_{r=1}^R w_r^m \cdot \bar{F}_r^m(k) \quad (14)$$

$$\hat{y}^m(k) = \alpha^F \hat{y}(k-1) + (1 - \alpha^F) \hat{y}^m(k-1) \quad (15)$$

Finally, τ time step ahead prediction from RIT2FIS, i.e. $\hat{y}(k + \tau)$ is computed using the weighted summation of $\hat{y}(k)$ and $\hat{y}^m(k)$ as given below,

$$\begin{aligned} \hat{y}(k + \tau) &= \hat{y}(k) + \beta \cdot \hat{y}^m(k) \\ &= \sum_{r=1}^R w_r \cdot \bar{F}_r(k) + \sum_{r=1}^R w_r^m + \beta \hat{y}^m(k) \end{aligned} \quad (16)$$

where β is the weight of the link connecting the second output from the memory neuron in the output layer to the final node. Here $0 \leq \beta \leq 1$ and weight of the link between the first output of the dynamic neuron and the final node is kept as 1.

C. Rule Base estimation and Initialization

Estimation of an optimal fuzzy rule base is a challenge for any fuzzy systems. Expert knowledge can be adopted for the same but most often such knowledge is rare while designing a neural fuzzy system for real-world applications. Here a k means clustering algorithm [27] and an 'Elbow Method' is employed on the training data to group the data points into different optimal clusters in an unsupervised manner. Once the optimal clustering is achieved each of these clusters can be regarded as the representation of the fuzzy rules and their cluster centers can be initialized as the fuzzy rule centers. A brief discussion on the clustering algorithm is provided below (referred here as R means algorithm instead of k-means for notational simplicity).

Clustering using R Means algorithm:

In a direct R Means clustering algorithm the number of clusters R is assumed to be fixed. Then the algorithm is executed for a different number of clusters till Within Class Sum of Square (WCSS) doesn't decrease significantly indicating the optimal clustering. Please note clustering is unsupervised hence doesn't require the actual output for each data point. The algorithm is explained with pseudocode for better comprehension and clarity.

Start with the input training data $\mathbf{v}(k) \in R^{\hat{P}}$;

Set number of clusters to R ;

Assign R randomly selected samples (from total S samples) to each of the R clusters as their cluster centers, i.e. $C_r = \mathbf{v}(k)$ where $r = 1, 2, \dots, R$;

while $WCSS > \text{Pre-defined Threshold}$ **do**

while All samples $k = 1, 2, \dots, S$ are visited **do**

Assign k^{th} sample to the cluster for which $\|\mathbf{v}(k) - C_r\|$ is minimum, C_r being the cluster center of r^{th} cluster;

end

Recompute the cluster centers for each clusters as $C_r^{new} = \sum_{k \in C_r} \mathbf{v}(k) / |C_r|$ where $|C_r|$ denotes the number of data points assigned to the r^{th} cluster;

Compute the clustering quality index (WCSS);

$$WCSS_R = \sum_{r=1}^R \sum_{k \in C_r} \|\mathbf{v}(k) - C_r\|^2 \quad (17)$$

end

Algorithm 1: R Means Clustering

Cluster Number Determination using The Elbow Method:

Once the clustering is completed and the final $WCSS$ value is available then we employ 'The Elbow Method' to compute the optimum number of clusters.

In this method the number of cluster is initialized at 1 (i.e. $R = 1$) and then the R Means clustering is performed for incremental R values (i.e. $R = 1, 2, 3, \dots$) and the corresponding WCSS value is recorded. It can be observed that after a certain value of R the WCSS value doesn't decrease significantly, this value of R is regarded as the optimal number of clusters.

The elbow plot for SID data is given in Fig. 3 where WCSS value doesn't decrease significantly after $R = 4$. Hence, from visual inspection of the plot, the optimal number of clusters is estimated to be 4 for the SID problem.

Figure 3: The elbow plot for SID problem

Let's assume that the elbow method indicates R to be the optimal number of clusters. In case of R' (s.t. $R' < R$) clusters, distance between cluster centers and data points will be greater leading to a bigger WCSS value (from Eqn. 17). Hence, for R' number of clusters $|x_i(k) - \mu_{ri}|$ will be higher which will result in a lower membership value and in turn lower firing strengths.

$$\begin{aligned} WCSS_{R'} &> WCSS_R \\ \phi'_{ri} &< \phi_{ri} \\ \bar{F}'_r &< \bar{F}_r \end{aligned} \quad (18)$$

The weight update factor Δw_r is proportional to $\bar{F}_R$ (Eqn. 26) or $\Delta w_r \propto \bar{F}_r$. Hence continuing above inequality,

$$\Delta' w_r < \Delta w_r \quad (19)$$

Now as the update factor is small then evidently it will take much more iterations (epochs) to reach the optima so,

$$epochs' > epochs \quad (20)$$

Remark 2: Convergence will be slower when number of rules are smaller than optimal rules provided by the elbow method.

Type 2 Fuzzy Rule Centers Initialization from R-Means:

Optimal number of clusters R , indicates the optimal size of the fuzzy rule base to be R . In a gradient descent learning process the certainty and speed of convergence often heavily relies on the initialization of its parameters. Here, the centers of the fuzzy rule will be initialized as per the cluster centers instead of a random initialization (as per common practise) resulting in an efficient learning. From the optimal clustering we already have R centers as $C_1, C_2 \dots C_R$. In a type 2 FIS (as described in the previous section) each of the rules requires two membership functions to represent its footprint of uncertainty. Hence both of the centers $\mu_{ri}^{lef}, \mu_{ri}^{rig}$ are initialized,

$$\begin{aligned} \mu_{ri}^{lef} &= C_{ri} - 0.1 \\ \mu_{ri}^{rig} &= C_{ri} + 0.1 \end{aligned} \quad (21)$$

where $C_r = [C_{r1}, C_{r2} \dots C_{r\hat{p}}]^T$. Here 0.1 is chosen empirically with a trial and error method.

D. Self-Regulated Gradient Descent Based Backpropagation

A new variant of self-regulated learning method ([28], [29]) is adopted with which the coarse knowledge is captured at the beginning followed by the finer knowledge towards the end of the learning process. A self-regulating learning threshold is put in use for such selective learning. At the q^{th} training epoch the network will learn from a sample only if,

$$e(k) \geq E_L (1 - q/Q) \quad (22)$$

Here the initial learning threshold E_L decreases its value with every epoch and Q is the total number of epochs. This method ensures that RIT2FIS learns from fewer samples with higher errors in the beginning and more samples at the later stage of

the learning process to fine tune the network. Self-regulation avoids the overfitting case.

To train recurrent neural networks, backpropagation through time method [16] has been found to be most effective. Here the parameters are updated by unfolding the network and propagating the error backwards through time. The rules to update the parameters are computed in such a way that the squared prediction error J (i.e. the performance metric) is minimized. The cost function J is defined as,

$$J(k) = \frac{1}{2} \sum_k e(k)^2 \quad (23)$$

where $e(k) = y(k) - \hat{y}(k)$. Minimization of the cost function $J(k)$ (i.e. global quadratic error energy) is performed by travelling in the negative direction of the gradient of J in the parameter spaces as shown below.

Updating the Consequent Parameters : First the update rules are computed for the consequent parameters of the network. To minimize the cost function, the required change in the weight w_r is proportional to the negative gradient of J with respect to w_r as shown below,

$$\Delta w_r = -\eta \cdot \frac{dE}{dw_r} \quad (24)$$

where η is the learning rate and is kept close to 0. Utilizing chain rule of partial derivative one can simplify the above equation to,

$$\frac{dE}{dw_{1r}} = \frac{dE}{d\hat{y}} \cdot \frac{d\hat{y}}{dw_{1r}} = -e(k) \cdot \bar{F}_{1r}(k) \quad (25)$$

Hence the final update rule for the weight w_r is,

$$w_r(k+1) = w_r(k) + \eta \cdot e(k) \cdot \bar{F}_r(k) \quad (26)$$

In the similar manner, the gradient rules can be computed for the other consequent weights. The update rules for the w_r^m and β are,

$$w_r^m(k+1) = w_r^m(k) + \eta \cdot e(k) \cdot \bar{F}_r^m(k) \quad (27)$$

$$\beta(k+1) = \beta(k) + \eta \cdot e(k) \cdot \hat{y}^m(k) \quad (28)$$

where $r = 1, 2, \dots, R$. The update rule for the memory neuron parameters in the memory neurons are derived using a backpropagation through time approach. The difference here with the basic back-propagation is, the network unfolds in time hence at any point of time the required change in the particular parameter depends on its past instances. The update rules for α^F (for output layer memory neuron) and α_r^R (for rule firing layer memory neurons) are given by,

$$\alpha^F(k+1) = \alpha^F(k) + \eta \cdot e(k) \cdot \beta \cdot [\hat{y}(k-1) - \hat{y}^m(k-1)] \quad (29)$$

$$\alpha_r^R(k+1) = \alpha_r^R(k) + \eta \cdot e(k) \cdot w_r^m \cdot [\bar{F}_r(k-1) - \bar{F}_r^m(k-1)] \quad (30)$$

and finally the type reduction parameter γ is updated as,

$$\gamma(k+1) = \gamma(k) + \eta \sum_{r=1}^R w_r \frac{\sum_{r=1}^R F_r(k) - F_r(k)}{(\sum_{r=1}^R F_r(k))^2} (F_r^{lo} - F_r^{up}) \quad (31)$$

Updating the Antecedent Parameters: In the interval type 2 fuzzy engine incorporated here we have two membership functions to form the upper and lower membership functions. The update rule for the left membership centers of the r^{th} rule (i.e. μ_{ri}^{lef}) are derived as below,

$$\mu_{ri}^{lef}(k+1) = \mu_{ri}^{lef}(k) - \eta \frac{dJ}{d\mu_{ri}^{lef}} \quad (32)$$

Where

$$\begin{aligned} \frac{dJ}{d\mu_{ri}^{lef}} = & -e(k)w_r \cdot \frac{\sum_{r=1}^R F_r(k) - F_r(k)}{(\sum_{r=1}^R F_r(k))^2} \\ & \times \left(\left(\gamma \frac{dF_r^{lo}}{d\mu_{ri}^{lef}} + (1-\gamma) \frac{dF_r^{up}}{d\mu_{ri}^{lef}} \right) \right) \end{aligned} \quad (33)$$

with,

$$\frac{dF_r^{lo}}{d\mu_{ri}^{lef}} = \begin{cases} F_r^{lo} \times \frac{2(x_i(k) - \mu_{ri}^{lef})}{\sigma_r^2}, & x_i(k) > \frac{\mu_{ri}^{lef} + \mu_{ri}^{rig}}{2} \\ 0, & \text{Otherwise} \end{cases} \quad (34)$$

and

$$\frac{dF_r^{up}}{d\mu_{ri}^{lef}} = \begin{cases} F_r^{up} \times \frac{2(x_i(k) - \mu_{ri}^{lef})}{\sigma_r^2}, & x_i(k) \leq \mu_{ri}^{lef} \\ 0, & \text{Otherwise} \end{cases} \quad (35)$$

Following the same method the gradient rule for the other membership center (μ_{ri}^{rig}) is found,

$$\mu_{ri}^{rig}(k+1) = \mu_{ri}^{rig}(k) - \eta \frac{dJ}{d\mu_{ri}^{rig}} \quad (36)$$

Where

$$\begin{aligned} \frac{dJ}{d\mu_{ri}^{rig}} = & -e(k)w_r \cdot \frac{\sum_{r=1}^R F_r(k) - F_r(k)}{(\sum_{r=1}^R F_r(k))^2} \\ & \times \left(\left(\gamma \frac{dF_r^{lo}}{d\mu_{ri}^{rig}} + (1-\gamma) \frac{dF_r^{up}}{d\mu_{ri}^{rig}} \right) \right) \end{aligned} \quad (37)$$

with,

$$\frac{dF_r^{lo}}{d\mu_{ri}^{rig}} = \begin{cases} F_r^{lo} \times \frac{2(x_i(k) - \mu_{ri}^{rig})}{\sigma_r^2}, & x_i(k) \leq \frac{\mu_{ri}^{lef} + \mu_{ri}^{rig}}{2} \\ 0, & \text{Otherwise} \end{cases} \quad (38)$$

and

$$\frac{dF_r^{up}}{d\mu_{ri}^{rig}} = \begin{cases} F_r^{up} \times \frac{2(x_i(k) - \mu_{ri}^{rig})}{\sigma_r^2}, & x_i(k) > \mu_{ri}^{rig} \\ 0, & \text{Otherwise} \end{cases} \quad (39)$$

And lastly update rules for the membership function spreads (σ_{ri}) are computed next,

$$\sigma_{ri}(k+1) = \sigma_{ri}(k) - \eta \frac{dJ}{d\sigma_{ri}} \quad (40)$$

Where

$$\begin{aligned} \frac{dJ}{d\sigma_{ri}} = & -e(k)w_r \cdot \frac{\sum_{r=1}^R F_r(k) - F_r(k)}{(\sum_{r=1}^R F_r(k))^2} \\ & \times \left(\left(\gamma \frac{dF_r^{lo}}{d\sigma_{ri}} + (1-\gamma) \frac{dF_r^{up}}{d\sigma_{ri}} \right) \right) \end{aligned} \quad (41)$$

with,

$$\frac{dF_r^{lo}}{d\sigma_{ri}} = \begin{cases} F_r^{lo} \times \frac{(x_i(k) - \mu_{ri}^{rig})^2}{\sigma_{ri}^3}, & x_i(k) \leq \frac{\mu_{ri}^{lef} + \mu_{ri}^{rig}}{2} \\ F_r^{lo} \times \frac{(x_i(k) - \mu_{ri}^{lef})^2}{\sigma_{ri}^3}, & x_i(k) > \frac{\mu_{ri}^{lef} + \mu_{ri}^{rig}}{2} \end{cases} \quad (42)$$

and

$$\frac{dF_r^{up}}{d\sigma_{ri}} = \begin{cases} F_r^{up} \times \frac{(x_i(k) - \mu_{ri}^{lef})^2}{\sigma_{ri}^3}, & x_i(k) < \mu_{ri}^{lef} \\ 0, & \mu_{ri}^{lef} \geq x_i(k) \leq \mu_{ri}^{rig} \\ F_r^{up} \times \frac{(x_i(k) - \mu_{ri}^{rig})^2}{\sigma_{ri}^3}, & x_i(k) > \mu_{ri}^{rig} \end{cases} \quad (43)$$

VI. EXPERIMENTAL RESULTS

In the first category, performance comparison is carried out on a synthetic nonlinear dynamic system identification problem ([6], [30], [24]) followed by the second category with the benchmark Mackey Glass [31] problem and finally on a real world wind prediction problem. The goal of the proposed RIT2FIS method is to provide superior predictive accuracy while maintaining a compact fuzzy rule base. Hence for performance comparison, Root Mean Square Error (RMSE) or Non-Dimensional Error Index (NDEI) and fuzzy rule base size are used as indexes according to the standard practice. Four popular alternate neuro-fuzzy models i.e. eTS [5], SimpleTS [32], SAFIS [6] and McFIS [7] (type 1) are rebuilt in MATLAB 2016 for fair comparison, while type 2 results are quoted from literature when available. The learning rate η is kept at 0.0001 throughout all the experiments and the initial learning threshold E_L to 0.1. The number of epochs Q is set between 100-1000 empirically depending on the problem.

A. Dynamical System Identification Problem

First, a **nonlinear dynamical system** identification problem (SID) and its performance comparison results are presented. **Nonlinear System Identification Problem:** SID is a popular synthetic problem with details available in literature [6], [24], [30]. Basically, the prediction of one step ahead output $y(k+1)$ is the main goal here. $y(k+1)$ depends on two past inputs (i.e. $\mathbf{u}(k), \mathbf{u}(k-1)$) and three past outputs (i.e. $y(k), y(k-1), y(k-2)$). Training is done feeding 900 samples while testing is performed using 1000 according to the standard practice in literature. Complete details are to be found in [24]. Figure 4 compares the actual output of the discrete

Figure 4: SID target vs actual plot with RIT2FIS

time dynamical system vs RIT2FIS predicted output. From the figure, one can easily observe that the RIT2FIS follows the actual output very closely. The memory neurons in the defuzzification and output layer helps RIT2FIS to approximate the system dynamics closely. Training time (CPU time in second) is not a performance metric here (as RIT2FIS is offline in nature) however the same is provided in the Table I along with testing RMSE and number of fuzzy rules for comparison. It is evident from the table that RIT2FIS is superior.

Table I: Performance comparison on SID problem

Problem	Network	#Rules	CPU Time (s)	Test RMSE
SID1	eTS [5]	49	30	0.021
	SimpleTS [32]	22	25	0.030
	SAFIS [6]	17	45	0.022
	McFIS [7]	10	22	0.030
	RIT2FIS	4	24	0.019

B. Benchmark Time Series Problems

Next, performance comparison is performed on the well known benchmark **chaotic long term** time series forecasting problem, viz., the Mackey-Glass [31].

Mackey-Glass Time Series Prediction: Mackey-Glass (MG) data is produced from the differential equation given in [31] shown below,

$$\frac{dy}{dk} = \frac{0.2y(k - \bar{\tau})}{1 + y^{10}(k - \bar{\tau})} - 0.1y(k) \quad (44)$$

The problem stated that the 85 time steps ahead future value $y(k + 85)$ is predicted based on past ($y(k - 18)$, $y(k - 12)$, $y(k - 6)$) and current values, $y(k)$. For the differential equation stated above, $\bar{\tau} = 17$ and $y(0) = 1.2$ are used as parameter value. The training used 3000 samples and the testing used 500 samples, which totaled to 3500 samples, which were used for testing the performance of RIT2FIS.

Non-dimensional error index (NDEI, which is defined as the ratio of the Test RMSE and the standard deviation of the test data; in this case, the standard deviation is 0.51) is used as the performance indicator for this prediction problem. Table II provides the rule numbers and test-NDEI for this problem.

Table II: Performance comparison on benchmark problems

Problem	Network	#Rules	CPU Time (s)	Test NDEI
MG	eTS [5]	14	0.3	0.380
	SimpleTS [32]	11	0.4	0.394
	SAFIS [6]	16	0.5	0.376
	McFIS [7]	10	0.9	0.100
	SIT2FNN [33]	-	-	0.206
	RIT2FIS	10	1.5	0.090

RIT2FIS is compared with several type 1 neuro-fuzzy methods and one type 2 neuro-fuzzy methods available in the literature and from the presented result in table II, it can be observed that RIT2FIS was able to attain better results than the other methods.

C. Real World Problems

Next, performance comparison is carried on a real-world wind speed prediction problem [2] which involves a greater amount of **uncertainty, impreciseness and noise** compared to the benchmark problem.

Wind Speed Prediction Problem: Due to recent development, the wind has been considered as one of the viable alternatives for a sustainable source of green energy. Wind machine generated energy which has uncertain and volatile characteristics due to the stochastic nature of climate condition. The data used is gathered from February 1, 2011, to February 28, 2011, which is available in the Iowa (USA) Department of Transport website. The data is known to be noisy and imprecise. The data is taken every 10 minutes. The training data contains 500 instances and the test set contains 1000 instances. In Table III the performance of RIT2FIS is compared with several type 1 and 2 fuzzy systems. The result shows that RIT2FIS shows better test RMSE compared to others.

Figure 5: Wind speed target vs actual plot with RIT2FIS

Table III: Performance comparison on real world problem

Problem	Network	#Rules	CPU Time (s)	Test RMSE
SID1	eTS [5]	12	37	0.380
	SimpleTS [32]	11	42	0.394
	SAFIS [6]	16	45	0.376
	McFIS [7]	10	31	0.230
	SIT2FNN [33]	-	-	0.194
	McIT2FIS [34]	-	-	0.132
	RIT2FIS	8	30	0.057

Studying the result tables from I-III one can easily infer the superior performance of RIT2FIS over its peers and this can be attributed to the following,

- Inclusion of memory neurons for dynamic handling
- Utilization of k-means clustering for rule base estimation
- Self-regulatory learning for better generalization

VII. OUR RELATED WORK

Most relevant works of the authors are summarized in this section as an attempt to provide comparative evidence of novelty in RIT2FIS. In NFIS-DN [35] we have proposed a type 1 FIS with a different recurrent structure than RIT2FIS. In McRFIS-MN [36] a projection based learning was adopted (while RIT2FIS uses GDBP), in an evolving setting whereas RIT2FIS doesn't require to evolve its structure. SPATFIS [37]

proposes a new evolving FIS focusing on online learning for data streams and unlike RIT2FIS it doesn't employ either a type 2 FIS (for better handling of uncertainty) or a self-regulated GDBP (to ensure generalization). Neither of the aforementioned researches utilize the unsupervised k-means clustering algorithm for optimal rule base optimization.

CONCLUSION

The article here studies a Recurrent Interval Type 2 Fuzzy Inference System namely RIT2FIS, its k-mean clustering based rule base estimation technique and self-regulatory backpropagation through time learning algorithm. The superiority of RIT2FIS in handling uncertainty and temporal variability is demonstrated with the aid of three popular time series problems of different categories. In all those cases, RIT2FIS is able to produce better predictive accuracy while using a fewer number of fuzzy rules than other state-of-the-art neuro-fuzzy systems. Extension of RIT2FIS in a deep fuzzy framework for classification and high dimensional regression problem will be explored in the future direction of this research.

REFERENCES

- [1] N. Haldrup, F. S. Nielsen, and M. Ø. Nielsen, "A vector autoregressive model for electricity prices subject to long memory and regime switching," *Energy Economics*, vol. 32, no. 5, pp. 1044–1058, 2010.
- [2] I. Colak, S. Sagioglu, and M. Yesilbudak, "Data mining and wind power prediction: A literature review," *Renewable Energy*, vol. 46, pp. 241–247, 2012.
- [3] E. M. Azoff, *Neural network time series forecasting of financial markets*. John Wiley & Sons, Inc., 1994.
- [4] J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, May 1993.
- [5] P. P. Angelov and D. P. Filev, "An approach to online identification of takagi-sugeno fuzzy models," *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 1, pp. 484–498, Feb 2004.
- [6] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran, "Sequential adaptive fuzzy inference system (safis) for nonlinear system identification and prediction," *Fuzzy sets and systems*, vol. 157, no. 9, pp. 1260–1275, 2006.
- [7] K. Subramanian and S. Suresh, "A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system," *Applied soft computing*, vol. 12, no. 11, pp. 3603–3614, 2012.
- [8] M. Pratama, S. G. Anavatti, P. P. Angelov, and E. Lughofer, "Panfis: A novel incremental learning machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 55–68, 2014.
- [9] M. Pratama, S. G. Anavatti, and E. Lughofer, "Genefis: Toward an effective localist network," *IEEE Trans. Fuzzy Systems*, vol. 22, no. 3, pp. 547–562, 2014.
- [10] "The concept of a linguistic variable and its application to approximate reasoning," *Information Sciences*, 1975.
- [11] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems," in *Ninth IEEE International Conference on Fuzzy Systems. FUZZ- IEEE 2000 (Cat. No.00CH37063)*, 2000.
- [12] "A type-2 neuro-fuzzy system based on clustering and gradient techniques applied to system identification and channel equalization," *Applied Soft Computing*, 2011.
- [13] J. Tavoosi and M. A. Badamchizadeh, "A class of type-2 fuzzy neural networks for nonlinear dynamical system identification," *Neural Computing and Applications*, 2013.
- [14] A. K. Das, K. Subramanian, and S. Suresh, "A computationally fast interval type-2 neuro-fuzzy inference system and its meta-cognitive projection based learning algorithm," in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014.
- [15] M. Pratama, J. Lu, E. Lughofer, G. Zhang, and M. J. Er, "An incremental learning of concept drifts using evolving type-2 recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1175–1192, 2017.
- [16] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
- [17] J. J. Hopfield, "Artificial neural networks," *IEEE Circuits and Devices Magazine*, vol. 4, no. 5, pp. 3–10, 1988.
- [18] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," in *Readings in Fuzzy Sets for Intelligent Systems*. Elsevier, 1993, pp. 387–403.
- [19] P. A. Mastorocostas and J. B. Theoharis, "A recurrent fuzzy-neural model for dynamic system identification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 2, pp. 176–190, 2002.
- [20] C.-H. Lee and C.-C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 349–366, Aug 2000.
- [21] C.-J. Lin and C.-C. Chin, "Prediction and identification using wavelet-based recurrent fuzzy neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2144–2154, Oct 2004.
- [22] Q.-Z. Zhang, W.-S. Gan, and Y.-I. Zhou, "Adaptive recurrent fuzzy neural networks for active noise control," *Journal of Sound and Vibration*, vol. 296, no. 4–5, pp. 935–948, 2006.
- [23] C.-F. Juang and C.-T. Lin, "A recurrent self-organizing neural fuzzy inference network," *IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 828–845, 1999.
- [24] C.-F. Juang, "A tsf-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms," *IEEE Trans. on Fuzzy Systems*, vol. 10, no. 2, pp. 155–170, Apr 2002.
- [25] P. Sastry, G. Santharam, and K. Unnikrishnan, "Memory neuron networks for identification and control of dynamical systems," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 306–319, 1994.
- [26] M. Nie and W. W. Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008 (IEEE World Congress on Computational Intelligence)*. IEEE International Conference on. IEEE, 2008, pp. 1425–1432.
- [27] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 1979.
- [28] S. Suresh, K. Dong, and H. Kim, "A sequential learning algorithm for self-adaptive resource allocation network classifier," *Neurocomputing*, vol. 73, no. 16, pp. 3012–3019, 2010.
- [29] A. K. Das, K. Subramanian, and S. Sundaram, "An evolving interval type-2 neurofuzzy inference system and its metacognitive sequential learning algorithm," *IEEE Trans. on Fuzzy Systems*, vol. 23, no. 6, pp. 2080–2093, 2015.
- [30] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.
- [31] R. S. Crowder, "Predicting the mackey-glass time series with cascade-correlation learning," in *Proc. 1990 Connectionist Models Summer School*. Carnegie Mellon Univ., Pittsburgh, PA, 1990, pp. 117–123.
- [32] P. Angelov and D. Filev, "Simpl_ets: a simplified method for learning evolving takagi-sugeno fuzzy models," in *Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on*. IEEE, 2005, pp. 1068–1073.
- [33] C.-F. Juang and Y.-W. Tsao, "A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1411–1424, 2008.
- [34] A. K. Das, N. Anh, S. Suresh, and N. Srikanth, "An interval type-2 fuzzy inference system and its meta-cognitive learning algorithm," *Evolving Systems*, 2016.
- [35] S. Samanta and S. Sundaram, "A new neuro-fuzzy inference system with dynamic neurons (nfis-dn) for system identification and time series forecasting," *Under Review*, 07 2018.
- [36] S. Samanta, S. Ghosh, and S. Sundaram, "A meta-cognitive recurrent fuzzy inference system with memory neurons (mcrfis-mn) and its fast learning algorithm for time series forecasting," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 2231–2237.
- [37] S. Samanta, M. Pratama, and S. Sundaram, "A novel spatio-temporal fuzzy inference system (spatfis) and its stability analysis," *Under Review*, 12 2018.