2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# A Computationally Fast Interval Type-2 Neuro-Fuzzy Inference System and its Meta-Cognitive Projection Based Learning Algorithm

A.K. Das, K. Subramanian and S. Suresh

***Abstract*—In this paper, a computationally efficient Interval Type-2 Neuro-Fuzzy Inference System (IT2FIS) and its Meta-Cognitive projection based learning (PBL) algorithm is presented, together referred as PBL-McIT2FIS. A six layered network with computationally cheap type-reduction technique is proposed, rendering the inference mechanism faster. During learning, the projection based learning algorithm assumes that IT2FIS has no rules in the beginning, and the learning algorithm adds rules to the network and updates it depending on the prediction error and relative knowledge present in the current sample. As each sample is presented to the network, the meta-cognitive component of the learning algorithm decides *what-to-learn*, *when-to-learn* and *how-to-learn* it, depending on the instantaneous error and spherical potential of the current sample. Whenever a new rule is added or an existing rule is updated, a projection based learning algorithm computes the optimal output weights by minimizing the total error in the network in a computationally efficient manner. The performance of PBL-McIT2FIS is evaluated on a set of benchmark problem and compared to other state-of-the-art algorithms available in literature. The results indicate superior performance of PBL-McIT2FIS.**

***Index Terms*—Interval Type-2 fuzzy systems, Meta-cognition, Self-regulation, Projection based learning**

## I. INTRODUCTION

In the domain of soft-computing, artificial neural networks and fuzzy rule based systems are being increasingly employed to solve various problems such as wind speed prediction [1], human action emotion recognition [2], power system analysis [3]. This is due to the exceptional predictability of neural networks (by virtue of its learning ability) and interpretability of fuzzy systems. In order to combine the advantages of these two, Neuro-Fuzzy Inference Systems (NFISs) have been employed widely in literature. Traditionally, Type-1 fuzzy sets were employed in neuro-fuzzy inference systems since they can handle vagueness in data. Many NFISs have been proposed based on these Type-1 fuzzy sets (known as Type-1 fuzzy systems) [4–6]. An evolving clustering method was used in dynamic evolving fuzzy inference system [4] to add rules. An evolving Takagi-Sugeno model (eTS) is propsed in [5] which used the concept of potential to update the structure. The influence of rule has been used in Sequential Adaptive Fuzzy Inference System (SAFIS) [6] for growing/pruning the rules. However, information uncertainty such as measurement uncertainty, non-stationary noise, cannot be modeled using the above mentioned algorithms due to the use of Type-1 membership functions.

In literature, this issue has been solved by the use of Type-2 fuzzy sets [7]. Recently Type-2 Fuzzy Logic Systems (FLS) have been studied in [8, 9]. Type-2 fuzzy sets handle uncertainty by using a secondary membership degree but are computationally expensive. The computational effort involved in employing Type-2 fuzzy systems was reduced with the development of Interval Type-2 fuzzy sets [10]. In Interval Type-2 Fuzzy sets, the secondary memberships are set either to zero or one. Based on these Interval Type-2 fuzzy sets, various NFIS have been proposed [11–18]. A Takagi-Sugeno-Kang type fuzzy inference has been realized in [11]. It uses gradient descent for antecedent parameter learning and rule-ordered Kalman filter is used to tune consequent parameters. Three different architectures of Interval Type-2 fuzzy neural network (IT2FNN) have been presented in [12]. A Type-2 neuro-fuzzy network has been developed in [15], where fuzzy clustering is used to construct the structure of the network and the consequent parameters are tuned by employing a gradient descent based algorithm. In [14], authors have presented an IT2FNN where antecedent part is modeled by gradient descent algorithm while functional-link-based in the consequent part. A seven layered IT2FNN and its gradient descent based learning algorithm has been developed in [16]. A mamdani fuzzy inference mechanism for evolving Interval Type-2 NFIS is proposed in [17]. The learning algorithm employs gradient descent approach to adapt centers and width. It uses all the samples to approximate the functional relationship between input and output data. In [18], a self-evolving IT2FNN in which a simplified type-reduction step is proposed which involves adaptive adjustment of the upper and the lower values.

The above mentioned NFIS address the issue of *how-to-learn* the rules of fuzzy system, efficiently, by employing learning algorithms derived from artificial neural networks. These learning algorithms however assume that the training data is distributed uniformly, and learn all the samples as it is presented to them. In the literature of artificial neural networks, it has been shown that the order in which the sample is presented to the network influences its performance,

A.K. Das, K. Subramanian and S. Suresh are with the School of Computer Engineering, Nanyang Technological University, Singapore.

significantly [19]. Thus, we need to develop a learning algorithm for IT2-NFIS, which is able to self-regulate its learning by deciding *what-to-learn*, *when-to-learn* and *how-to-learn*, a given training sample, efficiently. In literature, it has been shown that learning algorithms employing the concepts of meta-cognitive self-regulation has better generalization ability. Accordingly many meta-cognitive learning algorithms [19–27] have been proposed.

Similar to the above works, we propose a meta-cognitive learning algorithm for an IT2FIS. A computationally fast, IT2FIS is the cognitive component and a self-regulatory learning mechanism, which controls the learning of the cognitive component forms the meta-cognitive component. The IT2FIS is realized as a six layered network, consisting of an input layer with as many nodes as the number of input features ($m$), a fuzzification layer employing Gaussian membership function and a firing layer, with each layer consisting of twice as many nodes as number of rules ($2 \times K$), a type-reduction layer and a normalization layer consisting of $K$ nodes, each and an output layer with as many nodes as number of output features ($n$). The type-reduction layer employs a modified version of a simplified type-reduction technique proposed in [28], rendering the inference mechanism fast. The proposed IT2FIS begins with zero rules, and as each sample is presented, either adds, updates or deletes rules from the network, depending on the relative knowledge represented by the training sample and the network. A projection based learning mechanism is employed to analytically determine the output weight during rule addition or rule update.

As each sample is presented to the network, the meta-cognitive component monitors the knowledge in the cognitive component and effectively controls the learning in it. Prediction error and sample novelty is employed to monitor the knowledge in the cognitive component. The prediction error is measured by employing root mean squared error (RMSE) and sample novelty is measured using spherical potential [29]. Spherical potential is defined as the average distance of the current sample from all the rules in the network. The meta-cognitive component controls the learning process in cognitive component by deciding *what-to-learn*, *when-to-learn* and *how-to-learn* through *sample delete strategy*, *sample reserve strategy* and *sample learn strategy*, respectively. The proposed approach is similar to that proposed in [27]. However, the work in [27] is an efficient classifier, but cannot solve regression or approximation problems. In this work the proposed algorithm can be employed to solve regression problems.

The performance of McIT2FIS is evaluated on a benchmark system identification problem, identification of a time-varying system and Mackey-Glass chaotic system identification problem. Performance comparison clearly indicates the improved generalization and noise resistance of proposed PBL-McIT2FIS.

Rest of the paper is arranged as follows. In the next Section II, we describe the architecture of the employed IT2NFIS. In Section III, the meta-cognitive projection based learning algorithm is presented. The proposed systems is evaluated on a set of benchmark regression problems with different levels of added noise, in Section IV. The paper is concluded in Section V.

## II. Meta-Cognitive Interval Type-2 Neuro-Fuzzy Inference System Structure

In this section, the structure and inference mechanism of the Meta-Cognitive Interval Type-2 Neuro-Fuzzy Inference System (McIT2FIS) is presented. Let us assume that the network consists of $m$ input features, $n$ output features and has grown $K$ rules after processing $t-1$ samples. The proposed McIT2FIS, as shown in Fig.1, is a six-layered network which realizes Takagi-Sugeno-Kang Type-0 fuzzy inference mechanism. The detailed output inference mechanism when presented with the $t$-th sample, $\mathbf{x}(t)$, is as follows:

**Layer 1**- *Input layer*: This layer consists of as many nodes as the number of input features ($m$) in the data. The input layer passes the input data to the fuzzification layer directly. The output of $i$-th input node is :

$$u_i(t) = x_i(t) \quad i = 1, 2, \cdots, m. \tag{1}$$

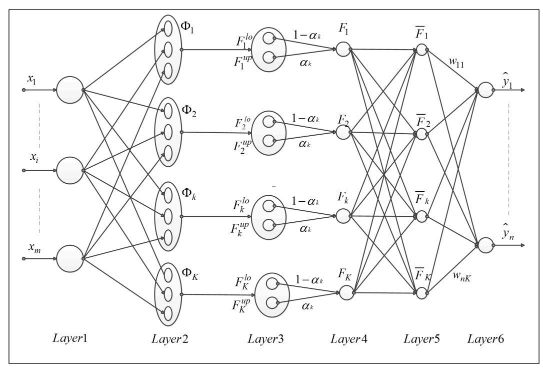


Fig. 1. Structure of a six layered Interval Type-2 Neuro-Fuzzy Inference System

**Layer 2**- *Fuzzification layer*: Each node in this layer calculates the membership of the input data with the rule antecedents by employing the Interval Type-2 Gaussian membership function. The membership of $i$-th input feature with $k$-th rule is given by :

$$\phi_{ki}(t) = \exp\left\{-\frac{(u_i(t) - \mu_{ki})^2}{2\sigma_{ki}^2}\right\} \equiv \phi\left(\mu_{ki}, \sigma_{ki}, u_i(t)\right) \tag{2}$$

where, $\mu_{ki} \in \left[\mu_{ki}^l, \mu_{ki}^r\right]$ and $\sigma_{ki} \in \left[\sigma_{ki}^l, \sigma_{ki}^r\right]$ are left and right limits of the center and width of $k$-th rule's $i$-th feature, respectively.

The footprint of uncertainty of this membership function can be represented as an Interval in terms of upper membership function $\phi^{up}$ and lower membership function $\phi^{lo}$, as given below:

$$\phi_{ki}^{up}(t) = \begin{cases} \phi\left(\mu_{ki}^l, \sigma_{ki}^l, u_i(t)\right) & u_i(t) < \mu_{ki}^l \\ 1 & \mu_{ki}^l \le u_i(t) \le \mu_{ki}^r \\ \phi\left(\mu_{ki}^r, \sigma_{ki}^r, u_i(t)\right) & u_i(t) > \mu_{ki}^r \end{cases} \tag{3}$$

$$\phi_{ki}^{lo}(t) = \begin{cases} \phi\left(\mu_{ki}^{r}, \sigma_{ki}^{r}, u_i(t)\right) & u_i(t) \leq \frac{(\mu_{ki}^{l} + \mu_{ki}^{r})}{2} \\ \phi\left(\mu_{ki}^{l}, \sigma_{ki}^{l}, u_i(t)\right) & u_i(t) > \frac{(\mu_{ki}^{l} + \mu_{ki}^{r})}{2} \end{cases} \quad (4)$$

The output of each node can be represented by the Interval $\Phi_{ki} = [\phi_{ki}^{lo}(t), \phi_{ki}^{up}(t)]$.

**Layer 3**- *Firing layer*: Each node in this layer calculates the firing strength of a rule. This layer consists of $2 \times K$ nodes where each set of $K$ nodes represent the upper and lower firing strength of $K$ rules. The algebraic product operation is applied upon the rule antecedents to calculate the firing strength of a rule and is given by :

$$\left[F_k^{lo}(t), F_k^{up}(t)\right]; \quad k = 1, \cdots, K \quad (5)$$

where

$$F_k^{lo}(t) = \prod_{i=1}^{m} \phi_{ik}^{lo} \ \text{ and } \ F_k^{up}(t) = \prod_{i=1}^{m} \phi_{ik}^{up}; \ k = 1, \cdots, K \quad (6)$$

**Layer 4**- *Type reduction layer*: This layer consists of $K$ nodes. Each node in this layer performs type-reduction of Interval Type-1 fuzzy set to Type-1 fuzzy number. For type-reduction, we employ a variant of Nie-Tan [28] type-reduction procedure as proposed in [27]

$$F_k(t) = \alpha F_k^{lo}(t) + (1 - \alpha) F_k^{up}(t); \ k = 1, \cdots, K \quad (7)$$

where $\alpha$ is the design vector. In our study $\alpha$ is chosen as 0.5.

**Layer 5**- *Normalization layer*: Each node in this layer normalizes the firing strengths of rule that is generated by the type reduction layer. This layer consists of $K$ nodes. The normalized firing strength is given by :

$$\bar{F}_k(t) = \frac{F_k(t)}{\sum_{p=1}^{K} F_p(t)}, \ k = 1, \cdots, K \quad (8)$$

**Layer 6**- *Output layer*: This layer calculates the network output which is weighted sum of the normalized firing strength of the $K$ rules obtained from the previous layer and is given by :

$$\hat{y}_j(t) = \sum_{k=1}^{K} w_{jk} \bar{F}_k(t); \quad j = 1, \cdots, n \quad (9)$$

where, $w_{jk}$ is the output weight connecting $k$-th rule with $j$-th output node.

Next, we describe the Projection based learning algorithm for the proposed network.

## III. Meta-cognitive Projection Based Learning Algorithm for IT2FIS

In this section, we present a fast meta-cognitive projection based learning algorithm for the proposed network. Here the training samples arrive in a sequential manner. Let us assume that we have a set of training data $\{(\mathbf{x}(1), \mathbf{y}(1)), \cdots, (\mathbf{x}(t), \mathbf{y}(t)), \cdots\}$, where, $\mathbf{x}(t) = [x_1(t), \cdots, x_m(t)]^T \in \Re^m$ is the $m$-dimensional input vector of $t$-th sample and $\mathbf{y}(t) = [y_1(t), \cdots, y_n(t)]^T \in \Re^n$ is n-dimensional output vector. The functional relationship between the input and the output ($\mathbf{x} \rightarrow \mathbf{y}$) can be represented by $\mathbf{f}[.]$

$$\mathbf{y} = \mathbf{f}[\mathbf{x}]. \quad (10)$$

The objective of PBL-McIT2FIS is to approximate **f[.]** such that, the predicted output

$$\hat{\mathbf{y}} = \hat{\mathbf{f}}[\mathbf{x}, \mathbf{w}] \quad (11)$$

is as close as possible to the desired target $\mathbf{y}$. Here, $\mathbf{w}$ is the parameter vector of PBL-McIT2FIS.

The difference between the actual and the predicted output for $t$-th sample, error $\mathbf{e}(t) = [e_1(t), \cdots, e_n(t)]$ is defined as,

$$e_j(t) = y_j(t) - \hat{y}_j(t); \quad j = 1, 2, \cdots, n \quad (12)$$

Root Mean Squared Error (RMSE) and spherical potential [29] are the two measures which are used to quantify the difference in knowledge between the current sample and the network. The RMSE of the current prediction at instant $t$ is given as

$$E(t) = \|\mathbf{e}(t)\| \quad (13)$$

Spherical potential [23, 30] quantifies the novelty of the current incoming data sample. It is defined as the average distance of the current sample from existing rules in a hyper-dimensional feature space.

$$\psi(t) = \frac{1}{K} \sum_{k=1}^{K} F_k(t). \quad (14)$$

Here, $K$ represents the number of rules in the neuro-fuzzy inference system.

The meta-cognitive learning algorithm employs RMSE and spherical potential to decide either to delete the sample without learning, or learn the knowledge in the sample or reserve the sample for future use. During sample learning, either a new rule is being added to the network, or network structure and parameters are tuned to accommodate the knowledge in the sample. Next we shall describe the projection based learning algorithm and the three strategies in detail.

### A. Projection Based Learning Algorithm

The projection based learning algorithm estimates the optimal output weight corresponding to the minimum energy of the energy function. In the literature of artificial neural networks, such a projection based learning algorithm was first proposed in [21] and was extended in Type-1 neuro-fuzzy inference systems in [25]. In this work, we extend this idea to Type-2 neuro-fuzzy inference systems. For the $t$-th sample, the energy function is defined as:

$$J_t = \sum_{j=1}^{n} \left(e_j(t)\right)^2 \quad (15)$$

$$= \sum_{j=1}^{n} \left(y_j(t) - \sum_{k=1}^{K} w_{jk} \bar{F}_k(t)\right)^2. \quad (16)$$

Assuming that McIT2FIS has employed $N$ samples until now, the total energy of the system is given as,

$$J(\mathbf{W}) = \frac{1}{2}\sum_{t=1}^{N} J_t \tag{17}$$

The aim of projection based learning algorithm is to estimate the optimal output weights ($\mathbf{W} = \mathbf{W}^*$) such that the total energy of the network is minimized.

The optimal output weights ($\mathbf{W}^*$) is obtained by equating the first order partial derivative of $J(\mathbf{W})$ with respect to output weight to zero. By equating the first order partial derivative to zero and re-arranging, we get,

$$\sum_{k=1}^{K}\sum_{t=1}^{N} \bar{F}_k(t)\bar{F}_p(t)w_{jk} = \sum_{t=1}^{N} \bar{F}_p(t)y_j(t),$$
$$\text{where } p = 1,\cdots,K;\ j = 1,\cdots,n \tag{18}$$

which in turn could be written as

$$\sum_{k=1}^{K} w_{jk}a_{kp} = b_{pj} \ \equiv\ \mathbf{A}\mathbf{w} = \mathbf{B} \tag{19}$$

where the projection matrix $\mathbf{A} \in \mathbb{R}^{K\times K}$ is given by

$$a_{kp} = \sum_{t=1}^{N} \bar{F}_k(t)\bar{F}_p(t),\ k = 1,\cdots,K;\ p = 1,\cdots,K \tag{20}$$

and the output matrix $\mathbf{B} \in \mathbb{R}^{K\times n}$ is

$$b_{pj} = \sum_{t=1}^{N} \bar{F}_p(t)y_j(t),\ p = 1,\cdots,K;\ j = 1,\cdots,n \tag{21}$$

From Eqn. (19), the optimal output weights $\mathbf{W}^*$ could be obtained as

$$\mathbf{W}^* = \mathbf{A}^{-1}\mathbf{B} \tag{22}$$

### *B. Sample Delete Strategy*

A sample is deleted from the network without being learnt if the RMSE for the current sample is below a particular delete threshold $E_d$, which implies that similar knowledge is already present. A sample is deleted if it satisfies the following condition:

$$E(t) \ <\ E_d \tag{23}$$

Here, $E_d$ is the delete threshold, and it is set in the range [0.0001, 0.001].

### *C. Sample Learning Strategy*

In this strategy either a new rule is added to the network or the parameters of the existing rules are updated. We shall now describe each of these sub-strategies.

- **Rule Growing for PBL-McIT2FIS**: If for the current sample, the prediction error is very high and if the knowledge present in the sample is novel to the network, a new rule is added to capture the new knowledge. The rule addition criterion is given as :

$$E(t) > E_a \text{ AND } \psi(t) \ <\ E_S. \tag{24}$$

where, $E_a$ is the add threshold and $E_S$ is the novelty threshold. Lower value of $E_S$ indicates higher resistance to rule addition. In the experiments in this work, this parameter is set in the range [0.4, 0.7]. $E_a$ is the self-adaptive add threshold which is initially set in the range [0.05, 0.1]. When a new rule is added to the network, $E_a$ is the self-adapted as

$$E_a = (1-\delta)E_a + \delta E(t) \tag{25}$$

The aim of self-adaptive threshold is to initially let the network add rules to gain knowledge, and later to fine tune it. Here, $\delta$ is the slope parameter that decides the slope at which $E_a$ increases, and is set close to 0.
When the $K+1$-th fuzzy rule is added to the network the rule center is initialized as,

$$\mu_{K+1} = [\mathbf{x}(t) * 0.9,\ \mathbf{x}(t) * 1.1] \tag{26}$$

The width is assigned as,

$$\sigma^l_{K+1} = \kappa \times \min_{\forall k}\left[\|(\mu^l_{K+1} - \boldsymbol{\mu}^l_k)\|, \|(\mu^l_{K+1} - \boldsymbol{\mu}^r_k)\|\right]; \tag{27}$$

$$\sigma^r_{K+1} = \kappa \times \min_{\forall k}\left[\|(\mu^r_{K+1} - \boldsymbol{\mu}^l_k)\|, \|(\mu^r_{K+1} - \boldsymbol{\mu}^r_k)\|\right]; \tag{28}$$

where, $\kappa$ determines the overlap between the newly added rule and nearest rule. $\kappa$ is chosen in the range [0.5,0.9], so as to induce overlap among rules.

When a new rule is added based on the $t$-th sample, the projection matrix $\mathbf{A}$ is updated as,

$$\mathbf{A}_{(K+1)\times(K+1)} = \begin{bmatrix} \mathbf{A}_{K\times K} + \left(\bar{F}(t)\right)^T \bar{F}(t) & \mathbf{a}^T_{K+1} \\ \mathbf{a}_{K+1} & a_{K+1,K+1} \end{bmatrix} \tag{29}$$

where, $\mathbf{a}_{K+1} \in \mathbb{R}^{1\times K}$ is

$$\mathbf{a}_{K+1,p} = \sum_{t=1}^{N} \bar{F}_{K+1}(t)\bar{F}_p(t),\ p = 1,\cdots,K \tag{30}$$

and

$$a_{K+1,K+1} = \sum_{t=1}^{N} \bar{F}_{K+1}(t)\bar{F}_{K+1}(t).i \tag{31}$$

The output matrix $\mathbf{B}$ is updated as,

$$\mathbf{B}_{(K+1)\times n} = \begin{bmatrix} \mathbf{B}_{K\times n} + \left(\bar{F}(t)\right)^T (\mathbf{y}(t))^T \\ \mathbf{b}_{K+1} \end{bmatrix} \tag{32}$$

and $\mathbf{b}_{K+1} \in \mathbb{R}^{1\times n}$ is given as,

$$b_{K+1,j} = \sum_{t=1}^{N} \bar{F}_{K+1}(t)y_j(t),\ j = 1,\cdots,n \tag{33}$$

The output weights are re-estimated as given in Eqn. (19).

**Parameter Update for PBL-McIT2FIS**: When a sample contains significant knowledge which is not novel, but is less familiar to the network, the output weights are updated. The output weight of the network is updated if the following condition is satisfied:

$$E(t) > E_L \tag{34}$$

Here, $E_L$ is self-adaptive parameter update threshold, which is set in the range [0.3,0.5]. It self-adapts according to the equation

$$E_L = (1-\delta)E_L + \delta E(t). \tag{35}$$

If a sample is used for updating the parameters of the network, the matrices $\mathbf{A} \in \Re^{K\times K}$ and $\mathbf{B} \in \Re^{K\times n}$ are updated as

$$\mathbf{A} = \mathbf{A} + \left(\bar{\mathbf{F}}(t)\right)^T \mathbf{F}(t) \tag{36}$$

$$\mathbf{B} = \mathbf{B} + \left(\bar{\mathbf{F}}(t)\right)^T (\mathbf{y}(t))^T \tag{37}$$

and the output weights are updated according to:

$$\mathbf{W}_K = \mathbf{W}_K + \mathbf{A}^{-1}\left(\bar{\mathbf{F}}(t)\right)^T (\mathbf{e}(t))^T \tag{38}$$

### D. Sample Reserve Strategy

If the current sample does not satisfy sample delete or sample learning strategy, it is reserved to be considered for learning at a different time.

## IV. Performance Evaluation

In the previous section, a meta-cognitive Interval Type-2 neuro-fuzzy inference system and its projection based learning algorithm was presented in details. In this section, the performance of the proposed algorithm is evaluated on non-linear system identification [31], online identification of a time-varying system and Mackey-Glass time series prediction problem [32]

### A. Performance Measures

Root mean square error (RMSE) is employed as a performance measure. RMSE is defined as,

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N} e^2(t)}{N}} \tag{39}$$

where $N$ is the total number of samples, $e(t)$ is the prediction error for the $t$-th sample.

### B. Non-linear System Identification Problem

First, we shall evaluate our algorithm on non-linear system identification problem [31]. The dataset is generated by a difference equation :

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) \tag{40}$$

Here the current output of the system, $y(t+1)$, depends on its past output, $y(t)$, and an input, $u(t) = sin(2\pi t/100)$. The network is trained using 50,000 samples and 200 samples are used for testing. The output, $y(t)$ is in the range $[-1.5, 1.5]$. The parameters for this problem are: meta-cognitive parameters: delete threshold, $E_d$ = 0.0001, add threshold, $E_a$ = 0.1189, parameter update threshold, $E_l$ = 0.0011, novelty threshold, $E_S$ = 0.2647, rule antecedent overlap constant, $\kappa$ = 0.8015.

TABLE I
PERFORMANCE COMPARISON FOR NON-LINEAR SYSTEM IDENTIFICATION PROBLEM

| Algorithm | Number of Rules | RMSE | |
|---|---|---|---|
| | | Train | Test |
| et2FIS | 14 | - | 0.053 |
| PBL-McIT2FIS | 16 | 0.03 | 0.03 |

The result of PBL-McIT2FIS is compared against another Type-2 algorithm et2FIS [17]. Table I shows the number of rules used, training and testing RMSE. From the table we can observe that PBL-McIT2FIS performs better than the existing algorithm.

### C. Online Identification of Time-Varying System

The experiment is performed to demonstrate the ability of PBL-McIT2FIS to handle disturbance in the system with dataset generated as in

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) + f(t) \tag{41}$$

where a disturbance $f(t)$ is introduced into the system, such that $f(t)$ is described as:

$$f(t) = \begin{cases} 0 & 1 \le t \le 1000 \ and\ t \ge 2001 \\ 1 & 1000 \le t \le 2000 \end{cases} \tag{42}$$

TABLE II
ONLINE IDENTIFICATION OF A TIME-VARYING SYSTEM

| Algorithm | Rules | Test RMSE |
|---|---|---|
| eT2FIS | 12 | 0.18 |
| PBL-McIT2FIS | 16 | 0.03 |

The network is trained using 50,000 samples and 200 samples are used for testing. The output, $y(t)$ is in the range $[-1.5, 1.5]$. The parameters for this problem are: meta-cognitive parameters: delete threshold, $E_d$ = 0.0001, add threshold, $E_a$ = 0.1277, parameter update threshold, $E_l$ = 0.0636, novelty threshold, $E_S$ = 0.2092, $\kappa$ = 0.8559. Table II shows the number the rules used, training and testing RMSE. From the table we can observe that even after the presence of disturbance in the training data PBL-McIT2FIS is able to generalize well.

### D. Mackey-Glass Chaotic Time Series Problem

The noise resistance ability of PBL-McIT2FIS is evaluated on the Mackey-Glass time series prediction [32]. This time series data is generated using the differential equation:

$$\frac{dx}{dt} = \frac{0.2x(t-\tau)}{1+x^10(t-\tau)} - 0.1x(t) \tag{43}$$

where $\tau$ = 30 and initial condition $x(0)$ = 1.2. The four past values are used to predict the current value; where input vector is $[x(t-24), x(t-18), x(t-12), x(t-6)]$ and the corresponding output vector is $[x(t)]$. Out of the one thousand samples that were generated 500 were used for training and last 500 were used for testing. The noise resistance ability of the proposed McIT2FIS is studied by performing several experiments. We add noise of Standard Deviation (S.D) 0.1, 0.2 and 0.3 with mean 0 to the generated training data set and noise of S.D 0.1, 0.3 to the test data set.In table III we can see that test RMSE increases with increase in noise levels. However, the increase in the RMSE is not significantly higher for McIT2FIS. It shows that McIT2FIS is able to learn the underlying functional relationship, inspite of varying noise. The use of meta-cognition and Interval Type-2 fuzzy system has helped the network to generalize well.

## V. Conclusion

In this paper, we have presented a computationally fast Interval Type-2 Fuzzy Inference System (IT2FIS) and its meta-cognitive projection based learning algorithm. The meta-cognitive learning algorithm monitors the knowledge in the current sample with respect to the knowledge contained in the network to decide on *what-to-learn*, *when-to-learn* and *how-to-learn*, efficiently. During how-to-learn, either a new rule is added to the network or parameters of the network are updated using projection based learning algorithm, which finds the analytical minima of the total energy function. The performance comparison on a set of benchmark applications indicates the superior performance of McIT2FIS.

## VI. Acknowledgment

The authors would like to thank the Nanyang Technological University- Ministry of Defence, Singapore (Grant No: MINDEF-NTU-JPP/13/02/01) for the financial support to conduct the study.

## References

[1] K. Subramanian, R. Savitha, S. Suresh, Complex-valued neuro-fuzzy inference system for wind prediction, in: Intl. Joint Conf. on Neural Net., 2012, pp. 1 – 7.

[2] K. Subramanian, S. Suresh, Human action recognition using meta-cognitive neuro-fuzzy inference system, International Journal of Neural Systems 22 (6) (2012) 1250028 (15).

[3] E. Kayacan, Y. Oniz, A.-C. Aras, O. Kaynak, R. Abiyev, A servo system control with time varying and nonlinear load conditions using type-2 tsk fuzzy neural system, Applied Soft Computing 11 (8) (2011) 5735 – 5744.

[4] Q. Song, N. Kasabov, Dynamic evolving neuro-fuzzy inference system DENFIS: Online learning and application for time-series prediction, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 144 – 154.

[5] P. Angelov, P. Filev, An approach to online identification of Takagi-Sugeno fuzzy models, IEEE Transactions on System, Man and Cybernetics, Part B: Cybernetics 34 (1) (2004) 484 – 498.

[6] H. Rong, N. Sundararajan, G. Huang, P. Saratchandran, Sequential adaptive fuzzy inference system SAFIS for nonlinear system identification and prediction, Fuzzy Sets and Systems 157 (9) (2006) 1260 – 1275.

[7] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Information Sciences 8 (1975) 199 – 249.

[8] N. N. Karnik, J. M. Mendel, Type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 7 (6) (1999) 643 – 658.

[9] J. M. Mendel, R. I. B. John, Type-2 fuzzy sets made simple, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 117 – 127.

[10] Q. Liang, J. Mendel, Interval type-2 fuzzy logic systems: Theory and design, IEEE Trans. on Fuzzy Systems 8 (5) (2000) 535 – 550.

[11] C. F. Juang, A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning, IEEE Transactions on Fuzzy Systems 16 (6) (2008) 1411 – 1424.

[12] J. R. Castro, O. Castillo, P. Melin, A. Rodriguez-Diaz, A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks, Information Sciences 179 (13) (2009) 2175 – 2193.

[13] C. F. Juang, An interval type-2 fuzzy neural network with support vector regerssion for noisy regression problems, IEEE Transactions on Fuzzy Systems 18 (4) (2010) 686 – 699.

[14] J. Y. Chang, Y. Y. Lin, M. F. Han, C. T. Lin, A functional-link based interval type-2 compensatory fuzzy neural network for nonlinear system modeling, IEEE International Conference on Fuzzy Systems (2011) 939 – 943.

[15] R. H. Abiyev, O. Kaynak, T. Alshanableh, F. Mamedov, A type-2 neuro-fuzzy system based on clustering and gradient techniques applied to system identification and channel equalization, Applied Soft Computing 11 (1) (2011) 1396 – 1406.

[16] J. Tavoosi, M. A. Badamchizadeh, A class of type-2 fuzzy neural networks for nonlinear dynamical system identification, Neural Computation and Application 23 (3 – 4) (2012) 707 – 717.

[17] S. W. Tung, C. Quek, C. Guan, eT2FIS: An evolving type-2 neural fuzzy inference system, Information Sciences 220 (1) (2013) 124 – 148.

[18] Y. Y. Lin, S. H. Liao, J.-Y. Chang, Simplified Interval Type-2 Fuzzy Neural Networks, IEEE Transactions on Neural networks and Learning Systems-doi:10.1109/TNNLS.2013.2284603.

[19] S. Suresh, K. Dong, H. Kim, A sequential learning algorithm for self-adaptive resource allocation network classifier, Neurocomputing 73 (16 – 18) (2010) 3012 – 3019.

TABLE III
MACKEY-GLASS SERIES WITH MCIT2FIS

| Noise Level | | Testing |
|---|---|---|
| in Training Data | in Testing Data | RMSE |
| 0.1 | clean | 0.04 |
| | 0.1 | 0.06 |
| | 0.3 | 0.21 |
| 0.2 | clean | 0.08 |
| | 0.1 | 0.08 |
| | 0.3 | 0.16 |
| 0.3 | clean | 0.09 |
| | 0.1 | 0.10 |
| | 0.3 | 0.10 |

TABLE IV
PERFORMANCE COMPARISON ON MACKEY-GLASS SERIES

| Algorithm | Testing RMSE | Testing RMSE | Testing RMSE |
|---|---|---|---|
| Noise Level with S.D | 0.1 | 0.2 | 0.3 |
| eT2FIS | 0.127 | 0.154 | 0.177 |
| PBL-McIT2FIS | 0.104 | 0.106 | 0.09 |


[20] G. S. Babu, S. Suresh, Meta-cognitive neural network for classification problems in a sequential learning framework, Neurocomputing 81 (1) (2012) 86 – 96.
[21] G. S. Babu, S. Suresh, Meta-cognitive RBF network and its projection based learning algorithm for classification problems, Applied Soft Computing 13 (1) (2013) 654 – 666.
[22] K. Subramanian, S. Suresh, R. Venkatesh Babu, Meta-cognitive neuro-fuzzy inference system for human emotion recognition, in: Intl. Joint Conf. on Neural Net., 2012, pp. 1 – 7.
[23] K. Subramanian, S. Suresh, N. Sundararajan, A meta-cognitive neuro-fuzzy inference system (McFIS) for sequential classification problems, IEEE Transactions on Fuzzy Systems 21 (6) (2013) 1080 – 1095.
[24] G. S. Babu, S. Suresh, Sequential projection based metacognitive learning in a radial basis function network for classification problems, IEEE Trans. Neural Networks and Learning Systems 24 (2) (2013) 194 – 206.
[25] K. Subramanian, S. Suresh, A projection based learning algorithm for meta-cognitive neuro-fuzzy inference system, in: IEEE Intl. Conf. on Fuzzy Systems, 2013, pp. 1 – 8.
[26] K. Subramanian, R. Savitha, S. Suresh, A meta-cognitive interval type-2 fuzzy inference system classifier and its projection based learning algorithm, in: Evolving and Adaptive Intelligent Systems (EAIS), 2013 IEEE Conference on, IEEE, 2013, pp. 48–55.
[27] K. Subramanian, A. K. Das, S. Sundaram, S. Ramasamy, A meta-cognitive interval type-2 fuzzy inference system and its projection based learning algorithm, Evolving Systems (2013) 1–12.
[28] M. Nie, W. W. Tan, Towards an efficient type-reduction method for interval type-2 fuzzy logic systems, in: IEEE Intl. Conf. on Fuzzy Systems, 2008, pp. 1425 – 1432.
[29] H. Hoffmann, Kernel PCA for novelty detection, Pattern Recognition 40 (3) (2007) 863–874.
[30] K. Subramanian, S. Suresh, A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system, Applied Soft Computing 12 (11) (2012) 3603 – 3614.
[31] C. F. Juang, C. T. Lin, An online self-constructing neural fuzzy inference network and its applicatons, IEEE Transactions on Fuzzy Systems 6 (1) (1998) 12 – 32.
[32] R.-S. Croder, Prediction the Mackey-Glass time series with cascase correlation learning, in: Proc. connectionist models summer school, Carnegic Mellon University, 1990, pp. 117 – 123.