

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

eT2FIS: An Evolving Type-2 Neural Fuzzy Inference System

S.W. Tung, C. Quek*, C. Guan

Center for Computational Intelligence,¹ Block N4 #2A-32, School of Computer Engineering, Nanyang Technological University,
Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:

Available online 27 March 2012

Keywords:

Evolving system
Online system
Neural fuzzy system
Incremental sequential learning
Type-2 fuzzy system
Mamdani system

ABSTRACT

There are two main approaches to design a neural fuzzy system; namely, through expert knowledge, and through numerical data. While the computational structure of a system is manually crafted by human experts in the former case, self-organizing neural fuzzy systems that are able to automatically extract generalized knowledge from batches of numerical training data are proposed for the latter. Nevertheless, both of these approaches are static where only parameters of a system are updated during training. On the other hand, the demands and complexities of real-life applications often require a neural fuzzy system to adapt *both* its parameters and structure to model the changing dynamics of the environment. To counter these modeling bottlenecks, intense research efforts are subsequently channeled into the studies of evolving/online neural fuzzy systems. There are generally two classes of evolving neural fuzzy systems: the Takagi–Sugeno–Kang (TSK) systems and the Mamdani systems. While most existing literature consists of evolving Type-1 TSK-typed and Type-1 Mamdani-typed models, they may not perform well in noisy environment. To improve the robustness of these neural fuzzy systems, recent efforts have been directed to extend evolving Type-1 TSK-typed neural fuzzy systems to Type-2 models because of their better known noise resistance abilities. In contrast, minimum similar effort has been made for evolving Mamdani-typed models. In this paper, we present a novel evolving Type-2 Mamdani-typed neural fuzzy system to bridge this gap. The proposed system is named *evolving Type-2 neural fuzzy inference system* (eT2FIS), and it employs a data-driven incremental learning scheme. Issues involving the online sequential learning of the eT2FIS model are carefully examined. A new rule is created when a newly arrived data is novel to the present knowledge encrypted; and an obsolete rule is deleted when it is no longer relevant to the current environment. Highly over-lapping fuzzy labels in the input–output spaces are merged to reduce the computational complexity and improve the overall interpretability of the system. By combining these three operations, eT2FIS is ensured a compact and up-to-date fuzzy rule base that is able to model the current underlying dynamics of the environment. Subsequently, the proposed eT2FIS model is employed in a series of benchmark and real-world applications to demonstrate its efficiency as an evolving neural fuzzy system, and encouraging performances have been achieved.

© 2012 Published by Elsevier Inc.

¹ Formerly Intelligent Systems Laboratory.

* Corresponding author. Tel.: +65 6790 4926; fax: +65 6792 6559.

E-mail addresses: swtung@pmail.ntu.edu.sg (S.W. Tung), ashcquek@ntu.edu.sg (C. Quek), ctguan@i2r.a-star.edu.sg (C. Guan).

1. Introduction

There are two main issues when performing structure learning in a neural fuzzy system; namely, the fuzzy partitioning of the input and output spaces, and the rule generation scheme adopted. Fuzzy partitioning determines the numbers, the positions and the spreads of the fuzzy labels in the input–output spaces; while the rule generation scheme determines the set of fuzzy rule base governing the neural fuzzy system. Existing neural fuzzy systems can then be classified into two types. That is, design by expert knowledge, and design from data. In the former approach, the computational structure of a neural fuzzy system is manually crafted by human experts. Prior knowledge regarding the fuzzy partitioning and the fuzzy rule base is determined by the human experts. Some examples of such neural fuzzy systems include the ANFIS [15] and the GARIC [5] models. Self-organizing neural fuzzy systems, which are based on the latter approach, adopt a more automated design using numerical data. By integrating self-organizing numerical methods [7,6,25,50,52,44] into the learning mechanisms, knowledge is automatically extracted from batches of raw numerical data to determine the positions and the spreads of the fuzzy labels. Self-automated rule generation schemes [50,38,36,34] have also been proposed in the literature. Despite the diverse philosophies behind the learning principles of these systems, they all embrace a uniform assumption that the dynamics of an application is non-changing over time. Subsequently, only the parameters of the systems are fine-tuned during training after the computational structures have been established. This, unfortunately, restricts the usefulness of these neural fuzzy systems to a static environment. On the other hand, real-life applications with time-varying dynamics can range from financial trading instruments [46,42], to assistive biomedical instruments [49,40], to physical phenomena [28,14]. Hence, it is pragmatic to explore alternative neural fuzzy systems with incremental evolutionary structure and parameters to model such non-stationaries.

Evolving neural fuzzy systems, with their online learning abilities, have been developed to address the issue of time-varying application environment (see Table 1). They adopt a data-centric incremental learning mode² where data is sequentially arriving. Structure and parameter learning of the systems are then performed based only on the current data sample. This allows an evolving neural fuzzy system to incorporate new knowledge that emerges after the system has identified its computational structure. It also provides the system with a life-long learning cycle to detect temporal shifts in the data patterns. Through this incremental processing of data, evolving neural fuzzy systems address the stability–plasticity trade-off [11] of a learning system. This means, while a progressive learning of new knowledge modifies an initial trained system (i.e. a process known as *plasticity*, that allows the continuous learning of new knowledge through changes to the structure and parameters of the initial trained system), the system is able to avoid catastrophically erasing existing knowledge that is still relevant (i.e. a process known as *stability*, that allows the system to recall previously learnt knowledge encoded in the structure and parameters of the initial trained system).³ As stated by Abraham and Robins [1], a dynamic learning system can retain memory and minimize memory losses through “a regulated balance between stability and plasticity to solve the trade-off between the stability required to retain information and the plasticity required for new learning”.

There are generally two classes of evolving neural fuzzy systems: the Takagi–Sugeno–Kang (TSK) systems and the Mamdani systems. Table 1 presents a list of some existing works in the literature. As seen, most existing works consist of evolving Type-1 TSK-typed and Type-1 Mamdani-typed neural fuzzy systems, where the fuzzy labels in the antecedent/consequent parts of the systems are Type-1 fuzzy sets. This may result in an unsatisfactory performance when modeling is performed in a noisy environment [8]. On the other hand, Type-2 systems are extensions of Type-1 systems where the membership grades of the fuzzy labels are Type-1 fuzzy sets [21,22,26]. With an additional degree of knowledge incorporated in the systems (analogous to the information provided by variance to a mean value in probability theory), Type-2 neural fuzzy systems appear more promising for handling uncertainties present in noisy environment [32]. As such, recent efforts have been directed to extend evolving Type-1 TSK-typed neural fuzzy systems to Type-2 models. In contrast, there is minimum parallel effort for evolving Mamdani-typed neural fuzzy systems. Although the T2SONFS [18] model has been listed as an evolving Type-2 Mamdani-typed neural fuzzy system in Table 1 because of its sequential learning, it is not a fully online system since the inputs have to be normalized prior to the design of the system. In addition, existing models of evolving Type-2 systems encounter one or more of the following major problems: (1) absence of rule pruning mechanism; and (2) lack of merger operation. A system without a rule pruning mechanism will continuously learn new rules without the removal of irrelevant rules, thus resulting in a complex and ever-expanding rule base; while a system without a merger operation might result in a highly-overlapping partitioning, thus leading to a degradation in the level of interpretability⁴ of the system.

This paper presents the *evolving Type-2 neural fuzzy inference system* (eT2FIS), a novel evolving Type-2 Mamdani-typed neural fuzzy system, which addresses the abovementioned deficiencies faced by existing systems. An incremental sequential learning scheme is employed for the structure and parameter learning of the eT2FIS model. Through its carefully crafted

² Static neural fuzzy systems adopt a batched learning mode, where the structures of the systems are fixed, and learning proceeds with cycling through a collected set of observed data a number of epochs to fine-tune the parameters.

³ On the other hand, static neural fuzzy systems have fixed structures, with parameter fine-tuning abilities. Subsequently, when new training data are presented to a system, re-training is needed to construct a new system with the updated set of training data. This, unfortunately, leads to an erase of knowledge from the initial trained system.

⁴ Online learning typically results in systems that become order-dependent during training [4]. Here, interpretability of a neural fuzzy system adopts the two conditions mentioned in [12,10]; namely, (1) the fuzzy partition must be readable/distinguishable in the sense that the fuzzy sets can be interpreted as linguistic labels; and (2) the set of rules must be compact and consistent with good generalization capability.

Table 1

A summary of evolving neural fuzzy systems in the literature.

	TSK systems (publication year)		Mamdani systems (publication year)
Type-1	SONFIN (1998) [16] GD-FNN (2001) [51] eTS (2004) [3]	NeuroFAST (2001) [48] DENFIS (2002) [24] SAFIS (2006) [39]	FALCON-ART (1997) [27] EFuNN (2001) [23] eFSM (2010) [47]
Type-2	SEIT2FNN (2008) [17] IT2FNN-SVM (2010) [20]	ORGQACO (2009) [19]	T2SONFS (2008) [18]

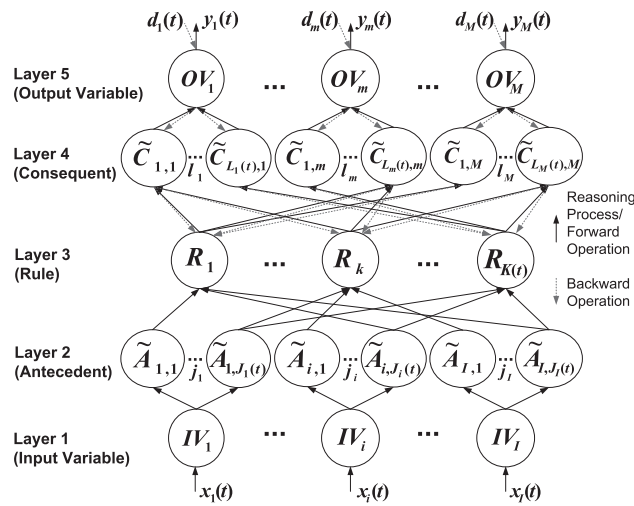


Fig. 1. Computational structure of the evolving Type-2 neural fuzzy inference system (eT2FIS).

learning mechanism, eT2FIS is able to transform low-level raw numerical data to high-level human interpretable fuzzy rules. The set of rule base governing the computational structure is of Type-2 IF-THEN Mamdani-typed fuzzy rules [30]. There are three main operations in the learning of eT2FIS; namely, (1) generation of new fuzzy rules; (2) deletion of obsolete rules; and (3) merger of highly over-lapping fuzzy labels in the input–output spaces. A new rule is created when a newly arrived data is novel to the present knowledge encrypted; and an obsolete rule is deleted when it is no longer relevant to the current environment. This addition-removal approach maintains an up-to-date fuzzy rule base in eT2FIS for the modeling of current dynamics of the environment, where a regulated balance is maintained such that old knowledge (in the system) and new knowledge (from an incoming training data) is able to co-define the structure of the model. This addresses the stability-plasticity trade-off of the system. Highly over-lapping fuzzy labels in the input–output spaces are merged to maintain a compact rule base in eT2FIS.

The rest of the paper is organized as follows. The computational structure, reasoning process and neural computations of the eT2FIS model are described in Section 2. The proposed learning mechanisms of eT2FIS are introduced in Section 3. Section 4 evaluates the learning and adaptation abilities of eT2FIS through a series of experimental simulations. Lastly, Section 5 concludes the paper.

2. eT2FIS: Evolving Type-2 Neural Fuzzy Inference System

This section describes the computational structure, the reasoning process and the neural computations of the proposed *evolving Type-2 neural fuzzy inference system* (eT2FIS).

2.1. Computational structure of eT2FIS

The proposed eT2FIS model is a five layers neural fuzzy system as shown in Fig. 1. Layer 1 consists of the input variable nodes; layer 2 consists of the antecedent nodes; layer 3 is the rule nodes; layer 4 is the consequent nodes; and layer 5 consists of the output variable nodes. In its initial form, there are no fuzzy partitioning or fuzzy rules in the system, i.e., there are no nodes in the hidden layers 2–4. Learning for eT2FIS is performed incrementally where $X(t) = [x_1(t), \dots, x_i(t), \dots, x_I(t)]^T$ is the input vector at a time step t . The corresponding desired output vector is denoted as $D(t) = [d_1(t), \dots, d_m(t), \dots, d_M(t)]^T$, and the computed output vector is denoted as $Y(t) = [y_1(t), \dots, y_m(t), \dots, y_M(t)]^T$. The notations used in Fig. 1 are defined as follows:

I	number of input dimensions;
M	number of output dimensions;
IV_i	i th input variable node;
OV_m	m th output variable node;
$J_i(t)$	number of fuzzy clusters in IV_i at time t ;
$L_m(t)$	number of fuzzy clusters in OV_m at time t ;
\tilde{A}_{i,j_i}	j_i th fuzzy cluster in IV_i ;
$\tilde{C}_{l_m,m}$	l_m th fuzzy cluster in OV_m ;
$K(t)$	number of fuzzy rules at time t ;
R_k	k th fuzzy rule.

Layer 3 of eT2FIS encrypts the rule base of the system where each rule node encodes a Type-2 IF-THEN Mamdani-typed fuzzy rule given as in (1):

$$R_k : \text{IF } x_1 \text{ is } \tilde{A}_{1,j_1}^{(k)} \text{ and } \dots \text{ and } x_i \text{ is } \tilde{A}_{i,j_i}^{(k)} \text{ and } \dots \text{ and } x_l \text{ is } \tilde{A}_{l,j_l}^{(k)} \\ \text{THEN } y_1 \text{ is } \tilde{C}_{1,1}^{(k)} \text{ and } \dots \text{ and } y_m \text{ is } \tilde{C}_{l_m,m}^{(k)} \text{ and } \dots \text{ and } y_M \text{ is } \tilde{C}_{M,M}^{(k)} \quad (1)$$

where $\tilde{A}_{i,j_i}^{(k)}$ (resp. $\tilde{C}_{l_m,m}^{(k)}$) is the j_i th antecedent (resp. l_m th consequent) node associated with the i th input (resp. m th output) variable that is connected to the rule node R_k . The fuzzy label in each antecedent/consequent node is an interval Type-2 membership function (see Fig. 2) whose footprint of uncertainty (FOU) [33] is defined as in (2):

$$\mu(x) = [\underline{\mu}(x), \bar{\mu}(x)] \quad (2)$$

where $\underline{\mu}(x)$ and $\bar{\mu}(x)$ are the lower and upper membership functions respectively. They are defined accordingly as in (3):

$$\underline{\mu}(x) = \begin{cases} \mu^R(c^R, \sigma; x) & \text{if } x \leq 1/2 (c^L + c^R) \\ \mu^L(c^L, \sigma; x) & \text{otherwise} \end{cases}; \quad \bar{\mu}(x) = \begin{cases} \mu^L(c^L, \sigma; x) & \text{if } x \leq c^L \\ 1 & \text{if } c^L < x \leq c^R \\ \mu^R(c^R, \sigma; x) & \text{if } x > c^R \end{cases} \quad (3)$$

where $\mu^L(c^L, \sigma; x)$ and $\mu^R(c^R, \sigma; x)$ are the left and right formation gaussian functions respectively. The gaussian function is defined as: $\mu(c, \sigma; x) = e^{-(x-c)^2/\sigma^2}$ such that c is the center and σ is the width of the function. Adaptation of the centers is performed using neural-network-based gradient descent approach [13,32] (see Appendix B); while the widths vary incrementally as eT2FIS performs learning (see Section 3.2).

2.2. Reasoning process of eT2FIS

As seen from Fig. 1, the reasoning process of the eT2FIS model is represented by solid arrows where the input vector $X(t)$ is presented to the system at layer 1. The proposed system then performs inference based on the input vector by propagating the information through layers 2–4. Consequently, the system produces a computed output vector $Y(t)$ at layer 5. The details on the reasoning process of eT2FIS are discussed here.

The generic operations for the reasoning process of eT2FIS are defined as follows: the forward activation functions of each layer $P \in \{1, \dots, 5\}$ are denoted as $fa^{(P)}$, and the corresponding forward output for an arbitrary node is denoted as fo .

Layer 1: The function of the input nodes is to directly pass on the input vector to the next layer. Hence, the neural operations of IV_i can be described as in (4):

$$fo_i = fa^{(1)}(x_i(t)) = x_i(t) \quad (4)$$

Layer 2: The fuzzy labels \tilde{A}_{i,j_i} define the antecedent segment of the Type-2 Mamdani-typed fuzzy rule described as in (1), where each fuzzy label is defined as an interval Type-2 gaussian function described as in (2). The function of layer 2 of eT2FIS

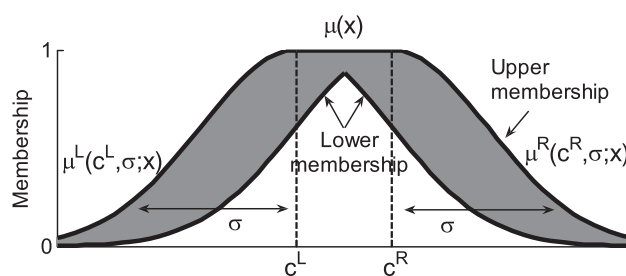


Fig. 2. An interval Type-2 membership function in the antecedent/consequent node.

is to perform similarity matching of the input value with the respective input labels. Hence, the neural computations of \tilde{A}_{i_j} can be described as in (5):

$$fo_{i_j} = fa^{(2)}(fo_i) = [f_{i_j}, \bar{f}_{i_j}] \tag{5}$$

where the bounds of the interval Type-1 output set are computed as in (6):

$$f_{i_j} = \underline{\mu}_{i_j}(x_i); \quad \bar{f}_{i_j} = \bar{\mu}_{i_j}(x_i) \tag{6}$$

such that $\mu_{i_j}(x) = [\underline{\mu}_{i_j}(x), \bar{\mu}_{i_j}(x)]$ refers to the Type-2 membership function embedded in \tilde{A}_{i_j} . For simplicity, the time index t has been dropped for the remaining parts of the reasoning process.

Layer 3: The set of Mamdani-typed rules that is induced incrementally from the numerical data is defined in the rule layer of the eT2FIS model. Each rule node R_k computes the overall degree of similarity between the input vector and the antecedent part of the k th fuzzy rule. Hence, the firing rate of R_k is computed as in (7), where

$$fo_k = fa^{(3)}(fo_{i_{j_1}}^{(k)}, \dots, fo_{i_{j_l}}^{(k)}) = [f_k, \bar{f}_k] \tag{7}$$

The bounds of the interval Type-1 output set are computed as in (8):

$$f_k = \min_{i \in \{1, \dots, l\}} f_{i_{j_i}}^{(k)}; \quad \bar{f}_k = \min_{i \in \{1, \dots, l\}} \bar{f}_{i_{j_i}}^{(k)} \tag{8}$$

where $fo_{i_{j_i}}^{(k)} = [f_{i_{j_i}}^{(k)}, \bar{f}_{i_{j_i}}^{(k)}]$ is the computed output of $\tilde{A}_{i_{j_i}}^{(k)}$ described as in (5).

Layer 4: This layer of eT2FIS consists of the fuzzy labels $\tilde{C}_{l_m, m}$ that define the consequent segments of the Mamdani-typed fuzzy rules in the system. The function of layer 4 of the system is to perform consequent derivation for the fuzzy rules based on the information from the current input vector. Since $\tilde{C}_{l_m, m}$ may serve as output to more than one fuzzy rule, the cumulative neural computations of $\tilde{C}_{l_m, m}$ can be described as in (9):

$$fo_{l_m, m} = fa^{(4)}(fo_{i_{j_1}}^{(l_m, m)}, \dots, fo_{i_{j_{K_{l_m, m}}}}^{(l_m, m)}) = [f_{l_m, m}, \bar{f}_{l_m, m}] \tag{9}$$

where $K_{l_m, m}$ is the total number of fuzzy rules in eT2FIS that shares the same consequent node $\tilde{C}_{l_m, m}$. The bounds of the interval Type-1 output set are computed accordingly as in (10):

$$f_{l_m, m} = \max_{k \in \{1, \dots, K_{l_m, m}\}} f_k^{(l_m, m)}; \quad \bar{f}_{l_m, m} = \max_{k \in \{1, \dots, K_{l_m, m}\}} \bar{f}_k^{(l_m, m)} \tag{10}$$

such that $fo_k^{(l_m, m)} = [f_k^{(l_m, m)}, \bar{f}_k^{(l_m, m)}]$ is the output of the k th fuzzy rule that shares $\tilde{C}_{l_m, m}$.

Layer 5: The output nodes perform a two-steps computation to obtain a crisp output value: type-reduction, followed by defuzzification. A modified height-type-reduction [9] is adopted in the eT2FIS model, and it is computed using Karnik–Mendel (KM) iterative algorithm [22]. Hence, the neural computations of OV_m can be described as in (11):

$$fo_m = fa^{(5)}(fo_{i_{j_1}}, \dots, fo_{l_m, m}) = y_m \tag{11}$$

where $y_m = 1/2(\underline{Y}_m + \bar{Y}_m)$ is the defuzzied value of the type-reduced set $Y_m = [\underline{Y}_m, \bar{Y}_m]$. The type-reduced set is defined as: $\int \dots \int \rho_{l_m} 1 / (\sum_{l_m} \theta_{l_m} \rho_{l_m} / \sum_{l_m} \rho_{l_m})$ such that θ_{l_m} is the midpoint of $\tilde{C}_{l_m, m}$ and $\rho_{l_m} \in fo_{l_m, m}$. Please refer to Appendix A for a detailed implementation of the KM algorithm for determining the type-reduced set Y_m .

2.3. Neural computations of eT2FIS

The neural computations defined in the proposed eT2FIS model are bi-directional in the forward and backward sense. The neural computations of eT2FIS are discussed here.

Forward Operation: As seen from Fig. 1, the forward operation of eT2FIS coincides with its reasoning path. In particular, the forward operation is defined as the neural computations of layers 1 to 3 of the system.

Backward Operation: The backward operation of eT2FIS is represented by dotted arrows in Fig. 1, where the desired output vector $D(t)$ is presented to the system in layer 5. The proposed system then passes the information towards the rule layer. The backward operation is a mirrored computation of the forward operation as discussed below.

The generic operations for the backward operation of eT2FIS are defined as follows: the backward activation functions of each layer $P \in \{3, \dots, 5\}$ are denoted as $ba^{(P)}$, and the corresponding backward output for an arbitrary node is denoted as bo .

Layer 5: The output nodes directly pass on the output vector to the next layer such that the neural operations of OV_m can be described as in (12):

$$bo_m = ba^{(5)}(d_m(t)) = d_m(t) \tag{12}$$

Layer 4: The consequent nodes perform similarity matching of the desired output values with the respective output labels such that the neural computations of $\bar{C}_{l_m,m}$ can be described as in (13):

$$bo_{l_m,m} = ba^{(4)}(bo_m) = [\underline{b}_{l_m,m}, \bar{b}_{l_m,m}] \quad (13)$$

where $\underline{b}_{l_m,m} = \mu_{l_m,m}(d_m)$ and $\bar{b}_{l_m,m} = \bar{\mu}_{l_m,m}(d_m)$.

Layer 3: A rule node computes the overall degree of similarity between the desired output vector and the consequent part of the k th fuzzy rule such that the neural computations of R_k are described as in (14):

$$bo_k = ba^{(3)}(bo_{l_1,1}, \dots, bo_{l_M,M}) = [\underline{b}_k, \bar{b}_k] \quad (14)$$

where $\underline{b}_k = \min_{m \in \{1, \dots, M\}} \underline{b}_{l_m,m}^{(k)}$ and $\bar{b}_k = \min_{m \in \{1, \dots, M\}} \bar{b}_{l_m,m}^{(k)}$.

The forward-and-backward neural computations are defined to compute the activation levels of the fuzzy rules in eT2FIS when a data pair $[X(t); D(t)]$ is presented. The objective is two-folds; namely, (1) to establish the *certainty factors* of the fuzzy rules, and (2) to determine the creation of a new rule (see Section 3.2). Certainty factor of a fuzzy rule reflects the modeling potential of the rule for the current environment, and it is defined as in (15):

$$\zeta_k(t) = \max[age_k(t), act_k(t)]; \quad \zeta_k(0) = 1 \quad (15)$$

where the certainty factor ζ of a newly created rule k is initialized to unity. This means that a newly created rule in eT2FIS is assumed the greatest modeling potential since $0 < \zeta \leq 1$. Subsequently, the computation of ζ consists of two parts: the forgetting and the enhancement components. They are computed respectively as in (16):

$$age_k(t) = \eta_k \zeta_k(t-1); \quad act_k(t) = \min[F(X), B(D)] \quad (16)$$

where η_k is a decaying constant; while $F(X) = 1/2 (f_k + \bar{f}_k)$ and $B(D) = 1/2 (\underline{b}_k + \bar{b}_k)$ are the forward and backward activations of R_k respectively. The *age* component mimics the biological phenomenon of *long-term depression* (LTD) [31]. Intuitively, information from $[X(t); D(t)]$ can be retained by a one-to-one retention of a fixed synaptic configuration encoding the data pair. However, a more realistic approach is to consider passive decomposition of the configuration because the relevance of a particular data pair is temporal under a dynamically changing environment. On the other hand, the *act* component mimics the biological phenomenon of *long-term potentiation* (LTP) [31]. That means, information storage in the network of a human brain often outlasts its initial synaptic configurations because long-term memory of knowledge can be enhanced through repeated rehearsals of related information. Hence, the modeling potential of a forgotten fuzzy rule can be enhanced via a repeated rehearsal of related information that first elicited its formation.

Through an incremental update of the certainty factors, eT2FIS is ensured a current and up-to-date rule base that is able to model the current underlying dynamics of the environment. If a fuzzy rule is able to generalize the recent encountered set of numerical data, the dominating factor in the computation of its certainty factor is the enhancement component. This, subsequently, maintains a high value for the computed certainty factor, thus enhancing the presence of the rule in the fuzzy rule base. On the other hand, the forgetting mechanism kicks in if a fuzzy rule fails to give a satisfactory generalization of the recent set of encountered numerical data. Subsequently, the modeling potential of the rule decreases over time until the rule becomes irrelevant to the environment or it gets restored through a rehearsal episode.

3. Learning mechanism in eT2FIS

Initialization is first performed when there are no existing fuzzy rules in the proposed eT2FIS model. An incremental learning scheme is subsequently employed for the structure and parameter learning of eT2FIS where the system learns and evolves with the arrival of each new data pair. There are three key operations in the structure learning of eT2FIS: (1) generation of new fuzzy rules, (2) deletion of obsolete rules, and (3) merger of highly over-lapping fuzzy labels in the input-output spaces. Following that, parameter learning is performed in eT2FIS using a gradient descent approach. The computational structure of the eT2FIS model is then established, and the system can be employed to model an application or further training is performed. Details on the structure learning mechanism of eT2FIS are presented in this section, and the parameter learning mechanism is presented in Appendix B.

3.1. Initialization

Prior to the commencement of learning, there are no nodes in the hidden layers 2–4 of the proposed eT2FIS model. Learning of the system begins with extracting and utilizing knowledge from the first incoming data pair $[X(0); D(0)]$ to establish an initial structure of the eT2FIS model. Subsequently, the formation of a first fuzzy cluster $\bar{A}_{i,1}$ in an input dimension i can be described using its FOU as: $\mu_{i,1}(x) = [\underline{\mu}_{i,1}(x), \bar{\mu}_{i,1}(x)]$ where the left and right formation gaussian functions can be described accordingly as in (17):

$$c_{i,1}^L = x_i - \Delta; \quad c_{i,1}^R = x_i + \Delta; \quad \sigma_{i,1} = \sigma_0 \quad (17)$$

such that Δ defines a small perturbation and the width is initialized as σ_0 . The same fuzzy clustering process is performed for each output dimension m to formulate a first fuzzy cluster $\tilde{C}_{1,m}$. In addition, a new fuzzy rule R_1 is formulated to encode the knowledge from $[X(0); D(0)]$ where $\{\tilde{A}_{i,1}\}_{i=1}^I$ and $\{\tilde{C}_{1,m}\}_{m=1}^M$ form the antecedent and consequent segments of R_1 respectively.

3.2. Rule generation

The proposed eT2FIS model proceeds with the generation of a new fuzzy rule when a data pair $[X(t); D(t)]$ is presented to a non-empty system at a time t . The activation levels of the fuzzy rules are computed using (16). If an existing fuzzy rule is able to represent the presented data, the system proceeds with the update of the certainty factors for the fuzzy rules as in (15). A fuzzy rule is deemed as being able to represent the presented data if the computed activation level of the rule exceeds a rule generation threshold Δ , i.e., $act_k > \Delta$.

On the other hand, a new fuzzy rule is created if existing rules in eT2FIS fail to provide a representation of the presented data, i.e., $act_k \leq \Delta$ for all k . The eT2FIS model proceeds to identify the best matched fuzzy clusters in the input dimensions via the computed similarity values. Deriving from (5), the similarity value between an input value x_i and an existing cluster in the i th input dimension is given as $SV(x_i, \tilde{A}_{i,j_i}) = 1/2 (f_{i,j_i}^L + f_{i,j_i}^R)$. Subsequently, the best matched fuzzy cluster in an input dimension i is denoted as \tilde{A}_{i,j_i^*} where $j_i^* = \arg \max_{j_i} SV(x_i, \tilde{A}_{i,j_i})$. A new fuzzy rule $R_{K(t+1)}$, where $K(t+1) = K(t) + 1$, is formulated to encode the knowledge from $[X(t); D(t)]$ such that each identified best matched fuzzy cluster can be categorized into three operations as follows:

1. *No action is required for the best matched fuzzy cluster and it is defined as part of the antecedent segment of $R_{K(t+1)}$.* This scenario occurs when the similarity value between the presented value and the best matched fuzzy cluster is greater than Δ . That means, the best matched fuzzy cluster is able to represent the presented value.
2. *Modify the best matched fuzzy cluster and the modified cluster is defined as part of the antecedent segment of $R_{K(t+1)}$.* This scenario occurs when the computed similarity value is less than Δ , but an updated similarity value between the presented value and the modified best matched fuzzy cluster is greater than Δ . That means, the modified best matched fuzzy cluster is able to represent the presented value although it is initially unable to prior to an update. There are two types of modifications: (a) *Increase the spread of the fuzzy cluster:* The spread between the centers of the left and right formation gaussian functions of the best matched fuzzy cluster is denoted as s_{i,j_i^*} , i.e., $s_{i,j_i^*} = c_{i,j_i^*}^R - c_{i,j_i^*}^L$. The spread s_{i,j_i^*} is then modified as described in (18):

$$s_{i,j_i^*}(t+1) = s_{i,j_i^*}(t) + \eta \cdot (1 - SV(x_i, \tilde{A}_{i,j_i^*})) \cdot s_{i,j_i^*}(t) \tag{18}$$

where η is a modification rate. Subsequently, the centers of the left and right functions are updated accordingly as in (19):

$$\begin{aligned} c_{i,j_i^*}^L(t+1) &= c_{i,j_i^*}^R(t) - s_{i,j_i^*}(t+1) \quad \text{if } x_i < c_{i,j_i^*}^L(t) \\ c_{i,j_i^*}^R(t+1) &= c_{i,j_i^*}^L(t) + s_{i,j_i^*}(t+1) \quad \text{if } x_i \geq c_{i,j_i^*}^R(t) \end{aligned} \tag{19}$$

- (b) *Increase the width of the fuzzy cluster:* The width of the best matched fuzzy cluster is modified as described in (20):

$$\sigma_{i,j_i^*}(t+1) = \sigma_{i,j_i^*}(t) + \eta \cdot (1 - SV(x_i, \tilde{A}_{i,j_i^*})) \cdot \sigma_{i,j_i^*}(t) \tag{20}$$

where η is a modification rate. Modification is performed on the best matched cluster by *either* increasing the spread or the width of the membership function under two conditions: the updated similarity value between the presented value and the modified best matched fuzzy cluster is greater than Δ , and the updated similarity value results in the largest increase in the computed similarity value. Otherwise, modification to the best matched cluster is performed by increasing *both* its spread and width.

3. *A new fuzzy cluster is created and it is defined as part of the antecedent segment of $R_{K(t+1)}$.* This scenario occurs when both the computed similarity value and the updated similarity value (as discussed in 2) are less than Δ . That means, the presented value is novel as compared to the existing clusters in the system. The formation of a new fuzzy cluster $\tilde{A}_{i,j_i(t+1)}$, where $J_i(t+1) = J_i(t) + 1$, in the i th input dimension can be described using its FOU as: $\mu_{i,J_i(t+1)}(x) = [\underline{\mu}_{i,J_i(t+1)}(x), \bar{\mu}_{i,J_i(t+1)}(x)]$ where the left and right formation gaussian functions can be described accordingly as in (21):

$$c_{i,J_i(t+1)}^L = x_i - \Delta; \quad c_{i,J_i(t+1)}^R = x_i + \Delta; \quad \sigma_{i,J_i(t+1)} = \begin{cases} \sigma^R & \text{if } J_i^L = \text{NULL} \\ \sigma^L & \text{if } J_i^R = \text{NULL} \\ R(\sigma^R, \sigma^L) & \text{otherwise} \end{cases} \tag{21}$$

such that

$$\sigma^R = R\left(\sqrt{-\frac{(c_{i,J_i^R} - x_i)^2}{\log \alpha}}, \sigma_{i,J_i^R}(t)\right); \quad \sigma^L = R\left(\sqrt{-\frac{(c_{i,J_i^L} - x_i)^2}{\log \alpha}}, \sigma_{i,J_i^L}(t)\right) \tag{22}$$

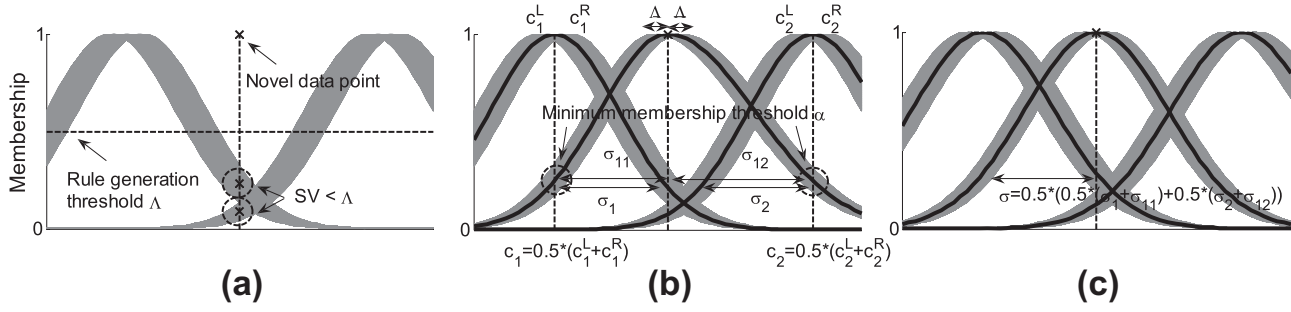


Fig. 3. Fuzzy partitioning: (a) introduction of a novel data point; (b) creation of a new cluster based on the novel point before regulation; and (c) final appearance of the fuzzy partitionings in the input–output dimension after regulation.

The width of the newly created fuzzy cluster is determined based on the notion of behavioral category learning process exhibited by humans [43]. $R(\sigma_1, \sigma_2) = \frac{1}{2} [\sigma_1 + \sigma_2]$ defines a regulator function that ensures a newly created fuzzy cluster has distinct semantic meaning, and α is a minimum membership threshold. Following [43], the minimum membership threshold α is fixed as 0.2 in this paper. The immediate left and right neighbors of the newly created cluster are denoted as j_i^L and j_i^R respectively, where

$$j_i^L = \begin{cases} \text{NULL} & \text{if } c_{i,j_i} \geq x_i \text{ for } 1 \leq j_i \leq J_i(t) \\ \arg \min_{c_{i,j_i} < x_i} |c_{i,j_i} - x_i| & \text{otherwise} \end{cases}$$

$$j_i^R = \begin{cases} \text{NULL} & \text{if } c_{i,j_i} \leq x_i \text{ for } 1 \leq j_i \leq J_i(t) \\ \arg \min_{c_{i,j_i} > x_i} |c_{i,j_i} - x_i| & \text{otherwise} \end{cases} \quad (23)$$

such that $c_{i,j_i} = 1/2 (c_{i,j_i}^L + c_{i,j_i}^R)$. Refinements are subsequently made to the immediate left and right neighbors of the newly created cluster as follows:

1. If the newly created cluster has no left neighbor (i.e. $j_i^L = \text{NULL}$), then only the right neighbor is updated: $\sigma_{i,j_i^R}(t+1) = \sigma_{i,j_i(t+1)}$.
2. If the newly created cluster has no right neighbor (i.e. $j_i^R = \text{NULL}$), then only the left neighbor is updated: $\sigma_{i,j_i^L}(t+1) = \sigma_{i,j_i(t+1)}$.
3. If the newly created cluster has both left and right neighbors, then they are updated: $\sigma_{i,j_i^L}(t+1) = \sigma_{i,j_i^R}(t+1) = \sigma_{i,j_i(t+1)}$.

Fig. 3 illustrates the fuzzy partitioning process in an input–output dimension of the proposed eT2FIS model. A novel data point is encountered in Fig. 3a, where the computed similarity values between the presented point and the existing clusters fall below Λ . In addition, it should be checked that the computed similarity values between the presented point and the modified versions of the existing clusters also fall below Λ . A new cluster is then created using information derived from this novel data point as seen in Fig. 3b. The spreads of the cluster on either side of the center depend on the distance of the presented point to the respective centers of its immediate neighbors. The centers of the immediate neighbors have the minimum membership value α . Regulation is performed to preserve a distinct semantic meaning of the newly created cluster as shown in Fig. 3c. Existing clusters are simultaneously refined to incorporate the newly created cluster.

The same identification process is performed for each output dimension to determine the consequent segment of $R_{K(t+1)}$. Following that, certainty factors of the fuzzy rules are computed based on the modified structure of the eT2FIS model.

3.3. Merger of fuzzy labels

The second operation in the structure learning of eT2FIS is the merger of highly over-lapping fuzzy labels in the input–output spaces. The objective is two-folds: to improve the semantic interpretation of the fuzzy clusters, and to reduce the computational complexity of the system with a more compact rule base. The system proceeds to identify the most similar fuzzy labels in the input dimensions via similarity matches. The similarity match between two existing clusters in the input dimension i is denoted as $SM(\tilde{A}_{i,j_i^1}, \tilde{A}_{i,j_i^2})$ and its computation is described as in (24):

$$SM(\tilde{A}_{i,j_i^1}, \tilde{A}_{i,j_i^2}) = \min(A_{j_i^1 \cap j_i^2} / A_{j_i^1}, A_{j_i^1 \cap j_i^2} / A_{j_i^2}) \quad (24)$$

where $A_{j_i^q}$ is the area enclosed by the isosceles triangle with center $c_{i,j_i^q} = 1/2 (c_{i,j_i^q}^L + c_{i,j_i^q}^R)$ and base $4\sigma_{i,j_i^q} \sqrt{\ln 2}$, $q = 1, 2$; and $A_{j_i^1 \cap j_i^2}$ is the triangular area enclosed by the intersection of the two isosceles triangles. Fig. 4A. illustrates the graphical meanings of the definitions given in (24). Subsequently, the most similar fuzzy labels in an input dimension i are denoted

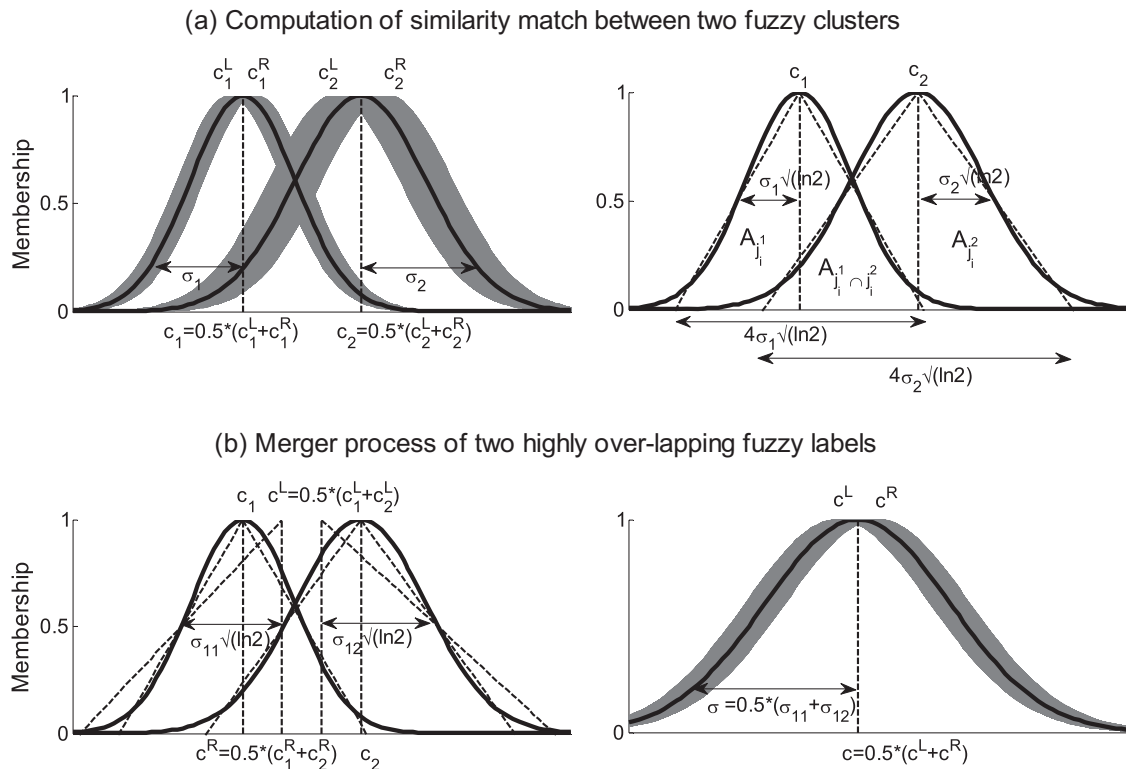


Fig. 4. (A) Computation of similarity match between two fuzzy clusters and (B) merger process of two highly over-lapping fuzzy labels.

as $\tilde{A}_{i_j^i}$ and $\tilde{A}_{i_j^{i'}}$ such that the similarity match between them is the greatest among all existing clusters in the input dimension i .

The most similar fuzzy clusters $\tilde{A}_{i_j^i}$ and $\tilde{A}_{i_j^{i'}}$ are merged if the similarity match between them exceeds a merger threshold Ω , i.e., $SM(\tilde{A}_{i_j^i}, \tilde{A}_{i_j^{i'}}) > \Omega$. Fig. 4B. illustrates the merger process of two highly over-lapping fuzzy labels. The merged fuzzy label is denoted as $\tilde{A}_{i_j^i \cup i_j^{i'}}$ and it can be described using its FOU as: $\mu_{i_j^i \cup i_j^{i'}}(x) = [\underline{\mu}_{i_j^i \cup i_j^{i'}}(x), \bar{\mu}_{i_j^i \cup i_j^{i'}}(x)]$ where the left and right formation gaussian functions can be described accordingly as in (25):

$$\begin{aligned} c_{i_j^i \cup i_j^{i'}}^L &= 1/2 (c_{i_j^i}^L + c_{i_j^{i'}}^L); & c_{i_j^i \cup i_j^{i'}}^R &= 1/2 (c_{i_j^i}^R + c_{i_j^{i'}}^R) \\ \sigma_{i_j^i \cup i_j^{i'}} &= 1/2 (\sigma' + \sigma'') \end{aligned} \tag{25}$$

such that $\sigma' = 1/\sqrt{\ln 2} (c_{i_j^i \cup i_j^{i'}}^L - c_{i_j^i} + \sigma_{i_j^i} \sqrt{\ln 2})$ and $\sigma'' = 1/\sqrt{\ln 2} (c_{i_j^i \cup i_j^{i'}}^R - c_{i_j^{i'}} + \sigma_{i_j^{i'}} \sqrt{\ln 2} - c_{i_j^i}^R)$. Without loss of generality, it is assumed that $c_{i_j^i} < c_{i_j^{i'}}$ where $c_{i_j^i} = 1/2 (c_{i_j^i}^L + c_{i_j^i}^R)$.

The same merger process is repeated for each output dimension to merge highly over-lapping fuzzy labels in the output spaces. Subsequently, the proposed eT2FIS model performs consistency check on the rule base. An inconsistent rule base occurs when there exists two rules in the rule base of the system such that the antecedent conditions are similar but the resultant consequences differ. If the rule base is inconsistent, then inconsistent rules with lower certainty factors are deleted. This approach not only ensures a consistent resultant rule base, it also ensures that eT2FIS provides the most aptly description of the current environment.

3.4. Deletion of obsolete rules

The last operation in the structure learning of eT2FIS is the deletion of obsolete rules. This is necessary to remove irrelevant rules that do not participant in the modeling of the current environment, hence ensuring an up-to-date rule base that is able to model the current underlying dynamics of the environment. This approach spares the proposed system from an explosively increasing rule base by maintaining a compact rule base as it performs a life-long incremental learning of the environment. In addition, by incrementally adding relevant new rules and deleting obsolete rules in the system, a regulated balance is maintained between old knowledge in the system and new knowledge from incoming data pairs. This addresses the stability-plasticity trade-off of the system.

As explained previously, the certainty factor of a fuzzy rule is high if it is able to generalize the recent encountered set of numerical data. The inverse is also true where the certainty factor of a fuzzy rule is low if it fails to give a satisfactory generalization of the recent set of encountered numerical data. Consequently, a fuzzy rule in eT2FIS is regarded as an obso-

lete rule if its certainty factor falls below a deletion threshold Γ , i.e., delete R_k if $\zeta_k(t) < \Gamma$ such that $K(t+1) = K(t) - 1$. Finally, some of the fuzzy labels might be “orphaned” when all fuzzy rules associated with them have been deleted. The orphaned fuzzy labels are then removed to ensure that the resultant computational structure of eT2FIS is compact.

4. Experimental results and analysis

This section illustrates the learning and adaptation abilities of the proposed eT2FIS model by employing it in four benchmarking and real-world applications: (1) identification of a nonlinear system [16]; (2) online tracking of a financial stock price; (3) modeling of highway traffic flow density [41]; and (4) prediction of a chaotic system [29].

4.1. Example 1 – Identification a nonlinear system

There are three parts to this application problem [16]: (a) the online identification of a non-time-varying system; (b) the online identification of a time-varying system; and (c) the identification of a noisy system.

4.1.1. Online identification of a non-time-varying system

This experiment generalizes the dynamics of a nonlinear system, where the dataset is generated by a difference equation as described in (26):

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) \quad (26)$$

such that the present output of the system, $y(t+1)$, depends nonlinearly on its past output, $y(t)$, and an input, $u(t) = \sin(2\pi t/100)$. Following the description in [16], 50000 training and 200 testing data pairs are generated with initial conditions $(u(0), y(0)) = (0, 0)$. Subsequently, the eT2FIS model is applied to identify the nonlinear system in an online mode, where no prior offline knowledge is used to train the system. The system performs structure and parameter learning upon the arrival of each incoming data pair. The thresholds for the three key operations are set as follows: rule generation threshold, $\lambda = 0.4$; merger threshold, $\Omega = 1$; and deletion threshold, $\Gamma = 0$. In addition, the spreads between the left and right formation functions of the fuzzy labels in the system are initialized to zero, since there is no noise added to this problem. This means that the proposed model is being used as an evolving Type-1 Mamdani-typed neural fuzzy system. Subsequently, six, four and three fuzzy labels are identified by eT2FIS for this two inputs–one output application problem respectively. Fig. 5 illustrates the identified fuzzy clusters in the input spaces $u(t)$ and $y(t)$. The respective fuzzy partitioning of the input spaces for the original problem derived by the SONFIN model [16] have also been included as a comparison. As clearly seen from Fig. 5a, the fuzzy clusters identified in eT2FIS are highly ordered and have clear semantic meanings with respect to the attached fuzzy labels, which provides an interpretable resultant system [35]. Comparatively, Fig. 5b shows the initial identified fuzzy clusters in SONFIN. As seen, the fuzzy clusters are highly over-lapping, making it difficult to induce any clear semantic meanings to the derived clusters. To tackle this problem, SONFIN performs an additional step of computing the similarity of a newly formed cluster with existing clusters in the input spaces, and subsequently align the new cluster. This improved result is shown in Fig. 5c. Although the number of fuzzy clusters identified in the input spaces have reduced (from an initial 10 fuzzy clusters to 5 fuzzy clusters in $u(t)$; and an initial 10 fuzzy clusters to 7 fuzzy clusters in $y(t)$), the resultant fuzzy clusters still have a significant amount of over-lap as seen in the first two clusters of the input space $y(t)$. From this illustration, a key strength of a Mamdani-typed neural fuzzy system is well-demonstrated in eT2FIS, where intuitive semantic labels can be directly identified from the raw numerical data for the input–output spaces. On the other hand, TSK models such as SONFIN generally report large over-lapping regions in the identified fuzzy clusters where one can hardly induce any underlying data structure from the observed fuzzy sets [47].

Subsequently, eT2FIS identifies fourteen fuzzy rules using the derived semantic fuzzy labels in the input–output spaces. Fig. 6a lists the identified fuzzy rule base in eT2FIS. One can easily verify that the derived rule base is consistent. Fig. 6b shows the modeling results of the eT2FIS model on the testing data. As seen, there is an almost perfect match between the computed outputs of the network and the desired outputs of the system, with eT2FIS achieving a root mean squared error (RMSE) of 0.053 on the testing data.

This modeling problem focuses on a non-time-varying system, thus the deletion operation is not necessary and it is switched off by setting $\Gamma = 0$. To study the effects of the *rule generation threshold*, λ , and the *merger threshold*, Ω , on the modeling abilities of the proposed eT2FIS model, different values are tested. The test results concerning the number of identified fuzzy labels, fuzzy rules and the RMSE on the testing data are listed in Table 2. Looking at the first table, it is observed that eT2FIS performs consistently within certain ranges of λ . When λ is set to a low value, the modeling abilities of eT2FIS suffer because fewer rules are created. On the other extreme, when λ is set to a high value, eT2FIS performs better with a lower reported RMSE value, but at the expense of a larger rule base. In general, a trade-off between interpretability and accuracy is needed [37], and this is achieved in eT2FIS as follows: better performance is expected when λ is high, with a decrease in the interpretability of the system due to a larger rule base; while a better interpretability is achieved with a low λ , resulting in a decrease in the modeling abilities. Meanwhile, the number of identified fuzzy labels also increase with an increasing λ value. When $\lambda = 0.5$, as much as eight fuzzy clusters are identified for the input space $u(t)$. To achieve a more compact structure,

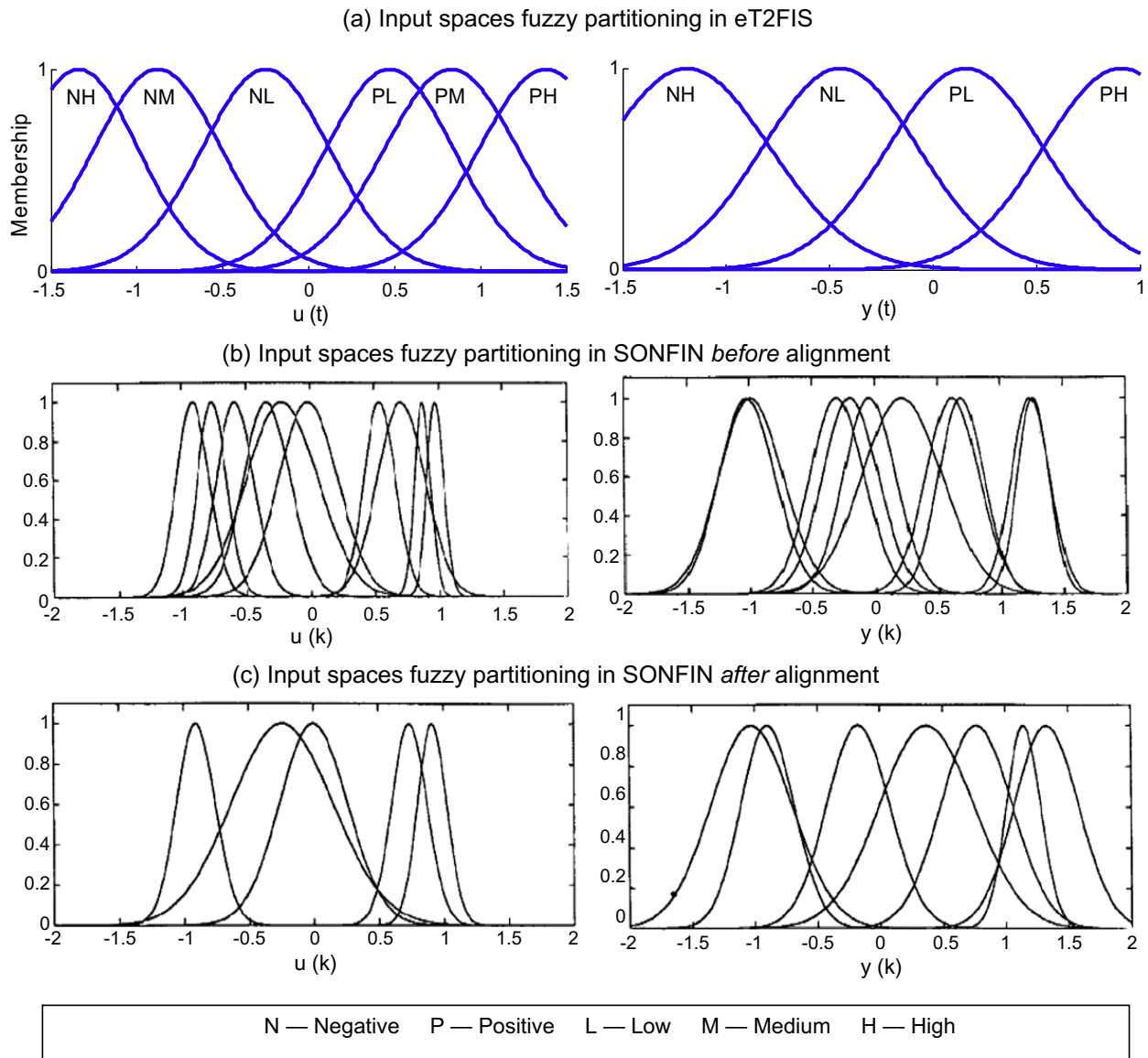


Fig. 5. Input spaces fuzzy partitioning for the online identification of a non-time-varying nonlinear system in: (a) eT2FIS; (b) SONFIN before alignment; and (c) SONFIN after alignment.

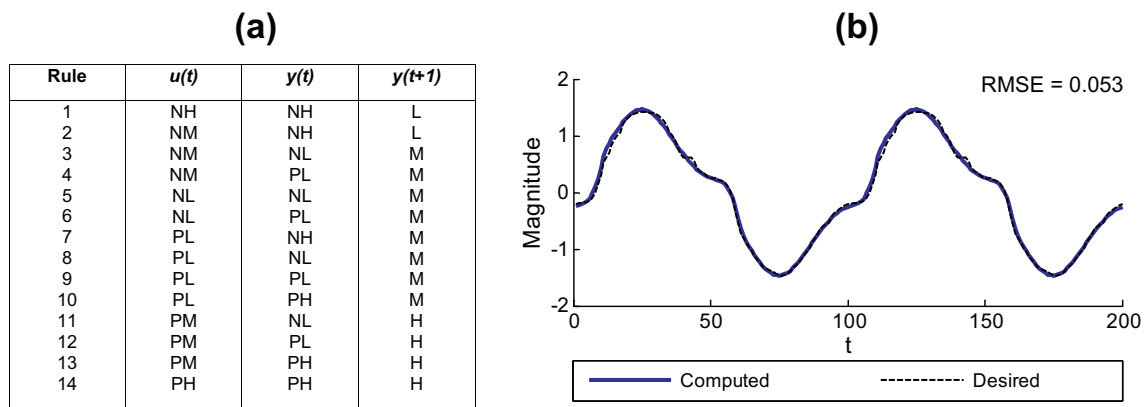


Fig. 6. (a) Identified fuzzy rulebase and (b) modeling results on testing data for the online identification of a non-time-varying nonlinear system using eT2FIS.

Table 2
Sensitivity test for different values of λ and Ω in eT2FIS.

	# Fuzzy labels		# Rules	RMSE
λ				
0.30	Inputs = 3, 4;	Output = 3	6	0.091
0.35	Inputs = 4, 4;	Output = 3	10	0.068
0.40	Inputs = 6, 4;	Output = 3	14	0.053
0.45	Inputs = 5, 5;	Output = 4	14	0.048
0.50	Inputs = 8, 5;	Output = 4	21	0.033
$\Omega = 1; \Gamma = 0$				
Ω				
0.9	Inputs = 8, 5;	Output = 4	21	0.033
0.8	Inputs = 7, 5;	Output = 4	20	0.032
0.7	Inputs = 6, 5;	Output = 4	17	0.034
0.6	Inputs = 4, 5;	Output = 4	12	0.055
0.5	Inputs = 4, 3;	Output = 2	8	0.114
$\lambda = 0.5; \Gamma = 0$				

Table 3
Results of online identification of a non-time-varying system.

Model	Typed	# Rules	RMSE
SONFIN	TSK	10	0.013
SAFIS	TSK	8	0.012
eTS	TSK	19	0.008
FALCON-ART	Mamdani	289	0.138
EFuNN	Mamdani	18	0.058
eFSM	Mamdani	14	0.073
eT2FIS	Mamdani	14	0.053

merging of highly similar fuzzy clusters can be performed on the system. The test results using different values of Ω are listed in the bottom table. As seen, the performance of eT2FIS is consistent when very similar fuzzy clusters are merged. The result is a smaller rule base, and a lesser number of fuzzy clusters identified. However, when Ω is over-relaxed, not very similar clusters might be merged, leading to a significant jump in the RMSE value. Ideally, merging should only be performed for highly over-lapping fuzzy clusters, and this can be achieved by setting a larger merger threshold.

The performance of the proposed eT2FIS model is subsequently compared against the following evolving neural fuzzy systems; namely, SONFIN [16]; SAFIS [39]; eTS [3]; FALCON-ART [27]; EFuNN [23]; and eFSM [47]. The benchmarking measure is the RMSE achieved on the testing data. Table 3 shows the benchmarking results of the models. From the table, it can be seen that TSK-typed systems generally perform better with smaller RMSE values, using a similar number of rules. However, this is not unexpected since TSK-typed systems are generally more expressive and accurate compared to Mamdani-typed models, focusing on achieving good modeling performances at the expense of an opaque system structure [47]. This can be seen from the highly over-lapping fuzzy clusters generated in TSK-typed models as shown in Fig. 5. Comparing within the Mamdani-typed systems, the proposed eT2FIS model is able to deliver an outstanding performance, both in terms of the lowest RMSE value achieved and the lowest number of rules used. This result illustrates the excellent generalization abilities of the proposed model, with an equally well-balanced interpretability maintained in the system.

4.1.2. Online identification of a time-varying system

This experiment is performed to demonstrate the evolving ability of the proposed eT2FIS model, with the dataset generated as in (27):

$$y(t + 1) = \frac{y(t)}{1 + y^2(t)} + u^3(t) + f(t) \tag{27}$$

where a disturbance $f(t)$ is introduced into the system in (26), such that $f(t)$ is described as in (28):

$$f(t) = \begin{cases} 0 & \text{for } 1 \leq t \leq 1000 \text{ and } t \geq 2001 \\ 1 & \text{for } 1001 \leq t \leq 2000 \end{cases} \tag{28}$$

Since this is a modeling problem concerning the identification of a time-varying system, the eT2FIS model is evaluated in an online mode. The thresholds for the three key operations for the structure learning in eT2FIS are set as follows: $\lambda = 0.4$; $\Omega = 1$; $\Gamma = 0$.

Fig. 7 illustrates the online modeling performance of eT2FIS, zooming in on the modeling of eT2FIS when the disturbance $f(t)$ is added in part (a), and removed in part (b). As seen, there are very good matches between the computed output of eT2FIS and the desired output of the system, both after $f(t)$ is introduced at $t = 1001$ and removed at $t = 2001$. This indicates

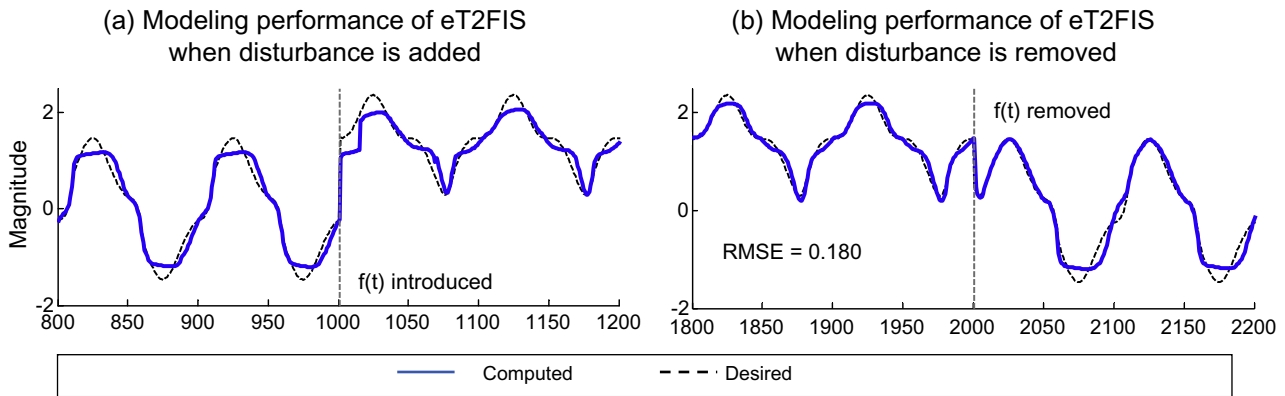


Fig. 7. Modeling performance of eT2FIS when disturbance is (a) added and (b) removed.

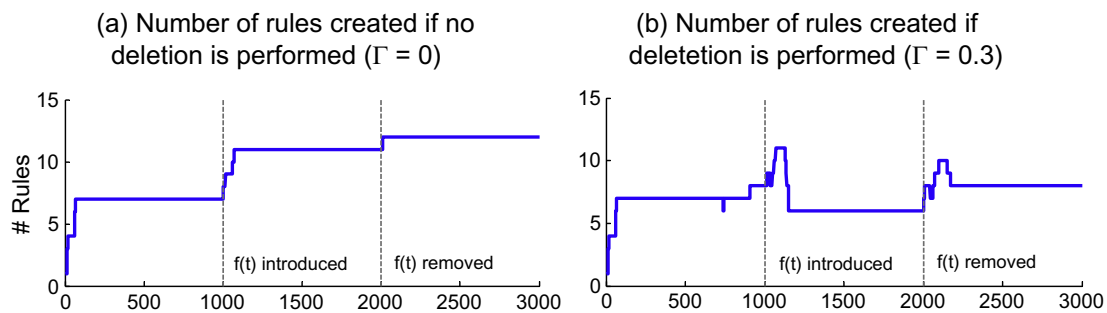


Fig. 8. Number of rules identified by eT2FIS throughout the course of online identification of a time-varying system for (a) when no deletion is activated (i.e. $\Gamma = 0$) and (b) when deletion is activated (i.e. $\Gamma = 0.3$).

Table 4

Sensitivity test for different values of Γ in eT2FIS.

Γ	# Fuzzy labels		# Rules	RMSE
0.0	Inputs = 6, 3;	Output = 5	12	0.180
0.1	Inputs = 7, 3;	Output = 5	11	0.172
0.2	Inputs = 6, 3;	Output = 4	11	0.175
0.3	Inputs = 5, 3;	Output = 4	8	0.174
0.4	Inputs = 10, 3;	Output = 4	13	0.202

$\lambda = 0.4; \Omega = 1.$

the prompt tracking of the proposed model to changes in the application environment, adapting to the changes in an efficient manner. Overall, eT2FIS achieves a RMSE of 0.180 over the duration $t \in [1, 3000]$. Subsequently, the change in the number of rules identified by eT2FIS throughout the course of online identification of this time-varying system is illustrated in Fig. 8a. From the figure, it can be seen that there is an increment in the number of fuzzy rules each time the disturbance is added or removed. Since the deletion threshold Γ has been set to zero, the deletion mechanism is switched off. This is illustrated in Fig. 8a, where the number of rules can only increase. At the end of the course of this experiment, a total of twelve rules have been identified by the eT2FIS model. Considering the short modeling span of 3000 time steps, the size of the fuzzy rule base is still interpretable. However, the system might be burdened by a very large rule base if a longer modeling time span is performed for time-varying application problems, affecting the interpretability of the system. A regulation on the size of the identified fuzzy rule base is thus desirable in such cases.

To study the effect of the *deletion threshold*, Γ , on the modeling abilities of the eT2FIS model, different values of Γ are tested, with the rest of the pre-selected variables unchanged. Deletion of obsolete rules is a feature in the proposed model to tackle the prevailing issue of an ever-expanding structure in evolving neural fuzzy systems. As before, the test results concerning the number of identified fuzzy labels, fuzzy rules and the RMSE values achieved are listed in Table 4. It is observed that eT2FIS performs consistently when the values of Γ are kept small. The identified numbers of fuzzy labels and fuzzy rules become more compact when deletion is performed, thus improving on the interpretability of the system. However, there is a great jump in the RMSE value when Γ is set too large. This is because, for this particular time-varying problem, the rate of change of the underlying dynamics is relatively slow. When the rate of updating the fuzzy rules through a deletion of ob-

Table 5
Performances of eT2FIS, eT1FIS-V1 and eT1FIS-V2 for the identification of a noisy system.

Noise level		[-0.1,0.1]		[-0.5,0.5]	
Model	# Epochs	# Rules	Average RMSE \pm STD	# Rules	Average RMSE \pm STD
eT2FIS	200	10	0.034 \pm 0.002	11	0.138 \pm 0.004
eT1FIS-V1	500	12	0.048 \pm 0.003	16	0.153 \pm 0.007
eT1FIS-V2	500	10	0.063 \pm 0.002	11	0.170 \pm 0.004

solete rules is too rapid, the system could end up forgetting the knowledge too soon after it has been learnt. In general, the deletion threshold is set proportional to the rate of change of the environment's dynamics. That is, a small Γ is sufficient for a slow changing application problem, while a larger Γ is needed for a rapid changing application problem. Subsequently, the change in the number of rules identified by eT2FIS throughout the course of online identification of the time-varying system for $\Gamma = 0.3$ is illustrated in Fig. 8b. As seen, there is a good regulation of the number of fuzzy rules identified throughout the experiment. An increase in the number of fuzzy rules is observed whenever $f(t)$ is added or removed. Following each increase, obsolete rules that are no longer participating in the modeling of the current environment are deleted. This is seen from the decrease in the number of rules. At the end of this experiment, a more compact fuzzy rule base (consisting of 8 rules) is identified, while the modeling performance is maintained (as seen from Table 4, where RMSE = 0.174).

4.1.3. Identification of a noisy system

This experiment is performed to illustrate the noise resistance ability of eT2FIS, where the same nonlinear system in (26) is considered. A total of 200 input–output data pairs are generated. Following the experimental description in [18], it is assumed that each measured $y(t)$ contains noise, and the noisy value is denoted as $y_n(t)$. The added noise is artificially generated white noise with uniform distribution. Simulations with noise being generated in the ranges [-0.1,0.1] and [-0.5,0.5] are conducted. Noise is added to the original 200 clean input–output data pairs. For training, the inputs are $y_n(t)$ and $u(t)$, and the desired output is $y_n(t+1)$. After training, another set of noise is added to the original 200 clean data pairs, and noisy values $y_n(t)$ are fed as inputs to test the noise resistance ability of eT2FIS. Subsequently, the RMSE between the computed output of eT2FIS and the original desired clean output $y(t+1)$ of the system is calculated. There are 20 Monte Carlo realizations for this experiment, where the mean and standard deviation values are averaged. The learning is performed in an offline manner where the training data is learnt by cycling through it a number of epochs.

In the first part of the benchmarking, eT2FIS is compared against two variations of its Type-1 counterparts. For the first variation, an evolving Type-1 Mamdani-typed model is built using the same learning mechanisms, where the spreads between the left and right formation functions of the fuzzy labels in the system are initialized to zero. This model is referred to as eT1FIS-V1. For the second variation, an eT2FIS model is built using the learning mechanisms, where the testing system is then converted to a Type-1 model by replacing all the Type-2 fuzzy sets by Type-1 fuzzy sets (center $c = 1/2(c^L + c^R)$; width σ remains unchanged). This model is then referred to as eT1FIS-V2. For noise level [-0.1,0.1], the thresholds in eT2FIS are set as follows: $\lambda = 0.35$; $\Omega = 0.55$; $\Gamma = 0$; and the initial spreads between the left and right functions of the fuzzy labels are 0.02; while considering a greater noise level [-0.5,0.5], λ is reduced to 0.3 and the initial spreads between the left and right functions of the fuzzy labels are increased to 0.024. Table 5 shows the performances of the three models. The Type-2 model outperforms both versions of its Type-1 counterparts under the two different noise levels, where lower RMSE values are reported. This is on top of the lesser number of training epochs required by eT2FIS. With the other pre-selected variables unchanged (except the spreads of the fuzzy labels), eT1FIS-V1 generally identifies more rules. Although this helps in reducing the RMSE achieved, the system is not as stable compared to eT2FIS as seen from the larger standard deviations of eT1FIS-V1. On the other hand, eT1FIS-V2 inherits the fuzzy rules from eT2FIS, with the fuzzy labels replaced as Type-1 sets. Since the original system is built with the aim of creating a Type-2 model, this could explain the poorest performance of eT1FIS-V2 among all the models. On the other hand, the performance of eT1FIS-V2 is more stable compared to eT1FIS-V1 as seen from the similar standard deviations achieved by eT1FIS-V2 from eT2FIS. This result illustrates the better and more stable noise resistance ability of the proposed eT2FIS model compared to its Type-1 counterparts, indicating that it is more robust when learning under noisy application environments.

In the second part of the benchmarking, the performance of the proposed eT2FIS model is compared against the following models; namely, a Type-1 system – SONFIN [16]; and a Type-2 Mamdani-typed system – T2SONFS [18]. Table 6 shows the results of the comparison. Being a Type-1 model, SONFIN performs the poorest among the three models for both different noise levels. As a fair comparison, 500 epochs of training are performed for the models. As seen, both the Type-2 Mamdani-typed models perform comparably in terms of the number of fuzzy rules identified, and RMSE values achieved. In fact, it should be noted that the proposed model is able to achieve similar performances within 200 epochs of training (see Table 5). This indicates that eT2FIS is able to perform faster learning of the application environment. From this experiment, the proposed Type-2 eT2FIS model has demonstrated better performances compared to both its Type-1 counterparts, as well as generic Type-1 models, under noisy learning environments; while achieving comparable performance compared to corresponding Type-2 models.

Table 6
Performances of eT2FIS and the benchmarking models for the identification of a noisy system.

Noise level		[-0.1,0.1]		[-0.5,0.5]	
Model	# Epochs	# Rules	Average RMSE	# Rules	Average RMSE
SONFIN	500	6	0.041	6	0.170
T2SONFS	500	6	0.034	6	0.138
eT2FIS	500	10	0.033	11	0.140

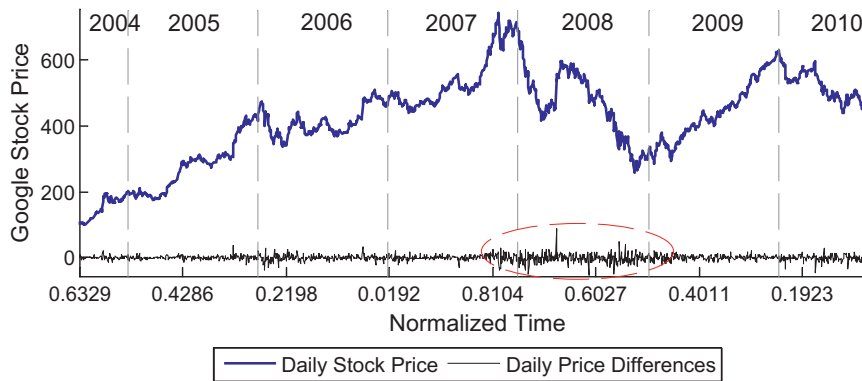


Fig. 9. Google stock price from 19 August 2004 to 21 September 2010.

4.2. Example 2 – Online tracking of a financial stock price

The online modeling ability of the proposed eT2FIS model is evaluated using a real-world financial time-series data, where the stock price of Google is investigated. The data was collected from Yahoo! Finance over a period of 6 years from 19 August 2004 to 21 September 2010, and a total of 1534 data points was collected. Fig. 9 shows the time-variant behavior of the daily stock price, covering over a range of [100.01, 741.79]. Also shown is the daily price differences of the stock, which covers a range of [-52.12, 89.87]. As seen from the figure, there is an exceptional period of volatility in the daily differences from the fourth quarter of 2007 to the first quarter of 2009 as indicated by the dotted circle. Both the maximum and minimum price differences occur within this period, with the mean and standard deviation of the price differences in this period achieving -0.5799 ± 13.97 . The standard deviation is significantly higher compared to that over the entire time horizon (0.2693 ± 9.321) and over the remaining time duration of the experiment (0.5470 ± 7.161).

The objective of this experiment is to track, in an online mode, the underlying movement of the stock price using the input and output vectors, where y is the daily stock price.

$$\text{input vector} = [y(t - 2), y(t - 1), y(t)]; \quad \text{output vector} = [y(t + 1)]$$

The benchmarks for comparisons are the accuracies on the numerical data (calculated as the mean squared error MSE) and the correspondence between the computed outputs with the desired outputs (computed as the Pearson correlation coefficient R). Due to availability constraint,⁵ the experimental results of eT2FIS are benchmarked against the following models; namely, EFuNN [23]; and DENFIS [24]. The thresholds in the eT2FIS model are set as follows: $\lambda = 0.35$; $\Omega = 1$; $\Gamma = 0.4$; and the initial spreads between the left and right functions of the fuzzy labels are 1.0. Since this is a modeling problem concerning a rapidly changing time-varying application, the deletion mechanism in eT2FIS has been switched on to regulate the size of the identified fuzzy rule base.

Fig. 10 shows the online tracking performance of the eT2FIS model for both the entire time horizon (A) and the period from 4th quarter 2007 to 1st quarter 2009 (B). From Fig. 10A-a, eT2FIS demonstrates a satisfying performance in modeling the time-varying stock price movement for the entire time frame of the experiment. The number of rules identified by eT2FIS during the course of the experiment is shown in Fig. 10A-b. The modeling squared error at each time step is also shown. As seen, there is a spike in the modeling error each time a change in the dynamics of the price movement is detected. More rules are then identified by eT2FIS to capture this change of dynamics. Over the 6 years, the underlying dynamics of the stock price movement changes at least 11 times. Subsequently, increments in the numbers of identified rules in eT2FIS are observed as indicated in the figure. Zooming into the volatile period from 4th quarter 2007 to 1st quarter 2009 (as indicated by the box-up in Fig. 10A), the online modeling ability of the proposed eT2FIS model is proven satisfactory as seen by the close map of the computed output of eT2FIS with the desired output in Fig. 10B-a. Prior to this period, there are approximately 31 rules

⁵ Non-evolving systems are not considered for benchmarking in this experiment because the experiment is performed in an online mode, where structure and parameter learning are performed with the arrival of each incoming data pair and it is discarded after learning. On the other hand, offline neural fuzzy systems assume that the set of training data is collected prior to learning, where structure and parameter learning are performed as two different stages using the training dataset.

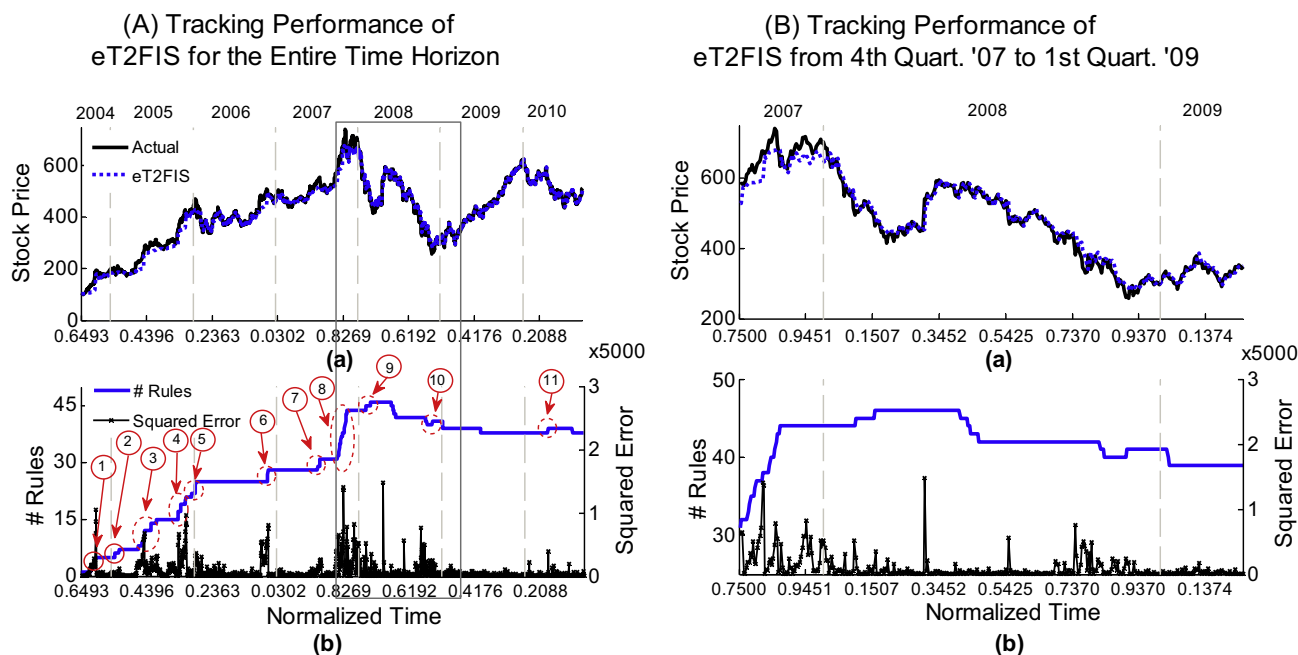


Fig. 10. (A) Tracking performance of eT2FIS for the entire time horizon and (B) tracking performance of eT2FIS from 4th quarter 2007 to 1st quarter 2009.

Table 7

Tracking performances for the financial stock price.

Model	Typed	# Rules	Entire time horizon		4th Quarter 2007–1st Quarter 2009	
			MSE	R	MSE (% Increment)	R (% Decrement)
EFuNN	Type-1 Mamdani	462	360.3	0.9907	575.0 (59.6)	0.9836 (0.717)
DENFIS	Type-1 TSK	203	153.2	0.9972	282.2 (84.2)	0.9937 (0.351)
eT2FIS	Type-2 Mamdani	34	340.7	0.9912	497.3 (46.0)	0.9870 (0.424)

identified in eT2FIS as seen from Fig. 10B-b. When a large modeling error is detected, the system identifies new rules to model the change in dynamics of the stock price movement. This can be seen by the rapid increment in the number of identified rules. Subsequently, the number of identified rules stabilizes and the system performs parameter learning. By the end of 2007, the modeling error decreases and is subsequently kept at a relatively low level throughout the period. This shows the responsiveness of the proposed eT2FIS model to changes in the environment, such that the system is able to adapt both its structure and parameters to model the changing dynamics of the environment.

The tracking performances for the financial stock price are shown in Table 7. The proposed eT2FIS model achieves a MSE value of 340.7 and a R value of 0.9912 for the online modeling of the stock price movement. This places the proposed system in second place, outperforming the Type-1 model EFuNN, but losing out to the TSK-typed DENFIS. Bringing focus to the volatile period from 4th quarter 2007 to 1st quarter 2009, a general trend is an increase in MSE and a decrease in R values achieved by the benchmarking models. In particular, eT2FIS achieves a MSE value of 497.3 and a R value of 0.9870, showing the lowest percentage increment in the MSE value among all three models. On the other hand, the percentage increment in MSE value is the highest for DENFIS, with a significant increase of 84.2%. Although the overall performance of the proposed model loses out to that of DENFIS, eT2FIS identifies a total of 34 fuzzy rules at the end of this experiment, as compared to DENFIS where 203 rules are used. This is a remarkable 6 times more fuzzy rules used compared to the proposed model, greatly decreasing the interpretability of DENFIS. It is noted that the DENFIS model does not prune obsolete rules, and new fuzzy rules are constantly added to the system each time the dynamics of the application environment changes. As a result, the size of the identified fuzzy rule base is ever-expanding. In retrospect, if the deletion mechanism is switched off in eT2FIS such that regulation of the size of the identified fuzzy rule base is not practiced, the number of fuzzy rules identified is 107 and the MSE value achieved is 199.8. In this case, the savings in the fuzzy rules achieved by eT2FIS is almost 50%, at the expense of 30% increment in the MSE value as compared to DENFIS. This result illustrates the modeling trade-off achieved in the eT2FIS model, where both satisfactory interpretability (indicated by the small rule base size as shown in Table 7) and accuracy of the system (indicated by the close map between the computed and desired outputs as shown in Fig. 10A-a) are achieved.

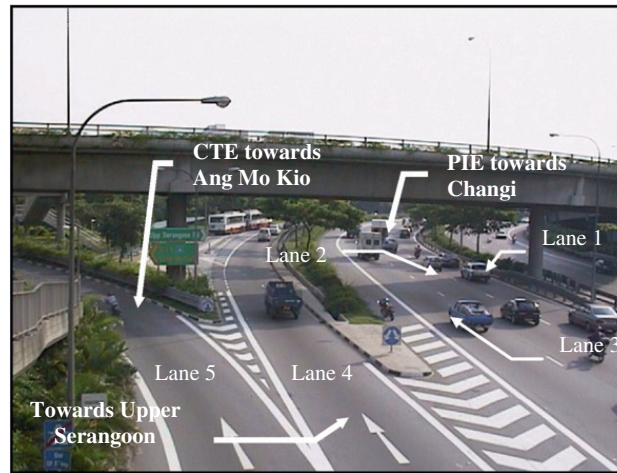


Fig. 11. Location of site 29 along the Pan Island Expressway in Singapore.

4.3. Example 3 – Modeling of highway traffic flow density

The learning and generalization abilities of the proposed eT2FIS model are evaluated by employing it in a real-life application involving the modeling of highway traffic flow density [41]. The data was collected from site 29 located at exit 15 along the east-bound Pan Island Expressway (PIE) in Singapore using loop detectors embedded beneath the road surface. The inductive loop detectors were pre-installed by the Land Transport Authority of Singapore (LTA) in 1996 along major roads to facilitate traffic flow data collection. Fig. 11 shows a picture of the location where the data was collected. There are a total of five lanes: three straight lanes for the main traffic (lanes 1–3), and two exit lanes (lanes 4–5). Only data from the three straight lanes (denoted as L1, L2 and L3 respectively) are used in this experiment. The data has four attributes: the time t at which the traffic flow data was measured, and the traffic flow densities for the three straight lanes during t . The eT2FIS model is used to model the traffic flow trend, and the trained model is then used to predict traffic flow density of a lane (L1, L2 or L3) at time $t + \tau$ for $\tau = 5, 15, 30, 45$ and 60 min.

Fig. 12 shows the traffic flow data for lanes L1–L3 spanning over a period of 6 days from 5th to 10th September 1996. The data is divided into three cross-validation groups (denoted as CV1, CV2 and CV3 respectively). The training data for each cross-validation group is extracted accordingly from the period labeled in Fig. 12. The benchmarking measurements are the Pearson correlation coefficient R and the mean squared error MSE. The performance of the proposed eT2FIS model is subsequently compared against the following models; namely, RSPOP [2]; MLP (with a configuration of 4 input nodes, 10 hidden nodes and 1 output node); GenSoFNN [45]; EFuNN [23]; DENFIS [24]; and eFSM [47]. The thresholds in the eT2FIS model are set as follows: $\lambda = 0.25$; $\Omega = 1$; $\Gamma = 0$; and the initial spreads between the left and right functions of the fuzzy labels are 0.16 for lanes L1 and L2; and $\lambda = 0.2$ for lane L3. Considering that lane L3 is the lane next to the exit lanes, the traffic density along lane L3 could be affected by cars filtering into L3 to exit the expressway. The novelty threshold has thus been made smaller to take into account the higher volatility effect encountered in lane L3.

Fig. 13 illustrates the identified fuzzy clusters in lane L1 for the 4 input-1 output spaces for training set CV1 when $\tau = 5$ min, together with the set of identified fuzzy rule base in eT2FIS. The distributions of the raw numerical data for lanes L1–L3 are also shown in the figure. The fuzzy clusters identified in the proposed eT2FIS model coincide with the peaks of the distributions as marked by the dotted circles, demonstrating the effectiveness of the incremental fuzzy partitioning employed in eT2FIS. Since the fuzzy clusters are highly ordered, clear semantic meanings can be attached to the fuzzy clusters. Subsequently, a total of 15 fuzzy rules are identified by the eT2FIS model as listed in Fig. 13. One can easily verify that the derived rule base is consistent, with a similar number of rules describing the morning and the evening traffic conditions. Fuzzy rules 1–6 reflect the off-peak morning traffic condition; while fuzzy rules 9–14 reflect the off-peak evening traffic condition; with the remaining rules (7, 8, 15) describe the peak morning-evening traffic conditions.

The consolidated traffic flow prediction results are shown in Fig. 14. The average R value and the average MSE value from the three cross-validation groups CV1–CV3 for each prediction horizon are plotted with respect to the lanes L1–L3. The performance of the proposed eT2FIS model is comparable with that achieved by the benchmarking models. In particular, the R values achieved by eT2FIS are among the top performers when $\tau = 60$ min, while most benchmarking models have lower correlations due to a longer time lag in the prediction horizon. This result demonstrates the good generalization abilities of the proposed eT2FIS model such that it is able to learn and generalize the traffic trend to subsequently perform good forecasting on unseen data.

Table 8 shows the average performances of all the models for this highway traffic flow density modeling task. Based on the benchmarking measures, it can be seen that the eT2FIS model outperforms both GenSoFNN and EFuNN. The computed average R and average MSE values of the proposed model are also comparable with fellow evolving models DENFIS and eFSM. Comparing to RSPOP, a comparable average R value is achieved by eT2FIS. Although a slight deterioration in the

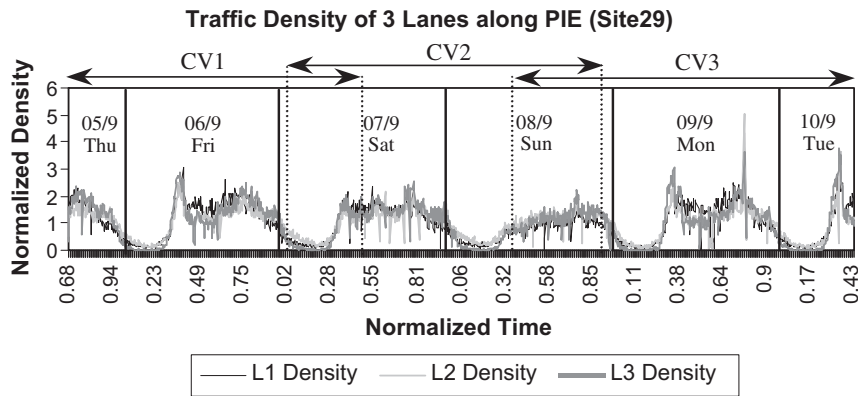


Fig. 12. Traffic flow densities of the three straight lanes along PIE at site 29.

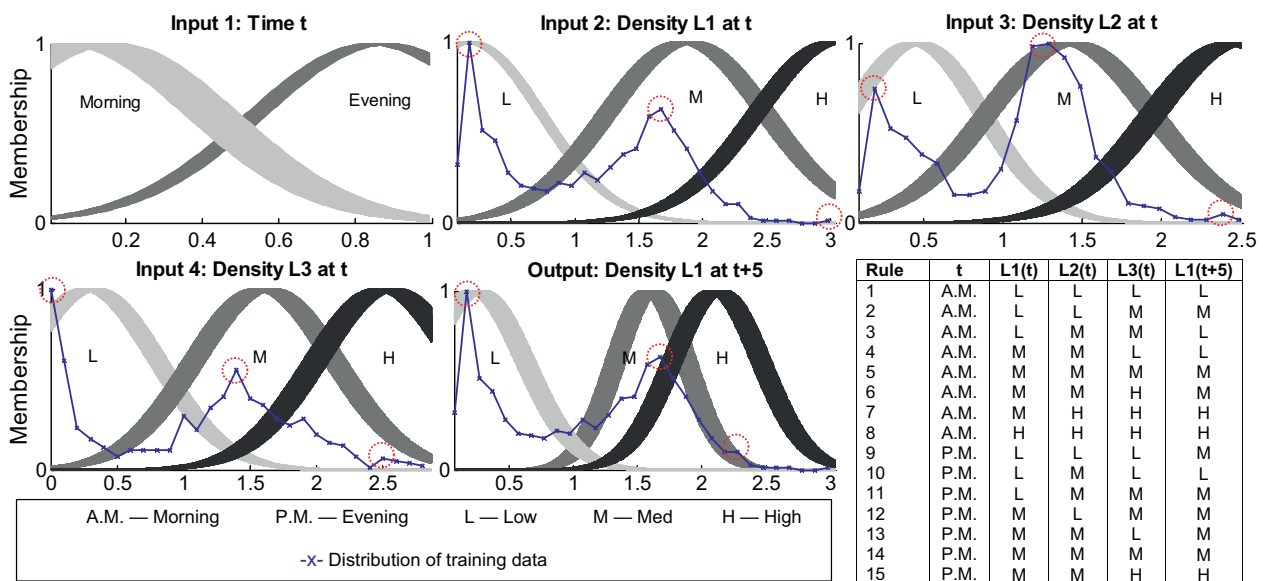


Fig. 13. Fuzzy clusters identified (with distributions of raw data) in lane L1 for training set of CV1 when $\tau = 5$ min, and the identified fuzzy rulebase in eT2FIS.

average MSE value is observed, it should be noted that RSPOP employs a batched learning mode while the proposed model employs an incremental learning mode. That means, while a batched mode system has the luxury of re-learning a set of training data with repeated cycles, eT2FIS extracts knowledge from the numerical data by a single pass. The MLP model achieves the best performance in terms of the computed benchmarking measures. Unfortunately, it also achieves the greatest standard deviations on the computed measures. That means, the performance of MLP is highly volatile, which makes it unpredictable for this modeling task. Overall, this result demonstrates an encouraging modeling potential of the proposed eT2FIS model; while maintaining a highly consistent and stable performance under varying conditions (i.e., time horizons).

4.4. Example 4 – Prediction of a chaotic system

The noise resistance ability of the proposed eT2FIS model is evaluated by employing it in a benchmark comparison involving the prediction of a chaotic system with noise. The original chaotic time-series is generated by a delay differential equation as described in (29):

$$\frac{\partial x(t)}{\partial t} = \frac{0.2x(t - \tau)}{1 + x^{10}(t - \tau)} - 0.1x(t) \tag{29}$$

which was first investigated by Mackey and Glass [29]. Following the problem described in [20], a fourth-order Runge–Kutta method was applied to compute the numerical approximation of the series with conditions as follows: $\tau = 30$ and initial condition $x(0) = 1.2$. Four past values are used to predict the present value, where

$$\text{input vector} = [x(t - 24), x(t - 18), x(t - 12), x(t - 6)]; \quad \text{output vector} = [x(t)]$$

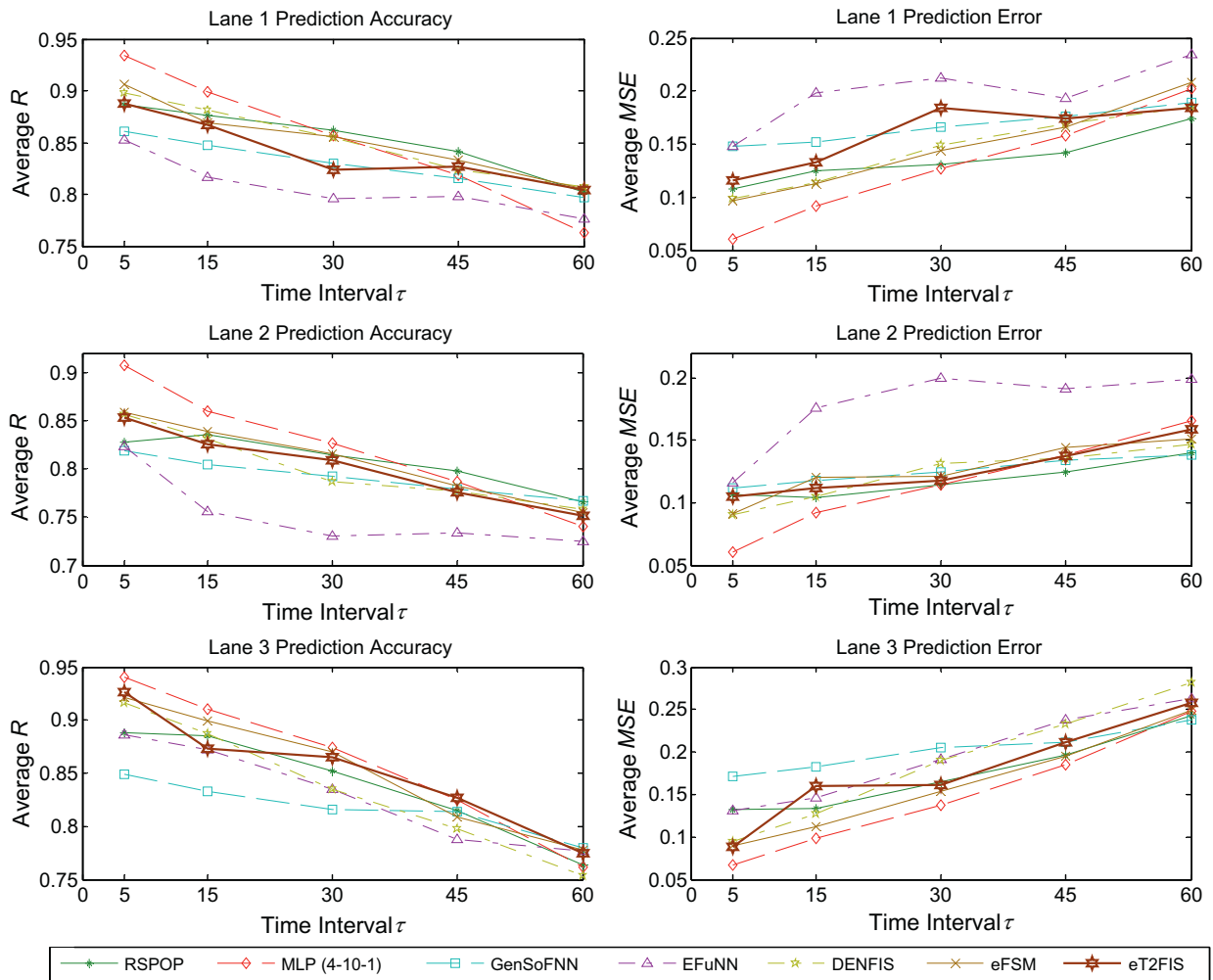


Fig. 14. Traffic flow prediction results.

Table 8

Average performances for the traffic flow prediction.

Model	Typed	Mode	# Rules	Average R (\pm STD)	Average MSE (\pm STD)
RSPOP	Mamdani	Batched	14.4	0.834 (\pm 0.041)	0.146 (\pm 0.038)
MLP (4-10-1)	-	Batched	-	0.847 (\pm 0.065)	0.130 (\pm 0.055)
GenSoFNN	Mamdani	Batched	50.0	0.813 (\pm 0.028)	0.164 (\pm 0.037)
EFuNN	Mamdani	Incremental	234.5	0.798 (\pm 0.050)	0.189 (\pm 0.041)
DENFIS	TSK	Incremental	9.7	0.831 (\pm 0.051)	0.153 (\pm 0.054)
eFSM	Mamdani	Incremental	20.3	0.840 (\pm 0.043)	0.154 (\pm 0.040)
eT2FIS	Mamdani	Incremental	21.0	0.833 (\pm 0.047)	0.153 (\pm 0.045)

A total of 1000 data pairs are extracted from the interval $t \in [124, 1123]$. The first 500 pairs are then used as training set, while the remaining 500 pairs are used for testing. In the training part, three training sets are created by adding Gaussian white noise with mean 0 and standard deviations 0.1, 0.2 and 0.3 to $x(t)$. On the other hand, for the testing part, the first set is the original clean data, with two other testing sets created by adding Gaussian white noise with mean 0 and standard deviations 0.1 and 0.3. The benchmarking measure is the root mean squared error RMSE. Subsequently, the performance of the proposed eT2FIS model is benchmarked against the following evolving neural fuzzy systems; namely, Type-1 models – SONFIN [16]; DENFIS [24]; EFuNN [23]; and Type-2 models – SEIT2FNN [17]; IT2FNN-SVM [20]. The thresholds in the eT2FIS model are set as follows: $\mathcal{A} = 0.3$; $\Omega = 1$; $\Gamma = 0$; and the initial spreads between the left and right functions of the fuzzy labels are 0.1.

In the first part of this experiment, intra-validation is performed within the training sets. The first 80% of the data is used to train the models, and the remaining 20% is used to validate the performances of the trained models. There are a total of 10 Monte Carlo realizations for the statistical analysis of the results. Table 9 shows the average RMSE values

Table 9
Average RMSE of the intra-validation on training sets for the prediction of chaotic system.

Model	Train STD			Average RMSE ± STD
	0.1	0.2	0.3	
SONFIN	0.113	0.226	0.302	0.214 ± 0.095
DENFIS	0.116	0.214	0.306	0.212 ± 0.095
SEIT2FNN	0.123	0.225	0.319	0.222 ± 0.098
IT2FNN-SVM	0.128	0.234	0.349	0.237 ± 0.110
EFuNN	0.126	0.252	0.366	0.248 ± 0.120
eT2FIS	0.120	0.225	0.327	0.224 ± 0.104

Table 10
Performance of eT2FIS for the prediction of chaotic system.

Train STD	0.1			0.2			0.3		
	Clean	0.1	0.3	Clean	0.1	0.3	Clean	0.1	0.3
RMSE	0.059	0.107	0.214	0.083	0.132	0.247	0.102	0.152	0.278

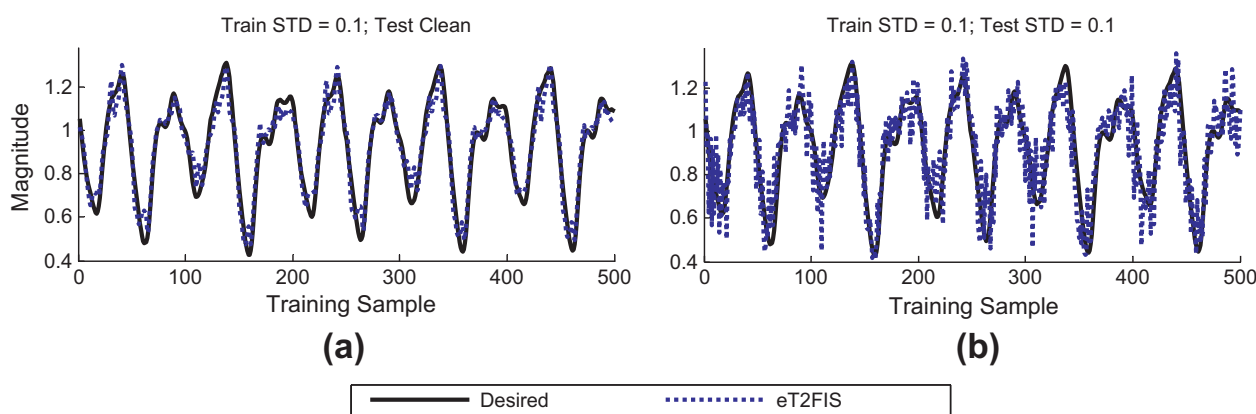


Fig. 15. Prediction results for a run of eT2FIS: (a) train STD = 0.1, test clean and (b) train STD = 0.1, test STD = 0.1.

achieved by the models in performing intra-validation. The best performer is DENFIS with an average RMSE of 0.212 ± 0.095 , and the worst performer is EFuNN with an average RMSE of 0.248 ± 0.120 . The differences between all benchmarking models are within 17% from the best performer, with eT2FIS claiming fourth position such that the difference between the proposed model and DENFIS is marginal (<6%). In addition, eT2FIS performs marginally better compared to fellow Type-2 model IT2FNN-SVM. Based on these results, the proposed model has very comparable performance with respect to all the benchmarking models.

In the second part of the experiment, cross-validations are performed between the training and testing sets. Table 10 shows the consolidated average performance of the proposed eT2FIS model for this prediction task. In general, the testing RMSE values increase as the noise levels in the training sets increase. Within each noise level of the training data, the testing RMSE values also increase with an increase in the noise levels of the testing data. Fig. 15 shows the prediction results for a particular run on eT2FIS, where the system is trained using noisy data with standard deviation 0.1 and then tested on: (a) clean data, and (b) noisy data with standard deviation 0.1. From the figure, eT2FIS demonstrates satisfactory performances when tested on both the clean data and the noisy data. This is seen from the close maps between the computed outputs of eT2FIS and the desired outputs of the system. As expected, larger fluctuations in the computed outputs can be seen when tested on the data that is contaminated with noise. This result illustrates that the eT2FIS model is able to perform relatively good modeling despite the presence of noise.

Table 11 shows the consolidated experimental results for the prediction of the chaotic system. The average measure reports on the central tendency of the performance for the models, while the standard deviation measures the stability of the performance of the models under varying noise conditions. The models are subsequently ranked according to the averages and standard deviations of their computed RMSE values under different noise levels. Being Type-2 models, SEIT2FNN, IT2FNN-SVM and eT2FIS are consistently ranked among the top three performers under different noise conditions. This implies that Type-2 models are generally more capable of handling uncertainties under noisy environment. Overall, the proposed eT2FIS model achieves an average ranking of 3.3, behind only fellow Type-2 models IT2FNN-SVM and SEIT2FNN.

Table 11
Consolidated experimental results for the prediction of chaotic system.

Model	Train STD = 0.1		Train STD = 0.2		Train STD = 0.3		Average rank (\sum Rank $i/6$)
	Average (Rank 1)	\pm STD (Rank 2)	Average (Rank 3)	\pm STD (Rank 4)	Average (Rank 5)	\pm STD (Rank 6)	
SONFIN	0.139 (4)	\pm 0.105 (5)	0.178 (5)	\pm 0.089 (5)	0.236 (5)	\pm 0.060 (4)	4.7
DENFIS	0.147 (6)	\pm 0.120 (6)	0.161 (4)	\pm 0.125 (6)	0.175 (2)	\pm 0.136 (6)	5.0
SEIT2FNN	0.119 (2)	\pm 0.084 (4)	0.141 (2)	\pm 0.077 (3)	0.216 (4)	\pm 0.033 (2)	2.8
IT2FNN-SVM	0.109 (1)	\pm 0.075 (1)	0.125 (1)	\pm 0.053 (1)	0.151 (1)	\pm 0.033 (1)	1
EFuNN	0.143 (5)	\pm 0.081 (3)	0.195 (6)	\pm 0.064 (2)	0.257 (6)	\pm 0.051 (3)	4.2
eT2FIS	0.127 (3)	\pm 0.079 (2)	0.154 (3)	\pm 0.084 (4)	0.177 (3)	\pm 0.091 (5)	3.3

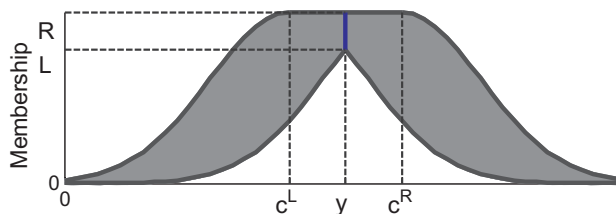


Fig. 16. An interval Type-2 membership function, reproduced from Fig. 2, with a singleton point.

This is hardly surprising since both IT2FNN-SVM and SEIT2FNN are TSK-typed models, and the proposed model is a Mamdani-typed model. TSK-typed models are precise fuzzy modeling systems whose main learning focus is on achieving good resultant accuracy. On the other hand, Mamdani-typed models are linguistic fuzzy modeling systems which focus on providing good interpretability of the systems, while maintaining a balance between the desired accuracies [37]. This result demonstrates the noise resistance ability of eT2FIS over Type-1 neural fuzzy systems, which is a main advantage of Type-2 models.

5. Conclusions

This paper proposes the *evolving Type-2 neural fuzzy inference system* (eT2FIS) – a Type-2 Mamdani-typed neural fuzzy system with an online learning mechanism. Compared to the existing evolving neural fuzzy systems, eT2FIS is one of the first few attempts in unifying the concepts of evolving neural fuzzy systems with Type-2 Mamdani-typed models. Learning is performed incrementally in eT2FIS such that the system evolves and learns with the arrival of a new data pair. This allows the proposed system to incorporate new knowledge progressively. Hence, allowing a life-long learning for an application environment. The eT2FIS model also addresses the stability-plasticity trade-off of a learning system by drawing inspiration from the human biological memory retention mechanism. This is demonstrated through the learning of eT2FIS where positive associations between new knowledge from the environment and existing knowledge in the system enhances the related knowledge (a biological phenomenon known as long-term potentiation), and negative associations between the new knowledge from the environment and existing knowledge in the system degrades the related knowledge (a biological phenomenon known as long-term depression). Subsequently, a new rule is created when the knowledge from the environment is novel with respect to the knowledge in the system and an obsolete rule is deleted when it is no longer actively participating in the modeling of the current environment. This addition-removal mechanism provides a regulated balance between old knowledge in the system and new knowledge from incoming data, such that old and new knowledge can co-exist to define the structure of the system. Another key feature in the proposed eT2FIS model is the merger of highly over-lapping fuzzy labels in the input–output spaces. This ensures a compact and more interpretable structure of the eT2FIS model. Finally, eT2FIS is employed in four benchmarking and real-world applications. From the experimental results, the proposed eT2FIS model has demonstrated: (1) it can effectively adapt both its structure and parameters to model complex real-world application problems; (2) it ensures a good level of human interpretability to the resultant system by providing highly semantic fuzzy

labels and easily comprehensible linguistic IF-THEN fuzzy rules; and (3) it possesses a good noise resistance ability when modeling under noisy conditions.

Appendix A. Type-reduction in eT2FIS

The height-type-reducer [9] replaces each Type-2 fuzzy set by a single point in the primary domain, referred to as a *singleton*, where it is denoted as y . Subsequently, for an interval Type-2 fuzzy set with a singleton y , the membership of y is an interval with bounds $[L, R]$. The notations can be explained using Fig. 16, where the interval Type-2 fuzzy set of Fig. 2 is reproduced. As seen, a possible singleton point for this fuzzy set is illustrated by the blue line. Subsequently, the height-type-reduction has been modified in this paper as follows: For each interval Type-2 fuzzy cluster $\tilde{C}_{l_m, m}$, the singleton point θ_{l_m} is chosen as $1/2 (c_{l_m, m}^L + c_{l_m, m}^R)$ where $c_{l_m, m}^L$ and $c_{l_m, m}^R$ are the centers of the left and right formation gaussian functions of the Type-2 membership function embedded in $\tilde{C}_{l_m, m}$ respectively; and the interval bounds $[L, R]$ are replaced by the interval Type-1 output set $fo_{l_m, m} = [\underline{f}_{l_m, m}, \bar{f}_{l_m, m}]$ (see (9)).

For a set of $L_m(t)$ interval Type-2 membership functions in the output space OV_m , let θ_{l_m} , $l_m = 1, \dots, L_m(t)$, denote the $L_m(t)$ singletons, with the corresponding interval bounds $[\underline{f}_{l_m, m}, \bar{f}_{l_m, m}]$. Assume that $\theta_1 \leq \dots \leq \theta_{L_m(t)}$. This can be achieved by applying an $L_m(t) \times L_m(t)$ matrix to sort the set $\{\theta_{l_m}\}$. For convenience, define the following function as in (30):

$$H(\rho_1, \dots, \rho_{L_m(t)}) = \frac{\sum_{l_m=1}^{L_m(t)} \theta_{l_m} \rho_{l_m}}{\sum_{l_m=1}^{L_m(t)} \rho_{l_m}} \quad (30)$$

To compute the upper bound \bar{Y}_m of the type-reduced set:

- Step 1: Set $\rho_{l_m} = 1/2 (\underline{f}_{l_m, m} + \bar{f}_{l_m, m})$ for $l_m = 1, \dots, L_m(t)$. Compute $H_1 = H(\rho_1, \dots, \rho_{L_m(t)})$.
- Step 2: Determine l^* , where $1 \leq l^* \leq L_m(t) - 1$, such that $\theta_{l^*} \leq H_1 \leq \theta_{l^*+1}$.
- Step 3: Set $\rho_{l_m} = \underline{f}_{l_m, m}$ for $l_m = 1, \dots, l^*$ and $\rho_{l_m} = \bar{f}_{l_m, m}$ for $l_m = l^* + 1, \dots, L_m(t)$. Compute $H_2 = H(\rho_1, \dots, \rho_{L_m(t)})$.
- Step 4: If $H_1 = H_2$, then the upper bound is found and $\bar{Y}_m = H_2$. Else, set H_1 as H_2 , and repeat by returning to Step 2.

To compute the lower bound \underline{Y}_m of the type-reduced set, the main difference is Step 3 of the above algorithm, where one needs to set $\rho_{l_m} = \bar{f}_{l_m, m}$ for $l_m = 1, \dots, l^*$ and $\rho_{l_m} = \underline{f}_{l_m, m}$ for $l_m = l^* + 1, \dots, L_m(t)$.

Appendix B. Parameter adaptation in eT2FIS

The parameter adaptation in eT2FIS follows the structure learning phase of the system, and it is performed for each incoming data tuple. The trainable parameters of eT2FIS are the centers of the left and right formation gaussian functions of the fuzzy labels embedded in the antecedent and consequent segments of the fuzzy rules. They are optimally tuned using a supervised learning algorithm based on a gradient descent approach to minimize the error function defined as in (31):

$$E = \frac{1}{2} \sum_m (d_m - y_m)^2 \quad (31)$$

where $D(t) = [d_1(t), \dots, d_M(t)]^T$ denotes the desired output vector, and $Y(t) = [y_1(t), \dots, y_M(t)]^T$ denotes the computed output vector at a time instance t (see Fig. 1). To simplify the notation, the time index shall be dropped. Nevertheless, it should be noted that parameter adaptation is performed at each time instance with the arrival of each data tuple.

For the rest of this section, the following generic notations are adopted:

$$N_m^L = \sum_{l_m=1}^{L_m} \theta_{l_m} \rho_{l_m}^L; \quad D_m^L = \sum_{l_m=1}^{L_m} \rho_{l_m}^L$$

$$N_m^R = \sum_{l_m=1}^{L_m} \theta_{l_m} \rho_{l_m}^R; \quad D_m^R = \sum_{l_m=1}^{L_m} \rho_{l_m}^R$$

such that the type-reduced set $Y_m = [\underline{Y}_m, \bar{Y}_m]$, see (11), is bounded by $\underline{Y}_m = N_m^L/D_m^L$ and $\bar{Y}_m = N_m^R/D_m^R$. As explained, the KM algorithm for computing Y_m requires the set $\{\theta_{l_m} = 1/2 (c_{l_m, m}^L + c_{l_m, m}^R)\}$ to be sorted in an ascending order. Let E_m be the matrix that performs such a sorting task where $E_m = (e)$ is a $L_m \times L_m$ matrix. Subsequently, the output sets of the consequent nodes $\{\underline{f}_{l_m, m}\}$ and $\{\bar{f}_{l_m, m}\}$, see (10), are sorted using E_m to form the sets $\{\underline{f}_{l_m, m}^S\}$ and $\{\bar{f}_{l_m, m}^S\}$ respectively. Then

$$\rho_{l_m}^L = \begin{cases} \bar{f}_{l_m, m}^S & \text{if } l_m = 1, \dots, l_m^* \\ \underline{f}_{l_m, m}^S & \text{if } l_m = l_m^* + 1, \dots, L_m \end{cases}; \quad \rho_{l_m}^R = \begin{cases} \underline{f}_{l_m, m}^S & \text{if } l_m = 1, \dots, l_m^* \\ \bar{f}_{l_m, m}^S & \text{if } l_m = l_m^* + 1, \dots, L_m \end{cases}$$

where l_m^* and l_m^{R*} are the left and right cross-over points determined using the KM algorithm (see Appendix A). Finally, reverse the sorting process of $\{\rho_{l_m}^L\}$ and $\{\rho_{l_m}^R\}$ using E_m^{-1} , and denote the sets as $\{\omega_{l_m}^L\}$ and $\{\omega_{l_m}^R\}$ respectively. Also, the error signal for an arbitrary node is given as $\delta = -\partial E/\partial fo$.

Layer 5: The error signal of OV_m can be computed as in (32):

$$\delta_m = -\frac{\partial E}{\partial fo_m} = -\frac{\partial E}{\partial y_m} = d_m - y_m \tag{32}$$

To derive the updating steps for tuning the centers of the fuzzy labels embedded in the consequent segments, one needs to compute the partial derivatives of (33):

$$\frac{\partial fo_m}{\partial c_{l_m,m}^L} = \frac{\partial fo_m}{\partial c_{l_m,m}^R} = \frac{1}{4} \left(\frac{\omega_{l_m}^L}{D_m^L} + \frac{\omega_{l_m}^R}{D_m^R} \right) \tag{33}$$

Subsequently, the updating steps for the centers of the left and right formation gaussian functions of the consequent labels are given as in (34):

$$\begin{aligned} c_{l_m,m}^L(t+1) &= c_{l_m,m}^L(t) + \eta \delta_m(t) \frac{\partial fo_m(t)}{\partial c_{l_m,m}^L(t)} \\ c_{l_m,m}^R(t+1) &= c_{l_m,m}^R(t) + \eta \delta_m(t) \frac{\partial fo_m(t)}{\partial c_{l_m,m}^R(t)} \end{aligned} \tag{34}$$

where $\eta > 0$ is a learning constant.

Layer 4: Using chain rule, the error signal of $\tilde{C}_{l_m,m}$ can be described as in (35):

$$\delta_{l_m,m} = -\frac{\partial E}{\partial fo_{l_m,m}} = [\underline{\delta}_{l_m,m}, \bar{\delta}_{l_m,m}] \tag{35}$$

where the bounds of the interval Type-1 set are computed accordingly as in (36):

$$\begin{aligned} \underline{\delta}_{l_m,m} &= -\frac{\partial E}{\partial \underline{f}_{l_m,m}} = \begin{cases} \frac{1}{2} \delta_m \left(\frac{1/2 (c_{l_m,m}^L + c_{l_m,m}^R) D_m^L - N_m^L}{(D_m^L)^2} \right) & \text{if } e_{b,l_m} = 1, b = l_m^* + 1, \dots, L_m \\ 0 & \text{otherwise} \end{cases} \\ &+ \begin{cases} \frac{1}{2} \delta_m \left(\frac{1/2 (c_{l_m,m}^L + c_{l_m,m}^R) D_m^R - N_m^R}{(D_m^R)^2} \right) & \text{if } e_{b,l_m} = 1, b = 1, \dots, l_m^{R*} \\ 0 & \text{otherwise} \end{cases} \\ \bar{\delta}_{l_m,m} &= -\frac{\partial E}{\partial \bar{f}_{l_m,m}} = \begin{cases} \frac{1}{2} \delta_m \left(\frac{1/2 (c_{l_m,m}^L + c_{l_m,m}^R) D_m^L - N_m^L}{(D_m^L)^2} \right) & \text{if } e_{b,l_m} = 1, b = 1, \dots, l_m^{L*} \\ 0 & \text{otherwise} \end{cases} \\ &+ \begin{cases} \frac{1}{2} \delta_m \left(\frac{1/2 (c_{l_m,m}^L + c_{l_m,m}^R) D_m^R - N_m^R}{(D_m^R)^2} \right) & \text{if } e_{b,l_m} = 1, b = l_m^{R*} + 1, \dots, L_m \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{36}$$

Layer 3: Since a rule node R_k can contribute to more than one consequent fuzzy label $\tilde{C}_{l_m,m}$ by (9), the cumulative error signal of R_k can be described as in (37):

$$\delta_k = -\frac{\partial E}{\partial fo_k} = [\underline{\delta}_k, \bar{\delta}_k] \tag{37}$$

where the bounds of the interval Type-1 set are computed accordingly as in (38):

$$\underline{\delta}_k = -\frac{\partial E}{\partial \underline{f}_k} = \sum_{l_m} \underline{\delta}_{l_m,m}^{(k)}; \quad \bar{\delta}_k = -\frac{\partial E}{\partial \bar{f}_k} = \sum_{l_m} \bar{\delta}_{l_m,m}^{(k)} \tag{38}$$

such that $\underline{\delta}_{l_m,m}^{(k)}$ (resp. $\bar{\delta}_{l_m,m}^{(k)}$) is the lower (resp. upper) bound of the error signal of $\tilde{C}_{l_m,m}^{(k)}$, and $\tilde{C}_{l_m,m}^{(k)}$ is a consequent node inheriting the lower (resp. upper) firing strength of R_k by (10).

Layer 2: As before, an antecedent node \tilde{A}_{i_ji} can contribute to more than one rule R_k by (7), hence the cumulative error signal of \tilde{A}_{i_ji} can be described as in (39):

$$\delta_{i_{ji}} = -\frac{\partial E}{\partial fo_{i_{ji}}} = [\underline{\delta}_{i_{ji}}, \bar{\delta}_{i_{ji}}] \quad (39)$$

where the bounds of the interval Type-1 set are computed accordingly as in (40):

$$\underline{\delta}_{i_{ji}} = -\frac{\partial E}{\partial \underline{f}_{i_{ji}}} = \sum_k \delta_k^{(i_{ji})}; \quad \bar{\delta}_{i_{ji}} = -\frac{\partial E}{\partial \bar{f}_{i_{ji}}} = \sum_k \bar{\delta}_k^{(i_{ji})} \quad (40)$$

where $\underline{\delta}_k^{(i_{ji})}$ (resp. $\bar{\delta}_k^{(i_{ji})}$) is the lower (resp. upper) bound of the error signal of $R_k^{(i_{ji})}$, and $R_k^{(i_{ji})}$ is a rule node inheriting the lower (resp. upper) firing strength of $\tilde{A}_{i_{ji}}$ by (8).

Next, to derive the updating steps for tuning the centers of the fuzzy labels embedded in the antecedent segments, one needs to compute the partial derivatives of (41):

$$\frac{\partial fo_{i_{ji}}}{\partial c_{i_{ji}}^L} = \begin{cases} \bar{f}_{i_{ji}} \frac{2(x_i - c_{i_{ji}}^L)}{\sigma_{i_{ji}}^2} & \text{if } x_i \leq c_{i_{ji}}^L \\ \underline{f}_{i_{ji}} \frac{2(x_i - c_{i_{ji}}^L)}{\sigma_{i_{ji}}^2} & \text{if } x_i > 1/2(c_{i_{ji}}^L + c_{i_{ji}}^R) \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial fo_{i_{ji}}}{\partial c_{i_{ji}}^R} = \begin{cases} \underline{f}_{i_{ji}} \frac{2(x_i - c_{i_{ji}}^R)}{\sigma_{i_{ji}}^2} & \text{if } x_i \leq 1/2(c_{i_{ji}}^L + c_{i_{ji}}^R) \\ \bar{f}_{i_{ji}} \frac{2(x_i - c_{i_{ji}}^R)}{\sigma_{i_{ji}}^2} & \text{if } x_i > c_{i_{ji}}^R \\ 0 & \text{otherwise} \end{cases} \quad (41)$$

Subsequently, the updating steps for the centers of the left and right formation gaussian functions of the antecedent labels are given as in (42):

$$c_{i_{ji}}^L(t+1) = c_{i_{ji}}^L(t) + \eta \begin{cases} \bar{\delta}_{i_{ji}}(t) & \text{if } x_i \leq c_{i_{ji}}^L \\ \underline{\delta}_{i_{ji}}(t) & \text{if } x_i > 1/2(c_{i_{ji}}^L + c_{i_{ji}}^R) \\ 0 & \text{otherwise} \end{cases} \left\{ \frac{\partial fo_{i_{ji}}(t)}{\partial c_{i_{ji}}^L(t)} \right.$$

$$c_{i_{ji}}^R(t+1) = c_{i_{ji}}^R(t) + \eta \begin{cases} \underline{\delta}_{i_{ji}}(t) & \text{if } x_i \leq 1/2(c_{i_{ji}}^L + c_{i_{ji}}^R) \\ \bar{\delta}_{i_{ji}}(t) & \text{if } x_i > c_{i_{ji}}^R \\ 0 & \text{otherwise} \end{cases} \left\{ \frac{\partial fo_{i_{ji}}(t)}{\partial c_{i_{ji}}^R(t)} \right. \quad (42)$$

where $\eta > 0$ is a learning constant.

References

- [1] W.C. Abraham, A. Robins, Memory retention – the synaptic stability versus plasticity dilemma, *Trends in Neurosciences* 28 (2005) 73–78.
- [2] K.K. Ang, C. Quek, RSPOP: rough set-based pseudo outer-product fuzzy rule identification algorithm, *Neural Computation* 17 (2005) 205–243.
- [3] P.P. Angelov, D.P. Filev, An approach to online identification of Takagi–Sugeno fuzzy models, *IEEE Transactions on Systems, Man, and Cybernetics – Part B* 34 (2004) 484–498.
- [4] A. Baraldi, P. Blonda, A survey of fuzzy clustering algorithms for pattern recognition – Part I, *IEEE Transactions on Systems, Man, and Cybernetics – Part B* 29 (1999) 778–785.
- [5] H.R. Berenji, P. Khedkar, Learning and tuning fuzzy logic controllers through reinforcements, *IEEE Transactions on Neural Networks* 3 (1992) 724–740.
- [6] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, New York, 1981.
- [7] J.C. Bezdek, E.C.-K. Tsao, N.R. Pal, Fuzzy Kohonen clustering networks, *IEEE Conference on Fuzzy Systems*, 1992, pp. 1035–1043.
- [8] M. Biglarbegian, W. Melek, J. Mendel, On the robustness of Type-1 and interval Type-2 fuzzy logic systems in modeling, *Information Sciences* 181 (2011) 1325–1347.
- [9] D. Driankov, H. Hellendoom, M. Reinfrank, *An Introduction to Fuzzy Control*, Springer-Verlag, New York, 1996.
- [10] M.J. Gacto, R. Alcalá, F. Herrera, Interpretability of linguistic fuzzy rule-based systems: an overview of interpretability measures, *Information Sciences* 181 (2011) 4340–4360.
- [11] S. Grossberg, *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*, Reidel Press, Boston, 1982.
- [12] S. Guillaume, Designing fuzzy inference systems from data: an interpretability-oriented review, *IEEE Transactions on Fuzzy Systems* 9 (2001) 426–443.
- [13] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 1998.
- [14] I. Horenko, On clustering of non-stationary meteorological time series, *Dynamics of Atmospheres and Oceans* 49 (2010) 164–187.
- [15] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference systems, *IEEE Transactions on Systems, Man, and Cybernetics* 23 (1993) 665–685.
- [16] C.F. Juang, C.T. Lin, An on-line self-constructing neural fuzzy inference network and its applications, *IEEE Transactions on Fuzzy Systems* 6 (1998) 12–32.
- [17] C.F. Juang, Y.W. Tsao, A self-evolving interval Type-2 fuzzy neural network with online structure and parameter learning, *IEEE Transactions on Fuzzy Systems* 16 (2008) 1411–1424.

- [18] C.F. Juang, Y.W. Tsao, A Type-2 self-organizing neural fuzzy system and its FPGA implementation, *IEEE Transactions on Systems, Man, and Cybernetics – Part B* 38 (2008) 1537–1548.
- [19] C.F. Juang, C.H. Hsu, Reinforcement interval Type-2 fuzzy controller design by online rule generation and Q-value-aided ant colony optimization, *IEEE Transactions on Systems, Man, and Cybernetics – Part B* 39 (2009) 1528–1542.
- [20] C.F. Juang, R.B. Huang, W.Y. Cheng, An interval Type-2 fuzzy-neural network with support-vector regression for noisy regression problems, *IEEE Transactions on Fuzzy Systems* 18 (2010) 686–699.
- [21] N.N. Karnik, J.M. Mendel, Introduction to Type-2 fuzzy logic systems, *IEEE World Congress on Computational Intelligence* (1998) 915–920.
- [22] N.N. Karnik, J.M. Mendel, Q. Liang, Type-2 fuzzy logic systems, *IEEE Transactions on Fuzzy Systems* 7 (1999) 643–658.
- [23] N. Kasabov, Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning, *IEEE Transactions on Systems, Man, and Cybernetics – Part B* 31 (2001) 902–918.
- [24] N. Kasabov, Q. Song, DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction, *IEEE Transactions on Fuzzy Systems* 10 (2002) 144–154.
- [25] T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological Cybernetics* 43 (1982) 59–69.
- [26] Q. Liang, J.M. Mendel, Interval Type-2 fuzzy logic systems: theory and design, *IEEE Transactions on Fuzzy Systems* 8 (2000) 535–550.
- [27] C.J. Lin, C.T. Lin, An ART-based fuzzy adaptive learning control network, *IEEE Transactions on Fuzzy Systems* 5 (1997) 477–496.
- [28] C.T. Lu, Y. Kou, J. Zhao, L. Chen, Detecting and tracking regional outliers in meteorological data, *Information Sciences* 177 (2007) 1609–1632.
- [29] M. Mackey, L. Glass, Oscillation and chaos in physiological control systems, *Science* 197 (1977) 287–289.
- [30] E.H. Mamdani, Application of fuzzy logic to approximate reasoning using linguistic systems, *IEEE Transactions on Computers* 26 (1977) 1182–1191.
- [31] S. Maren, M. Baudry, Properties and mechanisms of long-term synaptic plasticity in the mammalian brain: relationships to learning and memory, *Neurobiology of Learning and Memory* 63 (1995) 1–18.
- [32] J.M. Mendel, *Uncertain Rule-based Fuzzy Logic Systems: Introduction and New Directions*, Prentice Hall, Upper Saddle River, 2001.
- [33] J.M. Mendel, R.I.B. John, Type-2 fuzzy sets made simple, *IEEE Transactions on Fuzzy Systems* 10 (2002) 117–127.
- [34] S. Mitra, Y. Hayashi, Neuro-fuzzy rule generation: survey in soft computing framework, *IEEE Transactions on Neural Networks* 11 (2000) 748–768.
- [35] W. Pedrycz, F. Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*, MIT Press, 1998.
- [36] C. Quek, R.W. Zhou, The POP learning algorithms: reducing work in identifying fuzzy rules, *Neural Networks* 14 (2001) 1431–1445.
- [37] A. Riid, E. Rüstern, Identification of transparent, compact, accurate and reliable linguistic fuzzy models, *Information Sciences* 181 (2011) 4378–4393.
- [38] I. Rojas, H. Pomares, J. Ortega, A. Prieto, Self-organized fuzzy system generation from training examples, *IEEE Transactions on Fuzzy Systems* 8 (2000) 23–36.
- [39] H.J. Rong, N. Sundararajan, G.B. Huang, P. Saratchandra, Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction, *Fuzzy Sets and System* 157 (2006) 1260–1275.
- [40] P. Shenoy, M. Krauledat, B. Blankertz, R.P.N. Rao, K.R. Müller, Towards adaptive classification for BCI, *Journal of Neural Engineering* 3 (2006) 13–23.
- [41] G.K. Tan, Feasibility of predicting congestion states with neural network models, Final Year Project, School of Civil and Structural Engineering, Nanyang Technological University, 1997.
- [42] S.D. Teddy, E.M.-K. Lai, C. Quek, A cerebellar associative memory approach to option pricing and arbitrage trading, *Neurocomputing* 71 (2008) 3303–3315.
- [43] S.W. Tung, C. Quek, C. Guan, SaFIN: a self-adaptive fuzzy inference network, *IEEE Transactions on Neural Networks* 22 (2011) 1928–1940.
- [44] W.L. Tung, C. Quek, DIC: a novel discrete incremental clustering technique for the derivation of fuzzy membership functions, *Pacific Rim International Conferences on Artificial Intelligence* (2002) 178–187.
- [45] W.L. Tung, C. Quek, GenSoFNN: a generic self-organizing fuzzy neural network, *IEEE Transactions on Neural Networks* 13 (2002) 1075–1086.
- [46] W.L. Tung, C. Quek, GenSo-OPATS: a brain-inspired dynamically evolving option pricing model and arbitrage trading system, *IEEE Congress on Evolutionary Computation* (2005) 2429–2436.
- [47] W.L. Tung, C. Quek, eFSM – a novel online neural-fuzzy semantic memory model, *IEEE Transactions on Neural Networks* 21 (2010) 136–157.
- [48] S.G. Tzafestas, K.C. Zikidis, NeuroFAST: on-line neuro-fuzzy ART-based structure and parameter learning TSK model, *IEEE Transactions on Systems, Man, and Cybernetics – Part B* 31 (2001) 797–802.
- [49] C. Vidaurre, A. Schlogl, R. Cabeza, R. Scherer, G. Pfurtscheller, Study of on-line adaptive discriminant analysis for EEG-based brain computer interfaces, *IEEE Transactions on Biomedical Engineering* 54 (2007) 550–556.
- [50] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Transactions on Systems, Man, and Cybernetics* 22 (1992) 1414–1427.
- [51] S. Wu, M.J. Er, Y. Gao, A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks, *IEEE Transactions on Fuzzy Systems* 9 (2001) 578–594.
- [52] L.X. Wang, *Adaptive Fuzzy Systems and Control*, Prentice-Hall, New Jersey, 1994.