SaFIN: A Self-Adaptive Fuzzy Inference Network

Sau Wai Tung, *Student Member, IEEE*, Chai Quek, *Senior Member, IEEE*, and Cuntai Guan, *Senior Member, IEEE*

***Abstract*— There are generally two approaches to the design of a neural fuzzy system: 1) design by human experts, and 2) design through a self-organization of the numerical training data. While the former approach is highly subjective, the latter is commonly plagued by one or more of the following major problems: 1) an inconsistent rulebase; 2) the need for prior knowledge such as the number of clusters to be computed; 3) heuristically designed knowledge acquisition methodologies; and 4) the stability–plasticity tradeoff of the system. This paper presents a novel self-organizing neural fuzzy system, named *Self-Adaptive Fuzzy Inference Network* (SaFIN), to address the aforementioned deficiencies. The proposed SaFIN model employs a new clustering technique referred to as *categorical learning-induced partitioning* (CLIP), which draws inspiration from the behavioral category learning process demonstrated by humans. By employing the one-pass CLIP, SaFIN is able to incorporate new clusters in each input–output dimension when the existing clusters are not able to give a satisfactory representation of the incoming training data. This not only avoids the need for prior knowledge regarding the number of clusters needed for each input–output dimension, but also allows SaFIN the flexibility to incorporate new knowledge with old knowledge in the system. In addition, the self-automated rule formation mechanism proposed within SaFIN ensures that it obtains a consistent resultant rulebase. Subsequently, the proposed SaFIN model is employed in a series of benchmark simulations to demonstrate its efficiency as a self-organizing neural fuzzy system, and excellent performances have been achieved.**

***Index Terms*— Categorical learning-induced partitioning, fuzzy neural networks, hybrid intelligent systems, self-organizing.**

I. Introduction

NEURAL fuzzy systems are hybrid systems that capitalize on the functionalities of fuzzy systems and neural networks [1]. The black-box nature of a neural network can be resolved by injecting the interpretability of a fuzzy system into the connectionist structure, while marrying the learning powers of a neural network into a fuzzy system enables the system to automatically refine its parameters. Despite these improved advantages over its individual predecessors, there are two main concerns when designing a neural fuzzy system, namely, the fuzzy partitioning of the input and output spaces, and the rule-generation scheme adopted. Fuzzy partitioning for a neural fuzzy system determines the numbers, positions, and spreads of the linguistic labels in each input–output dimension, while the rule generation scheme determines the set of fuzzy rulebase/knowledge base governing the neural fuzzy system.

Existing neural fuzzy systems can be broadly classified into two classes: design by expert knowledge and design from data. In the former class, the computational structure of a neural fuzzy system is manually crafted by human experts. Prior knowledge for the design of fuzzy partitioning in the input–output spaces and the fuzzy rulebase of the system are determined by the human experts. Subsequently, only the parameters of the system are fine-tuned in the learning process. Since the primary source of knowledge for the design of such a neural fuzzy system stems from the limited competence of each user, the result is often highly subjective and speculative. In addition, the difficulties in verbally formalizing interactions in a complex application environment may result in an inconsistent rulebase, thus leading to a loss of accuracy [2]. Some examples of neural fuzzy systems in this class include the ANFIS [3] and the GARIC [4] models.

The second class of neural fuzzy systems, referred to as self-organizing neural fuzzy systems, is based on a self-reliant and automated design from numerical data. By integrating self-organizing numerical methods into the learning mechanisms, the positions and spreads of the linguistic labels in each input-output space can be determined from the training dataset. In addition, no initial rulebase needs to be specified prior to the training phase of the system due to the self-automated rule generation mechanism adopted. Fig. 1 shows the shift in the approaches used in designing neural fuzzy systems over the past two decades. As seen from the timeline, there has been an increasing interest in the study of self-organizing neural fuzzy systems mainly due to their automated and self-reliant designs which free them from the Achilles' heel suffered by the former class of neural fuzzy systems [2]. Nevertheless, existing self-organizing neural fuzzy systems suffer from one or more of the following major drawbacks: 1) an inconsistent rulebase; 2) the need for prior knowledge such as the number of clusters to be computed; 3) heuristically designed knowledge acquisition methodologies; and 4) the stability–plasticity tradeoff of the system.

First, a consistent and unique rulebase allows a logical and intuitive interpretation of the knowledge represented within the computational structure of the neural fuzzy system [5]. On the other hand, an inconsistent rulebase occurs when there exist two rules in the rulebase of the system such that the precedent conditions are similar but the resultant consequences differ. Subsequently, totally conflicting outcomes may appear when

Manuscript received October 3, 2010; revised June 16, 2011; accepted August 25, 2011. Date of publication October 18, 2011; date of current version December 1, 2011. This work was supported in part by the Agency for Science, Technology and Research (A*Star) under the A*Star Graduate Scholarship Program.

S. W. Tung and C. Quek are with the Centre for Computational Intelligence, School of Computer Engineering, Nanyang Technological University, 639798, Singapore (e-mail: swtung@pmail.ntu.edu.sg; ashcquek@ntu.edu.sg).

C. Guan is with the Institute for Infocomm Research, A*Star, 138632, Singapore (e-mail: ctguan@i2r.a-star.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2167720

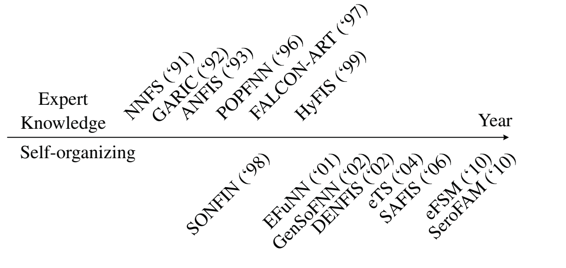


Fig. 1. Timeline reflecting the types of neural fuzzy systems in the literature. (Ref: NNFS [6], GARIC [4], ANFIS [3], POPFNN [7], FALCON-ART [8], SONFIN [9], HyFIS [10], EFuNN [11], GenSoFNN [12], DENFIS [13], eTS [14], SAFIS [15], eFSM [16], SeroFAM [17].)

the system encounters an input that triggers the activations of the two rules simultaneously. This inconsistent encryption of knowledge not only causes confusion for the human users, but also results in a meaningless and obscure description to the computational structure of the neural fuzzy system.

A second important consideration in the design of a neural fuzzy system is the choice of the fuzzy partitioning technique adopted. In order to directly capture and utilize the knowledge embedded in the raw numerical training data, numerical methods such as fuzzy Kohonen partitioning [18], fuzzy C-means [19], and linear vector quantization [20] were introduced to perform partitioning of the input–output spaces. In addition, arbitrary initialization of the input–output spaces was also proposed in [21]. A more comprehensive review on neural network-based clustering techniques is reported in [22]. Despite minimizing the amount of subjective information from human experts, these numerical methods still require prior knowledge regarding the number of clusters necessary in each input–output dimension. Such information is often arbitrarily determined at the beginning, and the numbers of clusters remain fixed throughout the entire learning process. Consequently, the system might not be able to achieve a desirable performance for the application problem due to either an over- or under-generalization restricted by the rigid amount of resources made available in the system. In addition, the predefined number of clusters in each input–output dimension also deprives the neural fuzzy system from subsequently incorporating new clusters upon the completion of the training process. This is an intricate situation known as the *stability–plasticity tradeoff* [23] where a regulated balance needs to be maintained for the coexistence of past knowledge and any future knowledge such that a current and up-to-date system is achieved for the modeling of the application environment.

This paper presents the *Self-Adaptive Fuzzy Inference Network* (SaFIN), a novel self-organizing neural fuzzy system, which addresses the above-stated deficiencies faced by existing neural fuzzy systems. A fully data-driven approach is employed for the automated formulation of fuzzy rulebase in the SaFIN model. This approach ensures that the system maintains a consistent knowledge encryption. In addition, the SaFIN model employs a new clustering technique known as the *categorical learning-induced partitioning* (CLIP), inspired from the behavioral category learning process demonstrated by humans. CLIP is a single-pass fuzzy partitioning technique that performs clustering of the input–output dimensions based on knowledge extracted from each training tuple. By considering the significance of the incoming sample with respect to any existing clusters in the input–output dimensions, the SaFIN model is able to automatically learn the presence of a new cluster. As a result, it frees the human user from predetermining the number of clusters present, while a balance is struck between the new knowledge from the current incoming training sample and the existing knowledge in the system such that old and new knowledge is able to codefine the structure of the model. Unlike previous heuristic clustering approaches proposed in the neural fuzzy paradigm [11]– [16], the SaFIN model employs a top-down approach in its clustering mechanism which draws inspiration from the human-based category learning process [24]– [29].

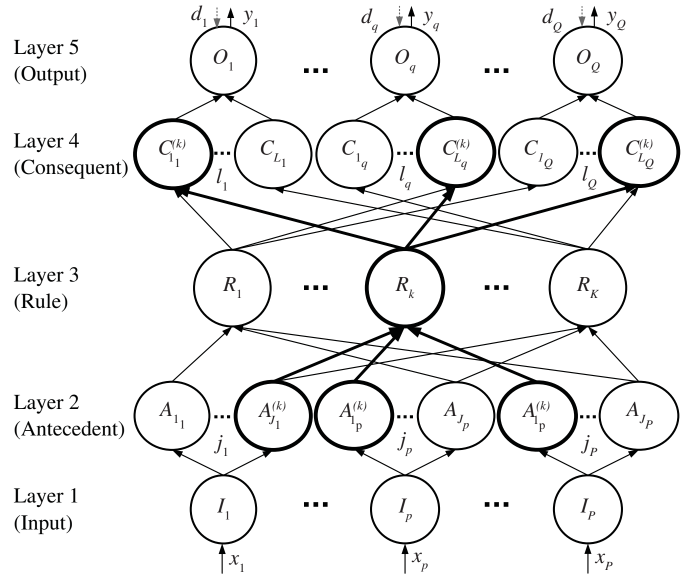


Fig. 2. Architecture of the SaFIN.

The rest of this paper is organized as follows. The computational structure and the reasoning process of the SaFIN model are described in Section II. The proposed learning mechanisms of SaFIN are introduced in Section III. Section IV evaluates the learning and self-organizing abilities of SaFIN through a series of benchmark experimental simulations. Lastly, Section V concludes this paper.

II. SaFIN

This section describes the computational structure and the reasoning process of the proposed SaFIN.

A. Architecture of SaFIN

The proposed SaFIN model is a five-layered neural fuzzy system as shown in Fig. 2. Layer 1 consists of the input (variable) nodes, layer 2 is the antecedent nodes, layer 3 is the rule nodes, layer 4 consists of the consequent nodes, and layer 5 is the output (variable) nodes. In the SaFIN model, the input vector is denoted as $x = \left(x_1, \ldots, x_p, \ldots, x_P\right)$. The corresponding desired output vector is denoted as $d = \left(d_1, \ldots, d_q, \ldots, d_Q\right)$, while the computed output is denoted as $y = (y_1, \ldots, y_q, \ldots, y_Q)$. The notations used in Fig. 2 are defined as follows:

P	number of input dimensions
Q	number of output dimensions
I_p	pth input node
O_q	qth output node
J_p	number of fuzzy clusters in I_p
L_q	number of fuzzy clusters in O_q
A_{j_p}	jth antecedent fuzzy cluster in I_p
C_{l_q}	lth consequent fuzzy cluster in O_q
K	number of fuzzy rules
R_k	kth fuzzy rule.

Layer 3 of SaFIN encrypts the rulebase of the system where each rule node encodes an IF-THEN Mamdani-type fuzzy rule [30] given as in (1)

$$\begin{aligned} R_k: \quad & \text{IF } x_1 \text{ is } A_{j_1}^{(k)} \text{ and } \ldots \text{ and } x_P \text{ is } A_{j_P}^{(k)} \\ & \text{THEN } y_1 \text{ is } C_{l_1}^{(k)} \text{ and } \ldots \text{ and } y_Q \text{ is } C_{l_Q}^{(k)} \end{aligned} \tag{1}$$

where $A_{j_p}^{(k)}$ (resp. $C_{l_q}^{(k)}$) is the jth antecedent (resp. lth consequent) node associated with the pth input (resp. qth output) variable that is connected to the rule node R_k, see Fig. 2. The tunable parameters of SaFIN are the centers and widths of the fuzzy labels embedded in the antecedent and consequent nodes, where each node defines a Gaussian membership function described as in (2)

$$\mu(c, \sigma; x) = e^{-((x-c)^2/\sigma^2)} \tag{2}$$

such that c and σ are the center and width of the function, respectively. Adaptation of the parameters is performed using the neural network-based gradient descent approach [31].

B. Reasoning Process of SaFIN

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

The generic operations for the proposed SaFIN model are defined as follows: the activation functions of each layer $M \in \{1 \ldots 5\}$ are denoted as $a^{(M)}$, and the corresponding output for an arbitrary node is denoted as o.

Layer 1: The function of the input nodes is to directly pass on the input vector to the next layer. Hence, the neural operation of I_p can be described as in (3)

$$o_p = a^{(1)}(x_p) = x_p \,. \tag{3}$$

Layer 2: The fuzzy labels A_{j_p} define the antecedent segments of the Mamdani-type rules described as in (1) where each label is defined as a Gaussian function as described in (2). The function of layer 2 of SaFIN is to perform antecedent matching of the input training value with the respective input labels such that the degree of similarity between x_p and the membership function embedded in A_{j_p} is computed. Hence, the neural computation of A_{j_p} can be described as in (4)

$$o_{j_p} = a^{(2)}(o_p) = \mu_{j_p}(c_{j_p}, \sigma_{j_p}; x_p) \tag{4}$$

where $\mu_{j_p}(c_{j_p}, \sigma_{j_p}; x)$ refers to the Gaussian membership function embedded in the node A_{j_p}.

Layer 3: The set of Mamdani-type rules that is induced from the training data is defined in the rule layer of the SaFIN model. Each rule node R_k computes the overall degree of similarity between the input training vector and the antecedent part of the kth fuzzy rule. Hence, the firing rate of R_k is computed as in (5), where

$$o_k = a^{(3)}\left(o_{j_1}^{(k)}, \ldots, o_{j_P}^{(k)}\right) = \min_{p \in \{1 \ldots P\}} o_{j_p}^{(k)} \,. \tag{5}$$

Layer 4: This layer of SaFIN consists of the fuzzy labels C_{l_q} that define the consequent segments of the Mamdani-type 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 input vector x. Since C_{l_q} may serve as output to more than one fuzzy rule, the cumulative neural computation for C_{l_q} can be described as in (6)

$$o_{l_q} = a^{(4)}\left(o_1^{(l_q)}, \ldots, o_{K_{l_q}}^{(l_q)}\right) = \max_{k \in \{1 \ldots K_{l_q}\}} o_k^{(l_q)} \tag{6}$$

where K_{l_q} is the total number of fuzzy rules in SaFIN that shares the same consequent node C_{l_q} and $o_k^{(l_q)}$ is the output of the kth fuzzy rule that shares C_{l_q}.

Layer 5: The function of the output nodes is to perform defuzzification to obtain a crisp output value. This is achieved using the center of averaging method [32] such that the neural operation of O_q can be described as in (7)

$$\begin{aligned} y_q = o_q &= a^{(5)}\left(o_{1_q}, \ldots, o_{L_q}\right) \\ &= \frac{\sum_{l_q \in \{1_q \ldots L_q\}} o_{l_q} c_{l_q} \sigma_{l_q}}{\sum_{l_q \in \{1_q \ldots L_q\}} o_{l_q} \sigma_{l_q}} \end{aligned} \tag{7}$$

where c_{l_q} and σ_{l_q} are the center and width of the Gaussian function embedded in C_{l_q}, respectively.

III. Learning Mechanisms in SaFIN

There are two main components to the learning process of the proposed SaFIN model: the fuzzy partitioning of the input–output dimensions and the rule generation procedure. Initially, there are neither fuzzy partitionings in the input–output spaces nor fuzzy rules in the system, i.e., there are no nodes in the hidden layers 2–4. A localized learning of the fuzzy labels is carried out such that the numbers of clusters in each input–output dimension can differ depending on the knowledge extracted from the numerical data. The numbers, positions, and spreads of the fuzzy labels are self-determined from the training dataset. New clusters are incorporated into the system when the knowledge extracted from the incoming training tuple is novel as compared to existing clusters in the system. Refinements are then made to the existing clusters such that old knowledge in the system and new knowledge from the incoming training tuple can coexist to provide a more accurate representation of the training data.

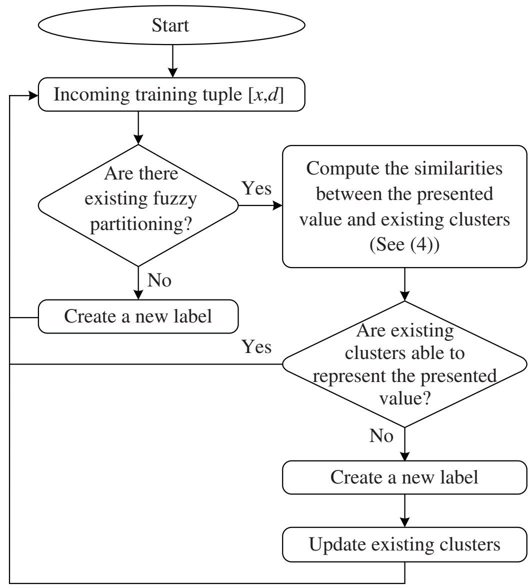


Fig. 3. Flowchart of the self-organizing clustering technique CLIP adopted in SaFIN.

This tailored approach addresses the stability–plasticity of SaFIN. Hence, these advantages over existing techniques serve as the main motivations to the development of the CLIP technique employed in the SaFIN model. As observed in [25], global conceptual categories such as *animals* and *vehicles* are the first level of distinctions made by young infants during category learning. Subsequently, high-contrast basic-level distinctions within each domain (for example, *dogs* versus *fish*, or *cars* versus *airplanes*) are being observed in older infants. This is then followed by moderate-contrast (*dogs* versus *rabbits*, or *cars* versus *motorcycles*) and low-contrast (*dogs* versus *horses*, or *cars* versus *trucks*) basic-level distinctions demonstrated in older children while performing category learning. By employing a similar divisive top-down approach, the CLIP technique draws inspiration from the behavioral category learning process exhibited by humans. (This is an efficient method of rapid knowledge acquisition that helps to expand the initial conceptual system formulated during infancy [25].) Details on the proposed self-organizing CLIP clustering technique will be presented in Section III-A.[1]

The second key component in the design process of SaFIN is the formulation of the rulebase. By employing a self-automated rule generation mechanism, no initial rulebase needs to be prespecified by the human expert. Firstly, a fuzzy rule is formulated to capture the knowledge from each of the incoming training tuples. Following that, each of the fuzzy rules in the rulebase is assigned a weightage depicting its significance in the modeling of the application environment. Conflicting rules with low influences are deemed as outliers and are subsequently deleted from the system. This approach ensures that the proposed SaFIN model maintains a consistent rulebase that is able to provide an aptly description to the application problem. Details on the self-automated rule generation mechanism will be presented in Section III-B.

[1]In this paper, CLIP is employed in a Type-1 neural fuzzy system. An application of CLIP in a Type-2 system can be found in [33].

A. Self-Organizing Clustering in SaFIN

Learning in the proposed SaFIN model begins with extracting and utilizing knowledge from each incoming training tuple to establish an initial fuzzy partitioning. Fig. 3 shows a flowchart of the self-organizing clustering technique, CLIP, adopted in SaFIN. With the arrival of each training tuple $[x, d]$, fuzzy partitioning is performed independently for each input–output dimension. If there is no existing fuzzy partitioning in an input dimension, then a new cluster spanning the entire domain is formed. This conforms to the initial global conceptual categories conceived by humans. For example, young infants first see a dog (a particular instance of the global concept *animal*) and associate the entire domain of *animal* as a dog. It is only much later in the process of category learning that the basic concept *dog* is established [29]. By translating this initial knowledge of the global concept, the formation of the first fuzzy cluster A_{1_p} in an input dimension p can be described using (8)

$$\begin{aligned} c_{1_p} &= x_p \\ \sigma_{1_p} &= R\left(\sqrt{-\frac{(\min_p - x_p)^2}{\log\alpha}}, \sqrt{-\frac{(\max_p - x_p)^2}{\log\alpha}}\right) \end{aligned} \quad (8)$$

where c_{1_p} and σ_{1_p} are the center and width of the Gaussian membership function embedded in A_{1_p}. A newly created membership function is centered upon the presented value, while $R(\sigma_1, \sigma_2) := 1/2\,[\sigma_1 + \sigma_2]$ defines a *regulator function* that ensures a fuzzy cluster has distinct semantic meaning by maintaining a reasonable amount of buffer on either sides of its center. Here, the boundary for the domain is given as $\left[\min_p, \max_p\right]$. The *minimum membership threshold* α is defined such that the membership value of any point in the domain should be at least α before regulation. The same initialization procedure is repeated for each output dimension.

Fig. 4(A) shows the formation of an initial cluster in each input–output dimension. For the special case where the presented value coincides with the midpoint of the domain, the corresponding fuzzy partitioning is depicted as Fig. 4(A.a). Since the center of the newly created cluster is equidistant from the lower and upper bounds, the membership function obtained before and after regulation is similar where the boundary points have the minimum membership value α. On the other hand, for an initial cluster created in an input–output dimension where the center is skewed from the midpoint of the domain, the spreads of the cluster on either sides of the center depend on the distance of the center to the respective boundary points. This is illustrated in Fig. 4(A.b). As seen from the figure, the spread on the side of the cluster nearer to a boundary point can be significantly smaller than the opposite side which is further away from the other boundary point. In the extreme case where the first presented value coincides with one of the boundary points, there will be no spread on one side of the cluster while the opposing side has a very wide spread. Such clusters deter a clear interpretation of the knowledge encoded in the proposed SaFIN model. Hence,

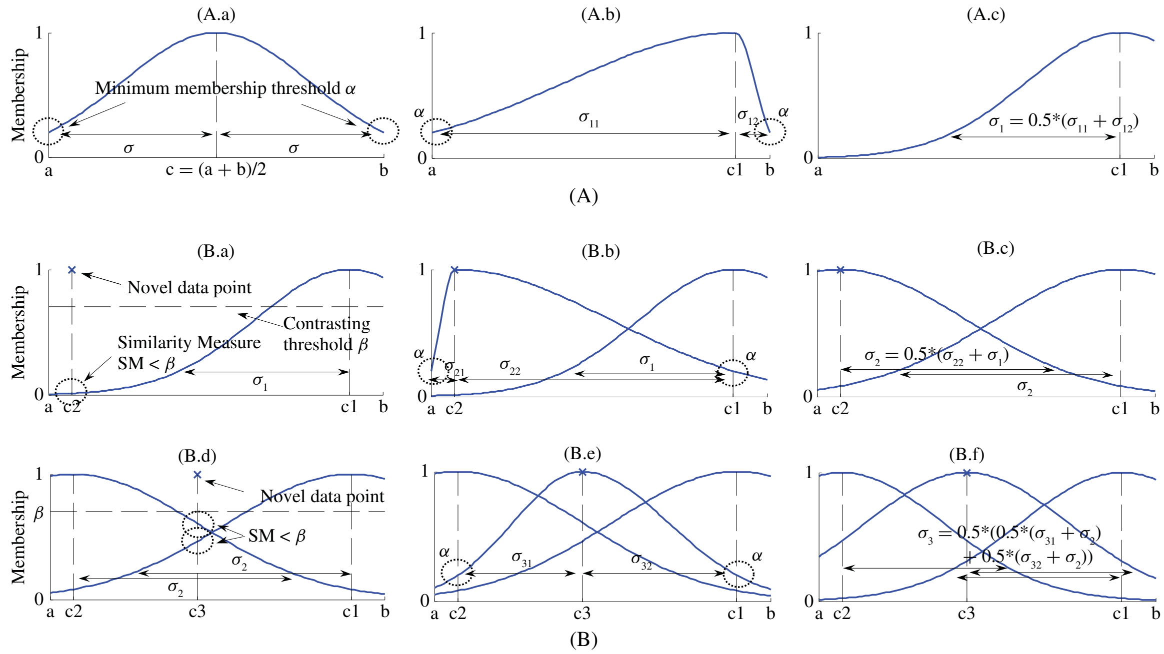


Fig. 4. (A) Initialization. (a) First cluster formed in each input–output dimension before/after regulation if the center coincides with the midpoint of the domain. (b) First cluster formed in an input–output dimension *before* regulation for a center that is skewed from the midpoint of the domain. (c) First cluster formed in an input–output dimension *after* regulation for a center that is skewed from the midpoint of the domain. (B) Clustering. (a) Introduction of a novel data point with no left neighbor. (b) Creation of a new cluster based on the novel point before regulation. (c) Final appearance of the fuzzy partitionings in the input–output dimension after regulation. (d) Introduction of a novel data point with both left and right neighbors. (e) Creation of a new cluster based on the novel point before regulation. (f) Final appearance of the fuzzy partitionings in the input–output dimension after regulation.

regulation is performed to ensure that each cluster created in an input–output dimension maintains its relevance and a distinct semantic meaning. Fig. 4(A.c) shows the resultant cluster after performing regulation on the initially created cluster depicted in Fig. 4(A.b). As clearly seen, an equal and reasonable amount of spread is maintained on both sides of the cluster center to ensure that the cluster has clear semantic meaning.

On the other hand, if there is an existing fuzzy partitioning in an input–output dimension, then SaFIN proceeds with computing the similarities between the presented value and existing clusters. As seen from (4), the similarity match between an input value x_p and an existing cluster in the pth input dimension is given as $SM\left(x_p, A_{j_p}\right) := \mu_{j_p}(c_{j_p}, \sigma_{j_p}; x_p)$. Subsequently, SaFIN identifies the best matched fuzzy cluster in an input dimension via the computed similarity values, i.e., the best matched fuzzy cluster is denoted as $A_{j_p^\star}$ where $j_p^\star = \arg\max_{j_p} SM\left(x_p, A_{j_p}\right)$. If the similarity match between the identified best matched cluster and the presented value in an input dimension exceeds a *contrasting threshold* β, the best matched cluster is deemed as being able to give a satisfactory description of the presented value. However, if the similarity match falls below β, then a new cluster is created in the input dimension based on the presented value. Similar approach is used in the SONFIN model [9] for the discovery of new information. However, unlike SONFIN, where old knowledge in the system remains unchanged prior to parameter learning, adjustments and refinements are subsequently made to the existing clusters in that dimension of SaFIN to incorporate the newly created cluster. This process effectively addresses the stability–plasticity of SaFIN by allowing the old knowledge to coexist with the new information derived from the training data to define the rulebase. Specifically, the existing concepts (fuzzy sets) in the input dimension are refined and adapted to enhance their representations of the training data encountered. As stated in [29], "every time a new category is learned within the global domains, it is necessarily learned as a subdivision of this larger division." By comparing a new exemplar with existing concepts (fuzzy sets) in an input dimension via the computed similarity values, a conceptually distinct category will be formed by SaFIN when a prominent distinction is observed. This divisive top-down category learning mechanism observed in humans is emulated for the creation of a new fuzzy cluster in the proposed SaFIN model. The formation of a new fuzzy cluster $A_{J_p(t)+1}$ in the pth input dimension can be described using (9)

$$\begin{aligned} c_{J_p(t)+1} &= x_p \\ \sigma_{J_p(t)+1} &= \begin{cases} \sigma^R & \text{if } j_p^L = \text{NULL} \\ \sigma^L & \text{if } j_p^R = \text{NULL} \\ R\left(\sigma^R, \sigma^L\right) & \text{otherwise} \end{cases} \end{aligned} \tag{9}$$

where

$$\sigma^R = R\left(\sqrt{-\frac{\left(c_{j_p^R} - x_p\right)^2}{\log\alpha}}, \sigma_{j_p^R}(t)\right)$$

$$\sigma^L = R\left(\sqrt{-\frac{\left(c_{j_p^L} - x_p\right)^2}{\log\alpha}}, \sigma_{j_p^L}(t)\right).$$

The immediate left and right neighbors of the newly created cluster are denoted as $A_{j_p^L}$ and $A_{j_p^R}$, respectively, where

$$j_p^L = \begin{cases} \text{NULL} & \text{if } c_{j_p} \geq x_p \\ & \quad \text{for } 1 \leq j_p \leq J_p(t) \\ \arg\min_{c_{j_p} < x_p} |c_{j_p} - x_p| & \text{otherwise} \end{cases}$$

$$j_p^R = \begin{cases} \text{NULL} & \text{if } c_{j_p} \leq x_p \\ & \quad \text{for } 1 \leq j_p \leq J_p(t) \\ \arg\min_{c_{j_p} > x_p} |c_{j_p} - x_p| & \text{otherwise.} \end{cases}$$

For simplicity, only refinements are made to the immediate left and right neighbors of the newly created cluster. The refinements made can be classified into three cases.

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

The same fuzzy clustering process is performed for each output dimension.

Fig. 4(B) illustrates the fuzzy clustering process in an input–output dimension of the proposed SaFIN model. After an initial cluster has been formed, SaFIN continues to perform clustering when a novel data point is encountered, i.e., the computed similarity values between the presented point and the existing clusters fall below β. This is illustrated as Fig. 4(B.a). A new cluster is then created using the information derived from this novel data point as seen in Fig. 4(B.b). Since the distance of the presented value from the lower bound is significantly smaller than that from the upper bound, the spreads on either sides of the cluster center are unbalanced. As explained before, this will decrease the interpretability of the knowledge encoded in the proposed SaFIN model. Hence, regulation is performed to preserve a distinct semantic meaning of the newly created cluster as seen from Fig. 4(B.c). The existing cluster is simultaneously refined to incorporate the newly created cluster in this figure. Following that, Fig. 4(B.d) shows the arrival of a novel data point to the input-output dimension. A new cluster is created where it is centered upon the presented value in Fig. 4(B.e). The spreads of the cluster on either sides of the center depend on the distance of the center to the respective centers of its immediate neighbors. The centers of the immediate neighbors have the minimum membership value α. Finally, regulation and refinements are performed in Fig. 4(B.f).

B. Self-Automated Rule Generation in SaFIN

The proposed SaFIN model proceeds with rule generation after achieving fuzzy partitionings for the input–output spaces. Fig. 5 shows the two stages of the proposed self-automated rule generation mechanism: 1) rule creation, and 2) consistency check. Rule creation formulates an initial rulebase using information derived from the training dataset. With the arrival of each training tuple $[x, d]$, the best matched fuzzy cluster is found for each input–output dimension (see computations in Section III-A). The best matched fuzzy cluster in the pth input dimension is denoted as $A_{j_p^\star}$, while the best matched fuzzy cluster in the qth output dimension is denoted as $C_{l_q^\star}$. Subsequently, a fuzzy rule $R^\star$ is formulated where $\left\{A_{j_p^\star}\right\}_{p=1}^{P}$ and $\left\{C_{l_q^\star}\right\}_{q=1}^{Q}$ are the antecedent and consequent segments of $R^\star$, respectively. If $R^\star$ is novel, then it is inserted into the rulebase such that the new rule $R_{K(t)+1} = R^\star$. By performing novelty check, SaFIN is ensured a unique rulebase. The new rule $R_{K(t)+1}$ is then assigned an initial weightage of $Wt_{K(t)+1} = 1$. In the SaFIN model, weightage of a fuzzy rule depicts its significance in modeling the application environment. Hence, the higher the weightage of a fuzzy rule, the greater is its potential in modeling the application environment. On the other hand, $R^\star$ is not inserted into the rulebase if it is not novel. That is, there exists a fuzzy rule $R_{k^\star}$ in SaFIN such that $R_{k^\star}$ has the same antecedent and consequent segments as $R^\star$. Subsequently, the weightage of $R_{k^\star}$ is enhanced as $Wt_{k^\star}(t+1) = Wt_{k^\star}(t) + 1$.

Following the formulation of an initial rulebase, the proposed SaFIN model performs consistency check on the rulebase. The design process of SaFIN is completed if

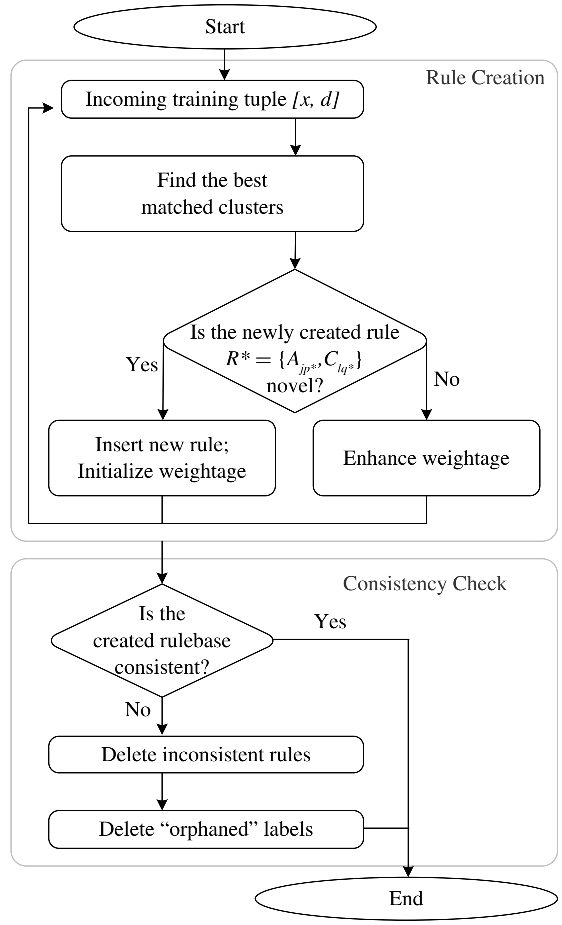


Fig. 5. Flowchart of the self-automated rule generation mechanism adopted in SaFIN.

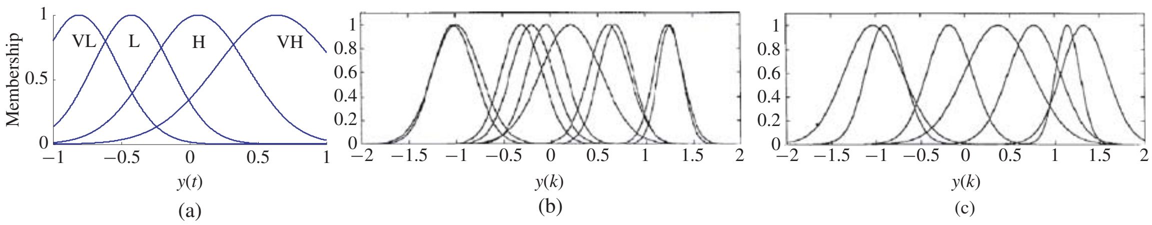


Fig. 6. Fuzzy partitioning for the nonlinear system in (a) SaFIN, (b) SONFIN *before* alignment, and (c) SONFIN *after* alignment.

the rulebase formulated is verified to be consistent. On the other hand, inconsistent rules with lower weightages are deleted for an inconsistent rulebase, i.e., if R_{k_1} and R_{k_2} are two rules in SaFIN with similar precedent conditions but differing consequences such that $Wt_{k_1} < Wt_{k_2}$, then $\{R_k\}_{k=1}^{K(t+1)} = \{R_k\}_{k=1}^{K(t)} \backslash R_{k_1}$. This approach of retaining the fuzzy rule with the strongest weightage not only ensures that the resultant rulebase is consistent but also ensures that the proposed system provides a most aptly description of the application environment. Finally, some of the fuzzy labels might be "orphaned" when all fuzzy rules associated with them have been deleted. For an input dimension p, an "orphaned" fuzzy label is identified by $A_{j_p^o} \notin R_k$ for all k. Subsequently, the "orphaned" label $A_{j_p^o}$ is removed to ensure that the resultant computational structure of SaFIN is compact, i.e., $\{A_{j_p}\}_{j_p=1_p}^{J_p(t+1)} = \{A_{j_p}\}_{j_p=1_p}^{J_p(t)} \backslash A_{j_p^o}$. The same deletion process is performed for each output dimension.

IV. Experimental Results and Analysis

This section illustrates the learning and self-organizing abilities of the proposed SaFIN model by employing it in four application areas: 1) the identification of a nonlinear system [9]; 2) the Nakanishi dataset [34], [35]; 3) the modeling of highway traffic flow density [36]; and 4) the UCI dataset [37].

A. Example 1 – Identification of a Nonlinear System

This experiment studies the sensitivity of the proposed learning mechanisms of SaFIN. The dataset is generated by a difference equation as described in (10)

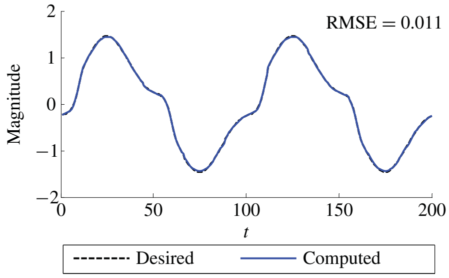
$$y\left(t+1\right) = \frac{y\left(t\right)}{1+y^2\left(t\right)} + u^3\left(t\right) \tag{10}$$

where 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 [9], 50 000 training and 200 testing data pairs are generated with initial conditions $(u(0), y(0)) = (0, 0)$. Subsequently, the SaFIN model is applied to this identification problem with $\alpha = 0.2$ and $\beta = 0.6$. Four fuzzy labels are identified by the CLIP algorithm for each of the input–output spaces respectively, while eight fuzzy rules are identified by the self-automated rule generation mechanism in SaFIN. As an illustration, Fig. 6 shows the fuzzy partitioning for the input space $y(t)$ for the proposed SaFIN model in (a), and that for the original problem derived by SONFIN in (b) and (c). As seen, the fuzzy clusters identified in SaFIN are highly ordered with distinct semantic meanings (i.e., VL–VERY LOW, L–LOW, H–HIGH, VH–VERY HIGH). Comparatively, Fig. 6(b) shows the initial identified fuzzy clusters in SONFIN. As seen, the fuzzy clusters are highly overlapping, making it difficult to induce any clear semantic meanings to the derived clusters. To tackle this problem, SONFIN performs an additional step to compute the similarity of a newly formed cluster with existing clusters in the input spaces, and subsequently aligns the new cluster. This improved result is shown in Fig. 6(c). Although the number of fuzzy clusters identified has reduced, the resultant fuzzy clusters still have a significant amount of overlap as seen in the first two clusters. This illustration shows the effectiveness of SaFIN as a neural fuzzy modeling tool where intuitive semantic labels can be directly identified from the raw numerical data for the input-output spaces. Fig. 7(a) shows the modeling results of the trained SaFIN model on the test dataset. As seen, there is a perfect match between the computed outputs of the network and the desired outputs of the system, with SaFIN achieving a root mean squared error (RMSE) of 0.011 on the testing dataset.

The proposed learning mechanisms of the SaFIN model are influenced by the preselected variables α and β. To provide a clearer understanding of the influences of these two parameters on the modeling performances of SaFIN, different values of them are tested. The test results concerning the number of identified fuzzy labels, fuzzy rules, and the RMSE on the test dataset are listed in Fig. 7(b). From the table, it is observed that the RMSE value remains consistent for certain ranges of the parameters. For a fixed value of $\alpha = 0.2$, the numbers of identified fuzzy labels in the two inputs–one output spaces increase as the contrasting threshold β increases. As a result, a higher value of β results in a decrease in the RMSE value, but at the expense of a larger rulebase. The same results hold for a constant value of $\beta = 0.6$ with a decreasing minimum membership threshold α. In conclusion, the larger the interval $[\alpha, \beta]$, the greater the numbers of identified fuzzy labels and fuzzy rules, and thus the lower the RMSE values. In this paper, the variables are chosen as follows: $\alpha \in [0.2, 0.4]$ and $\beta \in [0.5, 0.7]$ and/or with an average interval size of $[0.2, 0.5]$.

B. Example 2 – Nakanishi Dataset

The self-organizing abilities of the proposed SaFIN model are evaluated using three modeling experiments in the Nakanishi dataset [34], [35]: modeling of 1) a nonlinear system; 2) the human operation of a chemical plant; and 3) the daily pricing of a stock in a stock market. Following the descriptions from these papers, each of the three datasets is split into three groups A, B, and C, where A and B form the



(a)

β	# Fuzzy Labels		# Rules	RMSE
0.50	Inputs = 3,3;	Output = 3	6	0.017
0.55	4,4;	4	8	0.011
0.60	4,4;	4	8	0.011
0.65	6,6;	6	13	0.007
0.70	6,7;	6	14	0.007

$\alpha = 0.2$

α	# Fuzzy Labels		# Rules	RMSE
0.10	Inputs = 6,7;	Output = 6	14	0.006
0.15	6,6;	6	13	0.007
0.20	4,4;	4	8	0.011
0.25	4,4;	4	8	0.012
0.30	3,4;	3	6	0.023

$\beta = 0.6$

(b)

Fig. 7. (a) Modeling results for the nonlinear system when $\alpha = 0.2$, $\beta = 0.6$. (b) Sensitivity test for different values of α and β.

training dataset and C is the testing data. The benchmark for comparisons is the accuracies on the testing data (calculated as the mean squared error MSE) and the correspondence between the computed output with the testing data (calculated as the Pearson correlation coefficient R). The experimental results of SaFIN are subsequently benchmarked against the following models: Mamdani-type models—Hebb-R-R [38], POPFNN [7], RSPOP [39], and EFuNN [11], Reasoning models [34]— Sugeno P&P-G, Sugeno P, Sugeno P-G, Mamdani, and Turksen IVCRI, and Takagi–Sugeno–Kang (TSK)-type models [40]— ANFIS [3], DENFIS [13], anf FITSK [41].

1) Nonlinear System: The objective of this experiment is to identify and model the underlying principles of a nonlinear system. In the original dataset, there were four input (x_1–x_4) and one output (y) variables. Adopting the approach suggested in [34], only input variables x_1 and x_2 are subsequently used in the modeling. Correspondingly, a total of 3, 4, and 3 fuzzy clusters are identified by the proposed CLIP technique for the input–output dimensions. Fig. 8(a) illustrates the identified fuzzy clusters in the first input variable x_1. Since the fuzzy clusters are highly ordered, clear semantic meanings can be attached to the fuzzy clusters. This result demonstrates the excellent self-organizing clustering abilities of the proposed SaFIN model. A total of 10 Mamdani-type fuzzy rules are identified by SaFIN in this experiment. They are listed in Table I. The derived rulebase is consistent, as no two rules have similar antecedent conditions and different consequences. In addition, the knowledge embedded in the set of fuzzy rulebase is highly systematic and sound. This can be seen by examining the first four rules. Keeping x_1 as HIGH and varying x_2 from VERY HIGH to VERY LOW, the output y correspondingly varies from LOW to MED. This implies that y is inversely proportional to x_2 when x_1 is fixed. This relationship is repeatedly observed when x_1 is fixed at MED or LOW. Rearranging the rules, similar observations can be deduced between x_1 and y when x_2 is kept fixed. This result demonstrates the effectiveness of the self-automated rule generation mechanism proposed in SaFIN. Consolidated experimental results on the benchmarking measures are given in Table II.

2) Human Operation of a Chemical Plant: The proposed SaFIN model is employed to model the human operation of a chemical plant in this experiment. Although there were five input (x_1–x_5) and one output (y) variables in the original dataset, only selected input variables (x_1 and x_3) are used in this experiment [34]. Three fuzzy clusters are identified in each of the input–output dimensions. Fig. 8(b) illustrates the semantic fuzzy clusters in x_1 identified by SaFIN. By attaching semantic labels to the fuzzy clusters in each input–output dimension, Mamdani-type fuzzy rules can be subsequently extracted from the SaFIN model. The five extracted fuzzy rules are listed in Table I. Consistency of the fuzzy rulebase can be easily verified. Consolidated experimental results on the benchmarking measures are given in Table II.

TABLE I
MAMDANI-TYPE FUZZY RULES IDENTIFIED FOR THE NAKANISHI DATASET

	Nonlinear system			Chemical plant			Stock prediction			
Rule	x_1	x_2	y	x_1	x_3	y	x_4	x_5	x_8	y
1	H	VH	L	H	L	L	PH	PH	PH	NH
2	H	H	L	M	L	L	PH	PH	PL	NH
3	H	L	L	M	M	M	PM	PH	PH	NH
4	H	VL	M	L	M	H	PM	PH	PL	NL
5	M	VH	L	L	H	H	PM	PL	Z	NM
6	M	L	M				PL	PH	PH	NL
7	M	VL	M				PL	PL	Z	PH
8	L	VH	M				NL	PH	PH	NL
9	L	L	M				NL	PH	PL	NL
10	L	VL	H				NL	PH	Z	NL

L–LOW; M–MED; H–HIGH; Z–ZERO; N–NEG; P–POS; V–VERY

3) Daily Pricing of a Stock in a Stock Market: Using various statistics concerning a stock collected from a stock market, the proposed SaFIN model is employed to perform stock price prediction. Selected input variables (x_4, x_5, and x_8) are used although there were initially 10 input variables (x_1–x_{10}) [34]. Correspondingly, the number of fuzzy clusters identified for the three input variables and one output variable are 6, 4, 5, and 6. Fig. 8(c) illustrates the fuzzy clusters in x_4 identified by SaFIN. Clear and distinct semantic labels are attached to the fuzzy clusters. A total of 21 fuzzy rules are extracted from the SaFIN model, of which Table I lists the first 10 identified rules for this experiment. Consolidated experimental results on the benchmarking measures are given in Table II.

4) Discussion: Table II shows the consolidated experimental results for the Nakanishi dataset for the proposed SaFIN model and the benchmarking models. SaFIN outperforms all the benchmarking models in the first two tasks by ranking

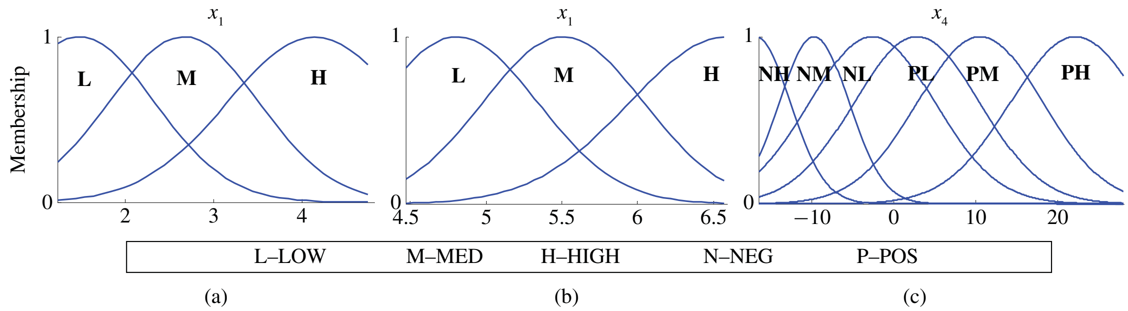


Fig. 8. Semantic fuzzy clusters in the first input dimensions of the Nakanishi dataset. (a) Nonlinear system. (b) Chemical plant. (c) Stock prediction.

TABLE II
CONSOLIDATED EXPERIMENTAL RESULTS FOR THE NAKANISHI DATASET

	Nonlinear system		Chemical plant		Stock prediction		
Model	MSE (Rank)	R (Rank)	MSE (Rank)	R (Rank)	MSE (Rank)	R (Rank)	Average Rank
Hebb-R-R	0.185 (2)	0.911 (2)	2.423×10^4 (2)	0.998 (2)	**15.14** (1)	**0.947** (1)	1.7
POPFNN	0.270 (3)	0.877 (3)	5.630×10^5 (8)	0.946 (9)	76.22 (10)	0.733 (10)	7.2
RSPOP	0.383 (7)	0.856 (4)	2.124×10^5 (5)	0.983 (7)	24.86 (3)	0.922 (2)	4.7
Sugeno P&P-G	0.345 (6)	0.828 (7)	2.897×10^5 (7)	0.973 (8)	94.58 (12)	0.706 (11)	8.5
Sugeno P	0.776 (12)	0.558 (12)	6.372×10^5 (9)	0.933 (12)	35.47 (5)	0.883 (4)	9
Sugeno P-G	0.467 (9)	0.845 (6)	1.931×10^6 (12)	0.990 (6)	168.9 (13)	0.700 (12)	9.7
Mamdani	0.862 (13)	0.490 (13)	6.580×10^5 (10)	0.937 (11)	40.84 (7)	0.865 (7)	10.2
Turksen IVCRI	0.706 (11)	0.609 (11)	2.581×10^5 (6)	0.993 (4)	93.02 (11)	0.661 (13)	9.3
ANFIS	0.286 (4)	0.853 (5)	2.968×10^6 (13)	0.780 (13)	38.06 (6)	0.875 (6)	7.8
EFuNN	0.566 (10)	0.720 (10)	7.247×10^5 (11)	0.946 (9)	72.54 (9)	0.756 (9)	9.7
DENFIS	0.411 (8)	0.805 (9)	5.240×10^4 (4)	0.995 (3)	69.82 (8)	0.810 (8)	6.7
FITSK	0.336 (5)	0.828 (7)	3.862×10^4 (3)	0.993 (4)	33.78 (4)	0.883 (4)	4.5
SaFIN	**0.057** (1)	**0.972** (1)	$\mathbf{1.354}\times10^4$ (1)	**0.999** (1)	23.49 (2)	0.918 (3)	**1.5**

first in terms of MSE and R. In the task of modeling a nonlinear system, SaFIN achieves a MSE value of 0.057, a remarkable 69.2% reduction compared to the second place Hebb-R-R. In addition, there is a slight improvement of 6.70% in the R value achieved, from 0.911 to 0.972. For the second task of modeling a chemical plant, the proposed SaFIN model delivers an outstanding performance by achieving a reduction of 44.1% in terms of calculated MSE and a slight improvement of 0.001 in terms of correlation R compared to the second position Hebb-R-R model. Although SaFIN loses out to Hebb-R-R in the final task of stock prediction, it should be noted that Hebb-R-R uses 8 (out of 10) input variables in this task [38]. Comparatively, SaFIN uses much less information by utilizing only three input variables, a significant reduction of 167%. This explains the slight compromise in the MSE and R values achieved by SaFIN. On the other hand, RSPOP uses five input variables [39]. Although it uses more information, the performances of SaFIN and RSPOP are comparable. On average, the proposed SaFIN model achieves a ranking of 1.5 for all the three tasks. This makes SaFIN the best performer in this set of experiment.

C. Example 3 – Highway Traffic Flow Density

The learning and generalization abilities of the proposed SaFIN model are evaluated by employing it in the modeling of a real-world application involving highway traffic flow density [36]. The data was collected from site 29 located at

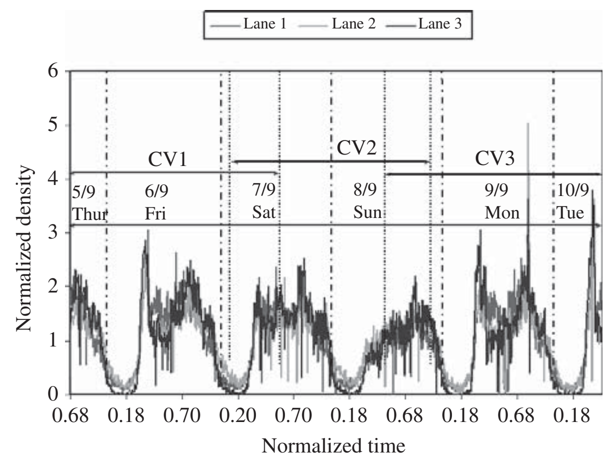


Fig. 9. Traffic flow densities of the three straight lanes along PIE at site 29.

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 preinstalled by the Land Transport Authority of Singapore in 1996 along major roads to facilitate traffic flow data collection. 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

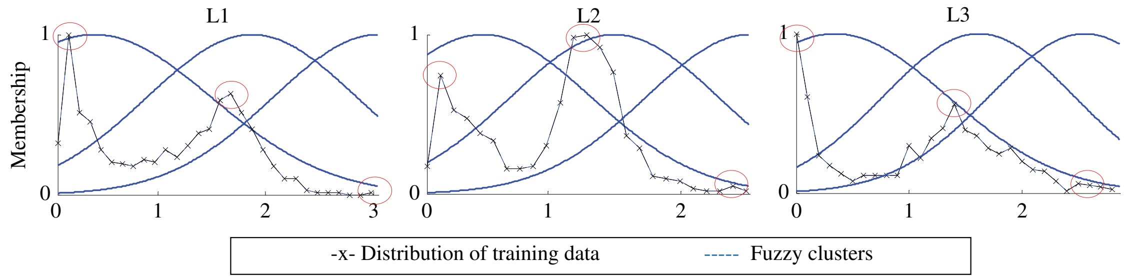


Fig. 10. Fuzzy clusters identified (with distributions of raw data) in lanes L1–L3 for training set of CV1 when $\tau = 5$ min.

flow densities for the three straight lanes during t. SaFIN is used to model the traffic flow trend. After that, the trained model is used to predict traffic flow density of a lane (L1, L2 or L3) at $t + \tau$ for $\tau = 5, 15, 30, 45$, and 60 min.

Fig. 9 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. 9. The benchmarking measurements are the Pearson correlation coefficient R and the mean squared error MSE. The performance of the proposed SaFIN model is subsequently compared against the following models: Hebb-R-R [38], RSPOP [39], MLP (with a configuration of 4 input nodes, 10 hidden nodes, and 1 output node), GenSoFNN [12], EFuNN [11], DENFIS [13], and eFSM [16].

Fig. 10 illustrates the identified fuzzy clusters in lanes L1–L3 for the training set of CV1 when $\tau = 5$ min. The distributions of the raw numerical data are also shown in the figure. The fuzzy clusters identified by the CLIP technique in the proposed SaFIN model coincide with the peaks of the distributions as marked by the dotted circles. Although there are very low distributions of data closer to the upper bounds of the three lanes, it is observed that the distributions immediately before the peaks are closest/at the zero mark. Hence, the CLIP technique identifies a cluster near the upper bound of each lane to cater for this peak. This figure demonstrates the tailored approach adopted by the CLIP technique during fuzzy partitioning.

The consolidated traffic flow prediction results are shown in Fig. 11. Only the average R values from the three cross-validation groups CV1–CV3 for each prediction horizon are plotted with respect to the lanes L1–L3, since the plots for the average MSE value are intrinsically the same. As seen, the general trend among the models is a decreasing R value as the time lag increases from 5 to 60 min. SaFIN is one of the top performers in this task of traffic flow prediction such that it is able to consistently achieve one of the highest average R under different time horizons. This is particularly prominent when $\tau = 60$ min where SaFIN is ranked either the first or second positions for all three lanes L1–L3, while most of the benchmarking models have greater errors due to a longer time lag in the prediction horizon. Although an anomaly is seen in the calculated R value in lane L1 where the R value achieved is slightly higher in $\tau = 60$ min compared to that in $\tau = 45$ min, the general trend of a decreasing R with an increasing time lag is still observed. A possible explanation to this anomaly could be the higher number of fuzzy rules identified in $\tau = 60$ min compared to that in $\tau = 45$ min, thus resulting in a marginal (< 0.01) increase in the calculated R. This result demonstrates the excellent generalization abilities of the proposed SaFIN model such that it is able to learn and generalize the traffic trend to subsequently perform good forecasting on unseen data.

TABLE III

AVERAGE PERFORMANCES FOR THE TRAFFIC FLOW PREDICTION

Model	Average R ($\pm$ Std. Dev.)	Average MSE ($\pm$ Std. Dev.)
Hebb-R-R	0.864 ($\pm$ 0.046)	0.114 ($\pm$ 0.042)
RSPOP	0.834 ($\pm$ 0.041)	0.146 ($\pm$ 0.038)
MLP (4-10-1)	0.847 ($\pm$ 0.065)	0.130 ($\pm$ 0.055)
GenSoFNN	0.813 ($\pm$ 0.028)	0.164 ($\pm$ 0.037)
EFuNN	0.798 ($\pm$ 0.050)	0.189 ($\pm$ 0.041)
DENFIS	0.831 ($\pm$ 0.051)	0.153 ($\pm$ 0.054)
eFSM	0.840 ($\pm$ 0.043)	0.154 ($\pm$ 0.040)
SaFIN	0.862 ($\pm$ 0.043)	0.118 ($\pm$ 0.037)

Table III shows the average performances of all the models for this highway traffic flow density modeling task. As clearly shown, SaFIN demonstrates superior modeling potential, second only to Hebb-R-R, in terms of the average benchmarking measures achieved. Despite employing a time-consuming and computationally intensive iterative post-training phase to recursively identify a reduced set of fuzzy rules with the aim of a good accuracy, Hebb-R-R performs only marginally better than the proposed SaFIN model. Comparatively, the performance of SaFIN is much more consistent and stable as shown by the small standard deviations about the average benchmarking indexes. Although GenSoFNN achieves a lower standard deviation in the R value, it should be noted that the average performance of GenSoFNN is among the poorest under both the benchmarking measures. This result demonstrates the excellent modeling potential of the proposed SaFIN model, while maintaining a highly consistent and stable performance under varying conditions (i.e., time horizons).

D. Example 4 – UCI Dataset

The classification and regression abilities of the proposed SaFIN model are evaluated using three real-life datasets from the UCI dataset, namely: 1) the servo data; 2) the wine data; and 3) the iris data. Both the wine and the iris data are classification problems, while the servo data is a regression

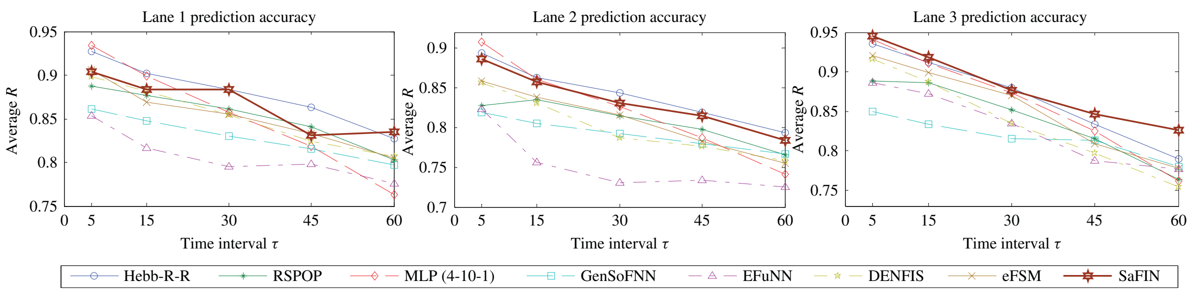


Fig. 11. Traffic flow prediction results.

TABLE IV

CONSOLIDATED EXPERIMENTAL RESULTS FOR THE UCI DATASET

	Servo			Wine			Iris			Average
Model	# Rules	RMSE (Rank)	Ts (Rank)	# Rules	CA% (Rank)	Ts (Rank)	# Rules	CA% (Rank)	Ts (Rank)	Rank
k-NN	N.A.	1.022 (6)	0.03 (1)	N.A.	81.69 (6)	0.03 (1)	N.A.	94.65 (± 1.60) (4)	0.02 (1)	3.2
MLP	N.A.	0.712 (2)	0.78 (2)	N.A.	82.54 (5)	4.52 (4)	N.A.	93.98 (± 3.53) (5)	0.54 (5)	3.8
HyFIS	84	0.742 (4)	2.26 (5)	96	94.37 (4)	12.09 (6)	13.3	95.36 (± 3.17) (3)	0.47 (3)	4.2
DENFIS	8	0.734 (3)	5.90 (6)	15	98.59 (3)	5.70 (5)	12	97.01 (± 0.95) (1)	0.70 (6)	4
EFuNN	75	0.804 (5)	0.90 (3)	100	100 (1)	1.00 (2)	28	92.63 (± 2.98) (6)	0.27 (2)	3.2
FMM*				-	100 (–)	-	48	97.33	-	-
GFMM*				-	100 (–)	-	29	97.33	-	-
SaFIN	12	0.575 (1)	1.09 (4)	93	100 (1)	3.98 (3)	13.7	96.34 (± 0.55) (2)	0.50 (4)	2.5

* For the iris classification, 50% of the data is used for training and testing, respectively.

problem. A detailed description of the datasets is found in [37]. In the tasks of servo regression and wine classification, the first 60% of the data are used as training set, while the remaining 40% are used for testing, while a threefold cross validation, with 34% of the data as training and 66% of the data as testing sets, is performed in the iris classification. The benchmarking measures are the accuracies on the testing data (calculated as the RMSE for the regression problem, and classification accuracies for the classification tasks) and the complexity of the networks (indicated by the size of the rulebase and the training time T). The proposed SaFIN model is subsequently benchmarked against the following models: k-NN [42], MLP; HyFIS [10], DENFIS [13], EFuNN [11], FMM [43], and GFMM [44]. All the models were running on the same computer platform, while results for FMM and GFMM are extracted from [44].

Table IV shows the consolidated experimental results for the UCI dataset. The testing results are listed for the servo and wine tasks, while the average performances are given for the iris classification. The performances of the proposed SaFIN model and the benchmarking models are subsequently ranked according to their testing accuracies and training time.[2] As clearly seen, the SaFIN model is one of the top performers when ranked against the testing accuracies for all three tasks. This illustrates the excellent modeling and generalization abilities of SaFIN. In addition, the SaFIN model uses much lesser rules compared to most of the benchmarking models in the tasks, second only to DENFIS. Despite that, it should also be noted that the training time of DENFIS is among the slowest for all the three experiments. On the other hand, SaFIN has a moderate training time when ranked against the benchmarking models, claiming the third or fourth positions out of six benchmarking models. Despite being among the fastest training models, both k-NN and EFuNN do not deliver comparable testing performances. On average, the proposed SaFIN model achieves a ranking of 2.5, making it the best performer in this set of experiment. On a second note, the testing performances of SaFIN is comparable with the classification models FMM and GFMM, in both the wine and iris classification tasks, while it utilizes significantly lesser rules. This result demonstrates that the proposed SaFIN model is able to deliver superior modeling performances, while maintaining a good balance in the complexity of the network.

[2]The models are not ranked for the size of their rulebase because the information is either not applicable or not available for most models. In addition, both FMM and GFMM do not participate in the servo regression task (and subsequently the ranking) because they are classification models.

V. CONCLUSION

This paper proposed a novel self-organizing neural fuzzy system framework named SaFIN. A key strength of SaFIN is the proposed knowledge acquisition methodology, which draws inspiration from the behavioral category learning process exhibited by humans. This translates to a new single-pass fuzzy partitioning technique known as CLIP, which is able to rapidly partition the input–output spaces. Knowledge is extracted from each arriving training tuple to perform fuzzy partitioning of the input–output spaces. This approach not only allows a tailored partitioning of the input–output dimensions but new clusters in each input-output dimension can also be automatically identified by SaFIN when a prominent distinction is observed. This frees human users from pre-defining the number of clusters necessary in the input–output dimensions. In addition, refinements are made to the existing clusters in an input–output space when a new cluster is added. This effectively addresses the stability–plasticity tradeoff of the model by maintaining a balance between the coexistence of past knowledge (in the system) and future knowledge (from the newly added cluster) in the computational structure. Finally, SaFIN handles the problem of a conflicting rulebase by retaining only the most significant rule in the set of inconsistent rules. This approach ensures that the proposed SaFIN model maintains a consistent rulebase that is able to give an aptly description of the application environment. SaFIN was subsequently employed in four benchmarking experiments to demonstrate its superiority as a self-organizing neural fuzzy system, and excellent performances have been achieved.

REFERENCES

[1] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*. New York: Wiley, 1997.

[2] S. Guillaume, "Designing fuzzy inference systems from data: An interpretability-oriented review," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 3, pp. 426–442, Jun. 2001.

[3] J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," *IEEE Trans. Syst. Man Cybern.*, vol. 23, no. 3, pp. 665–685, May–Jun. 1993.

[4] H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 724–740, Sep. 1992.

[5] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, *Interpretability in Fuzzy Modeling*. New York: Springer-Verlag, 2003, pp. 605–620.

[6] C. T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Compt.*, vol. 40, no. 12, pp. 1320–1336, Dec. 1991.

[7] C. Quek and R. W. Zhou, "POPFNN: A pseudo outer-product based fuzzy neural network," *Neural Netw.*, vol. 9, no. 9, pp. 1569–1581, Dec. 1996.

[8] C. J. Lin and C. T. Lin, "An ART-based fuzzy adaptive learning control network," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 4, pp. 477–496, Nov. 1997.

[9] C. F. Juang and C. T. Lin, "An online self-constructing neural fuzzy inference network and its applications," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 1, pp. 12–32, Feb. 1998.

[10] J. Kim and N. Kasabov, "HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems," *Neural Netw.*, vol. 12, no. 9, pp. 1301–1319, Nov. 1999.

[11] N. Kasabov, "Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning," *IEEE Trans. Syst. Man Cybern. B*, vol. 31, no. 6, pp. 902–918, Dec. 2001.

[12] W. L. Tung and C. Quek, "GenSoFNN: A generic self-organizing fuzzy neural network," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1075–1086, Sep. 2002.

[13] N. Kasabov and Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 144–154, Apr. 2002.

[14] P. P. Angelov and D. P. Filev, "An approach to online identification of Takagi-Sugeno fuzzy models," *IEEE Trans. Syst. Man Cybern.*, vol. 34, no. 1, pp. 484–498, Feb. 2004.

[15] H. J. Rong, N. Sundararajan, G. B. Huang, and P. Saratchandra, "Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction," *Fuzzy Sets Syst.*, vol. 157, no. 9, pp. 1260–1275, May 2006.

[16] W. L. Tung and C. Quek, "eFSM–A novel online neural-fuzzy semantic memory model," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 136–157, Jan. 2010.

[17] J. Tan and C. Quek, "A BCM theory of meta-plasticity for online self-reorganizing fuzzy-associative leaning," *IEEE Trans. Neural Netw.*, vol. 21, no. 6, pp. 985–1003, Jun. 2010.

[18] J. C. Bezdek, E. C.-K. Tsao, and N. R. Pal, "Fuzzy Kohonen clustering networks," in *Proc. IEEE Conf. Fuzzy Syst.*, San Diego, CA, Mar. 1992, pp. 1035–1043.

[19] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[20] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982.

[21] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst. Man Cybern.*, vol. 22, no. 1, pp. 1414–1427, Nov.–Dec. 1992.

[22] K.-L. Du, "Clustering: A neural network approach," *Neural Netw.*, vol. 23, no. 1, pp. 89–107, Jan. 2010.

[23] S. Grossberg, *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*. Boston, MA: Reidel, 1982.

[24] J. M. Mandler and P. J. Bauer, "The cradle of categorization: Is the basic level basic?," *Cognitive Develop.*, vol. 3, no. 3, pp. 247–264, 1988.

[25] J. M. Mandler, P. J. Bauer, and L. McDonough, "Separating the sheep from the goats: Differentiating global categories," *Cognitive Psychology*, vol. 23, no. 2, pp. 263–298, Apr. 1991.

[26] J. M. Mandler and L. McDonough, "Concept formation in infancy," *Cognitive Develop.*, vol. 8, no. 3, pp. 291–318, Jul.–Sep. 1993.

[27] J. M. Mandler and L. McDonough, "On developing a knowledge base in infancy," *Develop. Psychology*, vol. 34, no. 6, pp. 1274–1288, 1998.

[28] J. M. Mandler and L. McDonough, "Studies in inductive inference in infancy," *Cognitive Psychology*, vol. 37, no. 1, pp. 60–96, Oct. 1998.

[29] J. M. Mandler, "On the birth and growth of concepts," *Philosophical Psychology*, vol. 21, no. 2, pp. 207–230, Apr. 2008.

[30] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic systems," *IEEE Trans. Comp.*, vol. 26, no. 12, pp. 1182–1191, Dec. 1977.

[31] S. Haykin, *Neural Network: A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice Hall, 1998.

[32] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice Hall, 1996.

[33] S. W. Tung, C. Quek, and C. Guan, "eT2FIS: An evolving type-2 neural fuzzy inference system," *Inform. Sci.*, submitted.

[34] H. Nakanishi, I. Turksen, and M. Sugeno, "A review and comparison of six reasoning methods," *Fuzzy Sets Syst.*, vol. 57, no. 3, pp. 257–294, Aug. 1993.

[35] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7–31, Feb. 1993.

[36] G. K. Tan, "Feasibility of predicting congestion states with neural network models," M.S. thesis, School Civil Structural Engineering, Nanyang Technological Univ., Singapore, 1997.

[37] *UCI Machine Learning Repository* [Online]. Available: http://archive.ics.uci.edu/ml/

[38] F. Liu, C. Quek, and G. S. Ng, "A novel generic hebbian ordering-based fuzzy rule base reduction approach to Mamdani neuro-fuzzy system," *Neural Comput.*, vol. 19, no. 6, pp. 1656–1680, Apr. 2007.

[39] K. K. Ang and C. Quek, "RSPOP: Rough set-based pseudo outer-product fuzzy rule identification algorithm," *Neural Comput.*, vol. 17, no. 1, pp. 205–243, Jan. 2005.

[40] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst. Man Cybern.*, vol. 15, no. 1, pp. 116–132, Feb. 1985.

[41] K. H. Quah and C. Quek, "FITSK: Online local learning with generic fuzzy input Takagi-Sugeno-Kang fuzzy framework for nonlinear system estimation," *IEEE Trans. Syst. Man Cybern. B*, vol. 36, no. 1, pp. 166–178, Feb. 2006.

[42] T. M. Cover and P. E. Hart, "Nearest pattern classification," *IEEE Trans. Inform. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[43] P. K. Simpson, "Fuzzy min-max neural networks–Part I: Classification," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 776–786, Sep. 1992.

[44] B. Gabrys and A. Bargiela, "General fuzzy min-max neural network for clustering and classification," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 769–783, May 2000.

Sau Wai Tung (S'09) received the B.Sc. (honors-I) degree in pure mathematics from the National University of Singapore (NUS), Singapore, in 2006. She is currently pursuing the Ph.D. degree in computer engineering with Nanyang Technological University (NTU), Singapore. She is also a student at the Center for Computational Intelligence, NTU.

Her current research interests include intelligent architectures, neural networks, fuzzy systems, and fuzzy neural systems.

Ms. Tung is a recipient of the Agency for Science, Technology and Research Graduate Scholarship at NUS-NTU in 2007.

Chai Quek (SM'10) received the B.Sc. degree in electrical and electronics engineering and the Ph.D. degree in intelligent control from Heriot-Watt University, Edinburgh, U.K., in 1986 and 1990, respectively.

He is an Associate Professor and Assistant Chair with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. He is a Research Member with the Center for Computational Intelligence, NTU. His current research interests include intelligent controls, intelligent architectures, computational finance, neural networks, fuzzy neural systems, neurocognitive informatics, and genetic algorithms.

Dr. Quek is a member of the IEEE Technical Committee on Computational Finance.

Cuntai Guan (SM'03) received the Ph.D. degree in electrical and electronic engineering from Southeast University, Nanjing, China, in 1993.

He is a Principal Scientist and Program Manager with the Institute for Infocomm Research (I2R), Agency for Science, Technology and Research, Singapore. Since 2003, he has been establishing the Brain-Computer Interface Laboratory at I2R. His current research interests include brain-computer interfaces, neural signal processing, machine learning, pattern classification, and statistical signal processing, with application for cognitive training, rehabilitation, and health monitoring.

Dr. Guan is an Associate Editor of the IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING and *Frontiers in Neuroprosthetics*.