On Equivalence of FIS and ELM for Interpretable Rule-Based Knowledge Representation

Shen Yuong Wong, Keem Siah Yap, Hwa Jen Yap, Shing Chiang Tan, and Siow Wee Chang

***Abstract*— This paper presents a fuzzy extreme learning machine (F-ELM) that embeds fuzzy membership functions and rules into the hidden layer of extreme learning machine (ELM). Similar to the concept of ELM that employed the random initialization technique, three parameters of F-ELM are randomly assigned. They are the standard deviation of the membership functions, matrix-C (rule-combination matrix), and matrix-D [don't care (DC) matrix]. Fuzzy if-then rules are formulated by the rule-combination Matrix of F-ELM, and a DC approach is adopted to minimize the number of input attributes in the rules. Furthermore, F-ELM utilizes the output weights of the ELM to form the target class and confidence factor for each of the rules. This is to indicate that the corresponding consequent parameters are determined analytically. The operations of F-ELM are equivalent to a fuzzy inference system. Several benchmark data sets and a real world fault detection and diagnosis problem have been used to empirically evaluate the efficacy of the proposed F-ELM in handling pattern classification tasks. The results show that the accuracy rates of F-ELM are comparable (if not superior) to ELM with distinctive ability of providing explicit knowledge in the form of interpretable rule base.**

***Index Terms*— Extreme learning machine (ELM), fuzzy inference system (FIS), pattern classification, rule based.**

I. Introduction

INSPIRED by the biological morphologies, neural networks (NNs) have been a powerful computational tool in a diversity of applications, including classification, pattern recognition, function approximation, control, and modeling. Due to their inherent cognitive ability, NN models are well suited to represent complex functions such as those encountered in biological processes. In addition, the ability of NN to learn from examples has become one of the appealing features of NN in providing excellent framework for solving numerous problems for which no analytical solution is known.

Manuscript received October 15, 2013; revised March 10, 2014 and July 11, 2014; accepted July 16, 2014. Date of publication August 13, 2014; date of current version June 16, 2015. This work was supported by the University of Malaya Research Collaborative Grant Scheme PRP-UM-UNITEN under Grant CG026-2013.

S. Y. Wong and K. S. Yap are with the College of Graduate Studies & Department of Electronics and Communication Engineering, Universiti Tenaga National, Selangor 43009, Malaysia (e-mail: joeywsy77@yahoo.com; yapkeem@uniten.edu.my).

H. J. Yap is with the Department of Mechanical Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia (e-mail: hjyap737@um.edu.my).

S. C. Tan is with the Faculty of Information Science and Technology, Multimedia University, Malaka 75450, Malaysia (e-mail: sctan@mmu.edu.my).

S. W. Chang is with the Institute of Biological Sciences Faculty of Science, University of Malaya, Kuala Lumpur 50603, Malaysia (e-mail: siowwee@um.edu.my).

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/TNNLS.2014.2341655

Rule learning and extraction are an important area in NN research. Nonetheless, the inhibition to a widespread acceptance of NN is that it operates as a black box that reveals little or no information of the system they represent. NN demonstrates a deficiency in declarative knowledge structure, in other words, it lacks a explanatory facility to explain to user how a network arrives at a particular decision. Neither can one say something about the knowledge encoded in the black box [1]. In safety-critical problems, i.e., airlines and power plants, use of black-box models may not be acceptable.

A number of different strategies have been developed to overcome the black-box phenomenon. There are two main methodologies used, i.e., the pedagogical and decompositional methods [2]. The pedagogical treats the network as a black box, and aims to extract rules that map inputs directly into outputs. This method views the trained NN in translucency dimension. On the other hand, the decompositional method is devised based on transparency, where the computed output from each hidden and output unit in the trained network must be mapped into an outcome that corresponds to the notion of a rule consequent. Therefore, it provides an in-depth vision to the underlying trained NN by extracting rules at the level of individual units within the trained NN [3].

On the other hand, in the domain of machine learning, fuzzy inference system (FIS) has been popular in undertaking rule extraction tasks. It is owing to the ability of fuzzy logic to unveil the advantage of interpretability. FIS addressed this situation by extracting the embedded knowledge in trained NNs in the form of symbolic rules. FIS includes two important stages, i.e., the fuzzification and defuzzification, when interfacing with an environment. Fuzzification fetches inputs while defuzzification transforms fuzzy output to human interpretable format.

Nevertheless, FIS has been criticized for the requirement of an exhaustive collection of an expert knowledge and an enormous effort in maintaining the knowledge base [3]. In this regards, NN can be a better option to generate relatively fewer rules using all the available input attributes to cope with the real data sets that might contain noise or data sets of high-dimensional attribute space.

Therefore, a group of researchers combined the merits of both neural and fuzzy systems under a unified soft computing framework called the neuro-fuzzy approach to modeling expert behavior [4], [5]. Combination of both, however, allows the extraction of a set of rules exhibiting the property of transparency, which is a basic requirement for human to understand linguistic rule set. In particular, the neuro-fuzzy approach transforms and converts the NN in a FIS of the Takagi–Sugeno (TS) type [6]. Associated with the FIS there is a rule base formed by a set of fuzzy rules, which may be conveniently manipulated so that the knowledge implicitly captured by the NN becomes explicit. As pointed out in [6], this kind of knowledge discovery is expressed as opening the black box.

In addition to the black-box phenomenon that lies within NN, the fact that most NN approaches require a long training time in response to a large database has become another impediment to the NN algorithm in formulating an effective and insightful rule extraction model. It is particularly critical for real problems in large scale, complex processes in modern industrial environment today. This is the reason why conventional feedforward NNs tend not to be used in the industry. The learning method of conventional feedforward NNs is mainly trained by back-propagation (BP). BP algorithms always suffer from various issues, i.e., stopping criteria, learning rate, multiple learning epochs, and over tuning issue [4]. Training samples are also required to be presented repeatedly. Recently, a new learning algorithm for Single-layer hidden feedforward neural network (SLFN), known as the extreme learning machine (ELM) proposed in [7], has turned out to be the remedy for NN learning approaches as the speed of ELM can be thousands of times faster than the traditional network learning algorithms. The structure of ELM is rather simple and hence the learning process could be expedited.

However, the original ELM is a NN without fuzzy membership functions and linguistic rule set. It could be a limitation to applications particularly when the nature of the problem requires a pattern classification module that provides explicit knowledge in human understandable and intelligible form. In this respect, the need for such a system is now well established. It serves as our motivation to develop ELM based neuro-fuzzy system that is capable of handling pattern classification problems not only to produce high accuracy rates but a system endowed with an ability to formulate transparent and interpretable rules.

In this paper, a fuzzy ELM (F-ELM) is developed to realize the rule learning model within the original architecture of three-layer ELM. The objectives of this paper are four fold. First objective is to randomly assign input weights for automatic fuzzy rule generation in F-ELM without *a priori* knowledge from human experts. This is in agreement with the randomness mechanism advocated by the theory of ELM. In ELM, all the hidden neuron parameters are randomly generated (even before ELM looks at the training data) without tuning. This is because all the training data are linearly separable in the ELM feature space [8]. Second, this paper focuses on the interpretability and readability of the rules generated from the F-ELM. F-ELM properly defines five equally distributed membership functions to human understanding. To date, little has appeared in the form of transparent and consistent attempt to develop rule extraction models that deliver decipherable rules [2], [3], [5], [6]. Without understanding the underlying implication of the membership functions, domain experts could not fathom the meaning of the rules, not to mention the laymen. It ends up creating another black box inside an existing NN black box. Therefore, rules comprehensibility is the most prominent factor in the aspect of decision making. Third, a don't care (DC) approach is incorporated into the ELM architecture along with random rule combination (matrix-**C**). This approach is adopted to minimize the number of input attributes in the fuzzy rules. As such, the postprocessing methods that often executed by researchers [9], [10] to refine the rules can be eliminated. Last but not least, we want to demonstrate the operation of the F-ELM is significantly equivalent to a FIS in the three-layer ELM, while still preserving the outstanding property of ELM for being a straightforward, fast learning, and computationally efficient network learning algorithm.

The content of this paper is organized as follows: Section II introduces preliminaries of ELM, FIS, brief of previous works of neuro-fuzzy rule generation, and some ELM-related approaches. Section III explains the proposed F-ELM algorithm in detail. Section IV illustrates some experiments and results. Finally, Section V provides some concluding remarks.

II. PRELIMINARIES

A. Extreme Learning Machine

ELM is a new learning algorithm [7] for single-hidden-layer feedforward NNs (SLFNs) with addictive hidden neurons or radial basis function (RBF) hidden neuron that guarantees the universal approximation capability.

In ELM, the input weights (linking the input to the hidden layer) and hidden biases are randomly chosen, and the output weights (linking the hidden to the output layer) are analytically determined using Moore–Penrose (MP) generalized inverse. White [11], and Huang and Chen [12] found that random search over input to hidden layer connections is computationally efficient in SLFN with affine transformation and no additional learning is required. Tuning all the network parameters may cause learning complicated and inefficient.

Huang and Chen [13] later extended the preliminary ELM from the SLFN with addictive or RBF hidden neurons to generalized SLFN with a wide variety of hidden neurons. These hidden neurons can be any type of piecewise continuous nonlinear hidden neurons, including the addictive or RBF type of neurons, multiplicative neurons, fuzzy rules, fully complex neurons, hinging functions, high-order neurons, ridge polynomials, wavelets, Fourier terms, and so on.

In a perfect case, the output of this ELM, respectively, to $\mathbf{x}_j$ should be

$$f(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) = T_j \quad \text{for } j = 1, \ldots, N \tag{1}$$

where $\mathbf{a}_i$ and b_i are the input weights (linking the input to the first hidden layer) and bias (learning parameters) of the hidden nodes, β_i is the output weights (linking the hidden to output layer), T_j is the respective targeted output vectors, and $G(\mathbf{a}_i, b_i, \mathbf{x}_j)$ is the output of the ith hidden neuron, respectively, to the input vector $\mathbf{x}_j$, which can also be written as

$$\mathbf{H}\beta = \mathbf{T} \tag{2}$$

where

$$\mathbf{H}(a_1, \ldots, a_{\bar{N}}, b_1, \ldots, b_{\bar{N}}, x_1, \ldots, x_N) = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \ldots & G(\mathbf{a}_{\bar{N}}, b_{\bar{N}}, \mathbf{x}_1) \\ \vdots & \cdots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \ldots & G(\mathbf{a}_{\bar{N}}, b_{\bar{N}}, \mathbf{x}_N) \end{bmatrix}_{N\times\overline{N}} \tag{3}$$

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{\bar{N}} \end{bmatrix}_{\bar{N}xm} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}_{Nxm}. \tag{4}$$

As named in [10], $\mathbf{H}$ is called the hidden layer output matrix of the NN; the ith column of $\mathbf{H}$ is the ith hidden node output with respect to inputs.

Several improved versions of ELM have been proposed and can be found in [8]–[10], [13], and [14].

B. Fuzzy Inference System

FIS plays an important role in the application of fuzzy set theory. It is developed in [15] and [16], which deals with linguistic vague information based on fuzzy sets and fuzzy logic. Fuzzy logic is capable of handling vagueness, modeling uncertainty, supporting human-type reasoning, and could be applied to complex systems. FIS can be considered as fuzzy-rule-based systems, fuzzy models, fuzzy associative memories (FAMs), or fuzzy controllers when used as controllers [17].

In FIS of the TS type, the relationships among variables of the system are represented by fuzzy if-then rules in the form [6]

$$\begin{aligned} \text{Rule} \quad R_\ell &: \text{If } x_1 \text{ is } C_1^\ell \text{ and } x_2 \text{ is } C_2^\ell \ldots x_n \text{ is } C_n^\ell \\ &\text{then} \quad y^\ell = f(x_1 \ldots x_n) \end{aligned} \tag{5}$$

where C^ℓ are fuzzy sets, and x_n is the input of the system. When y^ℓ is a constant, the FIS is called a zero-order TS fuzzy model. The firing strength of each rule is calculated by

$$w_j = \bigcap_{i=1}^{n} \mu_{ij}(x_i) \tag{6}$$

where $\mu_{ij}(x_i)$ is the membership function associated to the fuzzy set C^ℓ and $\cap$ represent the product operator (AND operator). The consequent of a rule is an affine linear or nonlinear function of the input variables.

The output of the system is computed as the weighted sum of each rule's output

$$z = \sum_{j=1}^{N} y_j w_j. \tag{7}$$

Or more conventionally, as the weighted average

$$z = \frac{\sum_{j=1}^{N} y_j w_j}{\sum_{j=1}^{N} w_j} \tag{8}$$

where N is the number of rules of the system.

C. Revisiting Neuro-Fuzzy Rule Generation and ELM-Related Approaches

Rong *et al.* [18] described online sequential (OS)-fuzzy-ELM (OSELM for function approximation and classification problems) that extended the SLFN to a TS–Kang (TSK) FIS. The centers of the membership functions that were randomly generated had indicated that there was no specific definition of the membership values produced. Such algorithm had compromised on the interpretability of the rules. Similar to [19], the choice of the fuzzy logic used is the AND operation like most of the TSK model in the literature to get the firing strength (if part). The TSK fuzzy model output is achieved by the weighted sum of the output of each normalized rule. Interestingly, our proposed method is different from the common practice of FIS, we are implementing the Modified Probabilities OR (Modified PROBOR) and product for the computation of firing strength. In addition, we utilize and translate the output weights of the ELM NN itself into output class and its confidence factor.

Next, Sun *et al.* [4] proposed a neuro-fuzzy model that used ***k***-means clustering method to group the data. The membership of arbitrary input for each fuzzy rule was derived through an ELM and then normalized it. Multiple ELMs were used to obtain the consequent part of the fuzzy rules.

An interesting paper in [20] has proven the functional equivalence between RBF networks and FIS. Here, FIS is transformed into an adaptive network; however, the shapes and positions of the membership functions are updated using slow BP. And the output of each rule is identified through least square method. In addition, there is no weight associated with each link.

In recent times, not many research efforts have focused on the meaningful rule generations. Horikawa *et al.* [21] expanded the fuzzy NNs to a more general fuzzy modeling scheme. However, there is a dependency on the availability of the expert knowledge to create a proper initial rule base. Creating initial rule base is a difficult feat when the databases are large.

Kuo and Cohen [22] used a self-organizing and self-adjusting fuzzy model for manufacturing process control. The inputs and outputs were partitioned by Kohonen's feature mapping and the premise and consequent parameters were updated and tuned using BP. A major problem of using this method to refine rule-based knowledge is the preservation of symbolic knowledge under the weight tuning mechanism of the BP algorithm.

One way of applying fuzzy rule estimation was using the clustering algorithm [23], [24]. However, conflicts that were cropping up among the rules impeded the acceptance and acknowledgement of the generated rules. Wang and Mendel [25] had later introduced a method without conflicting rules. Yet the algorithm was still considered heuristic as the choice of fuzzy membership functions was using a trial and error procedure.

Jang [17] implemented a Sugeno-like fuzzy system called adaptive-network-based FIS (ANFIS). However, initial rule base must be acquired in advance, because ANFIS adjusted only the membership functions of the antecedent and consequent parameters. Due to the high flexibility of the adaptive NNs, ANFIS had lent itself to solve a vast number of applications. However, the shortcoming associated with this learning algorithm is that it is difficult to handle high-dimensional problems, as this would yield a huge number of input antecedents and consequent parameters, besides being computationally exhaustive to have an efficient implementation. In the ANFIS architecture, each linguistic term was represented by only one fuzzy set.

In addition, neuro-fuzzy hybrid approaches [26] were often used to design the fuzzy rule base of an intelligent system for a controller. ANFIS methodology was adopted to control the speed of a stepping motor drive [27], [28]. The advantage of this regulator was that the fuzzy control computation is inexpensive and it could be used for the control of the robotics manipulators, even for generation of walking motions, such as the design of a TS type fuzzy logic controller for a biped robot walking problem [29]. Another adaptive model using NNs, fuzzy logic, and fractal theory was used for control the chaotic and unstable behavior in aircraft systems [30].

In [31], fuzzy logic models were developed to aid in standardizing the overall decision of various Dissolved Gas-in-Oil Analysis (DGA) interpretation techniques. This overcomes the limitation of the traditional DGA interpretation techniques that were used only to analyze the collected DGA results to evaluate the consistency and accuracy of each interpretation technique.

Recently, NN with type-2 fuzzy weights was introduced to application in time series prediction. A new BP learning method enhanced with type-2 fuzzy logic was presented in [32]. Type-2 fuzzy weight adjustment in BP learning of NNs provided them the ability to manage uncertainty in data set. It was observed that NN with type-2 fuzzy weights shows better behavior at different levels of uncertainty than the monolithic NN and it had better tolerance to noise when different levels of noise were applied to the simulation data set. Besides that, interval type-2 fuzzy logic has also been used for medical analysis, such as to evaluate fetal health status through fetal phonocardiography [33], and as a powerful tool in handling uncertainties in the industrial applications [34].

Most methods engaged rule refinement in postprocessing stage. Nonetheless these methods are laborious and time consuming. Yap *et al.* [35] presented Improved Generalized Adaptive Resonance Theory (IGART) with pruning conducted postprocessingly by removing those categories that are insignificant to the overall output. In addition, further analysis by constructing an additional FIS classifier with the extracted rule set from IGART is carried out to access the quality of the extracted rule sets.

There is a tradeoff between the interpretability and accuracy. In many situations, one may adopt to an explosion of rules to achieve high accuracy. Efforts are much devoted to the tuning of membership functions in fuzzy rules identification [36]–[38]. Kuncheva [38] and Klawon and Klement [39] pointed out that if the choice of membership functions is not consistent throughout the implementation or the membership functions are of irregular shapes, then they are unlikely to associate with the linguistic labels precisely and unambiguously. With this, interpretability will not benefit.

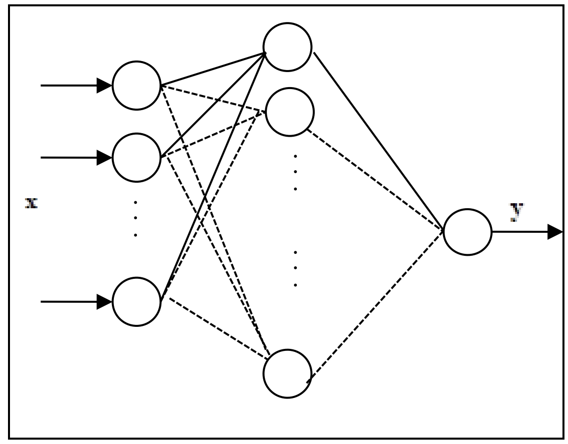

Fig. 1. ELM architecture.

III. F-ELM Algorithm

The architecture of F-ELM is similar to a three-layer ELM, in which it embeds the fuzzy rules into the hidden layer of ELM. Each hidden neuron in the structure of F-ELM represents a fuzzy rule. Therefore, the operations of F-ELM are equivalent to a FIS.

Unlike most of the algorithms given in the brief review above, we propose a method that guarantees the essence of interpretability by fixing the center of the membership functions that could iron out the issue of irregular shape of membership function and ambiguous linguistic labels [3]. Five Gaussian membership functions are introduced [35], with linguistic labels denoted as MF 1: very low, MF 2: low, MF 3: medium, MF 4: high, and MF 5: very high. They are centered at [0, 0.25, 0.5, 0.75, 1], respectively. Kuncheva [38] argued that rule interpretability is an essential part of NN-based classifiers. The interpretability of the membership functions associated with the rules has to be straightforward.

Similar to the randomness strategy exploited by ELM, F-ELM randomly assigns standard deviation of Gaussian membership functions, rule-combination (matrix-**C**) and DC (matrix-**D**). F-ELM engages binary bits from matrix-**C** to decide which membership function of an input attribute to be used in a rule. One distinctive advantage of F-ELM is that it allows the discontinuous use of membership function, i.e., for an input attribute of rule-1, based on the matrix-**C**, membership function 1 or 4 are selected. This implies that the inference engine of F-ELM will only consider MF 1 or MF 4, and exclude MF 2, MF 3, and MF 5. As such, the concept of noncontinuous membership function is successfully deployed. This is in contrast to the method proposed in [35], whereby their method is limited to using only the continuous membership functions. Thus, the above example will no longer be applicable, it is then changed to, i.e., for an input attribute of rule-1, membership function from 1 to 4 are to be included. In other words, every membership function between the MF 1 and MF 4 (MF 1, MF 2, MF 3, and MF 4) has to be treated in computation. This could inevitably draw many insignificant membership functions into calculation that may eventually jeopardize the accuracy.

Furthermore, matrix-D is injected to F-ELM during the rule generation. The input space is partitioned according to the number of antecedent. A five-linguistic antecedent (i.e., very low, low, medium, high, very high) is used, then there would be five partitions for each input attribute. An additional DC antecedent is then added. The fuzzy if-then rules can be rewritten by removing the attributes with DC. As such, each computed membership value represents an essential and valid fuzzified input attribute to be translated into useful premise parts of the rules. An example is delineated

$$\text{If } x_1 \text{ is low Or high And } x_2 \text{ is Medium And } x_3 \text{ is "Don't Care," then Class 1.} \tag{9}$$

Rewrite the above statement

$$\text{If } x_1 \text{ is low Or high And } x_2 \text{ is Medium, then Class 1.} \tag{10}$$

Although F-ELM encompasses fuzzy operation, it preserves and fulfills the briefness criteria of ELM, i.e., use of random input weights, the tuning of the number of hidden neurons, and computation of output weights by a fast and straightforward equation.

Fig. 1 shows typical three-layer ELM structure. With reference to Fig. 1, a new architecture (Fig. 2) is drawn to outline the concept of F-ELM. The dotted line represents the repeated process of the black straight line. Note that the proposed method can be readily applied to multiple-input and multiple-output systems, as shown in Section IV.

From Fig. 2, each hidden neuron of F-ELM resembles a rule that can be interpreted in the if-then format. The operation of the F-ELM is identical to a FIS.

By looking closely at Fig. 2, consider a set of N training data samples (with an input vector and, respectively, target output vector), $(\mathbf{x}_j, \mathbf{t}_j) \in \mathbf{R}^n \times \mathbf{R}^m$ are used to train F-ELM.

A. Training

The training algorithms can be summarized as follows.

Step 1: Define and initialize the parameters of F-ELM.

1) All input attributes are fuzzified to five membership functions. The centers of five membership function are fixed, i.e., $\mathbf{a} = [00.250.50.751]$. To explain further, the membership function centered at 0 is defined as very low (or in single label 1), the membership function centered at 0.25 is defined as low (or in single label 2), the membership function centered at 0.5 is defined as medium (or in single label 3), the membership function centered at 0.75 is defined as high (or in single label 4), and the membership function centered at 1 is defined as very high (or in single label 5).
2) Randomly assign training parameter of F-ELM, i.e., $\sigma \in R^+$ (standard deviation of Gaussian fuzzy membership function).
3) Select the number of hidden neurons (L), which is equivalent to number of fuzzy rules.
4) Randomly select the combination of membership functions for attributes of all hidden neuron (fuzzy rules). This can be done by randomly assign binary values to a 3-D matrix (with n attributes $\times$ five membership functions $\times$ L rules), known as rule-combination matrix (hereafter denoted as matrix-**C**). For example, rule-combination matrix $C(2, 3, 4) = 1$ represents the MF 3 of attribute-2 is active, thus it will be used in rule-4.
5) Randomly select the DC bits by assigning binary values to a 2-D matrix (with n attributes $\times$ L rules), known as DC matrix (hereafter, denoted as matrix-**D**). For example, $D(2,4) = 1$ represents attribute-2 is DC (not used) in rule-4.

Step 2: For all training pair $(\mathbf{x}_j, \mathbf{t}_j)$, do the following steps.

Step 2(a): Calculate the fuzzy values of using membership functions for all attributes x_{ji}

$$\mu(k, x_{ji}) = \exp\left[-\frac{(x_{ji} - a_k)^2}{2\sigma^2}\right] \tag{11}$$

where $j = 1$ to N, $i = 1$ to n, and $k = 1$ to 5.

Step 2(b): Calculate the Modified PROBOR values [6] of the input attributes in a rule

$$v_{i\ell} = \begin{cases} 1 & \text{if} \quad D(i, \ell) = 1 \\ 1 - \prod_{k=1}^{5} (1 - C(i, k, \ell)\mu(k, x_{ji})) & \text{else } D(i, \ell) = 0 \end{cases} \tag{12}$$

where $i = 1$ to n, ℓ =1 to L, and $k = 1$ to 5. Note that this is a Modified PROBOR function with some negligible fuzzy membership values based on random 3-D matrix-**C**. In the case that the DC is true, the results of Modified PROBOR should be set to 1.

Step 2(c): With the results of Modified PROBOR, compute the firing strength of each rule through a T-norm operator (generalized AND) that performs multiplication [17]

$$w_{j\ell} = \prod_{i=1}^{n} v_{i\ell}. \tag{13}$$

Step 3: Combined firing strength for all training samples to become a hidden layer output matrix

$$\mathbf{H} = \begin{bmatrix} w_{11} & \cdots & w_{1L} \\ \vdots & \cdots & \vdots \\ w_{N1} & \cdots & w_{NL} \end{bmatrix}_{N\times L}. \tag{14}$$

Step 4: Use MP generalized inverse [7] to analytically calculate the output weights

$$\boldsymbol{\beta} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{T} \tag{15}$$

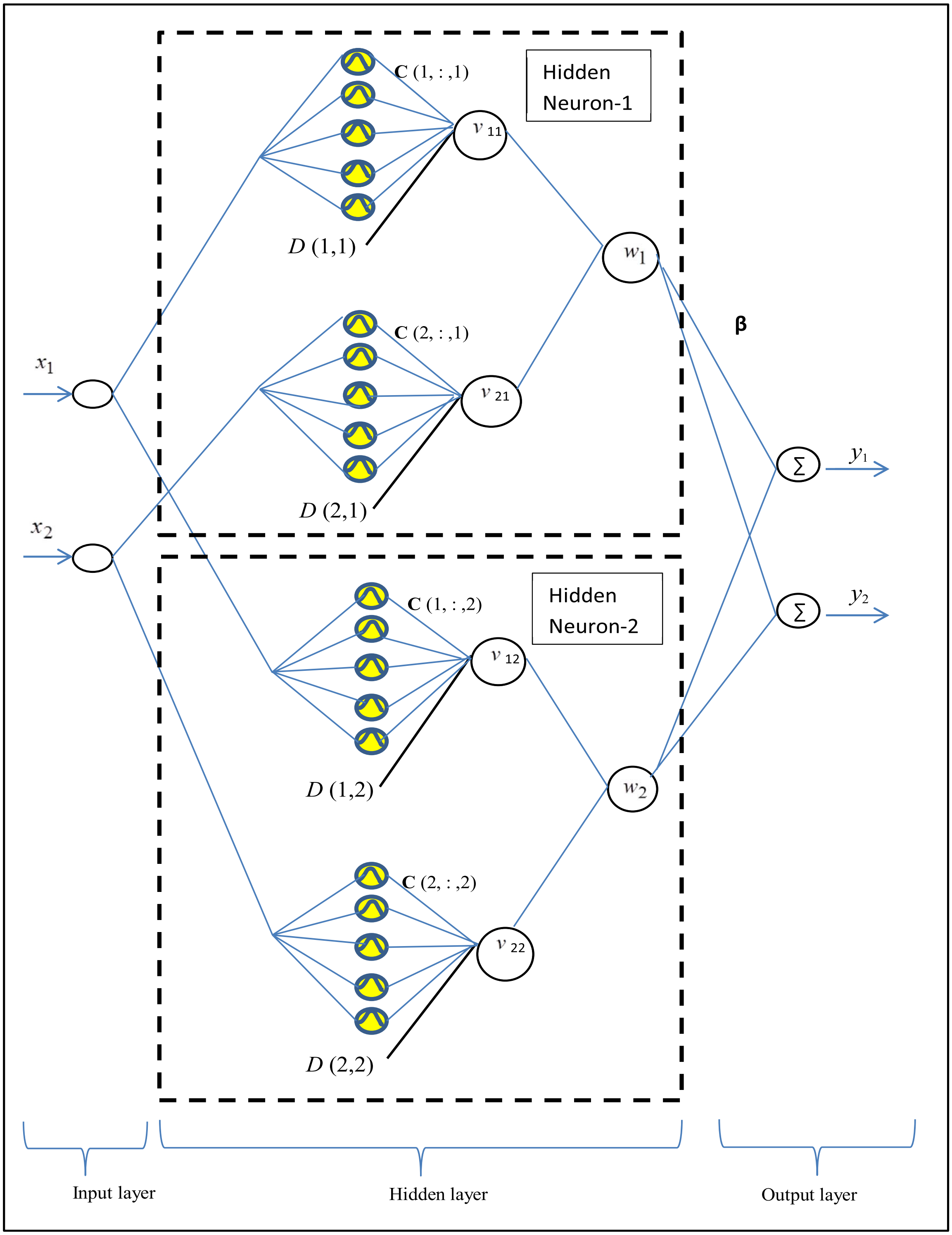

Fig. 2. Framework of F-ELM that exemplifies the case of two hidden neurons.

where $\mathbf{T} = [t_1 \ldots t_N]^T$. Note that since the operations of the proposed F-ELM is identical to the standard FISs, hence, $\boldsymbol{\beta}$ can be considered equivalent to output constants ($\mathbf{f}$) of FIS.

B. Prediction

Next, once the training cycle is completed, the F-ELM is ready for prediction to a new and unlabeled input vector $\mathbf{z} \in \mathbf{R}^n$. The prediction cycle can be summarized as follows.

Step 1(a): Calculate the membership functions for all attributes z_i

$$\mu(k, z_i) = \exp\left[-\frac{(z_i - a_k)^2}{2\sigma^2}\right] \tag{16}$$

where $i = 1$ to n, and $k = 1$ to 5.

Step 1(b): Calculate the Modified PROBOR [6] of the input attributes in a rule

$$v_{i\ell} = \begin{cases} 1 & \text{if} \quad D(i,\ell) = 1 \\ 1 - \prod_{k=1}^{5} (1 - C(i,k,\ell)\mu(k, x_{ji})) & \text{else } D(i,\ell) = 0 \end{cases} \tag{17}$$

where $i = 1$ to n, $\ell = 1$ to L, and $k = 1$ to 5. Note that this is a Modified PROBOR function with some negligible fuzzy membership values based on the matrix-**C**. In the case that the DC is true, the results of Modified PROBOR is set to 1.

Step 1(c): Compute the firing strength of each rule by combining membership values on the antecedent part through a T-norm operator (generalized AND by multiplication) [17], based on the results of Modified PROBOR

$$w_\ell = \prod_{i=1}^{n} v_{i\ell}. \tag{18}$$

Step 2: Combined firing strength for the input z_i to become a hidden layer output matrix

$$\mathbf{h} = \left[w_1 \; w_2 \; \ldots\ldots \; w_L \right]_{1\times L}. \tag{19}$$

Step 3: Compute the output of the F-ELM

$$y_k = \sum_{\ell=1}^{L} h_\ell \beta_{\ell k} \tag{20}$$

where $\mathbf{y}$ is the prediction of the F-ELM and $k = 1$ to m. Note that (20) is equal to weighted sum defuzzification of an FIS.

C. Interpretation of F-ELM to Rule Base Structure

Unlike the standard fuzzy logic operator as presented in (5) of FIS, F-ELM uses Modified PROBOR to compute effective membership values for each of the input attributes in the input fuzzification procedure. This is due to the introduction of additional elements, i.e., matrix-**C** and matrix-**D** that decide which membership function of an input attribute to be used in a rule.

An example is illustrated to provide mutual understanding of how F-ELM is translated into a rule base structure. Refer to Fig. 2, given 2 inputs (x_1 and x_2), matrix-**C** and matrix-**D** as follows:

$$\mathbf{C}(:,:,1) = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$
$$\mathbf{C}(:,:,2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{21}$$
$$\mathbf{D} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{22}$$

The above matrices can be delivered in the following if-then format:

$$R_1 : \text{IF } x_1 \text{ is Low or Very High and } x_2 \text{ is DC, THEN } \mathbf{f}_1 = \boldsymbol{\beta}_1$$
$$R_2 : \text{IF } x_1 \text{ is DC and } x_2 \text{ is Medium or High, THEN } \mathbf{f}_2 = \boldsymbol{\beta}_2. \tag{23}$$

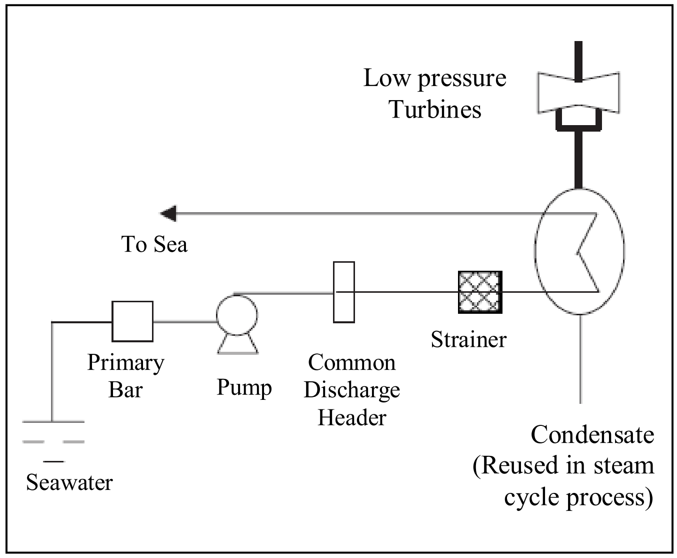

Fig. 3. CW system.

TABLE I

CLASSIFICATION OF THE OPERATING CONDITIONS OF THE CW SYSTEM

Class Label	Indication
1	Heat transfer in the condenser is efficient and there is no significant blockage in the piping system
2	Heat transfer in the condenser is not efficient and there is no significant blockage in the piping system
3	Heat transfer in the condenser is efficient and there is significant blockage in the piping system
4	Heat transfer in the condenser is not efficient, and there is significant blockage in the piping system

TABLE II

ABBREVIATIONS OF THE SENSOR PARAMETERS IN THE CW SYSTEM DATA SET

No.	Parameter	Description
1	LPT A	Low pressure cylinder exhaust temperature A
2	LPT B	Low pressure cylinder exhaust temperature B
3	GEN	Generator
4	CWIT A	Condenser circulating water inlet temperature A
5	CWIT B	Condenser circulating water inlet temperature B
6	CWOT A	Condenser circulating water outlet temperature A
7	CWOT B	Condenser circulating water outlet temperature B
8	CWIP A	Condenser circulating water inlet pressure A
9	CWOP A	Condenser circulating water outlet pressure A
10	CWIP B	Condenser circulating water inlet pressure B
11	CWOP B	Condenser circulating water outlet pressure B
12	VAC	Condenser vacuum

Output weights $\boldsymbol{\beta}$ obtained from the computation of F-ELM are used to represent the output constant of each rule. Here, $\boldsymbol{\beta}$ expresses the combination of class label and its respective confidence factor. It makes sense because the output weights are not random. They are determined analytically from the knowledge of hidden layer output matrix **H** and target values, which is introduced with the use of a MP generalized inverse of the matrix **H**. Please note computed $\boldsymbol{\beta}$ could be $\boldsymbol{\beta} > 1$ or $\boldsymbol{\beta} < -1$. To ensure confidence factor is within [0, 1],

TABLE III
TOTAL EIGHT RULES CAPTURED FROM THE F-ELM FOR THE CW DATA SET

	IF												THEN Class
	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	is
Rule 1	2,5	1,5	1,2,3,4	DC	DC	DC	1,2,3,4	DC	1,2,3,4,5	DC	DC	DC	1
Rule 2	2,4	DC	1,5	DC	2,3	1,3,5	DC	1,3,4,5	3,5	1,3,5	2,3,4,5	DC	2
Rule 3	1	2,3,4	DC	2,3,4,5	DC	DC	DC	DC	2,3,5	DC	2	DC	4
Rule 4	DC	DC	DC	DC	DC	4,5	DC	4,5	1,2,3,4,5	DC	DC	DC	2
Rule 5	DC	2	DC	2,3,4,5	1,2,3	DC	DC	1,2,4,5	DC	1,2,3,4	5	1,4,5	1
Rule 6	DC	DC	3,4,5	DC	DC	DC	DC	2	1,2,3,5	2	2	1,3	3
Rule 7	DC	DC	1,2,3,5	2,3,4,5	1,3	DC	5	DC	DC	1,4	DC	1,2,3	4
Rule 8	3,4,5	DC	3,4,5	DC	3	DC	DC	2,5	2,3,4	DC	1,2,5	DC	1

normalization is conducted

$$\alpha = \boldsymbol{\beta} / \max |\boldsymbol{\beta}| . \tag{24}$$

The normalized output weights are denoted as α. Therefore, (23) can be rewritten as

$$R_1 : \text{IF } x_1 \text{ is Low or Very High and } x_2 \text{ is DC, THEN } \mathbf{f}_1 = \alpha_1.$$
$$R_2 : \text{IF } x_1 \text{ is DC and } x_2 \text{ is Medium or High, THEN } \mathbf{f}_2 = \alpha_2. \tag{25}$$

IV. EXPERIMENTS AND RESULTS

To test the validity and the performance of the proposed F-ELM, we present in this section the experimental results with the real world problem of condition monitoring of circulating water (CW) system [40] and five benchmark data sets that include the binary classification, i.e., Pima Indian Diabetes (PID) and Wisconsin Breast Cancer (WBC), as well as multiclass classification with data sets, i.e., the Satellite image, Image segmentation, and Deoxyribonucleic Acid (DNA) [41]. More specifically, we give a concise and detailed explanation in the first experiment. These explanations are not repeated in the subsequent experiments. In the experiment, all the inputs (attributes) have been normalized into the range [0,1]. Fifty trials are conducted for each task.

A. Real World Problem: CW System

The system under consideration is the CW system of a power generation plant of Tenaga Nasional Berhad in Penang, Malaysia. The CW system plays an important role in the power generation station for providing sufficient and continuous amount of cooling water to the main turbine condensers and for condensing steam discharged from the turbine exhaust and other steam flows into the condenser.

Fig. 3 shows an overview of the CW system for a power plant [42], [43]. The CW system includes all piping and equipment (such as turbine condensers and drum strainer) between seawater intake and the outfall where water is returned to the sea.

A database based on the targeted power generation of 80 MW is established. The operating conditions of the CW system are categorized into four classes, as shown in Table I. A set of real sensor measurements of the CW system was collected from the power station. This data set consists of 2500 data samples, containing 12 attributes as tabulated in Table II. The data set is predivided into training and testing.

Following closely to ELM random input weights technique, F-ELM only tunes one parameter that is the number of hidden neurons. Note that the number of hidden neuron is equivalent to the number of rules. Each hidden neuron can be interpreted to a rule in the if-then format. For the sake of performance comparison with the methods published in [43], we consider rules computed by eight hidden neurons. Hence, Table III delineates the total eight rules captured from F-ELM for the CW data sets. Each row of the rule in Table III can be translated into a fuzzy rule.

Table IV portrays the third rule elicited from F-ELM in detail in the form of binary matrix. Note that each attribute encompasses different combination of membership functions. Obviously the last bit of each column denotes the DC bit for each of the attribute. If it is a 1, it means DC status is true such that the attribute can be negligible, if it is a 0, it means DC status is false and thus the respective input attribute and the membership functions should be remained as part of the rules. In addition to the detailed binary matrix of rule-3, interpretation of rule-3 in terms of the if-then format and the respective fuzzy equation from F-ELM is described in Table V.

For comparison purposes, three other FIS models based on the rule sets presented in [3]–[35] are included in Table VI. The results show that the performance of F-ELM is comparable with that of IGART-FIS when the number of rules of F-ELM is set the same as IGART-FIS method. Note that F-ELM becomes superior to other classifiers when the experiment allows more rules. As demonstrated by Table VI, F-ELM seems to improve the testing accuracy dramatically,

TABLE IV
CW RULE-3 FROM THE F-ELM IN BINARY MATRIX FORM

.	IF												THEN Class
Membership Functions	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	Is
very low	1	0	0	0	0	0	0	0	0	0	0	0	
low	0	1	0	1	0	0	0	0	1	0	1	0	
medium	0	1	0	1	0	0	0	0	1	0	0	0	4
high	0	1	0	1	0	0	0	0	0	0	0	0	
very high	0	0	0	1	0	0	0	0	1	0	0	0	
Don't Care	0	0	1	0	1	1	1	1	0	1	0	1	

TABLE V
INTERPRETATION OF RULE-3 AND ITS RESPECTIVE FUZZY EQUATIONS ELICITED FROM F-ELM FOR THE CW DATA SET

If
LPT A is very low
And LPT B is low Or medium Or high
And CWIT A is low Or medium Or high Or very high
And CWOP A is low Or medium Or very high
And CWOP B is low

Then Heat transfer in the condenser is not efficient and there is significant blockage in the piping system

Firing strength of Rule–3 is

$$w_3 = [\mu_1(LPTA)]*[\mu_2(LPTB) \otimes \mu_3(LPTB) \otimes \mu_4(LPTB)]*$$
$$[\mu_2(CWITA) \otimes \mu_3(CWITA) \otimes \mu_4(CWITA) \otimes \mu_5(CWITA)]*$$
$$[\mu_2(CWOPA) \otimes \mu_3(CWOPA) \otimes \mu_5(CWOPA)]*[\mu_2(CWOPB)]$$

Constant for the output variable of Rule–4 is

$\mathbf{f}_3 = \boldsymbol{\alpha}_3 = [-0.66\ \ -0.24\ \ -0.11\ \ 1\ ]$,

where class label vector is [-1 -1 -1 1] (class 4) with confidence factor [0.66 0.24 0.11 1].

TABLE VI
PERFORMANCE COMPARISON OF F-ELM AND OTHER CLASSIFIERS FOR THE CW DATA SET

Algorithm	**Number of Rules (Fuzzy Hidden Neurons)**	**Testing Accuracy (%)**
F- ELM	15	91.22
F- ELM	8	85.14
IGART-FIS [35]	8	86.10
FAM-FIS [3, 35]	18	79.90
FAM-RecBFN-FIS [3,35]	9	67.20

with 91.22% when the number of rules is equal to 15 (which are still less than the 18 rules used by FAM-FIS).

B. Benchmark Problems

Here, the performance of the proposed F-ELM is evaluated using a few benchmark problems, comprising the binary data sets, and multiclass data sets [41]. The results of F-ELM are compared against alternative approaches, as shown in Sections IV-B1 and IV-B2.

1) Performance Comparison of Binary Data Sets With Neuro-Fuzzy Approaches: Experimental results of F-ELM on two medical data sets [41], namely PID and WBC are presented. To benchmark the results against other neuro-fuzzy approaches, the setup is in accordance to [43]–[45] and the rule extraction method follows that in [43]–[45].

The PID data set consists of 768 data samples from two classes with eight attributes, in which 268 data samples are belong to patients diagnosed with diabetic, and the rest of the data samples are belong to patients considered as healthy. In accordance to [45], the data set was divided into three subsets, 50% for training, 25% for validation, and the remainder 25% for testing. In this experiment, the results of F-ELM are compared with Enhanced Generalized Adaptive Resonance Theory (EGART)-FIS, and Modified Fuzzy Min-Max Neural Network-FIS.

TABLE VII
PERFORMANCE COMPARISON OF F-ELM AND OTHER CLASSIFIERS FOR THE PID DATA SET

Algorithm	Number of Hidden Neuron	Training Accuracy	Testing Accuracy
F-ELM	9	75.35	74.09
EGART-FIS [45]	6	n/a	73.05
MFMM-FIS [43]	5	n/a	72.92

TABLE VIII
PERFORMANCE COMPARISON OF F-ELM AND OTHER CLASSIFIERS FOR THE WBC DATA SET

Algorithm	Number of Hidden Neuron	Training Accuracy	Testing Accuracy
F-ELM	6	93.94	94.17
EGART-FIS [45]	3	n/a	93.56
MFMM-FIS [43]	4	n/a	92.56

On the other hand, the WBC data set consists of 699 data samples of virtually assessed nuclear features of fine needle aspirates from patients, with nine attributes. Four hundred and fifty-eight data samples show benign while the remaining samples are diagnosed as malignant. In accordance with the experimental procedure in [45], the data set was divided into three subsets, 50% for training, 30% for validation, and 20% for testing.

The classification results of F-ELM for PID data set are recorded in Table VII. It is noticeable that F-ELM achieves the best testing accuracy of 74.9%. Although F-ELM constructs more rules than EGART-FIS and FAM-FIS, the total number of attributes in operation of F-ELM is lesser as compared to all other approaches listed in Table VII. It is due to the insertion of DC matrix, which helps in the attributes reduction. For example, although EGART-FIS yielded six rules, with all the eight attributes included, it results in the total attributes accumulated in the rule base are 48. In contrast, F-ELM only has a total of 32 attributes in its rule base, besides showing a 1.04% performance improvement as compared with EGART-FIS.

The training and testing accuracy of F-ELM for the WBC data set are summarized in Table VIII. F-ELM attains the best testing accuracy of 94.17% among all the three methods listed in Table VIII. Although F-ELM derives three more fuzzy rules as compared with EGART-FIS; the overall attributes size in the rule base of F-ELM is actually much smaller than the rule base of EGART-FIS with the incorporation of DC concept that facilitates attribute reduction. F-ELM does not keep every insignificant and redundant attribute in the rule base.

In short, the remarkable performance of F-ELM coupled with an advantage to bring about attributes reduction guarantees a good level of interpretability.

TABLE IX
SPECIFICATION OF BENCHMARK DATA SETS FOR EXPERIMENTS [46]

Data sets	# Attributes	# Class or output	# Training Samples	# Testing Samples
Image Segmentation	18	7	1,500	810
Satellite Image	36	6	4,435	2,000
DNA	180	3	2,000	1,186

2) Performance Comparison of Multiclass Data Sets With ELM Approaches: Three other benchmark problems [41], i.e., Image segmentation, Satellite Image, DNA, are evaluated. Table IX details the specifications of these problems.

The satellite image problem comprises data set generated from the Landsat multispectral scanner. There are four digital images of the same scene in four different spectral bands in one frame of the Landsat multispectral scanner imagery. The database is a tiny subarea of a scene, consisting of 82×100 pixels. Each sample in the data set corresponds to a region of 3×3 pixels. The aim is recognize the central pixel of a region into six categories, namely red soil, cotton crop, gray soil, damp gray soil, soil with vegetation stubble, and very damp gray soil given the multispectral value for each region. The training and test sets contain 4435 samples and 2000 samples, respectively.

The image segmentation data set consists of 2310 regions of 3×3 pixels, which were randomly drawn from seven outdoor images. The objective is to classify each region into one of the seven categories, namely brick facing, sky, foliage, cement, window, path, and grass, using 19 attributes extracted from each square region.

The data set of primate splice-junction gene sequences with associated imperfect domain theory is known as the DNA data set [46]. Splice junctions are points on a DNA sequence at which superfluous DNA is removed during the process of protein creation in higher organisms. The aim of the DNA data set is to recognize the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out) for a given sequence of DNA. This consists of three subtasks: recognizing exon/intron boundaries (referred to as EI sites), intron/exon boundaries (IE sites), and neither (sites). A given sequence of DNA consists of 60 elements (called nucleotides or base-pairs). Every symbolic variable representing nucleotides is coded as three binary indicator variables, thus resulting in 180 binary attributes.

In the simulation, we also studied the case where the F-ELM is given different number of hidden neurons, as tabulated in Table X. The number of hidden neurons and testing accuracy from other methods for the satellite image data set is recorded in Table XI. For a fair and transparent comparison, F-ELM applies the same number of fuzzy hidden neuron as what is presented in ELM. Note that F-ELM improves the accuracy rate of both the ELM considerably, which is 0.7% better than ELM with RBF activation function and 0.76% higher than ELM with sigmoid activation function.

TABLE X
TRAINING AND TESTING ACCURACY OF F-ELM FOR SATELLITE IMAGE DATA SET OVER 50 RUNS

Number of Rules (Fuzzy Hidden Neurons)	Training Accuracy (%)	Testing Accuracy (%)
200	89.64	89.02
250	90.19	89.17
300	91.05	89.80
350	91.55	89.82
400	91.80	89.73
450	92.29	89.99
500	92.53	89.81
550	92.75	89.87
600	93.20	90.19

TABLE XI
PERFORMANCE COMPARISONS OF F-ELM AND ELM-BASED CLASSIFIERS FOR SATELLITE IMAGE DATA SET

Algorithm	Number of Hidden Neuron	Training Accuracy (%)	Testing Accuracy (%)
F- ELM	600	93.20	90.19
F- ELM	400	91.80	89.73
ELM (RBF) [46]	400	92.94	89.03
ELM (Sigmoid) [46]	400	91.95	88.97
EM-ELM [47]	203	n/a	82.71
E-ELM [48]	200	n/a	88.44
E-OSELM [49]	400	92.79	89.01

TABLE XII
TRAINING AND TESTING ACCURACY OF F-ELM FOR IMAGE SEGMENTATION DATA SET OVER 50 RUNS

Number of Rules (Fuzzy Hidden Neurons)	Training Accuracy (%)	Testing Accuracy (%)
150	96.77	94.74
180	97.33	95.29
210	97.77	95.42
240	98.06	95.50
270	98.29	95.55
300	98.50	95.56
330	98.71	95.60

TABLE XIII
PERFORMANCE COMPARISON OF F-ELM AND ELM-BASED CLASSIFIERS FOR IMAGE SEGMENTATION DATA SET

Algorithm	Number of Hidden Neuron	Training Accuracy (%)	Testing Accuracy (%)
F- ELM	330	98.71	95.60
F- ELM	180	97.33	95.29
ELM (Sigmoid) [46]	180	96.75	95.07
ELM (RBF) [46]	180	96.22	94.91
EM-ELM [47]	35	n/a	88.17
E-ELM [48]	200	n/a	95.46
E-OSELM [49]	180	97.08	94.79

As enumerated in Table X, F-ELM accuracy rates continue to surge when the number of hidden neuron increases. F-ELM is also compared against other ELM-based approaches, such as the Error Minimized Extreme Learning Machine, evolutionary-ELM (E-ELM), and Ensemble of Online Sequential Extreme Learning Machine (E-OSELM). The results of F-ELM are seen superior to the aforementioned approaches. With 200 hidden neurons F-ELM outperforms E-ELM by 0.73%.

For the image segmentation data set, training and testing accuracy rates for 50 trials with number of fuzzy hidden neuron set to different values, are captured in Table XII. While the classification performance of F-ELM and other learning algorithms are compared and presented in Table XIII. F-ELM tops the performance of all methods with average testing accuracy of 95.60%, which is a significant improvement as compared with both the ELM (RBF and sigmoid). As narrated in Table XIII, even when F-ELM applies the same number of hidden neurons as the ELM methods, F-ELM still outperforms ELM (sigmoid) and ELM (RBF) by 0.22% and 0.38%, respectively. Besides, the result of F-ELM is also comparable with that of E-ELM and E-OSELM.

On the other hand, the effect of the number of hidden neurons on the network performance for the DNA data set is shown in the Table XIV. The average results of F-ELM, ELM (sigmoid), and ELM (RBF) in terms of classification

TABLE XIV
TRAINING AND TESTING ACCURACY OF F-ELM FOR DNA DATA SET OVER 50 RUNS

Number of Rules (Fuzzy Hidden Neurons)	Training Accuracy (%)	Testing Accuracy (%)
80	86.44	83.50
100	90.23	87.07
120	92.86	89.70
140	94.65	91.70
160	95.89	93.39
180	96.85	94.65
200	97.03	94.62

TABLE XV
PERFORMANCE COMPARISON OF F-ELM AND ELM-BASED CLASSIFIERS FOR DNA DATA SET

Algorithm	Number of Hidden Neuron	Training Accuracy (%)	Testing Accuracy (%)
F- ELM	200	96.85	94.65
ELM (Sigmoid) [46]	200	96.90	94.30
ELM (RBF) [46]	200	95.87	92.33
OSELM (Sigmoid) [46]	200	95.79	93.43
OSELM (RBF) [46]	200	96.12	94.37

accuracy and number of neurons are summarized in Table XV. Two levels of membership function are assigned, each centered at 0 and 1, to adapt to the binary form of all input attributes of the DNA data set. With 180 hidden neurons, obviously F-ELM yields the best testing accuracy rates of 94.65%. ELM with sigmoid activation function is achieving 94.30% and ELM with RBF activation function is getting only 92.33%. It should also be noted that, F-ELM obtains better testing accuracy rate as compared with both the OSELM-sigmoid and OSELM-RBF.

In general, the simulation results on the real world application and benchmark data sets have shown that the proposed F-ELM, is immensely efficient with good performance in undertaking the pattern classification tasks, and it comes with an extraordinary advantage of providing transparent and interpretable rule base.

V. CONCLUSION

In this paper, we have demonstrated that F-ELM leverages the merits of ELM that guarantees the universal approximation capability at an extremely fast speed and FIS for being a fuzzy reasoning model. F-ELM exhibits a methodology of embedding fuzzy rules into its hidden layer. As such, we do not have to constitute additional FIS to examine the performance of the extracted fuzzy if-then rules as presented by many researchers in the literature. The method of classification proposed here takes into consideration the quality of the fuzzy rules as it undergoes inference operation during learning.

The advantage of F-ELM has been put to evidence. The practical experiments have laid out F-ELM is capable of producing interpretable rules without compromising on the classification accuracy. The implementation of DC approach is proven useful for restructuring the knowledge acquired in F-ELM and eliminating those unnecessary input attributes such that the classification ability of F-ELM could be enhanced. The results indicate the potential of F-ELM in maintaining good prediction performance even when used without expert knowledge. All in all, the nature of fast learning ELM in combination to the highly comprehensible fuzzy rules makes the proposed F-ELM a promising candidate for approximation, modeling and control of complex processes.

As for future work, it would be interesting to exploit F-ELM with the incremental learning capabilities by applying it with the OSELM approach. The incremental learning strategy has its advantage especially when applied to the industrial applications where the data samples may arrive sequentially. In addition, incorporating F-ELM in the data regression analysis can also be another challenging application to be investigated in further studies.

REFERENCES

[1] J. M. Benitez, J. L. Castro, and I. Requena, "Are artificial neural networks black boxes?" *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1156–1164, Sep. 1997.

[2] R. Andrews, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowl. Based Syst.*, vol. 8, no. 6, pp. 373–389, Dec. 1995.

[3] S. C. Tan, C. P. Lim, and M. V. C. Rao, "A hybrid neural network model for rule generation and its application to process fault detection and diagnosis," *Eng. Appl. Artif. Intell.*, vol. 20, no. 2, pp. 203–213, Mar. 2007.

[4] Z. Sun, K.-F. Au, and T.-M. Choi, "A neuro–fuzzy inference system through integration of fuzzy logic and extreme learning machine," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 5, pp. 1321–1331, Oct. 2007.

[5] S. Mitra and Y. Hayashi, "Neuro–fuzzy rule generation: Survey in soft computing framework," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 748–768, May 2000.

[6] V. Miranda and A. R. G. Castro, "Improving the IEC table for transformer failure diagnosis with knowledge extraction from neural networks," *IEEE Trans. Power Del.*, vol. 20, no. 4, pp. 2509–2514, Oct. 2005.

[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, May 2006.

[8] G.-B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, nos. 1–3, pp. 155–163, Dec. 2010.

[9] Y. Lan, Y. C. Soh, and G.-B. Huang, "Two-stage extreme learning machine for regression," *Neurocomputing*, vol. 73, nos. 16–18, pp. 3028–3038, 2010.

[10] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, nos.1–3, pp. 359–366, Dec. 2008.

[11] H. White, "An additional hidden unit test for neglected nonlinearity in multilayer feedforward networks," in *Proc. Int. Joint Conf. Neural Netw.*, Washington, DC, USA, Jun. 1989, pp. 451–455.

[12] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, nos. 16-18, pp. 3460–3468, Oct. 2008.

[13] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, nos. 16–18, pp. 3056–3062, Oct. 2007.

[14] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: A survey," *Int. J. Mach. Lean. Cybern.*, vol. 2, no. 2, pp. 107–122, 2011.

[15] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 1, pp. 28–44, Jan. 1973.

[16] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Inf. Sci.*, vol. 8, no. 3, pp. 199–249, 1975.

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

[18] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran, "Online sequential fuzzy extreme learning machine for function approximation and classification problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 4, pp. 1067–1072, Aug. 2009.

[19] H. Ying, "Sufficient conditions on uniform approximation of multivariate functions by general Takagi–Sugeno fuzzy systems with linear rule consequent," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 28, no. 4, pp. 515–520, Jul. 1998.

[20] J.-S. R. Jang, and C. T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Tans. Neural Netw.*, vol. 4, no. 1, pp. 156–159, Jan. 1993.

[21] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 801–806, Sep. 1992.

[22] R. J. Kuo and P. H. Cohen, "Manufacturing process control through integration of neural networks and fuzzy model," *Fuzzy Sets Syst.*, vol. 98, no. 1, pp. 15–31, Aug. 1998.

[23] M. E. Cohen and D. L. Hudson, "Approaches to the handling of fuzzy input data in neural networks," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, San Diego, CA, USA, Mar. 1992, pp. 93–100.

[24] B. Kosko, *Neural Networks and Fuzzy Systems*. Eaglewood Cliffs, NJ, USA: Prentice Hall, 1992.

[25] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, Nov./Dec. 1992.

[26] J. E. Moreno, O. Castillo, J. R. Castro, L. G. Martínez, and P. Melin, "Data mining for extraction of fuzzy IF-THEN rules using Mamdani and Takagi–Sugeno–Kang FIS," *Eng. Lett.*, vol. 15, no. 1, pp. 82–88, 2007.

[27] L. Aguilar, P. Melin, and O. Castillo, "Intelligent control of a stepping motor drive using a hybrid neuro–fuzzy ANFIS approach," *Appl. Soft Comput.*, vol. 3, no. 3, pp. 209–219, 2003.

[28] P. Melin and O. Castillo, "Intelligent control of a stepping motor drive using an adaptive neuro–fuzzy inference system," *Inf. Sci.*, vol. 170, nos. 2–4, pp. 133–151, Feb. 2005.

[29] S. L. Cardenas-Maciel, O. Castillo, L. T. Aguilar, and J. R. Castro, "A T-S fuzzy logic controller for biped robot walking based on adaptive network fuzzy inference system," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2010, pp. 1–8.

[30] P. Melin and O. Castillo, "Adaptive intelligent control of aircraft systems with a hybrid approach combining neural networks, fuzzy logic and fractal theory," *Appl. Soft Comput.*, vol. 3, no. 4, pp. 353–362, 2003.

[31] A. Abu-Siada, S. Hmood, and S. Islam, "A new fuzzy logic approach for consistent interpretation of dissolved gas-in-oil analysis," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 20, no. 6, pp. 2343–2349, Dec. 2013.

[32] F. Gaxiola, P. Melin, F. Valdez, and O. Castillo, "Interval type-2 fuzzy weight adjustment for backpropagation neural networks with application in time series prediction," *Inf. Sci.*, vol. 260, pp. 1–14, Mar. 2014.

[33] V. S. Chourasia, A. K. Tiwari, and R. Gangopadhyay, "Interval type-2 fuzzy logic based antenatal care system using phonocardiography," *Appl. Soft Comput.*, vol. 14, pp. 489–497, Jan. 2014.

[34] S. O. Olatunji, A. Selamat, and A. A. A. Raheem, "Improved sensitivity based linear learning method for permeability prediction of carbonate reservoir using interval type-2 fuzzy logic system," *Appl. Soft Comput.*, vol. 14, pp. 144–155, Jan. 2014.

[35] K. S. Yap, C. P. Lim, and M. T. Au, "Improved GART neural network model for pattern classification and rule extraction with application to power systems," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2310–2323, Dec. 2011.

[36] H. Takagi and I. Hayashi, "NN-driven fuzzy reasoning," *Int. J. Approx. Reasoning*, vol. 5, no. 3, pp. 191–212, May 1991.

[37] T. Yamaguchi, N. Imasaki, and K. Haruki, "Fuzzy rule realization on associative memory system," in *Proc. Int. Joint Conf. Neural Netw.*, Washington, DC, USA, 1990, pp. 720–723.

[38] L. I. Kuncheva, "How good are fuzzy if-then classifiers?" *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 4, pp. 501–509, Aug. 2000.

[39] F. Klawon and E.-P. Klement, "Mathematical analysis of fuzzy classifiers," *Advances in Intelligent Data Analysis Reasoning About Data* (Lecture Notes in Computer Science), vol. 1280. Berlin, Germany: Springer-Verlag, 1997, pp. 359–370.

[40] *System Description and Operating Procedures*. Praí Power Station Stage 3, Piraí, Brazil, 1999.

[41] C. Blake and C. Merz. (1998). "UCI repository of machine learning databases," Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, USA. [Online]. Available: http://archive.ics.uci.edu/ml/

[42] S. C. Tan and C. P. Lim, "Application of an adaptive neural network with symbolic rule extraction to fault detection and diagnosis in a power generation plant," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 369–377, Jun. 2004.

[43] A. Quteishat and C. P. Lim, "A modified fuzzy min–max neural network with rule extraction and its application to fault detection and classification," *Appl. Soft Comput.*, vol. 8, no. 2, pp. 985–995, Mar. 2008.

[44] G. A. Carpenter and A. Tan, "Rule extraction: From neural architecture to symbolic representation," *Connection Sci.*, vol. 7, no. 1, pp. 3–27, 1995.

[45] K. S. Yap, C. P. Lim, and J. Mohamad-Salleh, "An enhanced generalized adaptive resonance theory neural network and its application to medical pattern classification," *J. Intell. Fuzzy Syst.*, vol. 21, nos. 1–2, pp. 65–78, Jan. 2010.

[46] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.

[47] G. Feng, G.-B. Huang, Q.-P. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 8, pp. 1352–1357, Aug. 2009.

[48] N. Liu and H. Wang, "Ensemble based extreme learning machine," *IEEE Signal Process. Lett.*, vol. 17, no. 8, pp. 754–757, Aug. 2010.

[49] Y. Lan, Y. C. Soh, and G.-B. Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, nos. 13–15, pp. 3391–3395, Aug. 2009.

Shen Yuong Wong received the bachelor's degree and the M.Sc. (Hons.) degree in electrical and electronic engineering from Universiti Tenaga Nasional, Selangor, Malaysia, in 2010 and 2012, respectively, where she is currently pursuing the Ph.D. degree in engineering.

Her current research interests include application of artificial intelligence, pattern recognition, fuzzy logic, and extreme learning machines.

Keem Siah Yap received the B.Eng. (Hons.) and M.Sc. degrees in electrical engineering from the University of Technology Malaysia, Johor Bahru, Malaysia, in 1998 and 2000, respectively, and the Ph.D. degree in electronics engineering from the Universiti Sains Malaysia, Pulau Penang, Malaysia.

He is currently an Associate Professor and the Program Head of the College of Graduate Studies with Universiti Tenanga Nasional, Selangor, Malaysia. His current research interests include theory and applications of artificial intelligence.

Hwa Jen Yap received the bachelor's degree in mechanical engineering, the M.Eng.Sc (Master of Engineering Science) degree, and the Ph.D. degree in engineering from the University of Malaya, Kuala Lumpur, Malaysia, in 2000, 2005, and 2012, respectively.

He is currently the Head of the Centre for Product Design and Manufacturing at the University of Malaya. He is also a Senior Lecturer with the Department of Mechanical Engineering, Faculty of Engineering, University of Malaya. His current research interests included virtual reality, virtual manufacturing, computer-aided design/computer-aided manufacturing/computer-aided engineering, human–computer interface, artificial intelligence, robotics, and automation.

Shing Chiang Tan received the B.Tech. (Hons.) and M.Sc. (Eng.) degrees from Universiti Sains Malaysia, Pulau Penang, Malaysia, in 1999, and the Ph.D. degrees from Multimedia University, Melaka, Malaysia, in 2002 and 2008, respectively.

He is currently an Associate Professor with the Faculty of Information Science and Technology, Multimedia University. His current research interests include computational intelligence (artificial neural networks, evolutionary algorithms, fuzzy logic, and decision trees) and its applications, which include but are not limited to, data classification, condition monitoring, fault detection and diagnosis, and biomedical disease classification.

Dr. Tan was a recipient of the Matsumae International Foundation Fellowship, Japan, in 2010.

Siow Wee Chang received the Ph.D. degree in computer science from the University of Malaya, Kuala Lumpur, Malaysia, in 2013.

She is currently a Senior Lecturer of Bioinformatics and Computational Biology with the Institute of Biological Sciences, Faculty of Science, University of Malaya. Her current research interests include artificial intelligence and machine learning, biomedical informatics, and human–computer interaction.