Fuzzy Broad Learning System: A Novel Neuro-Fuzzy Model for Regression and Classification

Shuang Feng and C. L. Philip Chen, *Fellow, IEEE*

***Abstract*—A novel neuro-fuzzy model named fuzzy broad learning system (BLS) is proposed by merging the Takagi–Sugeno (TS) fuzzy system into BLS. The fuzzy BLS replaces the feature nodes of BLS with a group of TS fuzzy subsystems, and the input data are processed by each of them. Instead of aggregating the outputs of fuzzy rules produced by every fuzzy subsystem into one value immediately, all of them are sent to the enhancement layer for further nonlinear transformation to preserve the characteristic of inputs. The defuzzification outputs of all fuzzy subsystem and the outputs of enhancement layer are combined together to obtain the model output. The *k*-means method is employed to determine the centers of Gaussian membership functions in antecedent part and the number of fuzzy rules. The parameters that need to be calculated in a fuzzy BLS are the weights connecting the outputs of enhancement layer to model output and the randomly initialized coefficients of polynomials in consequent part in fuzzy subsystems, which can be calculated analytically. Therefore, fuzzy BLS retains the fast computational nature of BLS. The proposed fuzzy BLS is evaluated by some popular benchmarks for regression and classification, and compared with some state-of-the-art nonfuzzy and neuro-fuzzy approaches. The results indicate that fuzzy BLS outperforms other models involved. Moreover, fuzzy BLS shows advantages over neuro-fuzzy models regarding to the number of fuzzy rules and training time, which can ease the problem of rule explosion to some extent.**

***Index Terms*—Broad learning system (BLS), classification, *k*-means, regression, Takagi–Sugeno (TS) fuzzy system.**

Manuscript received May 3, 2018; revised July 2, 2018; accepted July 13, 2018. Date of publication August 10, 2018; date of current version December 16, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61751202, Grant 61751205, and Grant 61572540, in part by the Macau Science and Technology Development Fund (FDCT) under Grant 019/2015/A1, Grant 079/2017/A2, and Grant 024/2015/AMJ, in part by the Multiyear Research Grants of University of Macau, and in part by the Teacher Research Capacity Promotion Program of Beijing Normal University, Zhuhai. This paper was recommended by Associate Editor C.-F. Juang. *(Corresponding author: C. L. Philip Chen.)*

S. Feng is with the School of Applied Mathematics, Beijing Normal University (Zhuhai Campus), Zhuhai 519085, China, and also with the Faculty of Science and Technology, University of Macau, Macau 99999, China (e-mail: fengshuang@bnuz.edu.cn).

C. L. P. Chen is with the Faculty of Science and Technology, University of Macau, Macau 99999, China, also with the College of Navigation, Dalian Maritime University, Dalian 116026, China, and also with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China (e-mail: philip.chen@ieee.org).

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/TCYB.2018.2857815

I. Introduction

THE COMBINATION of human-like reasoning style based on a set of IF-THEN fuzzy rules of fuzzy system with the learning and connecting structure of neural network (NN) leads to a hybrid system which is widely termed as fuzzy NN or neuro-fuzzy model in the literature.

The neuro-fuzzy approach can to some extent overcome the main problems existing in NNs and fuzzy systems, i.e., the deficiency of explaining the knowledge learned by an NN and the severe dependence on experience of experts for establishing fuzzy rule base. Specifically, the NN usually functions as a black box after trained by given data which lacks the ability to tell the user how and why this network produces the results and does not reveal enough information of the system it approximates. So far as fuzzy system is concerned, although it possesses better interpretability, the fuzzy rules are predefined according to expert knowledge in most cases and will not change with the updating input data which often results in low accuracy. Moreover, fuzzy systems require a huge amount of fuzzy rules (i.e.,“rule explosion”) to achieve a satisfactory accuracy when encountering some large databases or data with high dimension [1].

Neuro-fuzzy models inherit the merits of NN and fuzzy system. The main advantage is that they are universal approximators and capable of adapting interpretable IF-THEN rules to training data [2]. The parameters of membership functions corresponding to the fuzzy sets in fuzzy rules can be initialized randomly and updated iteratively by some traditional training algorithms of NN. Because of the fine properties of neuro-fuzzy models, different variants have been proposed until now and applied to diverse fields including nonlinear system identification and control [3]–[8], time series prediction [9], [10], regression [11], [12] and classification [13]–[19].

However, most of the neuro-fuzzy models follow the conventional way of training an NN, i.e., the parameters in fuzzy rules are learned by training data in an iterative manner. When the databases of real-world problems are in large scale which frequently appear in modern society, the learning process can be very time-consuming. Although the merit of NN can help to extract the suitable fuzzy rules without the intervention of human, it is still important to figure out how to reduce the number of fuzzy rules without affecting the accuracy when the dimension of inputs is relatively high.

To ease the aforementioned problems in neuro-fuzzy models, some improved structures and training algorithms concentrating on reducing fuzzy rules and speeding up the learning phase have been proposed recently. Hierarchical hybrid fuzzy NN [20], [21] uses some fuzzy subsystems to randomly aggregate several discrete input attributes into an intermediate output and an NN to handle the rest continuous input variables together with the intermediate outputs which can reduce the input dimension and fuzzy rules. But there is no common method of selecting the appropriate discrete attributes for combination. Online sequential fuzzy extreme learning machine (OS-F-ELM) [22] randomly assigns all the antecedent parameters of membership functions without further updating, and determines the corresponding consequent parameters like ELM to cut down the learning time. Fuzzy wavelet polynomial NN (FWPNN) [23] and fused fuzzy deep NN [24] employ k-means method to group the data and choose the clustering centers to initialize the parameters of Gaussian membership functions in premise part of fuzzy rules followed by an iterative adjustment using PSO or BP, which is very time-consuming. The neuro-fuzzy inference system developed in [25] also groups the data by k-means but derives membership functions in antecedent and consequent of fuzzy rules through an ELM. Fuzzy ELM (F-ELM) [26] and its improve version IF-ELM [27] use randomly generated rule-combination matrix and "don't care" matrix to reduce the number of fuzzy rules with fixed centers and random widths for Gaussian membership functions, and determine the consequent parameters analytically. However, most of the aforementioned neuro-fuzzy models consider only one fuzzy systems in their structures.

A fast and efficient network called broad learning system (BLS) [28] has been developed very recently. It expands the neurons consisting of feature and enhancement nodes in a broad manner without stacking layers in deep, and calculates the weights by pseudoinverse. The fast and accurate properties of BLS motivate our attempt to integrate it with fuzzy systems and design a novel neuro-fuzzy model.

We replace the left part of BLS consisting of feature nodes with Takagi–Sugeno (TS) fuzzy subsystems so as to establish a new neuro-fuzzy model named fuzzy BLS. The distinguish characteristics of fuzzy BLS that make it different from other neuro-fuzzy approaches are summarized as follows.

1) The fuzzy BLS contains a group of first-order TS fuzzy subsystems, and the input data are processed by each of them. All the fuzzy subsystems are engaged in producing the output of a fuzzy BLS, thus it can benefit from this "ensemble" structure.
2) The k-means algorithm is employed to group the input data and determine the number of fuzzy rules in each fuzzy subsystem, as well as the centers of Gaussian membership functions in antecedent part. Due to the property of k-means algorithm, different centers will be generated from the training data for each fuzzy subsystem which ensures that different results can be produced. Then the information of input data can be extracted as much as possible.
3) The outputs of fuzzy rules in a fuzzy subsystem are not aggregated into one value immediately. Instead, these intermediate values produced by all fuzzy subsystems are catenated as vectors and directly sent to the enhancement nodes for nonlinear transformation. Then the outputs of enhancement layer together with the defuzzification outputs of fuzzy subsystems are used to produce the final model output.
4) The parameters of a fuzzy BLS consist of the weights connecting the outputs of enhancement layer to the final output layer and the coefficients in the consequent part of fuzzy rules in every fuzzy subsystem, which can be calculated by pseudoinverse rapidly. Therefore, fuzzy BLS retains the fast computational nature of BLS.

The proposed fuzzy BLS is evaluated by some popular benchmarks for regression and classification. Its performance is compared with some state-of-the-art nonfuzzy and neuro-fuzzy approaches. The results indicate that fuzzy BLS can achieve the highest testing accuracies in almost all experiments among the models involved. Moreover, fuzzy BLS needs fewer rules and less training time yet behaves much better than the neuro-fuzzy models used for comparison, which demonstrates its superiority.

The rest of this paper is organized as follows. The brief ideas of BLS and TS fuzzy system are summarized in Section II. The structure and learning algorithm of proposed fuzzy BLS are introduced in Section III. And some regression and classification benchmarks are carried out in Section IV to compare the performance of fuzzy BLS with some recently proposed nonfuzzy and neuro-fuzzy models. Section V gives the conclusions and some further discussions on fuzzy BLS.

II. Preliminaries

A. Broad Learning System

BLS [28] is an improved flat framework [29] which inherits the major feature and superiority of a random vector functional-link NN [30]. Contrary to some popular deep networks suffering from a time-consuming learning for excessive parameters, BLS can provide an alternative algorithm for large-scale data classification that is much faster with a comparable or a slight loss of accuracy. The structure of a BLS expanded in a wide sense is illustrated in Fig. 1. Recently, an enhanced model of BLS with graph regularization is proposed to improve its performance on image recognition [31].

In a BLS, the original input vectors are first mapped into random features in feature nodes by some feature mappings. Then the random features are sent into the "enhancement nodes" for nonlinear transformation. Further, the random features together with the outputs of the enhancement nodes are connected to the output layer, and the connection weights are the model parameters which are calculated rapidly by ridge regression of the pseudoinverse.

Given a training dataset $\{\boldsymbol{X}, \boldsymbol{Y}\}$ and n feature mappings ϕ_i, then the ith mapped feature matrix is

$$\boldsymbol{Z}_i = \phi_i(\boldsymbol{X}\boldsymbol{W}_{e_i} + \boldsymbol{\beta}_{e_i}), \quad i = 1, 2, \ldots, n \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{N\times M}$, $\boldsymbol{Y} \in \mathbb{R}^{N\times C}$, N is the number of input samples, M is the dimension of each sample, C is the dimension of corresponding outputs, weights $\boldsymbol{W}_{e_i}$ and bias term $\boldsymbol{\beta}_{e_i}$ are randomly generated matrices with the proper dimensions. ϕ_i is usually a linear transformation.

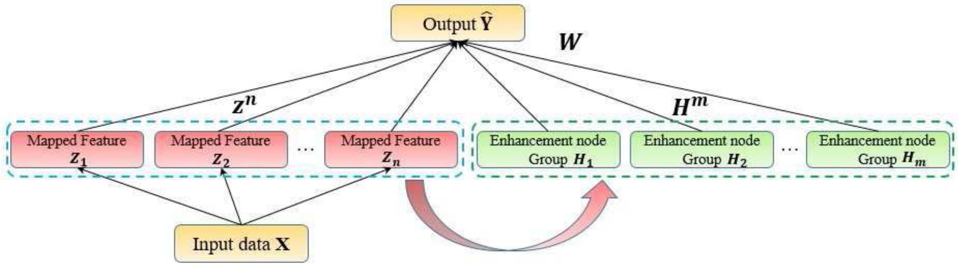

Fig. 1. Structure of a BLS.

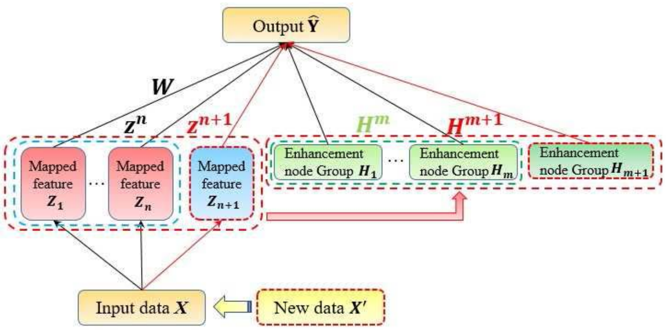

Fig. 2. Increments of inputs, feature nodes, and enhancement nodes for BLS.

We denote $\boldsymbol{Z}^n \triangleq (\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n)$ as the outputs of n groups of feature nodes. To obtain sparse features $\boldsymbol{Z}^n$ of input data, the randomly initialized weight matrix $\boldsymbol{W}_{e_i}$ is fine-tuned by a sparse autoencoder.

Then $\boldsymbol{Z}^n$ is connected to the layer of enhancement nodes for nonlinear transformation. The output matrix of the jth group of enhancement nodes is

$$\boldsymbol{H}_j \triangleq \xi_j\left(\boldsymbol{Z}^n \boldsymbol{W}_{h_j} + \boldsymbol{\beta}_{h_j}\right), \; j = 1, 2, \ldots, m \tag{2}$$

where the activation function $\xi_j = \tanh(x)$, $\boldsymbol{W}_{h_j}$ and $\boldsymbol{\beta}_{h_j}$ are weights and bias terms connecting the outputs of feature layer to the enhancement nodes which are randomly generated from [0, 1].

The output matrix of the enhancement layer is denoted by $\boldsymbol{H}^m \triangleq (\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m)$. Therefore, the output of a BLS $\hat{\boldsymbol{Y}}$ has the following form:

$$\begin{aligned}\hat{\boldsymbol{Y}} &= (\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n, \boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m)\boldsymbol{W} \\ &= \left(\boldsymbol{Z}^n, \boldsymbol{H}^m\right)\boldsymbol{W}\end{aligned} \tag{3}$$

where $\boldsymbol{W}$ are the weights connecting the layer of feature nodes and the layer of enhancement nodes to the output layer, and it can be calculated rapidly by the ridge regression approximation of pseudoinverse of $(\boldsymbol{Z}^n, \boldsymbol{H}^m)$ with actual outputs $\boldsymbol{Y}$, i.e.,

$$\boldsymbol{W} = \left(\boldsymbol{Z}^n, \boldsymbol{H}^m\right)^+ \boldsymbol{Y}. \tag{4}$$

Three incremental learning algorithms dealing with three scenarios including the increments of enhancement nodes, feature nodes, and input data (see Fig. 2) are also developed for a BLS without retraining the whole model (refer to [28] for more details).

B. Takagi–Sugeno Fuzzy System

There are two important and popular types of fuzzy systems: 1) the Mamdani and 2) TS fuzzy systems. The TS fuzzy system proposed in [32] is one of the most common fuzzy models, which has been widely applied to diverse fields, including nonlinear system modeling and identification, fuzzy control, fuzzy inference, and reasoning. The main difference from Mamdani type is that the consequent (then part) of every fuzzy rule in a TS fuzzy system is a function of inputs. The typical fuzzy if-then rules in a TS fuzzy system can be represented as

$$\begin{aligned}&\text{If } x_1 \text{ is } A_{k1} \text{ and } x_2 \text{ is } A_{k2} \ldots \text{ and } x_M \text{ is } A_{kM} \\ &\text{then } y_k = f_k(x_1, x_2, \ldots, x_M), \; k = 1, 2, \ldots, K\end{aligned}$$

where A_{kj} is a fuzzy set, x_j is the system input ($j = 1, 2, \ldots, M$), and K is the number of rules.

Usually the function f_k is a polynomial of the input variables, however, it can be any function supposing that it is capable of describing the output properly within the universe of discourse specified by the antecedent of fuzzy rules. The resulting fuzzy system is called a first-order TS fuzzy model with a first-order polynomial f_k. Another popular one is the zero-order TS fuzzy model when f_k is a constant, which can be viewed as a special case of the Mamdani fuzzy system where the consequent of each rule is represented by a fuzzy singleton (or a defuzzified consequent). A zero-order TS fuzzy system has proved functionally equivalent to a radial basis function network under certain minor constraints [33].

The activation level (fire strength) of the kth rule can be computed by

$$\tau_k = \prod_{j=1}^{M} \mu_{kj}\left(x_j\right) \tag{5}$$

where $\mu_{kj}(\cdot)$ is the membership function corresponding to fuzzy set A_{kj}.

Hence, the defuzzified output of a TS fuzzy system is

$$y = \frac{\sum_{k=1}^{K} \tau_k y_k}{\sum_{k=1}^{K} \tau_k} = \frac{\sum_{k=1}^{K} \prod_{j=1}^{M} \mu_{kj}\left(x_j\right) f_k(x_1, x_2, \ldots, x_M)}{\sum_{k=1}^{K} \prod_{j=1}^{M} \mu_{kj}\left(x_j\right)}. \tag{6}$$

III. Fuzzy Broad Learning System

A novel neuro-fuzzy model called fuzzy BLS is proposed by integrating BLS and TS fuzzy systems in this section.

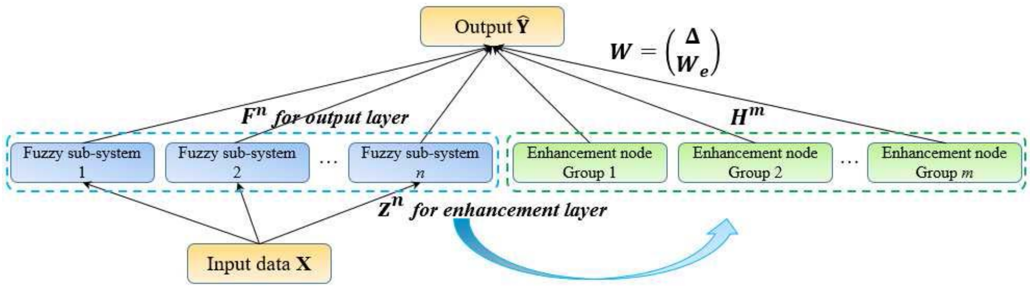

Fig. 3. Structure of a fuzzy BLS.

Fuzzy BLS retains the structure of a BLS, but replaces the feature nodes of a BLS with a group of TS fuzzy subsystems which eventually results in a hybrid neuro-fuzzy network (see Fig. 3). The sparse autoencoder for fine-tuning weights in the feature layer of a BLS is also removed in the fuzzy BLS to reduce the structure complexity.

Suppose that there are n fuzzy subsystems and m groups of enhancement nodes in a fuzzy BLS. The input data are $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)^T \in \mathbb{R}^{N\times M}$ where $\boldsymbol{x}_s = (x_{s1}, x_{s2}, \ldots, x_{sM}), s = 1, 2, \ldots, N$. Suppose that there are K_i fuzzy rules in the ith fuzzy subsystem which have the following form:

If x_{s1} is A^i_{k1} and x_{s2} is A^i_{k2} ... and x_{sM} is A^i_{kM}
then $z^i_{sk} = f^i_k(x_{s1}, x_{s2}, \ldots, x_{sM}),\ k = 1, 2, \ldots, K_i$.

We adopt the first-order TS fuzzy system and let

$$z^i_{sk} = f^i_k(x_{s1}, x_{s2}, \ldots, x_{sM}) = \sum_{t=1}^{M} \alpha^i_{kt} x_{st} \tag{7}$$

where α^i_{kt} is the coefficient.

The fire strength of the kth fuzzy rule in the ith fuzzy subsystem is

$$\tau^i_{sk} = \prod_{t=1}^{M} \mu^i_{kt}(x_{st}) \tag{8}$$

and we denote the weighted fire strength for each fuzzy rule as

$$\omega^i_{sk} = \frac{\tau^i_{sk}}{\sum_{k=1}^{K_i} \tau^i_{sk}}. \tag{9}$$

The Gaussian membership function is chosen for μ^i_{kt} corresponding to fuzzy set A^i_{kt} which is defined as follows:

$$\mu^i_{kt}(x) = e^{-\left(\frac{x - c^i_{kt}}{\sigma^i_{kt}}\right)^2} \tag{10}$$

where c^i_{kt} and σ^i_{kt} are, respectively, width and center.

In most neural-fuzzy models, the parameters of membership functions and the coefficients in the consequents have to be tuned during the learning process by BP or some other iterative algorithms. Unfortunately, the learning phase usually becomes very time-consuming when the training datasets are very large or the dimension of input variable is very high. Moreover, high-dimensional input variables will also bring along the explosion of fuzzy rules. Although some recent neuro-fuzzy

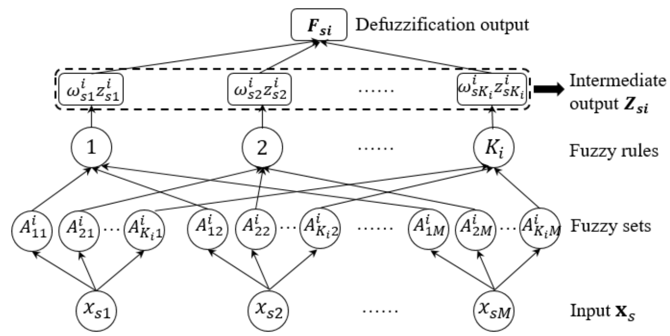

Fig. 4. Structure of the ith fuzzy subsystem in a fuzzy BLS.

models [34]–[36] have been proposed to deal with the issue of dimensionality, they are almost applied to some problems with relatively small number of input features, such as system identification and control.

To solve these problems, three strategies are employed in our proposed fuzzy BLS.

1) The coefficients α^i_{kt} are initialized by random numbers from uniform distribution of [0, 1] and then they are determined by pseudoinverse which will be discussed subsequently.
2) We set $\sigma^i_{kt} = 1$ for all fuzzy subsystems.
3) The k-means algorithm is applied to the training set to select K_i clustering centers for the ith fuzzy subsystem, and the centers c^i_{kt} of the Gaussian membership functions are initialized by these K_i clustering centers. Meanwhile, the number of fuzzy rules in the ith fuzzy subsystem is also determined by the number of centers K_i. Due to the randomness of initial conditions in k-means algorithm, different centers will be chosen from the training set in each fuzzy subsystem, thus the proposed fuzzy BLS can benefit from this ensemble structure consisting of a group of fuzzy subsystems.

A. Inputs for Enhancement Layer

In order to retain the information lying behind the input data as much as possible, we define a vector consisting of the outputs of all fuzzy rules in the ith fuzzy subsystem before aggregating them into one value as the fuzzy subsystem's defuzzification output. The intermediate vectors of all fuzzy subsystems are then fed into the layer of enhancement nodes for further nonlinear transformation. The structure of the ith fuzzy subsystem is displayed in Fig. 4. The output vector of the ith fuzzy subsystem for the sth training sample $\boldsymbol{x}_s$ without aggregation is denoted by

$$\boldsymbol{Z}_{si} = \left(\omega_{s1}^i z_{s1}^i, \omega_{s2}^i z_{s2}^i, \ldots, \omega_{sK_i}^i z_{sK_i}^i\right) \tag{11}$$

and the output matrix of the ith fuzzy subsystem for all the training samples $\boldsymbol{X}$ is

$$\boldsymbol{Z}_i = (\boldsymbol{Z}_{1i}, \boldsymbol{Z}_{2i}, \ldots, \boldsymbol{Z}_{Ni})^T \in \mathbb{R}^{N\times K_i}, i = 1, 2, \ldots, n. \tag{12}$$

To keep the consistence of notation, we denote the intermediate output matrix of n fuzzy subsystems by

$$\boldsymbol{Z}^n = (\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n) \in \mathbb{R}^{N\times(K_1+K_2+\cdots+K_n)}. \tag{13}$$

Then $\boldsymbol{Z}^n$ will be sent into the enhancement nodes for nonlinear transformation. Suppose that there are L_j neurons in the jth enhancement node group, and we represent the output matrix of enhancement layer by

$$\boldsymbol{H}^m = (\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m) \in \mathbb{R}^{N\times(L_1+L_2+\cdots+L_m)} \tag{14}$$

where $\boldsymbol{H}_j = \xi_j(\boldsymbol{Z}^n \boldsymbol{W}_{h_j} + \boldsymbol{\beta}_{h_j}) \in \mathbb{R}^{N\times L_j}$ is the output matrix of the jth enhancement node group, $\boldsymbol{W}_{h_j}$ and $\boldsymbol{\beta}_{h_j}$ are weights and bias terms connecting the outputs $\boldsymbol{Z}^n$ of fuzzy subsystems to corresponding enhancement nodes which are randomly generated from [0, 1] ($j = 1, 2, \ldots, m$).

This nonlinear transformation could make full use of the rule outputs $\boldsymbol{Z}^n$ rather than just aggregate them into one value by linear combination, which can be considered as a complementary part to the first-order polynomial adopted in consequent part.

B. Outputs of Fuzzy Subsystems

Now we calculate the defuzzification output of every fuzzy subsystem which will be sent into the top layer together with the output matrix $\boldsymbol{H}^m$ of enhancement layer. Since the training target $\boldsymbol{Y} \in \mathbb{R}^{N\times C}$ has C components, each fuzzy subsystem should be a multioutput model. The output vector of the ith fuzzy subsystem for training sample $\boldsymbol{x}_s$ is defined as

$$\begin{aligned}\boldsymbol{F}_{si} &= \left(\sum_{k=1}^{K_i}\omega_{sk}^i\left(\sum_{t=1}^{M}\delta_{k1}^i\alpha_{kt}^i x_{st}\right), \ldots, \sum_{k=1}^{K_i}\omega_{sk}^i\left(\sum_{t=1}^{M}\delta_{kC}^i\alpha_{kt}^i x_{st}\right)\right)\\ &= \sum_{t=1}^{M}\alpha_{kt}^i x_{st}\left(\omega_{s1}^i, \ldots, \omega_{sK_i}^i\right)\begin{pmatrix}\delta_{11}^i & \cdots & \delta_{1C}^i\\ \vdots & & \vdots\\ \delta_{K_i1}^i & \cdots & \delta_{K_iC}^i\end{pmatrix}\end{aligned} \tag{15}$$

where we introduce the parameters δ_{kc}^i to the consequent part of each fuzzy rule in the ith fuzzy subsystem, then the initial coefficient α_{kt}^i is changed into $\delta_{kc}^i\alpha_{kt}^i$ ($c = 1, 2, \ldots, C$).

Remark 1: The reason why we do not calculate the value of α_{kt}^i directly is to reduce the number of parameters. The total number of α_{kt}^i is $M\sum_{i=1}^n K_i$, while the total number of δ_{kc}^i is $C\sum_{i=1}^n K_i$. Because the output dimension C is always much smaller than the input dimension M in practice, it is faster and easier to calculate δ_{kc}^i than α_{kt}^i by pseudoinverse. Moreover, the final coefficients $\delta_{kc}^i\alpha_{kt}^i$ in f_k^i will be different from each other due to the random initial values of α_{kt}^i. Once we calculate the value of δ_{kc}^i by pseudoinverse, the coefficients in consequent will be determined accordingly.

Algorithm 1 Training a Fuzzy BLS

Input: Training samples $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathbb{R}^{N\times(M+C)}$, numbers of fuzzy rules K_i, enhancement nodes L_j, fuzzy subsystems n and enhancement node groups m.
Output: A Fuzzy BLS with parameter matrix $\boldsymbol{W}$.

initialize the coefficients α_{kt}^i in function f_k^i by uniform distribution in [0, 1].
for $i = 1$ to n **do**
  apply k-means algorithm to training samples $\boldsymbol{X}$ to obtain K_i clustering centers.
  initialize the centers of Gaussian membership functions by the values of K_i clustering centers.
  for $s = 1$ to N **do**
    calculate $\boldsymbol{Z}_{si}$ according to Eq. (11);
    calculate $\boldsymbol{F}_{si}$ according to Eq. (15);
  end for
  obtain $\boldsymbol{Z}_i$ according to Eq. (12);
  calculate $\boldsymbol{F}_i$ according to Eq. (16);
end for
obtain $\boldsymbol{Z}^n$ according to Eq. (13);
calculate $\boldsymbol{H}^m$ according to Eq. (14);
calculate $\boldsymbol{F}^n$ according to Eq. (17);
calculate $\boldsymbol{W}$according to Eq. (19).

Then the output matrix of the ith fuzzy subsystem for all the training samples $\boldsymbol{X}$ is

$$\boldsymbol{F}_i = (\boldsymbol{F}_{1i}, \boldsymbol{F}_{2i}, \ldots, \boldsymbol{F}_{Ni})^T \triangleq \boldsymbol{D}\Omega^i\boldsymbol{\delta}^i \in \mathbb{R}^{N\times C} \tag{16}$$

where $\boldsymbol{D} = diag\{\sum_{t=1}^M \alpha_{kt}^i x_{1t}, \ldots, \sum_{t=1}^M \alpha_{kt}^i x_{Nt}\}$, and

$$\Omega^i = \begin{pmatrix}\omega_{11}^i & \cdots & \omega_{1K_i}^i\\ \vdots & & \vdots\\ \omega_{N1}^i & \cdots & \omega_{NK_i}^i\end{pmatrix}, \quad \boldsymbol{\delta}^i = \begin{pmatrix}\delta_{11}^i & \cdots & \delta_{1C}^i\\ \vdots & & \vdots\\ \delta_{K_i1}^i & \cdots & \delta_{K_iC}^i\end{pmatrix}.$$

Then we can obtain the aggregative output of n fuzzy subsystems for the top layer, which is

$$\begin{aligned}\boldsymbol{F}^n &= \sum_{i=1}^n \boldsymbol{F}_i = \sum_{i=1}^n \boldsymbol{D}\Omega^i\boldsymbol{\delta}^i = \boldsymbol{D}\left(\Omega^1, \ldots, \Omega^n\right)\begin{pmatrix}\boldsymbol{\delta}^1\\ \vdots\\ \boldsymbol{\delta}^n\end{pmatrix}\\ &\triangleq \boldsymbol{D\Omega\Delta} \in \mathbb{R}^{N\times C}\end{aligned} \tag{17}$$

where $\boldsymbol{\Omega} = (\Omega^1, \ldots, \Omega^n) \in \mathbb{R}^{N\times(K_1+K_2+\ldots+K_n)}$ is the matrix consisting of fire strength ω_{sk}^i, and $\boldsymbol{\Delta} = ((\boldsymbol{\delta}^1)^T, \ldots, (\boldsymbol{\delta}^n)^T)^T \in \mathbb{R}^{(K_1+K_2+\ldots+K_n)\times C}$ consists of the parameters to be calculated later.

C. Inputs for the Top Layer

Now we send the output $\boldsymbol{F}^n$ of all the fuzzy subsystem together with $\boldsymbol{H}^m$ to the top layer of Fuzzy BLS. The weight matrix connecting the enhancement layer to top layer is denoted as $\boldsymbol{W}_e \in \mathbb{R}^{(L_1+L_2+\ldots+L_m)\times C}$, while the weights connecting the fuzzy subsystems to top layer are all set to be 1. Therefore, the final output of a Fuzzy BLS is

$$\begin{aligned}\hat{\boldsymbol{Y}} &= \boldsymbol{F}^n + \boldsymbol{H}^m\boldsymbol{W}_e\\ &= \boldsymbol{D\Omega\Delta} + \boldsymbol{H}^m\boldsymbol{W}_e\\ &= \left(\boldsymbol{D\Omega}, \boldsymbol{H}^m\right)\begin{pmatrix}\boldsymbol{\Delta}\\ \boldsymbol{W}_e\end{pmatrix}\\ &\triangleq \left(\boldsymbol{D\Omega}, \boldsymbol{H}^m\right)\boldsymbol{W}\end{aligned} \tag{18}$$

where $\boldsymbol{W}$ is the parameter matrix of a fuzzy BLS consisting of $\boldsymbol{\Delta}$ and $\boldsymbol{W_e}$.

Given the training targets $\boldsymbol{Y}$, matrix $\boldsymbol{W}$ can be calculated rapidly by pseudoinverse, i.e.,

$$\boldsymbol{W} = \left(\boldsymbol{D\Omega}, \boldsymbol{H}^m\right)^+ \boldsymbol{Y} \tag{19}$$

where

$$\left(\boldsymbol{D\Omega}, \boldsymbol{H}^m\right)^+ = \left(\left(\boldsymbol{D\Omega}, \boldsymbol{H}^m\right)^T \left(\boldsymbol{D\Omega}, \boldsymbol{H}^m\right)\right)^{-1} \left(\boldsymbol{D\Omega}, \boldsymbol{H}^m\right)^T.$$

The training algorithm of fuzzy BLS is summarized in Algorithm 1.

Remark 2: The computational complexity of fuzzy BLS can be easily deduced since there are mainly four parts: 1) k-means; 2) fuzzy subsystems; 3) enhancement layer; and 4) pseudoinverse. Therefore, the time complexity of fuzzy BLS is

$$O\left(NM\sum_{i=1}^{n} K_i^2 + N\sum_{i=1}^{n} K_i \sum_{j=1}^{m} L_j + \left(\sum_{i=1}^{n} K_i + \sum_{j=1}^{m} L_j\right)^3\right).$$

IV. Performance Evaluation of Fuzzy BLS

We employ different benchmarks to compare fuzzy BLS with BLS and other representative nonfuzzy models, as well as some state-of-the-art neuro-fuzzy models. For simplicity, each fuzzy subsystem of fuzzy BLS is set to have the same number of fuzzy rules. The results of other models are cited from the references directly if there is no special explanation.

A. Nonlinear System Identification

The nonlinear dynamic system used here is described as follows [37]:

$$y(n) = \frac{y(n-1)y(n-2)(y(n-1)+2.5)}{1+y^2(n-1)+y^2(n-2)} + u(n-1). \tag{20}$$

The training input $u(n)$ is uniformly produced from the range of $[-2, 2]$. We generate two training sets: one consists of 1000 observation data, and the other one following the way of [22] has 5000 observations with some noises uniformly distributing in $[-0.2, 0.2]$. And 200 testing data are generated by $u(n) = \sin(2\pi n/25)$.

The system inputs are $(y(n-2), y(n-1)$, and $u(n-1))$ and the corresponding output is $y(n)$. We first use 5000 noisy training data to train fuzzy BLS and BLS, and compare their performance with ANFIS [38], SAFIS [39], eTS [40], Simpl_eTS [41], DENFIS [42], and OS-F-ELM [22]. The best results are listed in Table I. Then the 1000 clean training data are used for training ANFIS, BLS, and fuzzy BLS, and the comparison results are reported in Table II. The parameters of BLS consist of the numbers of feature nodes N_f, mapping groups N_m, and enhancement nodes N_e. The parameters of fuzzy BLS are the numbers of rules N_r in each fuzzy subsystem, fuzzy subsystems N_t, and enhancement nodes N_e.

We can see from Tables I and II that fuzzy BLS dramatically reduces the testing RMSE. It obtains the smallest test errors on both data sets, and its performance has not been influenced too much by the noisy training data. Besides, when trained by 1000 observations, the total time consumed by ANFIS, BLS, and fuzzy BLS implies that fuzzy BLS can achieve much higher accuracy in less time than traditional fuzzy models. Moreover, there are the fewest fuzzy rules in each fuzzy subsystem of fuzzy BLS among the fuzzy models involved.

The outputs of ANFIS, BLS, and fuzzy BLS for the 200 testing data are also depicted in Fig. 5. Apparently, Fig. 5(b) reveals that fuzzy BLS can better approximate the system outputs, especially at some turning points of the curve than ANFIS and BLS.

B. Regression

We select 13 regression data sets from the UCI repository [43], which fall into three categories: 1) small size and low dimensions; 2) medium size and dimensions; and 3) large size and low dimensions. The details of the data sets are listed in Table III. These data sets will be employed to compare fuzzy BLS with some popular nonfuzzy models and neuro-fuzzy models.

TABLE I
Performance Comparison for Nonlinear System Identification (5000 Noisy Training Data)

Model	Parameter Settings	RMSE		Time (s)
		Training	Testing	
ANFIS	$\#Rules = 27$	0.1264	0.0479	26.89
DENFIS	$\#Rules = 276$	0.2246	0.1204	14.01
SAFIS	$\#Rules = 30$	0.1493	0.0533	3.836
eTS	$\#Rules = 31$	0.1620	0.0638	4.992
Simpl_eTS	$\#Rules = 42$	0.3305	0.1169	12.92
OS-F-ELM	$\#Rules = 30$	0.1217	0.0402	2.602
BLS	$N_f = 15, N_m = 8, N_e = 375$	0.1560	0.0264	<u>0.074</u>
Fuzzy BLS	$N_r = 24, N_t = 8, N_e = 90$	0.1587	**0.0243**	1.857

TABLE II
Performance Comparison for Nonlinear System Identification (1000 Training Data)

Model	Parameter Settings	RMSE		Time (s)
		Training	Testing	
ANFIS	$\#Rules = 27, \#epochs = 100$	0.0597	0.0473	46.67
BLS	$N_f = 13, N_m = 8, N_e = 210$	0.0242	0.0251	<u>0.056</u>
Fuzzy BLS	$N_r = 21, N_t = 9, N_e = 77$	0.0301	**0.0229**	0.765

TABLE III
Details of Data Sets for Regression

Datasets	No. of samples		Attributes
	Training	Testing	
Abalone	2784	1393	8
Basketball	64	32	4
Cleveland	202	101	13
Pyrim	49	25	27
Strike	416	209	6
Mortgage	524	210	15
Weather Izmir	730	293	9
California Housing	10,320	10,320	9
Auto MPG	196	196	6
Census (House8L)	10,000	12,784	8
2D Planes	10,000	30,768	10
Bank	4,500	3,692	8
Kinematics of Robot Arm	4,000	4,192	8

TABLE IV
PARAMETER SETTINGS OF SVM, LSSVM, ELM, BLS, AND FUZZY BLS FOR REGRESSION DATA SETS

Data set	SVM		LSSVM		ELM		BLS			Fuzzy BLS		
	C	γ	C	γ	C	γ	N_f	N_m	N_e	N_r	N_t	N_e
Abalone	2^2	2^{-1}	2.8932	3.0774	2^0	2^0	5	6	41	4	6	37
Basketball	2^0	2^0	6.0001	27.3089	2^{25}	2^{11}	6	7	4	5	1	26
Cleveland	2^2	2^2	0.7527	45.2507	2^{13}	2^{15}	10	1	3	16	1	44
Pyrim	2^{10}	2^8	52.5877	3.2463	2^2	2^6	2	6	26	1	7	48
Strike	2^0	2^{-4}	0.3167	0.7383	2^{-1}	2^5	9	11	30	3	2	23

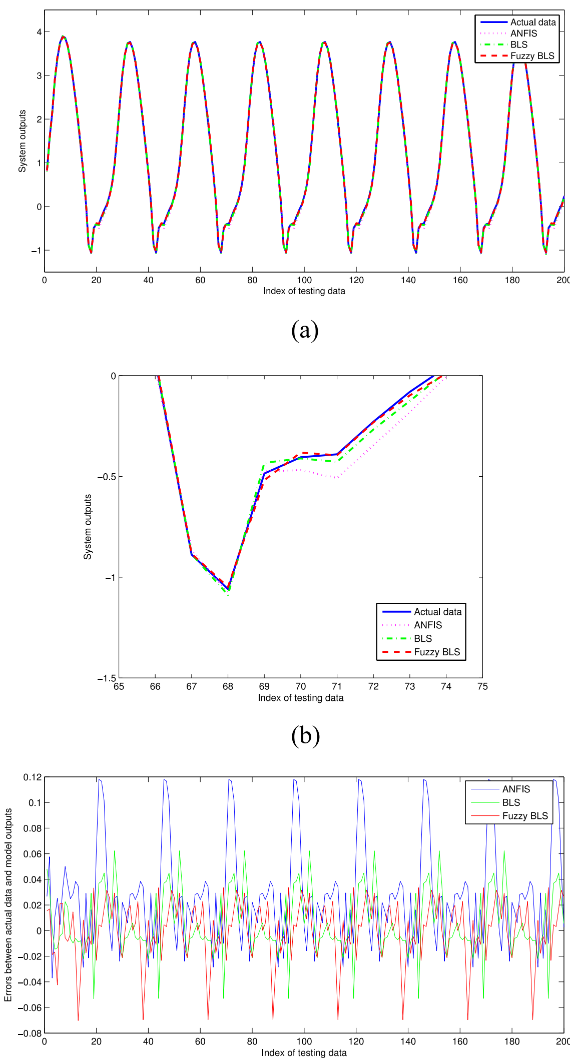

Fig. 5. Comparison of model outputs for 200 testing data. (a) Overall comparison. (b) Local comparison. (c) Errors for each testing samples.

1) Comparison With Nonfuzzy Approaches: We use the first five data sets in Table III to compare the performance of fuzzy BLS with SVM, LSSVM [44], ELM [45], and BLS. The cost parameter C and kernel parameter γ of SVM, LSSVM, and ELM play an important role in learning a good regression model, hence they have to be chosen appropriately for a fair comparison. In this paper, we carry out a grid search for the parameters (C, γ) from $\{2^{-24}, 2^{-23}, \ldots, 2^{24}, 2^{25}\}$ to determine the optimal settings for SVM (using *libsvm* [46]) and ELM, while the optimal values of (C, γ) for LSSVM are decided by itself using *LS-SVMlab* Toolbox. We also perform a grid search for the parameters of BLS from $[1, 10] \times [1, 30] \times [1, 200]$, and for the parameters of fuzzy BLS from $[1, 20] \times [1, 10] \times [1, 100]$, respectively. The parameter settings are shown in Table IV. We carry out 50 trials to get the statistical results.

The results of SVM, LSSVM, ELM, BLS, and fuzzy BLS are given in Table V. In the five regression benchmarks, fuzzy BLS can almost obtain the smallest testing errors in relatively short running time, which again demonstrates the advantage of the proposed fuzzy BLS over these classical models. Meanwhile, small number of fuzzy rules is needed in fuzzy BLS which can ease the rule explosion problem in fuzzy systems.

The Friedman test [47] is recommended for statistical comparison of more than two algorithms over multiple data sets which checks whether the measured average ranks are significantly different. We can easily obtain that $\chi_F^2 = 16.16$ and $F_F = 16.83$. With five algorithms and five data sets, F_F is distributed according to the F distribution with 4 and 16 degrees-of-freedom. The critical value of $F(4, 16)$ for $\alpha = 0.05$ is $3.01 < F_F = 16.83$, so we can conclude that the ranks of these models are significantly different. Meanwhile, we employ the one-way repeated ANOVA measure to check whether the mean of each algorithm differs significantly from the aggregate mean across all conditions. The F-score for the ANOVA test is $F_A = 3.25 > 3.01$, which also implies the five algorithms perform differently on different data sets.

2) Comparison With Neuro-Fuzzy Approaches: The Mortgage and Weather Izmir data sets are adopted to compare the performance of fuzzy BLS with MEA-FIS [48], MGA-FIS [49], and recently proposed FWPNN [23]. We follow the method in [23] to prepare the training and testing data. The parameters settings are listed in Table VI, and the comparison results of the above models are illustrated in Table VII.

We can also observe that fuzzy BLS obtains the best training and testing accuracies on both data sets. It shows great advantage over the newly proposed FWPNN. Also fuzzy BLS can further improve the performance of BLS on this benchmark.

We then use the last six datasets in Table III which are also adopted in [22] to compare some other neuro-fuzzy systems including OS-F-ELM, ANFIS, and Simpl_eTS with our fuzzy BLS. Table VIII summarizes the comparison results and parameter settings of fuzzy BLS. It is clear that the

TABLE V
PERFORMANCE COMPARISON (TESTING RMSE) OF SVM, LSSVM, ELM, BLS, AND FUZZY BLS FOR REGRESSION DATA SETS

Datasets	SVM		LSSVM		ELM		BLS		Fuzzy BLS	
	Aver±Std	Time (s)	Aver±Std	Time (s)	Aver±Std	Time (s)	Aver±Std	Time (s)	Aver±Std	Time (s)
Abalone	0.0757±0.0017	0.784	0.0748±0.0014	382.5	0.0754±0.0012	1.203	0.0746±0.0011	0.035	**0.0745±0.0011**	0.433
Basketball	0.0831±0.0055	0.020	0.0815±0.0067	0.484	0.0824±0.0064	0.024	0.0810±0.0069	0.027	**0.0808±0.0052**	0.009
Cleveland	0.1252±0.0058	0.026	0.1169±0.0060	2.273	0.1165±0.0118	0.029	**0.1100±0.0045**	0.005	0.1112±0.0041	0.016
Pyrim	0.1069±0.0080	0.021	0.0887±0.0077	0.421	0.0824±0.0081	0.026	0.0929±0.0117	0.024	**0.0767±0.0075**	0.042
Strike	0.0736±0.0142	0.038	0.0725±0.0096	3.577	0.0713±0.0169	0.036	0.0682±0.0132	0.047	**0.0665±0.0103**	0.034

TABLE VI
PARAMETER SETTINGS OF BLS AND FUZZY BLS FOR MORTGAGE AND WEATHER IZMIR DATA SETS

Data set	BLS			Fuzzy BLS		
	N_f	N_m	N_e	N_r	N_t	N_e
Mortgage	9	4	135	17	6	40
Weather Izmir	4	3	87	7	8	6

TABLE VII
PERFORMANCE COMPARISON (MSE/2) FOR MORTGAGE AND WEATHER IZMIR DATA SETS

Models	Mortgage		Weather Izmir	
	Training	Testing	Training	Testing
MEA-FIS	0.06±0.03	0.08±0.05	1.30±0.27	1.49±0.26
MGA-FIS	0.016±0.0002	0.022±0.0005	0.926±0.041	1.150±0.123
FWPNN	0.006±0.001	0.009±0.001	0.667±0.052	0.855±0.133
BLS	0.0031±0.0002	0.0038±0.0004	0.7691±0.0263	0.6990±0.0664
Fuzzy BLS	**0.0011±0.0001**	**0.0030±0.0002**	**0.6138±0.0233**	**0.6698±0.0394**

proposed fuzzy BLS behaves better than other models in the aspect of both accuracy and training time.

We also perform the Friedman test to check whether the measured average ranks are significantly different. We can obtain that $\chi_F^2 = 16.99$ and $F_F = 84.98$. With four algorithms and six data sets, F_F is distributed according to the F distribution with 3 and 15 degrees-of-freedom. The critical value of $F(3, 15)$ for $\alpha = 0.05$ is $3.29 < F_F$, so we can conclude that the performance of these models are significantly different. Since there is no result on 2-D planes data set for Simpl_eTS, we perform the repeated ANOVA test on the rest five data sets for the four models. The F-score for the ANOVA test is $F_A = 5.28$ which is greater than the critical value $F(3, 12)_{0.05} = 3.49$, thus we can reject the equality of mean hypothesis.

C. Classification

In this section, we evaluate the performance of fuzzy BLS through five benchmarks [43] for binary and multiclass classification, and compare it with EGART-FIS [50], MFMM-FIS [51], F-ELM [26], IF-ELM [27], and BLS. The details of the data sets are listed in Table IX.

1) Binary Datasets: The Pima Indians Diabetes (PID) and Breast Cancer Wisconsin (WBC) data sets consist of medical data. The samples of PID data set fall into two classes: 1) patients diagnosed with diabetic and 2) healthy people. The samples of WBC data set are also categorized into two classes: 1) patients who shows benign or 2) are diagnosed as malignant.

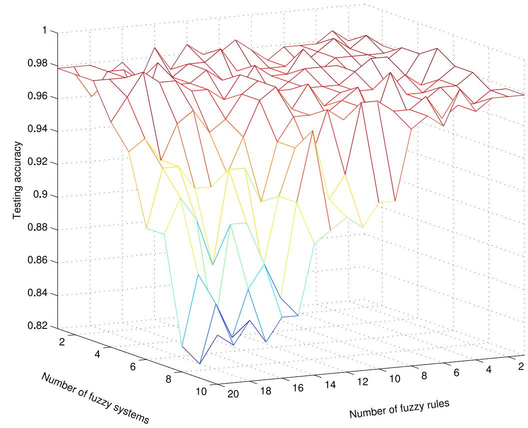

Fig. 6. Testing accuracy of fuzzy BLS with different N_r and N_t on WBC data set.

The training sets and testing sets are constructed according to [26]. The parameters of BLS are $N_f = 4, N_m = 2, N_e = 3$, and of fuzzy BLS are $N_r = 7, N_t = 1, N_e = 53$, respectively, when trained with PID data set. The values of parameters of BLS and fuzzy BLS are $N_f = 2$, $N_m = 5$, $N_e = 1$ and $N_r = 2$, $N_t = 6, N_e = 20$ for WBC data set. The experimental results are illustrated in Table X.

We can observe that the classification accuracies of the two benchmarks have been remarkably increased using fuzzy BLS by more than 4 percents compared to F-ELM. And there are almost 2 percents of improvement in accuracy from BLS to fuzzy BLS. Moreover, 9 and 6 fuzzy rules are used in F-ELM for PID and WBC data sets, but the fuzzy BLS only employ 7 and 2 rules in each fuzzy subsystem and obtains much better results.

In addition, we fix the number of enhancement nodes in fuzzy BLS as 20 and evaluate the testing accuracy on WBC data set for different numbers of fuzzy rules and subsystems in Fig. 6. It shows that when the dimension of attributes and the number of classes are small, fuzzy BLS can obtain satisfactory classification accuracy with a few fuzzy rules and fuzzy subsystems which is consistent with our intuition.

2) Multiclass Data Sets: The Image Segmentation, Statlog, and DNA data sets have more categories for classification. The samples of Image Segmentation data set are randomly drawn from seven outdoor images. The objective is to classify each region which has the size of 3×3 pixels into one of the seven classes: 1) brick facing; 2) sky; 3) foliage; 4) cement; 5) window; 6) path; and 7) grass. The 18 attributes are extracted from each square region.

TABLE VIII
PERFORMANCE COMPARISON (TESTING RMSE) OF OS-FUZZY-ELM, ANFIS, SIMPL_ETS AND FUZZY BLS FOR REGRESSION

Datasets	OS-Fuzzy-ELM		ANFIS		Simpl_eTS		Fuzzy BLS				
	Aver±Std	Time (s)	Aver±Std	Time (s)	Aver±Std	Time (s)	Aver±Std	Time (s)	N_r	N_t	N_e
Cal-Housing	0.1320±0.0015	3.751	0.1380±0.0017	914.3	0.1616±0.0046	34.94	**0.1283±0.0046**	4.562	50	4	20
Auto MPG	0.0765±0.0075	0.049	0.0803±0.0079	0.492	0.0806±0.0086	0.198	**0.0739±0.0030**	0.013	7	1	66
Census	0.0661±0.0019	8.946	0.0667±0.0027	191.5	0.0814±0.0030	106.5	**0.0655±0.0006**	5.553	65	2	20
2D Planes	0.0413±0.0008	84.06	0.0476±0.0008	126.3	N/A	N/A	**0.0412±0.0001**	18.46	85	4	60
Bank	0.0390±0.0016	20.93	0.0394±0.0019	208.2	0.0512±0.0033	245.2	**0.0365±0.0003**	3.548	60	5	40
Kinematics	0.0853±0.0023	574.8	0.0823±0.0031	1557	0.1460±0.0055	1092	**0.0698±0.0011**	7.733	450	2	40

TABLE IX
DETAILS OF DATA SETS FOR CLASSIFICATION

Datasets	No. of samples		Attributes	Classes
	Training	Testing		
Pima Indians Diabetes	384	192	8	2
Breast Cancer Wisconsin	350	140	9	2
Image segmentation	1,500	810	18	7
Statlog	4,435	2,000	36	6
DNA	2,000	1,190	180	3

TABLE X
PERFORMANCE COMPARISON FOR PID AND WBC DATA SETS

Models	PID		WBC	
	Training	Testing	Training	Testing
F-ELM	75.35%	74.09%	93.94%	94.17%
IF-ELM	N/A	N/A	96.93%	97.41%
EGART-FIS	N/A	73.05%	N/A	93.56%
MFMM-FIS	N/A	72.92%	N/A	92.56%
BLS	78.39%	76.56%	96.57%	97.85%
Fuzzy BLS	82.03%	**78.65%**	96.57%	**99.29%**

TABLE XI
PARAMETER SETTINGS OF BLS AND FUZZY BLS FOR IMAGE SEGMENTATION, STATLOG, AND DNA DATA SETS

Data set	BLS			Fuzzy BLS		
	N_f	N_m	N_e	N_r	N_t	N_e
Image Segmentation	6	5	191	14	6	500
Statlog	18	11	194	49	10	170
DNA	10	20	40	19	16	60

The Statlog data set comprises satellite images generated from the Landsat multispectral scanner. There are four digital images for the same scene in four different spectral bands in one frame of the Landsat multispectral scanner imagery. The data set consists of 82 × 100 pixels, in which every sample corresponds to a region of 3 × 3 pixels. We have to categorize the central pixel of a region into six classes: 1) red soil; 2) cotton crop; 3) gray soil; 4) damp gray soil; 5) soil with vegetation stubble; and 6) very damp gray soil.

Every sample of DNA data set consists of 60 nucleotides, which falls into one of the there categories: 1) EI sites; 2) IE sites; and 3) neither.

The parameter settings for BLS and fuzzy BLS are summarized in Table XI, and the classification results are displayed in Table XII. We can note that fuzzy BLS achieves the highest testing accuracy on the three data sets. Fuzzy BLS only needs 14 fuzzy rules in each fuzzy subsystem and 84(=14 × 6) fuzzy rules in total for Image Segmentation data set, however, there are 330 fuzzy rules in F-ELM [26] which is far more than fuzzy BLS. On DNA data set, fuzzy BLS even outperforms

TABLE XII
PERFORMANCE COMPARISON FOR IMAGE SEGMENTATION, STATLOG, AND DNA DATA SETS

Models	Image segmentation		Statlog		DNA	
	Training	Testing	Training	Testing	Training	Testing
OS-F-ELM	95.84%	94.41%	92.84%	89.40%	96.51%	94.21%
F-ELM	98.71%	95.60%	93.20%	90.19%	96.85%	94.65%
IF-ELM	N/A	N/A	93.95%	89.22%	N/A	N/A
BLS	97.80%	96.30%	90.48%	89.15%	97.05%	**95.21%**
Fuzzy BLS	98.87%	**96.79%**	92.76%	**90.50%**	96.15%	**95.21%**

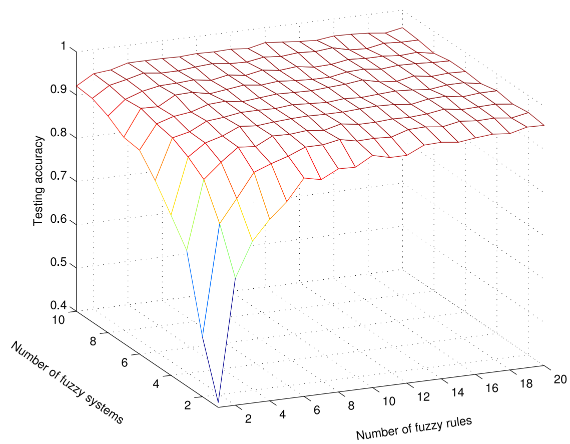

Fig. 7. Testing accuracy of fuzzy BLS with different N_r and N_t on Image segmentation data set.

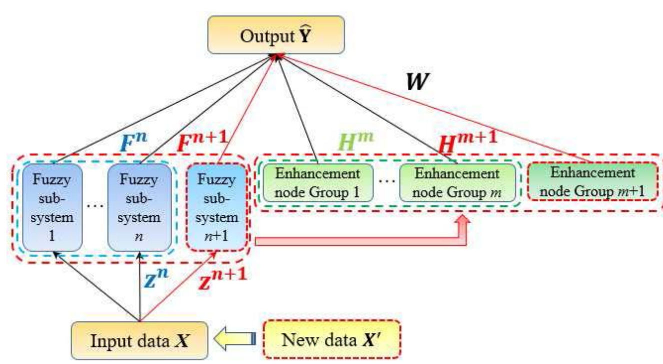

Fig. 8. Increments of input data, fuzzy subsystems, and enhancement nodes in a fuzzy BLS.

other three neuro-fuzzy models on testing accuracy with a slightly lower training accuracy, which demonstrates its good generalization capability.

We also evaluate the sensitivity of fuzzy BLS for the numbers of fuzzy rules and fuzzy subsystems on Image segmentation data set with 500 enhancement nodes, which is illustrated in Fig. 7. Due to the increase of training samples and attributes, fuzzy BLS needs more rules and subsystems to capture the characteristic of inputs. Nevertheless, we can see that its performance seems very stable in a large range of values of N_r and N_t.

V. Conclusion

By incorporating TS fuzzy systems into a BLS, a new neuro-fuzzy model named fuzzy BLS is proposed for regression and classification problems. To establish a fuzzy BLS, the feature nodes of a BLS are replaced by some fuzzy subsystems, and the outputs of every fuzzy subsystem are directly sent to the enhancement nodes without being aggregated first. Fuzzy BLS also completely gets rid of the tuning process of sparse autoencoder in BLS to reduce the structure complexity.

To overcome some limitations existing in traditional fuzzy systems including rule explosion and dimensional curse, k-means algorithm is employed to determine the number of fuzzy rules and the centers of Gaussian membership functions in antecedent part.

The parameters of fuzzy BLS that have to be computed during the training phase are the weights connecting the outputs of enhancement layer to the final output layer and the coefficients of first-order polynomials in consequent part of all rules in the fuzzy subsystems. And the calculation can be rapidly handled by ridge regression approximation of pseudoinverse in one step, which greatly reduces the learning time compared to other neuro-fuzzy models that adopt BP or other iterative training algorithms.

The performance of fuzzy BLS is evaluated and compared with both nonfuzzy and neuro-fuzzy approaches through some popular benchmarks for regression and classification. The experimental results reveal that fuzzy BLS can achieve higher accuracies in testing than the models involved. Fuzzy BLS needs fewer rules and less running time yet obtains much better results than the neuro-fuzzy models, which demonstrates its remarkable advantages.

The BLS can easily adapt to the increments of inputs, feature nodes, and enhancement nodes without retraining the whole network. Fortunately, fuzzy BLS retains the structure of BLS which implies that we can naturally generalize the incremental learning algorithms of BLS to the increments of fuzzy subsystems, as well as inputs and enhancement nodes (see Fig. 8). The online and incremental learning of a fuzzy BLS will be discussed in our future work, and we will also focus on how to improve its performance by integrating other techniques of computational intelligence.

Furthermore, we can consider of establishing some novel structures combining fuzzy systems and BLS which could be more interpretable and could better inherit the local modeling approach of fuzzy systems.

References

[1] L.-X. Wang and C. Wei, "Approximation accuracy of some neuro-fuzzy approaches," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 4, pp. 470–478, Aug. 2000.

[2] 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.

[3] W. A. Farag, V. H. Quintana, and G. Lambert-Torres, "A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 756–767, Sep. 1998.

[4] D. Kukolj and E. Levi, "Identification of complex systems based on neural and Takagi–Sugeno fuzzy model," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 272–282, Feb. 2004.

[5] J. S. Cervantes, W. Yu, S. Salazar-Cruz, and I. Chairez, "Takagi–Sugeno dynamic neuro-fuzzy controller of uncertain nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1601–1615, Dec. 2017.

[6] C.-W. Chang and C.-W. Tao, "A novel approach to implement Takagi–Sugeno fuzzy models," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2353–2361, Sep. 2017.

[7] Y. Li, L. Liu, and G. Feng, "Adaptive finite-time controller design for T–S fuzzy systems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2425–2436, Sep. 2017.

[8] D. Lee and J. Hu, "Local model predictive control for T–S fuzzy systems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2556–2567, Sep. 2017.

[9] K. K. Ang and C. Quek, "Stock trading using RSPOP: A novel rough set-based neuro-fuzzy approach," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1301–1315, Sep. 2006.

[10] A. Miranian and M. Abdollahzade, "Developing a local least-squares support vector machines-based neuro-fuzzy model for nonlinear and chaotic time series prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 2, pp. 207–218, Feb. 2013.

[11] V. S. Kumar, S. V. Chandra, and C. S. Kumar, "Neuro-fuzzy function approximations using feedforward networks—An application of sigmoidal signal," in *Proc. 9th Int. Conf. Mach. Learn. Appl.*, Washington, DC, USA, Dec. 2010, pp. 895–898.

[12] V. Sumati, C. Patvardhan, S. Paul, L. Singh, and V. M. Swarup, "Application of interval type-2 subsethood neural fuzzy inference system in control and function approximation," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2017, pp. 1–6.

[13] K. Cpalka, "A new method for design and reduction of neuro-fuzzy classification systems," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 701–714, Apr. 2009.

[14] C. L. P. Chen and Y. Lu, "FUZZ: A fuzzy-based concept formation system that integrates human categorization and numerical clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 27, no. 1, pp. 79–94, Feb. 1997.

[15] L. Chen, C. L. P. Chen, and M. Lu, "A multiple-kernel fuzzy C-means algorithm for image segmentation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 5, pp. 1263–1274, Oct. 2011.

[16] K. Subramanian, S. Suresh, and N. Sundararajan, "A metacognitive neuro-fuzzy inference system (MCFIS) for sequential classification problems," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 6, pp. 1080–1095, Dec. 2013.

[17] A. Segatori, A. Bechini, P. Ducange, and F. Marcelloni, "A distributed fuzzy associative classifier for big data," *IEEE Trans. Cybern.*, to be published.

[18] C. L. P. Chen, C.-Y. Zhang, L. Chen, and M. Gan, "Fuzzy restricted Boltzmann machine for the enhancement of deep learning," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 2163–2173, Dec. 2015.

[19] S. Feng and C. L. P. Chen, "A fuzzy restricted Boltzmann machine: Novel learning algorithms based on the crisp possibilistic mean value of fuzzy numbers," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 1, pp. 117–130, Feb. 2018.

[20] D. Wang, X.-J. Zeng, and J. A. Keane, "Hierarchical hybrid fuzzy-neural networks for approximation with mixed input variables," *Neurocomputing*, vol. 70, nos. 16–18, pp. 3019–3033, 2007.

[21] S. Feng, H. Li, and D. Hu, "A new training algorithm for HHFNN based on Gaussian membership function for approximation," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1631–1638, 2009.

[22] 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.

[23] W. Huang, S.-K. Oh, and W. Pedrycz, "Fuzzy wavelet polynomial neural networks: Analysis and design," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 5, pp. 1329–1341, Oct. 2017.

[24] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, "A hierarchical fused fuzzy deep neural network for data classification," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 4, pp. 1006–1012, Aug. 2017.

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

[26] S. Y. Wong, K. S. Yap, H. J. Yap, S. C. Tan, and S. W. Chang, "On equivalence of FIS and ELM for interpretable rule-based knowledge representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1417–1430, Jul. 2015.

[27] C.-S. Ouyang *et al.*, "An improved fuzzy extreme learning machine for classification and regression," in *Proc. Int. Conf. Cybern. Robot. Control (CRC)*, 2016, pp. 91–94.

[28] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[29] C. L. P. Chen and J. Z. Wan, "A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 1, pp. 62–72, Feb. 1999.

[30] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: Theory, system architecture, and functionalities," *Computer*, vol. 25, no. 5, pp. 76–79, May 1992.

[31] J. W. Jin, Z. L. Liu, and C. L. P. Chen, "Discriminative graph regularized broad learning system for image recognition," *SCIENCE CHINA Inf. Sci.*, to be published, doi: 10.1007/s11432-017-9421-3.

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

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

[34] C.-F. Juang, S.-H. Chiu, and S.-W. Chang, "A self-organizing T–S-type fuzzy network with support vector learning and its application to classification problems," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 5, pp. 998–1008, Oct. 2007.

[35] J. de Jesús Rubio, "SOFMLS: Online self-organizing fuzzy modified least-squares network," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 6, pp. 1296–1309, Dec. 2009.

[36] N. J. Cheung, X.-M. Ding, and H.-B. Shen, "OptiFel: A convergent heterogeneous particle swarm optimization algorithm for Takagi–Sugeno fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 4, pp. 919–933, Aug. 2014.

[37] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[38] 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.

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

[40] P. Angelov, J. Victor, A. Dourado, and D. Filev, "On-line evolution of Takagi–Sugeno fuzzy models," *IFAC Proc. Vol.*, vol. 37, no. 16, pp. 67–72, 2004.

[41] P. Angelov and D. Filev, "Simpl_eTs: A simplified method for learning evolving Takagi–Sugeno fuzzy models," in *Proc. 14th IEEE Int. Conf. Fuzzy Syst. (FUZZ)*, 2005, pp. 1068–1073.

[42] N. K. 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.

[43] C. L. Blake and C. J. Merz, *UCI Repository of Machine Learning Databases*, vol. 55, Dept. Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 1998. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html

[44] B. De Moor and J. Vandewalle, *Least Squares Support Vector Machines*. Singapore: World Sci., 2002.

[45] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[46] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[47] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.

[48] R. Alcalá, P. Ducange, F. Herrera, B. Lazzerini, and F. Marcelloni, "A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 5, pp. 1106–1122, Oct. 2009.

[49] R. Alcalá, M. J. Gacto, and F. Herrera, "A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 4, pp. 666–681, Aug. 2011.

[50] K. S. Yap, C. P. Lim, and J. Mohamad-Saleh, "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, 2010.

[51] 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, 2008.

Shuang Feng received the B.S. degree in mathematics and the M.S. degree in applied mathematics from Beijing Normal University, Beijing, China, in 2005 and 2008, respectively. He is currently pursuing the Ph.D. degree in computer science with the Faculty of Science and Technology, University of Macau, Macau, China.

He is currently an Associate Professor with the School of Applied Mathematics, Beijing Normal University (Zhuhai Campus), Zhuhai, China. His research interests include fuzzy systems and fuzzy neural networks, and their applications in computational intelligence.

C. L. Philip Chen (S'88–M'88–SM'94–F'07) received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1985 and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He was a tenured Professor, the Department Head, and the Associate Dean with two different universities in the U.S. for 23 years. He is currently a Chair Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China. The University of Macau's Engineering and Computer Science programs receiving Hong Kong Institute of Engineers' (HKIE) accreditation and Washington/Seoul Accord is his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty. His current research interests include systems, cybernetics, and computational intelligence.

Dr. Chen was a recipient of the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater at Purdue University. He has been the Editor-in-Chief of the IEEE TRANSACTION ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, since 2014 and an Associate Editor of several IEEE TRANSACTIONS. He was the Chair of TC 9.1 Economic and Business Systems of International Federation of Automatic Control from 2015 to 2017, and also a Program Evaluator of the Accreditation Board of Engineering and Technology Education of the U.S. for computer engineering, electrical engineering, and software engineering programs. He was the IEEE SMC Society President from 2012 to 2013 and the Vice President of the Chinese Association of Automation (CAA). He is a Fellow of AAAS, IAPR, CAA, and HKIE.