

# Robust Extreme Learning Fuzzy Systems Using Ridge Regression for Small and Noisy Datasets

Te Zhang<sup>1</sup>, Zhaohong Deng<sup>1,\*</sup>, Kup-Sze Choi<sup>2</sup>, Shitong Wang<sup>1</sup>

<sup>1</sup> School of Digital Media, Jiangnan University, Wuxi, Jiangsu, P.R. China

<sup>2</sup> Centre for Smart Health, Hong Kong Polytechnic University, Hong Kong

\* Corresponding author

**Abstract**—Fuzzy Extreme Learning Machine (F-ELM) constructs a fuzzy neural networks by embedding fuzzy membership functions and rules into the hidden layer of Extreme Learning Machine (ELM), that is, it can be interpreted as a fuzzy system with the structure of neural network. F-ELM generates fuzzy rules by randomly creating matrix-C (rule-Combination matrix), and matrix-DC (Don't Care matrix). The parameter optimization of F-ELM output layer is based on least square (LS), which is same as that used in the classical ELM. Although F-ELM has shown the characteristics of fast learning of model parameters, it has poor robustness to small and noisy datasets since its parameters connecting hidden layer with output layer are optimized by LS. In order to overcome this challenge, a Ridge Regression based Extreme Learning Fuzzy System (RR-EL-FS) is presented in this study, which has introduced the strategy of ridge regression into F-ELM to enhance the robustness. The experimental results also validate that the performance of RR-EL-FS is better than F-ELM and some related methods to small and noisy datasets.

**Index Terms**—Fuzzy system; Fuzzy extreme learning machine; Least square; Ridge regression; Robustness; Small and noisy dataset.

## I. INTRODUCTION

Fuzzy neural networks (FNNs) are a combination of fuzzy theory and neural networks, which have been extensively applied to learning, recognition, association, self-adaption and fuzzy information processing. For FNNs, the task of automatically extract knowledge for fuzzy system is implemented by adopting the learning technologies used in neural networks; Meanwhile, FNNs will be no longer deemed as a black box by due to the embedded fuzzy rules in it [1].

Extreme learning machine (ELM) is an effective learning algorithm for single-hidden-layer feedforward neural networks (SLFNs) which is presented by Huang etc. [2]. This technology randomly generates the hidden layer parameters of SLFNs and utilizes Least Square (LS) to calculate the weights connecting the hidden layer and the output layer. ELM has shown good performance in regression and classification [3,4], Which can overcomes many shortcomings in conventional neural network training techniques, such as the local minimum, iterations and slow training speed [5]. At present, ELM has attracted increasing attentions in related fields. Particularly, Fuzzy Extreme Learning Machine (F-ELM) which has integrated ELM with FNNs is proposed in [6]. In F-ELM, each hidden neuron corresponds to a fuzzy rule, and thus F-ELM can be regarded as a fuzzy rule based system that usually has the better interpretability.

In F-ELM, LS has been adopted for the parameter learning of consequents. This learning strategy make F-ELM face a challenge that the trained model by F-ELM lacks robustness to small and noisy datasets. When the size of a training dataset is small and the data in it are noisy, F-ELM is easy to make the trained model overfitting, resulting in the weak generalization abilities. This is because that the output matrix is usually singular or sick when facing small and noisy dataset by using LS to learn the

parameters involved [7]. For some practical applications, it is difficult to collect a large number of data. Meanwhile, the noise in data is unavoidable in some situations. Therefore, it is a significant work to study how to enhance the robustness of ELM based fuzzy system learning algorithm to small and noisy datasets.

This paper introduces the ridge regression (RR) strategy to enhance the robustness of ELM based fuzzy systems [8]. Ridge regression estimation is a modified LS estimation, which is the agonic estimation with least variance when facing large and low noise training dataset. However, the dataset acquired from real world may be small and has high noise, which causes the abnormality of the regressive coefficient for the parameter learning when using LS. As a modified LS estimation technique, RR has better robustness to small and noisy dataset. It can obtain more reliable regressive coefficient by abandoning the unbiasedness of LS, i.e., paying the price of reducing the training accuracy. By introducing RR, a novel algorithm, i.e., RR based extreme learning fuzzy system (RR-EL-FS) is proposed. Since the consequent parameters of fuzzy systems are evaluated by RR, RR-EL-FIS has overcome the weak robustness of F-ELM to small and noisy datasets to a great extent. The experimental studies also validate that the performance of RR-EL-FIS is better than F-ELM and some classical methods to small and noisy datasets.

The rest of this paper is organized as follows. In Section II, the related works are reviewed. In Section III, the ridge regression based extreme learning fuzzy systems learning method is presented. Experimental results are reported and discussed in Section IV. Finally, conclusions are given in section V.

## **II. REALATED WORK**

### ***A. Classical Takagi-Sugeno-Kang Fuzzy System***

Fuzzy systems are fuzzy-rule-based systems, in which knowledge base is composed of fuzzy if-then rules. The distinctive characteristic of fuzzy systems is that it introduces fuzzy set to simulate human reasoning and converts uncertain linguistic expertise into exact mathematical expression [9]. Classical fuzzy systems includes Takagi-Sugeno-Kang Fuzzy System (TSK-FS), Mamdani-Larsen Fuzzy System (ML-FS) and Generalized Fuzzy System (GFS) [10-12]. Among them, TSK-FS has been investigated extensively by many researchers because of its effectiveness and flexibility in practical applications [13-15]. In this study, we focus on 0-order TSK-FS, which just corresponds to the model trained by F-ELM.

In classical TSK-FS, the commonly used fuzzy if-then rules are defined as follows:

The  $k$ th fuzzy rule:

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge \dots \wedge x_d \text{ is } A_d^k, \\ &\text{THEN } y = f^k(x), k = 1, 2, \dots, K, \end{aligned} \quad (1)$$

where  $\mathbf{x} = (x_1 \ x_2 \ \dots \ x_d)^T$  is the input vector of system;  $A_i^k$  is a fuzzy subset subscribed by the input variable  $x_i$  for the  $k$ th rule;  $\wedge$  is a fuzzy conjunction operator, and  $K$  is the number of fuzzy rules. Each rule of TSK-FS is premised on the input vector  $\mathbf{x}$ , and maps the fuzzy sets in the input space  $A^k \subset R^d$  to a varying singleton denoted by  $f_k(\mathbf{x})$ . Let  $\mu^k(\mathbf{x})$  is the membership functions of  $A^k \subset R^d$  fuzzy set. Then  $\mu^k(\mathbf{x})$  can be calculated by the conjunction operation of membership of each dimension in *if part*, i.e. ,

$$\mu^k(\mathbf{x}) = \mu_1^k(x_1) \wedge \mu_2^k(x_2) \wedge \dots \wedge \mu_d^k(x_d). \quad (2)$$

When multiplicative conjunction is employed as the conjunction operator, along with multiplicative implication as the implication operator, and additive combination as the combination operator, the output of TSK-FS can be formulated as follows:

$$f(\mathbf{x}) = \sum_{k=1}^K \frac{\mu^k(\mathbf{x})}{\sum_{k'=1}^K \mu^{k'}(\mathbf{x})} \cdot f^k(\mathbf{x}) = \sum_{k=1}^K \frac{\prod_{j=1}^d \mu_j^k(x_j)}{\sum_{k'=1}^K \prod_{j=1}^d \mu_j^{k'}(x_j)} \cdot f^k(\mathbf{x}) \quad (3)$$

or

$$f(\mathbf{x}) = \sum_{k=1}^K \mu^k(\mathbf{x}) \cdot f^k(\mathbf{x}) = \sum_{k=1}^K \prod_{j=1}^d \mu_j^k(x_j) \cdot f^k(\mathbf{x}). \quad (4)$$

when TSK-FS is a 0-order TSK-FS,  $f^k(\mathbf{x}) = \beta_k$  is a constant.

## B. Fuzzy Extreme Learning Machine

F-ELM is an FNN learning algorithm, which introduces ELM learning technique to train a FNN that corresponds to a 0-order TSK-FS [6]. F-ELM generates fuzzy *if-then* rules by randomly assigning binary values to a 3-D rule-combination matrix C and a 2-D Don't care matrix DC, and optimizes the parameters connecting hidden layer with output layer (i.e., the consequent parameters of a 0-order TSK-FS) by LS. The details of F-ELM are described below.

### 1) Generation of Antecedent in F-ELM

In F-ELM, each dimension of input vector is divided into five fuzzy subset  $\mathbf{A}^i = (A_1^i, A_2^i, A_3^i, A_4^i, A_5^i)^T$  with Gaussian function as membership functions [16]. The center of five Gaussian membership functions are fixed to  $[0, 0.25, 0.5, 0.75, 1]$ , with linguistic labels denoted as very low, low, medium, high, very high [17]. Given an input vector  $\mathbf{x} = (x_1 \ x_2 \ \dots \ x_d)$ , the fuzzy values of all input attributes are calculated as follows:

$$\mu_i^k = \exp \left[ -\frac{(x_i - a_k)^2}{2\sigma^2} \right], \quad (5)$$

where  $k = 1, \dots, 5$ ,  $i = 1, \dots, d$ ,  $a_k \in \{0, 0.25, 0.5, 0.75, 1\}$ , and  $\mu_i^k$  is the fuzzy value of  $x_i$  to the  $k$ th fuzzy set. Standard deviation of Gaussian fuzzy membership function  $\sigma$  are assigned randomly in (5).

Similar to the randomness strategy exploited by ELM, F-ELM generate fuzzy rules by using rule-combination matrix-C and Don't care matrix-DC to decide which attribute and the associated membership function to be used in the antecedent of a rule. This can be done by randomly assigning binary values to rule-combination matrix C (with  $n$  attributes  $\times$  five membership functions  $\times L$  rules) and Don't care matrix DC (with  $n$  attributes  $\times L$  rules). For example, rule-combination matrix  $C(2, 3, 4) = 1$  indicates  $\mu_2^3$  in (5) is active for attribute 2 in rule-4 and  $D(2, 4) = 1$  indicates attribute-2 is don't care in rule-4. For an input vector  $x = (x_1, x_2)$ , if the matrix-C and matrix-DC of the  $k$ th rule are given as follows:

$$C(:, :, k) = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (6a)$$

$$D(:, k) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (6b)$$

then the  $k$ th fuzzy if-then rule can be expressed below.

$$R_k : \text{IF } x_1 \text{ is } A_2^k \text{ or } A_5^k \text{ and } x_2 \text{ is don't care, THEN } f_k = \beta_k, \quad (7)$$

where  $\beta_k$  is the consequent parameter of the  $k$ th rule in a 0-order TSK fuzzy system.

## 2) The Consequent Learning in F-ELM

The final output of F-ELM is calculated as follows:

$$f(x) = \sum_{l=1}^L w_l \beta_l, \quad (8)$$

where  $L$  is the number of hidden nodes, i.e., the number of rules;  $\beta_l$  is the

consequent parameter of the  $l$ th rule, and  $w_l$  is the firing strength of the  $l$ th rule which is calculated as follows:

$$w_l = \prod_{i=1}^d v_{il}, \quad (9)$$

$$v_{il} = \begin{cases} 1 & \text{if } D(i, l) = 1 \\ 1 - \prod_{k=1}^5 (1 - C(i, k, l) \mu_i^k) & \text{else } D(i, l) = 0 \end{cases}, \quad (10)$$

where  $\mu_i^k$  is the fuzzy value of  $x_i$  to the  $k$ th fuzzy set defined in (5).

Let

$$\mathbf{w} = (w_1 \ w_2 \ \dots \ w_l)^T, \quad (11)$$

$$\boldsymbol{\beta} = (\beta_1 \ \beta_2 \ \dots \ \beta_l)^T. \quad (12)$$

Then (8) can be written as

$$f(x) = \boldsymbol{\beta}^T \mathbf{w}. \quad (13)$$

Similar to the way in ELM, F-ELM uses LS to solve  $\boldsymbol{\beta}$ . Given a training dataset

$(\mathbf{x}_n, y_n) \in \mathbf{R}^d \times \mathbf{R}$  with  $N$  samples, combining firing strength of all training samples to construct a hidden layer output matrix, i.e.,

$$\mathbf{H} = \begin{bmatrix} w_{11} & \dots & w_{1L} \\ \vdots & \dots & \vdots \\ w_{N1} & \dots & w_{NL} \end{bmatrix}_{N \times L},$$

then the objective function of F-ELM used to solve  $\boldsymbol{\beta}$  is expressed below.

$$\min_{\boldsymbol{\beta}} J = \frac{1}{2} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|^2 \quad (14)$$

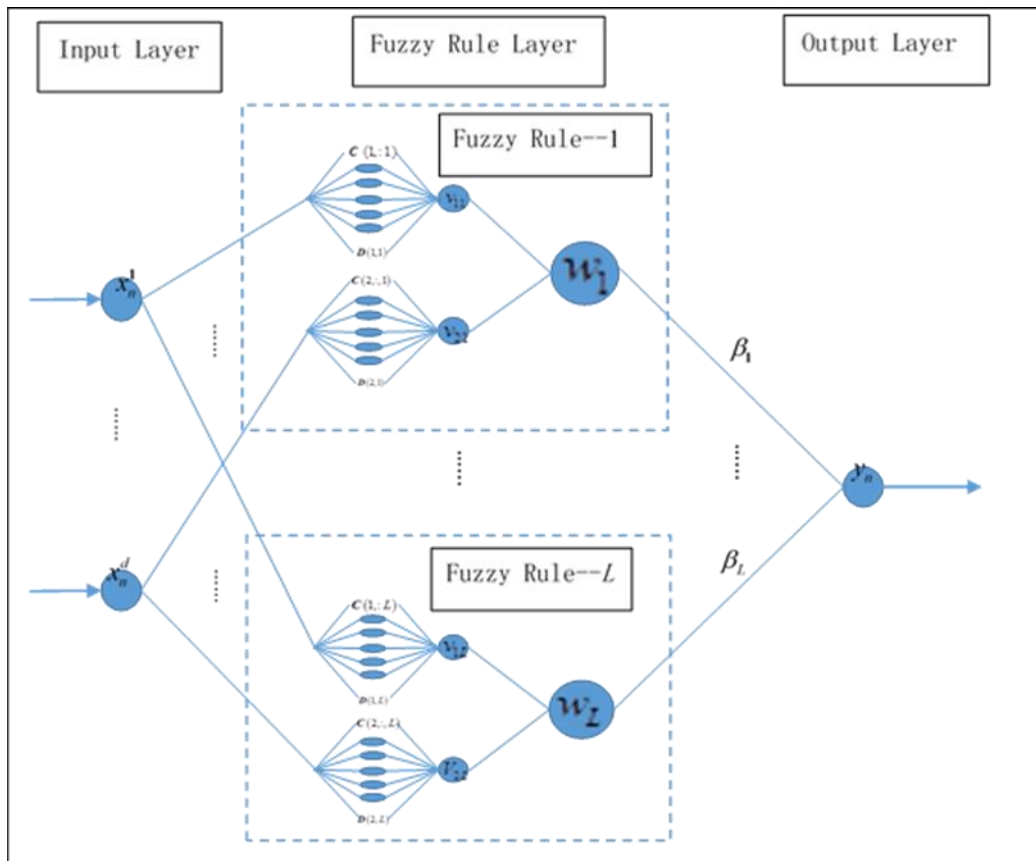
where  $\mathbf{T} = [y_1, y_2, \dots, y_N]^T$ . Furthermore, the output weights can be calculated based on (14):

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

### 3) Neural Networks Structure of F-ELM

F-ELM has represented a 0-order TSK-FS as a SLFN. The network structure is

shown in Fig. 1. The first layer of F-ELM is the input layer which corresponds to input vector directly. The function of this layer is to transmit the input vector  $\mathbf{x}_n = (x_1 \ x_2 \ \dots \ x_d)$  to hidden layer. The second layer is the fuzzy rule layer. This layer uses (5) to fuzzify the input vector, and complete the generation of the antecedents of fuzzy rule by randomly assigning binary values to matrix-C and matrix-DC. The Modified PROBOR[18] values for all attributes  $x_i$  of each rule, i.e.,  $v_{il}$ , will be calculated by (10) when the generation of fuzzy rule is finished. Then the firing strength of each rule, i.e.,  $w_l$ , is calculated by (9). The third layer is the output layer which calculates the final output by (8).



**Fig. 1. Network Structure of F-ELM**

### III. RIDGE REGRESSION BASED EXTREME LEARNING FUZZY SYSTEM

#### A. Challenge of F-ELM to Small and Noisy Datasets

F-ELM is capable of producing interpretable rules without loss of the performance in accuracy by integrating fuzzy system with ELM, and the implementation of DC (Don't Care) strategy is proven useful for eliminating those unnecessary input attributes in rules. The output weights of F-ELM are calculated by LS which is the agonic estimation with least variance when the training dataset has a large size and has low noise. However, there usually exist high noise in real world data and it is difficult to get sufficient data for model training in some situations. When the training dataset is small and noisy, the output weight calculated by LS will exist deviation, resulting in a weak generalization of the trained model. In order to overcome this shortcoming, RR will be introduced into the learning of F-ELM for output weights, i.e., the consequent parameters of a 0-order fuzzy system. By using RR estimation to learn the consequent parameters of fuzzy system, the shortcoming of F-ELM, i.e., the weak robustness to small and noisy dataset, can be overcome to a great extent.

#### B. Ridge Regression

Ridge regression, i.e., RR, is a modified LS estimation presented by American scholar A. E. Hoerl in 1962 [8]. In the past several decades, several RR methods have appeared [19-22]. According to [23], the classical ridge regression is described briefly below.

Given a multi-input single-output regression task with training dataset  $\mathbf{D}_{\text{reg}} = \{\mathbf{x}_n, y_n\}$ ,  $\mathbf{x}_n \in R^d$ ,  $y_n \in R$ ,  $n=1,2,\dots,N$ , the basic method of ridge regression is to obtain a linear regression model:

$$y = f(x) = \mathbf{x}^T \boldsymbol{\theta} \quad (16)$$

where the following loss function is adopted for parameter optimization:

$$\min_{\boldsymbol{\theta}} J = \frac{1}{2} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{Y}\|^2 + \frac{\lambda}{2} \|\boldsymbol{\theta}\|^2 \quad (17)$$

In (17)  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$  is a  $N \times d$  input matrix;  $\mathbf{Y} = [y_1, y_2, \dots, y_N]^T$ ;  $\lambda$  is the regularization parameters, which can be adjusted manually or determined with some strategies, such as cross validation. Based on (17), the solution of  $\boldsymbol{\theta}$  can be calculated as follows:

$$\boldsymbol{\theta} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}_d)^{-1} \mathbf{X}^T \mathbf{Y}, \quad (18)$$

where  $\mathbf{I}_d$  is a  $d \times d$  identity matrix.

RR has the better robustness to small and noisy dataset because it obtains more reliable regressive coefficients by abandoning the unbiasedness of LS. It is obvious that by abandoning the unbiasedness of LS and paying the price of reducing the training accuracy, the regressive coefficients obtained by RR is more reliable and the performance of the trained model is better than that trained based on LS to small and noisy datasets.

### ***C. Parameters Learning of Ridge Regression based Extreme Learning Fuzzy System***

In this subsection, RR-EL-FS is proposed. Firstly, RR-EL-FS uses rule-combination matrix-C and Don't care matrix-DC to construct the antecedents of fuzzy rules in a 0-order TSK-FS which is similar to F-ELM, and then utilize RR to optimize the consequents of fuzzy rules.

Given a multi-input single-output regression task with training dataset  $\mathbf{D}_{\text{reg}} = \{\mathbf{x}_i, y_i\}$ ,  $\mathbf{x}_i \in R^d$ ,  $y_i \in R$ ,  $i = 1, 2, \dots, N$ , the loss function of RR-EL-FS is as follows:

$$\min_{\beta} J = \frac{1}{2} \|W\beta - Y\|^2 + \frac{\lambda}{2} \|\beta\|^2, \quad (19)$$

where  $\beta$  is the consequence parameters of a 0-order TSK-FS defined in (12);

$Y = [y_1, y_2, \dots, y_N]^T$  and  $W = (w_1 \ w_2 \ \dots \ w_N)^T$  with  $w_j$  representing the firing strength vector of the  $j$ th training samples that is calculated using (11). Based on (19)  $\beta$  can be calculated as follows:

$$\beta^* = (W^T W + \lambda I_d)^{-1} W^T Y. \quad (20)$$

Finally, by combining the randomly generated antecedents and RR learning based consequents a 0-order TSK-FS is obtained.

#### ***D. Advantages of RR-EL-FIS***

The risks of learning for a model consist of the empirical risk and the confidence interval. F-ELM is to get the optimal vector  $\beta$  by minimizing the bias between model output and real output in the training dataset, i.e., only the empirical risk minimization is considered. For small and noisy datasets, the model trained by F-ELM will only have a good performance in the training dataset but a bad performance in the test dataset because the empirical risk minimization is apt to make the model overfitting. In order to get an appropriate solution of  $\beta$ , we need to minimize the empirical risk and the confidence interval simultaneously. According to statistical learning theory [24-25], RR criterion can introduce regularization term into LS to improve the generalization performance of the model. As shown in [4], using RR to optimize the consequents of a fuzzy system can loose the constraint conditions to get the better solution compared with some classical algorithms that have introduced confidence interval minimization in the objective function, such as

LS-SVM. Besides, similar to F-ELM, RR-EL-FS is based on fuzzy system such that it can keep the good interpretability and enhance the generalization performance of the trained model simultaneously.

## **IV. EXPERIMENTAL STUDIES**

In this section, we evaluate the effectiveness of the proposed RR-EL-FS by comparing it with several related methods on both synthetic and real-world datasets. The experimental studies are organized as follows. In Subsection A, the experimental settings are described. The experimental results on the synthetic datasets are reported in Subsection B. In Subsection C, the proposed RR-EL-FS is compared with F-ELM by using eleven real-world datasets. In Subsection D, the comparison between the proposed RR-EL-FS and the other five existing fuzzy system learning algorithms are reported, and in Subsection E the statistical analysis of the experimental results is reported.

### ***A. Experimental Setup***

#### ***1) Methods Adopted for Performance Evaluation***

In Subsections B and C, RR-EL-FS is compared with F-ELM to evaluate the improvement of robustness. The robustness of RR-EL-FS is further compared with other five fuzzy systems learning algorithms in section D. These five fuzzy system learning algorithms are L2-TSK-FS [26],  $\varepsilon$ -TSK-FS(IQP) [27],  $\varepsilon$ -TSK-FS(LSSLI) [27], FS-FCSVM[28] and GENFIS2 in MATLAB toolbox [29].

The same random parameters, i.e., matrix-C, matrix-DC in RR-EL-FS and F-ELM have been used to eliminate the influence of random parameters on two

algorithms. Besides, the numbers of fuzzy rules are set to 20 for all fuzzy rule based algorithms in Subsections C and D if not specified.

## 2) *Evaluation Index*

The generalization performance index  $J$  in (21) is used in all experiments to evaluate the generalization abilities [30] for regression.

$$J_{reg} = \sqrt{\frac{\frac{1}{N} \sum_{i=1}^N (y'_i - y_i)^2}{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2}} \quad (21)$$

where  $N$  is the number of samples in a test dataset;  $y_i$  and  $y'_i$  are the real output and the model output for the  $i$ th test input, respectively; and  $\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$ . The smaller the value of  $J$ , the better the generalization performance.

## 3) *Experimental Platform.*

All experiments are carried out in the following platform: 1) CPU: Intel(R) Core(TM) i7-4790 CPU; 2) CPU Clock Speed: 3.60GHz; 3) Memory: 16GB; 4) System: WIN7 64bit; 5) Programming Environment: MATLAB 8.1.0.604 (R2013a).

## B. *Synthetic Dataset*

In this subsection, a synthetic dataset is generated using the following *sinc* function.

$$y = \sin(x)/x + \sigma(0,0.3), \quad (22)$$

where  $\sigma(0,0.3)$  is the noise term, which was normally distributed with mean and variance set to 0 and 0.3, respectively. We set input attribute  $x$  in the interval  $[-10,10]$  with the step of  $\pi/40 + 10^{-5}$  to generate 255 pairs training data

$(x_i, y_i)$  . Meanwhile, we use *sinc* function below to generate 100 pairs test data  $(x_t, y_t)$ .

$$y = \sin(x)/x, \quad (23)$$

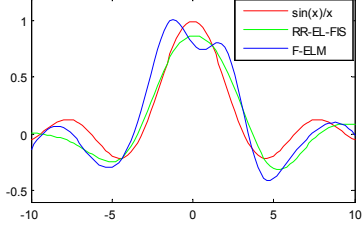
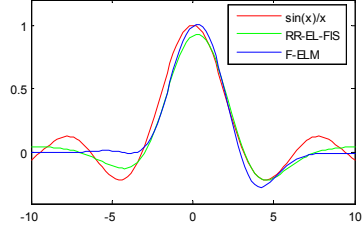
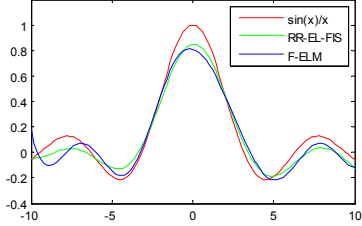
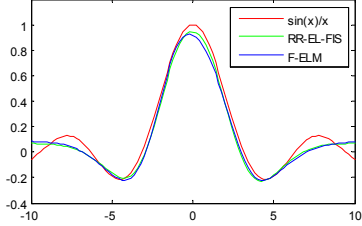
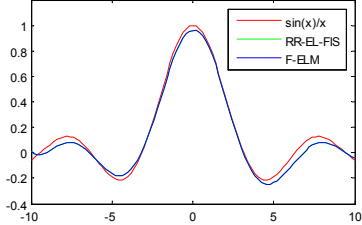
where we choose input attribute  $x_i$  in the interval  $[-10, 10]$  with the step of 0.2 to generate 100 pairs test data excluding  $x_i = 0$ .

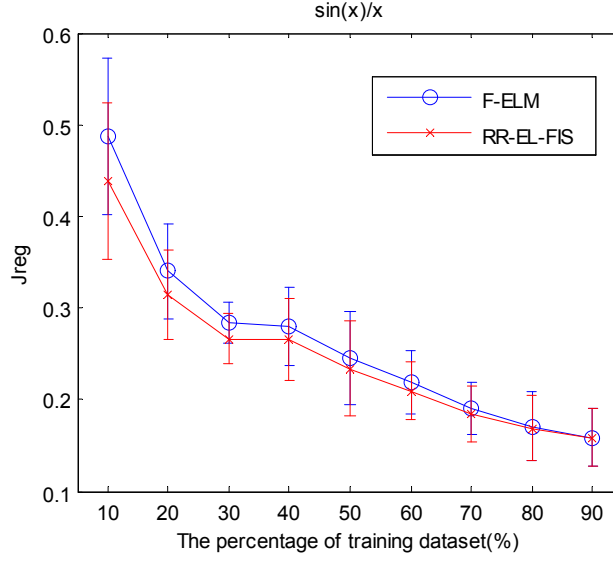
**1) *The performance comparison between RR-EL-FS and F-ELM under different sizes of training datasets***

For the generated training data, different percentages (10%-90%) have been used for model training in order to evaluate the influence of dataset size on RR-EL-FS and F-ELM algorithms. Fig. 2 and Table I show the experimental results.

From Fig. 2 and Table I we can give the following observations: (1) The values of  $J_{reg}$  of two algorithms both decrease with the increasing of the size of training dataset. (2) With the increasing of the size of training dataset, the generalization abilities of the models trained by two algorithms are both enhanced. (3) The values of  $J_{reg}$  of RR-EL-FS are always smaller than F-ELM, which means that RR-EL-FS has better robustness than F-ELM. (4) The values of  $J_{reg}$  of RR-EL-FS are smaller than that of F-ELM obviously when the sizes of the training datasets are smaller, which indicates that the proposed RR-EL-FS is more advantageous to F-ELM in small datasets. (5) When the size of the training dataset increases, the advantage of RR-EL-FS to F-ELM becomes less obvious.

**Table I Experimental result on synthetic datasets**

Percentage of training dataset	$J_{reg}$		Modeling effect
	F-ELM (Mean $\pm$ Std)	RR-EL-FS (Mean $\pm$ Std)	
10%	0.4870 $\pm$ 0.0847	0.4386 $\pm$ 0.0855	
30%	0.2836 $\pm$ 0.0227	0.2663 $\pm$ 0.0269	
50%	0.2460 $\pm$ 0.0511	0.2341 $\pm$ 0.0518	
70%	0.1899 $\pm$ 0.0281	0.1846 $\pm$ 0.0303	
90%	0.1582 $\pm$ 0.0317	0.1579 $\pm$ 0.0251	

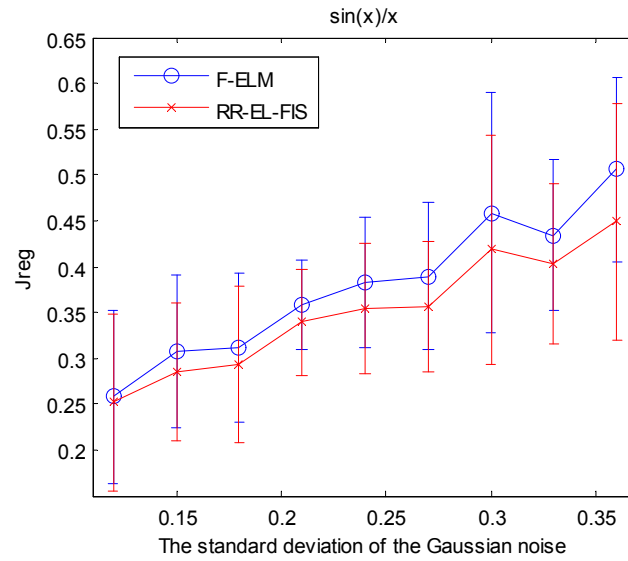


**Fig.2 The performance comparison between RR-EL-FS and F-ELM in different sizes of training dataset.**

## 2) *The Performance Comparison between RR-EL-FS and F-ELM under Different Degrees of Noise*

In this subsection nine datasets with different degrees of noise are generated to compare the robustness of RR-EL-FIS and F-ELM to noise. All datasets were generated based on (22) as follows: We set the input in the interval  $[-10, 10]$  with step of  $\pi/80 + 10^{-5}$  to generate 255 pairs training data  $(x_i, y_i)$ . Then Gaussian white noise with 0 mean and nine different standard deviations, i.e.,  $\sigma = 0.12, 0.15, 0.18, 0.21, 0.24, 0.27, 0.30, 0.33, 0.36$ , were added in the original data. The larger the standard deviation is, the larger the noise is. Finally, nine noisy datasets are obtained that were used for model training. In our experiment, we randomly select 10% of each dataset as training dataset. After the model has been trained, the noise-free test dataset generated in subsection IV. B was adopted to evaluate the generalization abilities. The above procedure is repeated ten times and the mean and

standard deviation of the results are reported for performance comparison as shown in Fig. 3 and Table II.



**Fig. 3 The performance comparison between RR-EL-FS and F-ELM under datasets with different degrees of noise.**

**Table II The performance comparison between RR-EL-FIS and F-ELM under datasets with different degrees of noise**

$\sigma^*$	F-ELM	RR-EL-FIS
0.12	$0.4250 \pm 0.0844$	$0.4155 \pm 0.0630$
0.15	$0.4335 \pm 0.0986$	$0.4221 \pm 0.0960$
0.18	$0.4634 \pm 0.0674$	$0.4475 \pm 0.0667$
0.21	$0.5290 \pm 0.1104$	$0.5057 \pm 0.1114$
0.24	$0.5139 \pm 0.0568$	$0.4903 \pm 0.0477$
0.27	$0.5871 \pm 0.1651$	$0.5493 \pm 0.1704$
0.30	$0.5202 \pm 0.1067$	$0.4921 \pm 0.1035$
0.33	$0.5632 \pm 0.0961$	$0.5143 \pm 0.1161$
0.36	$0.6124 \pm 0.1577$	$0.5350 \pm 0.1339$

\* Standard deviation of Gaussian noise.

From Fig. 3 and Table II we can give the following observations: 1) With the degree of noise increasing, the values of  $J_{reg}$  of two algorithms both increase. 2) The

values of  $J_{reg}$  of RR-EL-FIS are smaller than that of F-ELM under different degrees of noise. 3) The gaps of the curves of  $J_{reg}$  between two algorithms are small when the degree of noise is low, and with the degree of noise increasing the gaps between two algorithms become bigger as shown in Fig.3, which means that RR-EL-FIS has the better robustness than F-ELM to noise.

**Table III Descriptions of 11 real-world datasets**

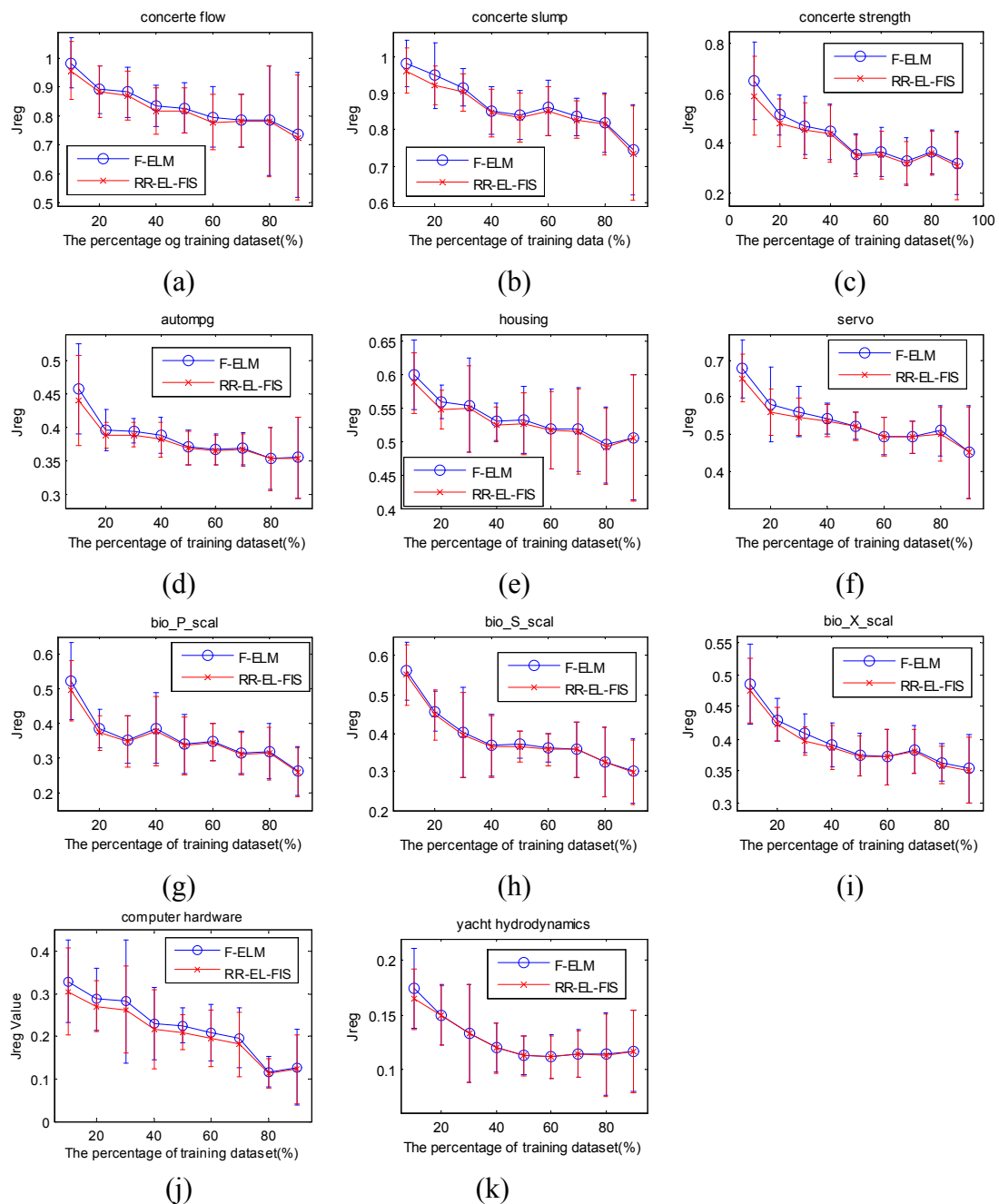
Dataset	Size	Number of attributes
Autompg	392	7
Concrete flow	103	7
Concrete slump	103	7
Concrete strength	103	7
Computer hardware	209	7
Housing	506	13
Servo	167	4
Yacht Hydrodynamics	308	6
bio_P_scal	293	6
bio_S_scal	293	6
bio_X_scal	293	6

### C. Real-world dataset

In this subsection, 11 real-world datasets are adopted to further evaluate the performance of RR-EL-FS. Eight of them come from the UCI database and the others are three fermentation datasets [26]. In our experiments, the inputs of all dataset have been normalized to  $[0,1]$ . The detailed information of all datasets are listed in Table III.

The similar way in subsection IV.B.1) is used in the experiments. Here, for each dataset different percentages of data were adopted for model training and the remaining data were used for test. For all the fuzzy rules based methods, the numbers of fuzzy rules were set to 20. The experimental results are reported in Fig. 4, Table IV

and Table V. In Fig. 4, the influence of size of training dataset is shown. Furthermore, the detailed results obtained based on 10% and 90% data as training set are given in Table IV and Table V, respectively. From these results, we can get the similar conclusions as in subsection IV.B.1), i.e., RR-EL-FS is usually more robust than F-ELM. In particular, when the size of training dataset is small the advantage is very obvious.



**Fig.4 Performance comparison between RR-EL-FS and F-ELM under different sizes of training datasets in 11 real-world datasets.**

**Table IV Performance comparison ( $J_{reg}$ ) between RR-EL-FS and F-ELM under 10% of data as training set in 11 real-world datasets**

Dataset	F-ELM (Mean $\pm$ Std)	RR-EL-FS (Mean $\pm$ Std)
Concrete flow	0.9831 $\pm$ 0.0856	0.9557 $\pm$ 0.1010
Concrete slump	0.9830 $\pm$ 0.0647	0.9631 $\pm$ 0.0632
Concrete strength	0.6504 $\pm$ 0.1551	0.5906 $\pm$ 0.1584
Autompg	0.4579 $\pm$ 0.0668	0.4408 $\pm$ 0.0666
Housing	0.5994 $\pm$ 0.0517	0.5873 $\pm$ 0.0452
Servo	0.6766 $\pm$ 0.0783	0.6514 $\pm$ 0.0653
bio_P_scal	0.5224 $\pm$ 0.1108	0.4946 $\pm$ 0.0874
bio_S_scal	0.5602 $\pm$ 0.0756	0.5502 $\pm$ 0.0782
bio_X_scal	0.4851 $\pm$ 0.0613	0.4752 $\pm$ 0.0500
Computer hardware	0.3300 $\pm$ 0.0972	0.3059 $\pm$ 0.1031
yacht hydrodynamics	0.1743 $\pm$ 0.0370	0.1646 $\pm$ 0.0276

**Table V Performance comparison ( $J_{reg}$ ) between RR-EL-FS and F-ELM under 90% of data as training data in 11 real-world datasets**

Dataset	F-ELM (Mean $\pm$ Std)	RR-EL-FS (Mean $\pm$ Std)
Concrete flow	0.7345 $\pm$ 0.2143	0.7253 $\pm$ 0.2167
Concrete slump	0.7431 $\pm$ 0.1247	0.7351 $\pm$ 0.1298
Concrete strength	0.3201 $\pm$ 0.1277	0.3073 $\pm$ 0.1361
Autompg	0.3553 $\pm$ 0.0601	0.3548 $\pm$ 0.0604
Housing	0.5058 $\pm$ 0.0940	0.5044 $\pm$ 0.0941
Servo	0.4527 $\pm$ 0.1251	0.4499 $\pm$ 0.1242
bio_P_scal	0.2620 $\pm$ 0.0704	0.2585 $\pm$ 0.0701
bio_S_scal	0.3031 $\pm$ 0.0832	0.2986 $\pm$ 0.0846
bio_X_scal	0.3544 $\pm$ 0.0530	0.3512 $\pm$ 0.0512
Computer hardware	0.1275 $\pm$ 0.0885	0.1233 $\pm$ 0.0809
yacht hydrodynamics	0.1168 $\pm$ 0.0378	0.1164 $\pm$ 0.0382

**Table VI Performance comparison of seven algorithms under 10% of data as training set in 11 real-world datasets (Mean $\pm$ Std)**

Dataset	RR-EL-FS	F-ELM	FS-FCSV M	FIS	L2-TSK-F S	LSSLI	IQP
Concrete flow	0.9557 $\pm 0.1010$	0.9831 $\pm 0.0856$	0.9882 $\pm 0.1261$	1.0318 $\pm 0.1082$	<b>0.8936</b> $\pm 0.0575$	1.0161 $\pm 0.1006$	1.0394 $\pm 0.1028$
Concrete slump	<b>0.9631</b> $\pm 0.0632$	0.9830 $\pm 0.0647$	0.9747 $\pm 0.0655$	1.1115 $\pm 0.0495$	0.9751 $\pm 0.0658$	1.1880 $\pm 0.2302$	1.1747 $\pm 0.1738$
Concrete strength	0.5906 $\pm 0.1584$	0.6504 $\pm 0.1551$	0.8856 $\pm 0.0985$	0.6952 $\pm 0.2215$	0.5473 $\pm 0.1268$	<b>0.5336</b> $\pm 0.1631$	0.5522 $\pm 0.1978$
Autompg	0.4408 $\pm 0.0666$	0.4579 $\pm 0.0668$	0.4136 $\pm 0.0303$	0.4922 $\pm 0.0296$	<b>0.4124</b> $\pm 0.0178$	0.4714 $\pm 0.0462$	0.4826 $\pm 0.0587$
Housing	0.5873 $\pm 0.0452$	0.5994 $\pm 0.0517$	<b>0.5637</b> $\pm 0.0400$	0.6226 $\pm 0.0265$	0.5681 $\pm 0.0289$	0.5913 $\pm 0.0502$	0.6257 $\pm 0.0504$
Servo	<b>0.6514</b> $\pm 0.0653$	0.6766 $\pm 0.0783$	0.8702 $\pm 0.1017$	0.7898 $\pm 0.1241$	0.7498 $\pm 0.1060$	0.7354 $\pm 0.1630$	0.7982 $\pm 0.1007$
bio_P_scal	<b>0.4946</b> $\pm 0.0874$	0.5224 $\pm 0.1108$	0.5645 $\pm 0.0788$	0.6816 $\pm 0.0496$	0.5562 $\pm 0.0594$	0.5619 $\pm 0.1250$	0.5703 $\pm 0.2250$
bio_S_scal	<b>0.5502</b> $\pm 0.0782$	0.5602 $\pm 0.0756$	0.6220 $\pm 0.0991$	0.7130 $\pm 0.0673$	0.6294 $\pm 0.1039$	0.6620 $\pm 0.0914$	0.6266 $\pm 0.1098$
bio_X_scal	0.4752 $\pm 0.0500$	0.4851 $\pm 0.0613$	0.4583 $\pm 0.0360$	0.4638 $\pm 0.0397$	<b>0.4238</b> $\pm 0.0139$	0.5248 $\pm 0.0677$	0.5679 $\pm 0.0989$
Computer hardware	0.3059 $\pm 0.1031$	0.3300 $\pm 0.0972$	0.8550 $\pm 0.1204$	0.4241 $\pm 0.2778$	0.6798 $\pm 0.1802$	<b>0.1514</b> $\pm 0.0454$	0.1843 $\pm 0.0834$
yacht	<b>0.1646</b>	0.1743	0.6024	0.6877	0.5905	0.6566	0.6608
hydrodynamics	$\pm 0.0276$	$\pm 0.0370$	$\pm 0.0645$	$\pm 0.0868$	$\pm 0.0516$	$\pm 0.0694$	$\pm 0.0768$
Average	<b>0.5618</b>	0.5839	0.7089	0.7012	0.6387	0.6448	0.6621
Times of the best	5	0	1	0	3	2	0

#### ***D. Performance Comparison with Related Methods***

In this Subsection, the proposed RR-EL-FS is further evaluated with several related methods that are described in section IV.A. All the adopted algorithms are fuzzy rule based methods and the numbers of rules for these methods are all set to 20 in the experiments. The adopted datasets are the 11 real-world datasets described in

Table III. In order to simulate the scene of small and noisy datasets, for each dataset 10% of data were used for training and the remaining were used for test. The above procedure is repeated 10 times and the mean and standard deviation of index  $J_{reg}$  are reported for performance evaluation. The experimental results are reported in Tables VI and VII. It is noted that the numbers of fuzzy rules of FS-FCSVM and L2-TSK-FS are set to 5 for datasets Concrete flow, Concrete strength, Concrete slump and servo because these two algorithms generate fuzzy rules based on fuzzy c-mean clustering which requires the number of training data is larger than the number of clusters, i.e., the number of rules.

**Table VII Training time of all algorithms (second)**

	RR-EL-FS	F-ELM	FS-FCSVM	FIS	L2-TSK-FS	LSSLI	IQP
Concrete flow	<b>7.14E-04</b>	7.38E-04	0.0015	0.0017	0.0036	0.0026	0.0083
Concrete slump	<b>4.48E-04</b>	8.17E-04	0.0013	0.0028	0.0043	0.0025	0.009
Concrete strength	<b>7.32E-04</b>	7.34E-04	0.0016	0.0017	0.0036	0.0029	0.004
Autompg	0.0036	<b>0.0023</b>	0.0082	0.0069	0.0143	0.0107	0.0145
Housing	0.0037	<b>0.0033</b>	0.0205	0.0127	0.0213	0.0312	0.0446
Servo	<b>0.001</b>	<b>0.001</b>	0.0019	0.0048	0.0052	0.0036	0.0112
bio_P_scal	0.0021	<b>0.0018</b>	0.0067	0.0042	0.0122	0.0109	0.0125
bio_S_scal	0.0025	<b>0.0017</b>	0.0058	0.0055	0.0105	0.0088	0.0111
bio_X_scal	0.003	<b>0.0018</b>	0.0061	0.0053	0.0126	0.0078	0.016
Computer hardware	0.0025	<b>0.0015</b>	0.0045	0.0035	0.0097	0.0091	0.0128
yacht hydrodynamics	0.002	<b>0.0019</b>	0.0072	0.0042	0.0142	0.0084	0.0163
Average	0.0020	<b>0.0016</b>	0.0059	0.0048	0.0101	0.0090	0.0146

Table VI shows that among seven algorithms the proposed RR-EL-FS has obtained minimal average of index  $J_{reg}$  and the maximum number of times for the best results that have been obtained. From Table VI we know that RR-EL-FS has obtained the best results in datasets Concrete slump, servo, bio\_P\_scal, bio\_S\_scal and yacht hydrodynamics. From Table III we can find that most of these six datasets

are smaller than other datasets, which also further indicates that RR-EL-FS has better robustness to small and noisy datasets.

In Table VII the training time of all algorithms is compared. We can find that the training speed of RR-EL-FS is highly competitive to F-ELM and obviously faster than other algorithms.

#### ***E. Statistical Analysis***

The experimental results in subsection IV.D are further evaluated with statistical analysis. Friedman test [31-32] is adopted for this purpose, which is a nonparametric test method to evaluate whether the significant difference exists between different methods. Our Friedman test is based on the index  $J_{reg}$  of all algorithm in Table VI.

For Friedman null hypothesis, it means that there are no significant difference between RR-EL-FS and other algorithms with the significant level  $\alpha = 0.05$ . The rankings of all algorithms are given in Table VIII and the statistical result of Friedman test is shown in Table IX. The lower ranking of the algorithm, the better performance of the algorithm. The following observations can be given from the results in Tables VIII and IX: (i) The ranking of RR-EL-FS is the lowest which means that the performance of RR-EL-FS is the best. (ii) The null hypothesis is rejected in Table IX, which indicates that the performance of all algorithms are significant different. As we can observe in Tables II, IV, V and VI, RR-EL-FS has shown the better robustness than other several related methods to small and noisy datasets.

**Table VIII The rankings of all algorithms in 11 real-world datasets**

Algorithms	Ranking
RR-EL-FS	2.1818
FS-FCSVM	4
FIS	5.8182
L2-TSK-FS	2.8182
LSSLI	4.2727
IQP	5.4545
F-ELM	3.4545

**Table IX Statistical result of Friedman test**

Index	Statistic	$p$ – value	Hypothesis
$J_{reg}$	24.74026	0.000381	Rejected

## V. CONCLUSIONS

In this study a ridge regression based extreme learning fuzzy system training algorithm is proposed to overcome the challenge that when training dataset is small and noisy, F-ELM will has weak robustness. By introducing the ridge regression learning mechanism, the proposed algorithm has overcome this shortcoming to a great extent, while it still keeps the advantages of F-ELM, such as the good interpretability and the fast training speed.

Although the proposed method has shown promising performance in small and noisy datasets, there are still many work that deserves to study in depth in future. For example, the propose algorithm is only available to 0-order TSK-FS, How to extend it to other types of fuzzy systems is a significant work.

## ACKNOWLEDGEMENT

This work was supported in part by the Outstanding Youth Fund of Jiangsu

Province (BK20140001), National Natural Science Foundation of China (61272210), National key research and development project (2016YFB0800803 ) and the Hong Kong Polytechnic University (G-UA68, G-UA3W).

## REFERENCES

- [1] J. M. Benitez, J. L. Castro, and I. Requena, "Are artificial neural networks black boxes ?," *IEEE Transactions on Neural Networks*, vol. 8, no. 5, pp. 1156-1164, 1997.
- [2] G.-B. Huang, Q.-Y. Zhu, C.K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," *Proc. int. joint Conf. neural Netw*, vol. 2, no. 2, pp. 985-990, 2004.
- [3] G.-B. Huang, D.-H. Wang, Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning & Cybernetics*, vol. 2, no. 2, pp. 107-122, 2011.
- [4] G.-B. Huang, H. Zhou, X. Ding, et al., "Extreme learning machine for regression and multiclass classification.," *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics*, vol. 42, no. 2, pp. 513-529, 2012.
- [5] G.-B. Huang, Q.-Y. Zhu, C.K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489-501, 2006.
- [6] S.-Y. Wong, K.-S. Yap, H.-J. Yap, et al, "On equivalence of FIS and ELM for interpretable rule-based knowledge representation," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 76, no. 7, pp. 1417-1430, 2015.

- [7] L. Zeng, X. Zhang, BU. Zhanyu, Y. Liu, "Extreme learning machine based on principal components estimation," *Computer Engineering and Applications*. vol. 52, no. 4, pp. 110-114, 2016.
- [8] A.E. Hoerl, "Application of ridge analysis to regression problems," *Chemical Engineering Progress*, vol. 58, no. 3, pp. 54-59, 1962.
- [9] L.A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338-353, 1965.
- [10] F.L. Chung, Z. Deng, S.T. Wang, "From minimum enclosing ball to fast fuzzy system training on large datasets," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 1, pp. 173-184, 2009.
- [11] M.F. Azeem, M. Hanmandlu and N. Ahmad, "Generalization of adaptive neuro-fuzzy inference systems," *IEEE Transactions on Neural Networks*, vol. 11, no. 6, pp. 1332-1346, 2000.
- [12] Z.H. Deng, K.S. Choi, F.L. Chung, et al., "Scalable TSK fuzzy modeling for very large datasets using minimal-enclosing-ball approximation," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 210-226, 2011.
- [13] Y.Z. Jiang, Z.H. Deng, S.T. Wang, "0-Order L2-Norm Takagi–Sugeno–Kang type transfer learning fuzzy system," *Acta Electronica Sinica*, vol. 41, no. 5, pp. 897-904, 2013.
- [14] Y.Y. Lin, J.Y. Chang, C.T. Lin, "A TSK-type-based self-evolving compensatory interval type-2 fuzzy neural network (TSCIT2FNN) and its applications," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 447-459, 2014.

- [15] Z.H. Deng, K.S. Choi, L.B. Cao, et al., "T2FELA: Type-2 fuzzy extreme learning algorithm for fast training of interval type-2 TSK fuzzy logic system," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 4, pp. 664-676, 2014.
- [16] K.S. Yap, C.P. Lim, M.T. Au, "Improved GART neural network model for pattern classification and rule extraction with application to power systems," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2310-2323, 2011.
- [17] S.C. Tan, C.P. Lim, M.V.C. Rao, "A hybrid neural network model for rule generation and its application to process fault detection and diagnosis," *Engineering Applications of Artificial Intelligence*, vol. 20, no.1, pp. 203-213, 2007.
- [18] V. Miranda, A.R.G. Castro, "Improving the IEC table for transformer failure diagnosis with knowledge extraction from neural networks," *IEEE Transactions on Power Delivery*, vol. 20, no. 4, pp. 2509-2516, 2005.
- [19] A.E. Hoerl, W.K. Robert, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, pp. 80-86, 2000.
- [20] C. Saunders, A. Gammerman, V. VOVK, "Ridge regression learning algorithm in dual variables," *Proc. of the 15th Int. Conf. on Machine Learning ICML-98, Madison-Wisconsin*, pp. 515-521, 1999.
- [21] B.J.R. Jang, C.T. Sun, E. Mizutani, "Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence," *Prentice-Hall, Upper Saddle River, NJ*, 2010.

- [22] A.E. Hoerl, W.K. Robert, "Ridge regression: applications to nonorthogonal problems" *Technometrics*, Vol. 12, no. 1, pp. 69-82, 1970.
- [23] Z.H. Deng, K.S. Choi, Y.Z. Jiang, et al., "Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2585-2599, 2014.
- [24] D.L. Jia, J.S. Zhang, "Niche particle swarm optimization combined with chaotic mutation," *Control & Decision*, vol. 22, no. 1, pp. 117-120, 2007.
- [25] C.A.C. Coello, D. Ing, M.S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm," 2003.
- [26] Z.H. Deng, J.B. Zhang, T.Z. Jiang, Y.Z. Shi, S.T. Wang, et al., "Fuzzy subspace clustering based zero-order L2-norm TSK fuzzy system," *Electronics & Information Technology*, vol. 37, no. 9, pp. 2082-2088, 2015.
- [27] J.M. Leski, "TSK-fuzzy modeling based on  $\epsilon$ -insensitive learning," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 2, pp. 181-193, 2005.
- [28] C.F. Juang, S.H. Chiu, S.J. Shiu, "Fuzzy system learned through fuzzy clustering and support vector machine for human skin color segmentation," *IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans*, vol. 37, no. 6, pp. 1077-1087, 2007.
- [29] S.L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & Fuzzy Systems*, vol. 2, no. 3, pp. 267-278, 1994.
- [30] Z.H. Deng, K.S. Choi, F.L. Chung, et al., "Scalable TSK fuzzy modeling for very

large datasets using minimal-enclosing-ball approximation,” *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 210-226, 2011.

[31]T. Joachims, “Transductive inference for text classification using support vector machines,” *Sixteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc.* pp. 200-209, 1999.

[32]J. H. Friedman, “On bias, variance, 0/1—loss, and the curse-of-dimensionality,” *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 55-77, 1997.