

Research Article

Fuzzy Weighted Least Squares Support Vector Regression with Data Reduction for Nonlinear System Modeling

Xiaoyong Liu ¹, Aijia Ouyang ², and Zhonghua Yun ³

¹College of Engineering, Zunyi Normal University, Zunyi 563006, China

²School of Information Engineering, Zunyi Normal University, Zunyi 563006, China

³School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

Correspondence should be addressed to Aijia Ouyang; ouyangaijia@163.com

Received 21 June 2018; Accepted 29 November 2018; Published 10 December 2018

Academic Editor: Piotr Jędrzejowicz

Copyright © 2018 Xiaoyong Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a fuzzy weighted least squares support vector regression (FW-LSSVR) with data reduction for nonlinear system modeling based only on the measured data. The proposed method combines the advantages of data reduction with some ideas of fuzzy weighted mechanism. It not only possesses the capability of illuminating local characteristic of the modeled plant but also can deal with the problem of boundary effects resulted from local LSSVR method when the modeled data is at the boundary of whole data subset. Furthermore, in comparison of the SVR, the proposed method only utilizes fewer hyperparameters to construct model, and the overlap factor λ can be chosen in relatively smaller value than SVR to further reduce more computational time. First of all, distilling the original input space into several regions with fuzzy partition by applying Gustafson-Kessel clustering algorithm (GKCA) is a foundation for data reduction and the overlap factor is introduced to reduce the size of subsets. Following that, those subset regression models (SRMs) which can be simultaneously solved by LSSVR are integrated into an overall output of the estimated nonlinear system by fuzzy weighted. Finally, the proposed method is demonstrated by experimental analysis and compared with local LSSVR, weighted SVR, and global LSSVR methods by using the index of computational time and root-mean-square error (RMSE).

1. Introduction

It is well known that, in a large number of applications such as advanced control, process simulation, fault detection, or other research areas, a significant problem is to construct mathematical model of estimated system only based on its measured data. Some major theories or methods on identifying nonlinear system have been independently developed in various research field including fuzzy system [1], neural networks [2], and other approach [3]. However, LSSVR method [4], like SVR [5], also adopts the structural risk minimization and has the better equilibrium between sparsity and modeling accuracy. Furthermore, LSSVR, by substituting a set of equality conditions for complex inequalities ones, translates complex quadratic optimization as a simple linear programming, which is greatly relieve computational load. In literature [6], the power of generalization for LSSVR is no worse than that SVR.

Therefore, the LSSVR has been attracting extensive attentions and has obtained successful application like time series prediction [7, 8], subspace identification [9, 10], signal processing [11, 12], and other applications [13, 14] during the past few years. In spite of the LSSVR approach, referred to as the global LSSVR (G-LSSVR) approach, has become an effective tool in various applications, and can identify an estimated model whose modeling accuracy is guaranteed by obtaining an appropriate mathematical model [15] and a proper hyperparameter set which usually consists of the two variables: the kernel width (σ) and the penalty factor (γ). Generally, it is an insignificant for global-LSSVR to derive a well local behavior. We discovers in literature [15, 16] that G-LSSVR has also some defective in illuminating local behavior.

Recently, local modeling approaches, as an alternative efficient algorithm, because of their superiorities to identify various areas of estimated nonlinear system, seem desirable. In the literature [17], a local fusion modeling method based

on LSSVR and nerofuzzy has been proposed. It employs LSSVR and a learning method named as layering two kind of problem to construct each local model. In the literature [18], it adopts another approach which makes use of interest training point to identify local model instead of all points and by applying vector-norm distance [19] to search for K nearest points. To seek a set of optimal data points, a Euclidian distance measurement method [20, 21] is proposed and the local model is set up according to these neighboring points. From the pointview of capturing the local behavior, our aims are to construct their connection between some models and the localized support vector regression (LSVR) method [22] has been proposed. Considering the much more computational time of G-LSSVR, a local grey SVR [23] is developed to speed up the calculational time. Further, by introducing regularization, a general local and global learning framework [24] formulates multiple classifier in each data of neighbours.

Nevertheless, the local modeling approaches or local-LSSVR have more superiorities in identifying local characteristics than that approaches such as global-SVR or LSSVR; it is still unsatisfactory in modeling global capability. First, due to the different criterion to select K nearest neighbor training data in subsets, the better performance for local-LSSVR is not derived when those training data are at the boundary area. Second, because the number of constructing all local models is equal to the size of testing set, the local LSSVR approach generally leads to a heavy computation load [16]. Third, it generates boundary effects resulting from local LSSVR method when the modeled data is at the boundary of whole data subset.

Based on the above consideration, our aims present a FW-LSSVR method for nonlinear system modeling based only on the obtained measure data. The paper integrates the superiorities of GKCA, weighted average mechanism, and some ideas from LSSVR. First of all, distilling the original input space into several regions with fuzzy partition by applying GKCA is a foundation for data reduction. Following that, those subset regression models (SRMs) which can be simultaneously solved by LSSVR are integrated into an overall output of the estimated nonlinear system by fuzzy weighted. The proposed method not only possesses the capability of illuminating local characteristic of the estimated models but also can deal with the problem of boundary effects resulted from local LSSVR method. Furthermore, in comparison of the support vector regression (SVR), the proposed method only utilizes fewer hyperparameters to construct model, and the overlap factor λ is chosen in relatively smaller value than SVR to further reduce more computational time. Finally, experimental analysis demonstrates that our approach not only overcomes the disadvantages of local LSSVR, weighted SVR, and global LSSVR methods in the process of modeling nonlinear system but also has better root-mean-square error (RMSE) performance and needs less computational time.

The paper is organised as follows: brief descriptions for LSSVR and GKCA in Section 2 are firstly given, the proposed method is introduced in detail in Sections 3 and 4 shows several examples for demonstrating our approach, and Section 5 summarizes the whole paper.

2. Preliminaries

2.1. Least Squares Support Vector Regression. It has been shown that the generalization performance for LSSVR presented by [25] is comparable to that of the SVR through a meticulous empirical study [6]. Next, we will concisely introduce LSSVR with the following training points, where $\mathbf{x}_k \in \mathbf{R}^n$ is the input pattern and y_k is the corresponding target.

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N) \quad (1)$$

The LSSVR can be represented for a test input \mathbf{x} as

$$f(\mathbf{x}) = \sum_{k=1}^N \alpha_k K(\mathbf{x}, \mathbf{x}_k) + b \quad (2)$$

Because of adopting the Gaussian kernel width, kernel function $K(\mathbf{x}, \mathbf{x}_k)$ in (2) can further be rewritten as

$$f(\mathbf{x}) = \sum_{k=1}^N \alpha_k \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_k\|^2}{2\sigma^2}\right) + b \quad (3)$$

where support-value-vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$ and bias b are solved by formulating the following optimization [25]:

$$\begin{aligned} \min_{\mathbf{w}, b, \xi_k} \quad & J_1(\mathbf{w}, b, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \frac{1}{2} \sum_{k=1}^N \xi_k^2 \\ \text{s.t.} \quad & y_k = \mathbf{w}^T \Phi(\mathbf{x}_k) + b + \xi_k, \quad k = 1, 2, \dots, N. \end{aligned} \quad (4)$$

where $\Phi(\cdot)$ represents a feature mapping which nonlinear space is transformed to a high-dimensional linear space and parameter $\gamma \in \mathbf{R}^+$ is regularization constant which governs the relative importance between the data fitting and the smoothness of the solution. Using Lagrange multiplier method for (4) gives rise to an unconstrained optimization problem:

$$\begin{aligned} L_1(\mathbf{w}, b, \xi; \boldsymbol{\alpha}) = & J_1(\mathbf{w}, b, \xi) \\ & - \sum_{k=1}^N \alpha_k (\mathbf{w}^T \Phi(\mathbf{x}_k) + b + \xi_k - y_k) \end{aligned} \quad (5)$$

In terms of the KKT condition, one derives

$$\begin{aligned} \frac{\partial L_1}{\partial \mathbf{w}} = 0 & \longrightarrow \mathbf{w} = \sum_{k=1}^N \alpha_k \Phi(\mathbf{x}_k) \\ \frac{\partial L_1}{\partial b} = 0 & \longrightarrow \sum_{k=1}^N \alpha_k = 0 \\ \frac{\partial L_1}{\partial \xi_k} = 0 & \longrightarrow \alpha_k = \gamma \xi_k \\ \frac{\partial L_1}{\partial \alpha_k} = 0 & \longrightarrow (\mathbf{w}^T \Phi(\mathbf{x}_k) + b) - y_k + \xi_k = 0, \end{aligned} \quad (6)$$

$$k = 1, 2, \dots, N$$

Consequently, learning process of LSSVR corresponding to (5) is implemented by solving

$$\left(\begin{array}{c|c} \mathbf{0} & \mathbf{1}_N^T \\ \hline \mathbf{1}_N & \Omega + \gamma^{-1} \mathbf{I}_N \end{array} \right) \left(\begin{array}{c} b \\ \alpha \end{array} \right) = \left(\begin{array}{c} 0 \\ \gamma \end{array} \right) \quad (7)$$

where $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$, $\mathbf{1}_N = [1, 1, \dots, 1]^T$, $\Omega_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$, and $\forall i, j = 1, 2, \dots, N$, with $K(\cdot, \cdot)$ a positive-definite Mercer kernel function meeting the Mercer's theorem.

2.2. Gustafson-Kessel Clustering Algorithm. Clustering analysis plays an important role in classification and regression problem. In order to study some important characteristics of complex system, it is crucial for researchers to decompose an original data set into several subsets which is well reflect a system's behavior. Especially, GKCA [26] used for extracting various clustering center in different shape and direction for a larger data set [27] and is superior to conventional FCM. GKCA can be achieved by minimizing the following objective function:

$$J = \sum_{k=1}^R \sum_{i=1}^N \mu_{ik}^m (\mathbf{x}_i - \mathbf{v}_k)^T \mathbf{M}_k (\mathbf{x}_i - \mathbf{v}_k) \quad (8)$$

$$\sum_{k=1}^R \mu_{ik} = 1, \quad i = 1, 2, \dots, N \quad (9)$$

μ_{ik} is a component from fuzzy matrix \mathbf{U} , \mathbf{M}_k is defined by (12), R describes the number of clustering center \mathbf{v}_k , and it needs to be predefined. In a nutshell, GKCA can be boiled down to the following steps:

(1) calculating the cluster centers

$$\mathbf{v}_k^{(l)} = \frac{\sum_{i=1}^N (\mu_{ik}^{(l-1)})^m \mathbf{x}_i}{\sum_{i=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq k \leq R \quad (10)$$

where l denotes iteration number and N is the number of all data points.

(2) computing \mathbf{F}_k according to the definition of covariance

$$\mathbf{F}_k = \frac{\sum_{i=1}^N (\mu_{ik}^{(l-1)})^m (\mathbf{x}_i - \mathbf{v}_k^{(l)}) (\mathbf{x}_i - \mathbf{v}_k^{(l)})^T}{\sum_{i=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq k \leq R \quad (11)$$

(3) computing the distance

$$\mathbf{M}_k = \det(\mathbf{F}_k)^{1/n} (\mathbf{F}_k)^{-1} \quad (12)$$

$$d_{ik\mathbf{M}_k}^2 = (\mathbf{x}_i - \mathbf{v}_k^{(l)})^T \mathbf{M}_k (\mathbf{x}_i - \mathbf{v}_k^{(l)})$$

(4) revising the components μ_{ik} of fuzzy matrix

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^R (d_{ik\mathbf{M}_k} / d_{jk\mathbf{M}_k})^{2/(m-1)}}, \quad 1 \leq i \leq N, \quad 1 \leq k \leq R \quad (13)$$

The iteration stops when the difference between the fuzzy partition matrices $\mathbf{U}^{(l)}$ and $\mathbf{U}^{(l-1)}$ in the following iterations are lower than δ .

3. Fuzzy Weighted Least Squares Support Vector Regression

The paper develops a new method combining respective advantages both global and local learning method to formulate overall framework. The procedures of the proposed FW-LSSVR approach are depicted by Figure 1.

3.1. Constructing Fuzzy Weighted with Triangle Membership Functions. Applications of fuzzy concepts were early developed by Zadeh [28]. A triangular fuzzy number \tilde{A} can be parametrized by a triplet (a^L, a^M, a^R) , where a^L and a^R denote the left and right bounds, respectively, and a^M represents the mode of \tilde{A} . The membership function of the triangular fuzzy number \tilde{A} is defined by

$$\mu_{\tilde{A}}(u) = \begin{cases} 0, & u < a^L \\ \frac{(u - a^L)}{(a^M - a^L)}, & a^L \leq u \leq a^M \\ \frac{(a^R - u)}{(a^R - a^M)}, & a^M \leq u \leq a^R \\ 0, & u \geq a^R \end{cases} \quad (14)$$

The α -cut A_α of the fuzzy set \tilde{A} in the universe of discourse U is defined by

$$A_\alpha = \{u_i \mid \mu_{\tilde{A}}(u_i) \geq \alpha, u_i \in U\} \quad (15)$$

where $\alpha \in [0, 1]$.

In generally, fuzzy partition is implemented by some clustering methods and GKCA [26] is common used to decompose the original data set. It discovers that GKCA is superior to that of FCM(fuzzy c-means) and subtractive clustering. GKCA extended the standard FCM algorithms by adopting a flexible distance measure that is calculated using covariance matrices as exhibited in (11). Meanwhile, various difformities and orientation in original data set are detected by GKCA.

In this paper, GKCA is used, in which it is based on the minimization of (8). As stated in Section 2.2, the iteration is to be stopped when the termination criterion is satisfied, namely, $\|\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}\| < \delta$, and an appropriate fuzzy membership matrices is obtained finally. Following that, the cluster centers \mathbf{v}_{kj} and spread widths ρ_{kj} are calculated, respectively, as [29]

$$\mathbf{v}_{kj} = \frac{\sum_{i=1}^N x_{ij} u_{ki}}{\sum_{i=1}^N u_{ki}} \quad (16)$$

$$\rho_{kj} = \frac{\sum_{i=1}^N u_{ki} \|\mathbf{x}_{ij} - \mathbf{v}_{kj}\|^2}{\sum_{i=1}^N u_{ki}} \quad (17)$$

$$i = 1, 2, \dots, N; \quad k = 1, 2, \dots, R; \quad j = 1, 2, \dots, n \quad (18)$$

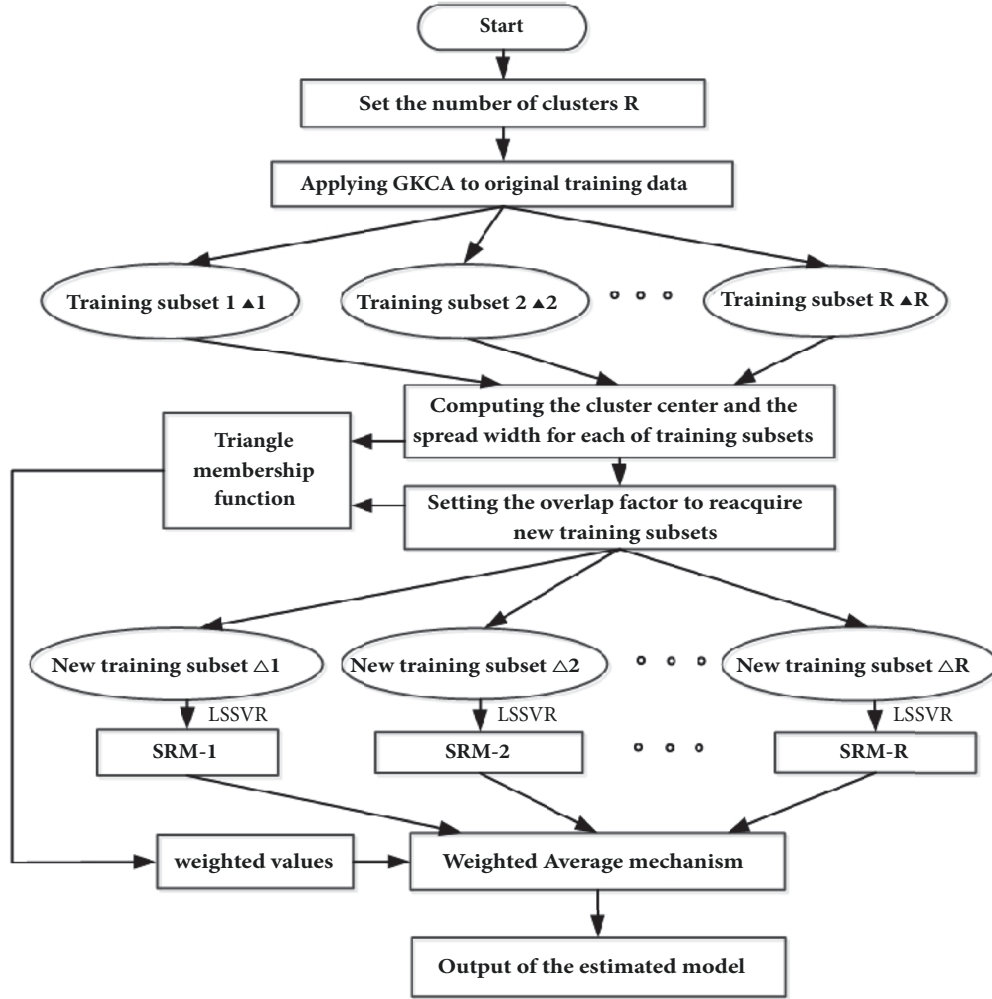


FIGURE 1: An illustration for the proposed FWA-LSSVR approach.

where N is the number of training data, R is the number of clusters, u_{ik} is the degree of membership of \mathbf{x}_i in the cluster k , \mathbf{x}_i is the i th training data, n is a feature dimensionality, and $\| \cdot \|$ measures the distance between two vectors.

From (16) and (17), the weighted values can be calculated by applying triangle membership functions. In order to derive the weighted values, triangle membership function $\mu_k(x_{ij})$ is constructed as follows according to (14):

$$\mu_k(x_{ij}) = \begin{cases} 0 & x_{ij} \leq v_{kj} - \lambda \cdot \sigma_{kj} \\ \frac{x_{ij} - (v_{kj} - \lambda \cdot \sigma_{kj})}{v_{kj} - (v_{kj} - \lambda \cdot \sigma_{kj})} & v_{kj} - \lambda \cdot \sigma_{kj} < x_{ij} < v_{kj} \\ \frac{v_{kj} + \lambda \cdot \sigma_{kj} - x_{ij}}{(v_{kj} + \lambda \cdot \sigma_{kj}) - v_{kj}} & v_{kj} < x_{ij} < v_{kj} + \lambda \cdot \sigma_{kj} \\ 0 & x_{ij} \geq v_{kj} + \lambda \cdot \sigma_{kj} \\ 1 & x_{ij} = v_{kj} \end{cases} \quad (19)$$

Instead of (15), α -cut A_α of the fuzzy set \tilde{A} , the overlap factor λ is introduced into triangle membership functions to more readily dominate the size of original data subsets, and the degree of fulfilment $\beta_k(\mathbf{x}_i)$ is calculated in terms of

$$\beta_k(\mathbf{x}_i) = \mu_k(x_{i1}) \cdot \mu_k(x_{i2}) \cdots \mu_k(x_{in}) = \prod_{j=1}^n \mu_k(x_{ij}) \quad (20)$$

$$i = 1, 2, \dots, N; \quad j = 1, 2, \dots, n \quad k = 1, 2, \dots, R$$

By the normalized firing level of the k th fuzzy sets, weighted values $\omega_k(\mathbf{x}_i)$ is finally calculated as

$$\omega_k(\mathbf{x}) = \frac{\beta_k(\mathbf{x})}{\sum_{k=1}^R \beta_k(\mathbf{x})} \quad (21)$$

In some applications [30], Gaussian membership function is adopted

$$\mu_k(x_{ij}) = \exp(-\theta \|x_{ij} - u_{kj}\|^2) \quad (22)$$

$$i = 1, 2, \dots, N; \quad k = 1, 2, \dots, R; \quad j = 1, 2, \dots, n$$

where $\theta = 4/\tau^2$ and $\tau \geq 0$ describing the width of the Gaussian fuzzy function, which is usually chosen as a interval $[0.3, 0.5]$.

3.2. FW-LSSVR with Data Reduction. Takagi-Sugeno fuzzy models [31] have recently become a powerful practical instrument in identifying the complex system. Based on the fuzzy partition, nonlinear description of estimated system can well be expanded into several simple linear descriptions by applying rules of if-then

$$\begin{aligned} R_i : & \text{If } x_{i1} \text{ is } \mu_{i1}(x_{i1}) \text{ and } \dots \text{ and } x_{in} \text{ is } \mu_{in}(x_{in}) \\ & \text{then } y_i = \mathbf{a}_i^T \mathbf{x} + b_i \quad i = 1, 2, \dots, R \end{aligned} \quad (23)$$

Here $\mu_{i1}, \mu_{i2}, \dots, \mu_{in}$ are the fuzzy membership function of x_n , y_n is corresponding output, and a and b are defined as consequent parameter.

There are the fuzzy sets assigned to corresponding input variables, variable y_i represents the value of the i th rule output, and \mathbf{a}_i and b_i are parameters of the consequent function.

For the input \mathbf{x} , fuzzy weighted output \hat{f} can be summarized as

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^R \omega_i(\mathbf{x}) (\mathbf{a}_i^T \mathbf{x} + b_i) \quad (24)$$

where $\omega_i(\mathbf{x})$ represents the normalized firing strength of the i th rule for the k th sample and is computed by (20) and (21).

Next, substituting linearizing around a point, the proposed method make use of the subset regression models (SRMs) which are simultaneously solved by LSSVR in each fuzzy partition area. Firstly, the original input data set is divided into several subsets with fuzzy partition. In each region, SRM is independently trained by LSSVR. Based on the obtained centers ν_{kj} and the spread width ρ_{kj} from (16) and (17), the old data set is once again decomposed into a new one Δ_k by introducing the overlap factor λ to reduce the size of original subsets. We can perform the partition by the following pseudo code:

```

for   k = 1 : R
    for   i = 1 : N

        if   ( $\nu_{k1} - \lambda \cdot \rho_{k1} \leq x_{i1} \leq \nu_{k1} + \lambda \cdot \rho_{k1}$  &&
              ...
              &&  $\nu_{kn} - \lambda \cdot \rho_{kn} \leq x_{in} \leq \nu_{kn} + \lambda \cdot \rho_{kn}$ )
            ( $x_{ij}, y_i$ )  $\in \Delta_k$ 
        else
            ( $x_{ij}, y_i$ )  $\notin \Delta_k$ 
    end
end

```

end

end

(25)

where the overlap factor λ is introduced to reduce the size of subsets and obtained a new training set with data reduction.

Then, the obtained new training subsets will be used to construct each \mathbf{SRM}_k by (3) as follows:

$$\begin{aligned} \mathbf{SRM}_k(\mathbf{x}) &= f(\mathbf{x})|_{\mathbf{x} \in \Delta_k} \\ &= \sum_{i=1}^{m_k} \alpha_{ki} \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right) + b_k \end{aligned} \quad (26)$$

$k = 1, 2, \dots, R$

where \mathbf{SRM}_k is termed as the k th subset regression model, the parameters α_{ki} and b_k are derived by LSSVR approach, and m_k describes the size of new subset Δ_k . Following that, the weighted values $\omega_i(\mathbf{x})$ computed by (21) are combined with the \mathbf{SRM}_k to form the global predicted output as follows:

$$\hat{y}(\mathbf{x}_i) = \sum_{k=1}^R \omega_k(\mathbf{x}_i) \mathbf{SRM}_k(\mathbf{x}_i) \quad (27)$$

It is clear from (27) that each \mathbf{SRM}_k is solved by LSSVR and can be completed simultaneously. As a result, it can largely improve computational efficiency of the proposed method. In brief, the proposed approach can be summarized as follows.

Step 1. Define the overlap factor λ and select the size of clustering subsets R where R is generally selected to 2.

Step 2. Obtain the appropriate fuzzy partition matrices U by applying GKCA until the termination criterion $U^{(l)} - U^{(l-1)} < \delta$ is satisfied, namely, $U = U^{(l)}$ finally.

Step 3. Compute the cluster centers ν_{kj} and the spread width ρ_{kj} using (16) and (17) from the obtained matrices U and the training data \mathbf{X} .

Step 4. Determine new training subsets Δ_k by (25) based on the overlap factor λ , the cluster centers, and the spread width.

Step 5. Set two hyperparameters $\{\sigma$ and $\gamma\}$ in the LSSVR.

Step 6. Construct each subset regression model $\mathbf{SRM}_k(\mathbf{x})$ by the LSSVR approach and (26) is thus obtained.

Step 7. Compute the global predicted output according to (26) and (27) finally.

4. Experimental Studies

For the purpose of illustrating our approach, both RMSE (root mean squares error) and computational time are considered by four simulated data experiments. All numerical experiments are carried out on the personal computer with

TABLE 1: Comparison results of the proposed method, [15], G-LSSVR and Local-LSSVR with the hyper-parameter set $\{1.5, 1000\}$, and the overlap factor λ as 1.5 are shown for Example 1.

R: the number of the SRMs	RMSE (Training)		RMSE (Testing)	
	our approach	[15]	our approach	[15]
3-SRMs	0.2172	0.8184	0.2151	0.8186
5-SRMS	0.0368	0.5446	0.0363	0.5447
7-SRMs	0.0146	0.3554	0.0140	0.3556
9-SRMs	0.0079	0.3389	0.0076	0.3391
12-SRMS	0.0042	0.1957	0.0040	0.1961
15-SRMs	0.0026	0.1471	0.0026	0.1472
G-LSSVR	1.7295		1.7298	
Local-LSSVR				
M=21 training points	3.7635		3.7630	
M=41 training points	1.1742		1.1622	
M=61 training points	0.0054		0.0050	
M=81 training points	0.0028		0.0026	

a 2.50GHz Intel(R) Celeron(R) CPU and 2 Gbytes memory. This computer runs on Windows XP, with MATLAB R2012a and VC++ 6.0 compiler installed. The LSSVR from Matlab Toolbox was used (this toolbox has been obtained via the Internet at <http://www.esat.kuleuven.be/sista/lssvmlab/>).

We evaluate the performance of the proposed approach on four benchmark data sets [15]. The index adopted for measuring modeling accuracy is selected as

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2} \quad (28)$$

Another index is the total computational time for constructing the proposed method and the local running time for constructing those SRMs. The two indexes of the proposed method are compared with G-LSSVR, local-LSSVR, and [15]. In addition, the importance of the selected different overlap factor λ is also compared. To obtain a fair comparison, their hyperparameters are set as the same values. The local-LSSVR is shortly introduced in the following. Let the training data $\mathbf{X} = \{(\mathbf{x}_i, y_i) \mid i = 1, 2, \dots, N\}$ be obtained by experiment or a real system and \mathbf{x}_t be generated from testing data set and devoted to the test input of predicted output. In the closest regions of \mathbf{x}_t , there are P training inputs to be selected by applying the norm-distance approach. As a result, local-LSSVR models corresponding to all testing output are derived by P training inputs in those regions.

Example 1. The approximated function is

$$y(x) = 10 \cdot \exp^{-x} \cdot \cos(2 \cdot \pi \cdot x), \quad x \in [0, 5] \quad (29)$$

In this function, 501 training points and 1001 testing points are obtained from (29). Due to the use of the proposed approach, there are only two hyperparameters (i.e., σ and γ) to be chosen, whereas SVR approach needs to choose three hyperparameters (i.e., σ , γ , and ϵ). For comparison, Figure 2 shows the results of the WFA-LSSVR and G-LSSVR method. The two indexes both RMSE and computational time

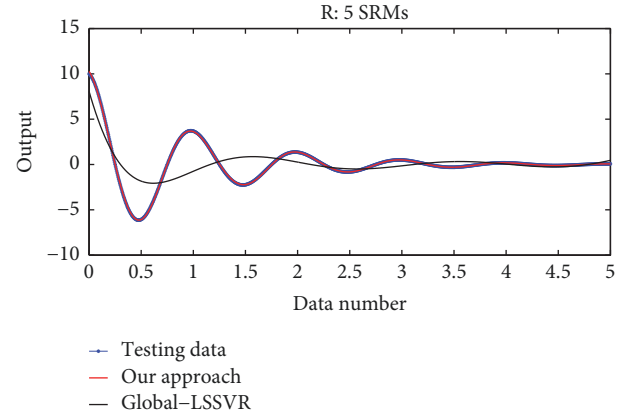


FIGURE 2: Comparison of the proposed model outputs, Global-LSSVR outputs, and testing output data for Example 1.

including local-LSSVR, G-LSSVR, [15] and our approach, are summarized in Tables 1 and 2, respectively. Additionally, the importance of selected different overlap factor λ is also compared by Table 3.

If we take these tables into account, it discovers that our approach obtains a better nonlinear function approximation comparing to RMSE in Table 1 for G-LSSVR, local-LSSVR and [15]. In addition, the running time of the proposed methods in Table 2 is approximately 10-times shorter comparing with local-LSSVR method at least. In other words, the proposed method leads to a less computational time than local-LSSVR. As shown in Table 2, local-LSSVR needs more computational time. The main reason is that the number of required local models is too large and is equal to the size of all testing set. From Table 3 we also see that the comparison results on RMSE and computational time corresponding to the overlap factor λ as 1.5 performed better than as 2.5. That is to say, under the circumstances to cover the training data, larger λ does not necessarily lead to a better performance. These results confirm the superiority of our proposed method over other methods.

TABLE 2: Computational time of the proposed method, G-LSSVR and Local-LSSVR with the hyperparameter set $\{1.5, 1000\}$, and the overlap factor λ as 1.5 are shown for Example 1, where the T-T represents total computational time for building the overall process of the proposed method and L-T represents the computational time for constructing all SRMs.

R:the number of SRMs		3	5	7	9	12	15
Proposed approach	L-T(Second)	0.1406	01563	0.1875	0.2344	0.2813	0.2656
	T-T(Second)	0.6406	0.8750	1.2813	1.2969	1.5469	1.7031
G-LSSVR		T-T(Second): 0.2500			L-T(Second): —		
M: Training points		21	41	61	81	—	—
Local-LSSVR	L-T(Second)	—	—	—	—	—	—
	T-T(Second)	16.2500	17.8125	17.3125	18.0625	—	—

TABLE 3: Comparison results of the selected different overlap factor λ for our approach are shown for Example 1, where R represents the number of SRMs, L-T the computational time for constructing all SRMs, and Δ_i the number of data points for each training subset.

The total training data points $\Delta=501$								RMSE	RMSE	L-T
$R=5$	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7	(Training)	(Testing)	(second)
$\lambda=1.5$	128	159	160	160	128	—	—	0.0146	0.0142	0.1875
$\lambda=2.5$	268	183	266	183	267	—	—	0.2086	0.2081	0.3906
$R=7$	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7			
$\lambda=1.5$	101	134	134	133	101	133	133	0.0132	0.0129	0.2156
$\lambda=2.5$	147	215	222	222	147	215	223	0.1875	0.1869	0.5103

TABLE 4: Comparison results of the proposed method, [15], G-LSSVR and Local-LSSVR with the hyper-parameter set $\{3, 15\}$, and the overlap factor λ as 1.5 are shown for Example 2.

R: the number of the SRMs	RMSE (Training)		RMSE (Testing)	
	our approach	[15]	our approach	[15]
4-SRMs	0.0179	0.0325	0.0180	0.0327
6-SRMS	0.0180	0.0323	0.0180	0.0325
8-SRMs	0.0177	0.0316	0.0178	0.0317
10-SRMs	0.0175	0.0305	0.0175	0.0306
G-LSSVR	0.0289		0.0284	
Local-LSSVR				
M=49 training points	0.1210		0.1288	
M=81 training points	0.0477		0.0558	
M=121 training points	0.0249		0.0282	
M=169 training points	0.0160		0.0190	

Example 2. The approximated function with two variables was

$$f(x_1, x_2) = \frac{\sin\left(\sqrt{x_1^2 + x_2^2}\right)}{\sqrt{(x_1^2 + x_2^2)}}. \quad (30)$$

In this function, x_1 and x_2 equally sampling on interval $[-5, 5]$ are used as training inputs. The number of the training data obtained is 1681 (i.e., 41×41). The number of the used test data is 6561 (i.e., 81×81). For comparison, the same indexes in Example 1 are used including our approach, local-LSSVR, and Global-LSSVR and [15] is summarized in Tables 4 and 5. Additionally, the importance of selected different overlap factor λ is also compared by Table 6. These results show that the proposed method (WFA-LSSVR) outperforms G-LSSVR, local-LSSVR, and [15].

Furthermore, it demonstrates that the predicted outputs of local modeling approach base on LSSVR lead to the problem of boundary effects under the different number of $M=49, 81, 121$, and 169 , as shown in Figure 3. Figure 4 gives the estimated value of the proposed FW-LSSVR approach with 4, 6, 8 and 10 SRMs. From Tables 4 and 5 we can also see that only slightly worse results (training RMSE) were obtained by our approach than the local-LSSVR method with $M=169$ training data points, but running time of the local-LSSVR is significantly longer, in which has no less than 107.3438 seconds in the experiment. From Table 6, the comparison results on RMSE and computational time corresponding to the overlap factor λ as 1.5 performed better than as 3.0. That is to say, under the circumstances to cover the training data, larger λ does not necessarily lead to a better performance and conversely a large number of training data points and more computational time are required to construct all

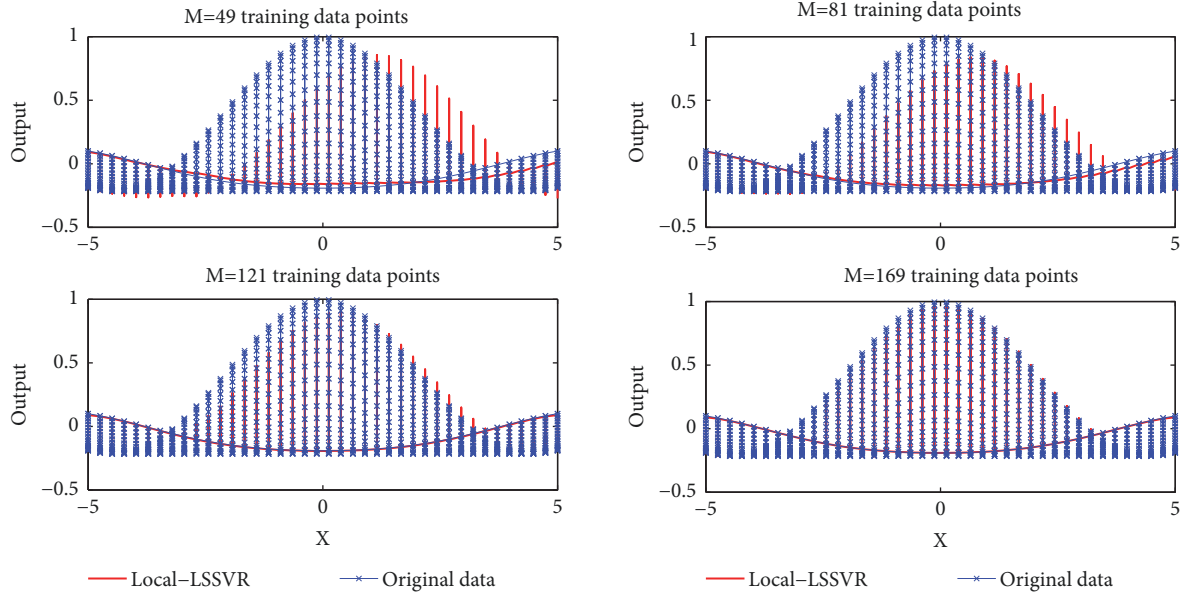


FIGURE 3: The predicted output corresponding to the local-LSSVR technique, where $M=49, 81, 121$, and 169 .

TABLE 5: Computational time of the proposed method, G-LSSVR and Local-LSSVR with the hyperparameter set $\{3, 15\}$, and the overlap factor λ as 1.5 are shown for Example 2, where the T-T represents total computational time for building the overall process of the proposed method and L-T represents the computational time for constructing all SRMs.

R:the number of SRMs		4	6	8	10
Proposed approach	L-T(Second)	0.4688	0.7031	0.7969	1.0156
	T-T(Second)	3.0469	4.8125	5.1875	6.5781
G-LSSVR		T-T(Second): 3.2969			L-T (Second): —
M: Training points		49	81	121	169
Local-LSSVR	L-T(Second)	—	—	—	—
	T-T(Second)	107.3438	115.6563	127.4063	155.1563

TABLE 6: Comparison results of the selected different overlap factor λ for our approach are shown for Example 2, where R represents the number of SRMs, L-T the computational time for constructing all SRMs, and Δ_i the number of data points for each training subset.

The total training data points $\Delta=1681$									RMSE	RMSE	L-T
$R=6$	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7	Δ_8	(Training)	(Testing)	(second)
$\lambda=1.5$	880	880	880	920	920	1000	—	—	0.0180	0.0180	0.7031
$\lambda=3.0$	1320	1320	1360	1600	1360	1600	—	—	0.0189	0.0189	1.0781
$R=7$	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7	Δ_8			
$\lambda=1.5$	760	840	760	880	840	760	760	880	0.0177	0.0178	0.7969
$\lambda=3.0$	1280	1600	1280	1280	1280	1360	1360	1600	0.0179	0.0179	1.4375

SRMs. Therefore, compared with other methods, our method achieved better testing RMSE with less computational time and also had relatively good generalization ability.

Example 3. In this example, the following nonlinear dynamic system was used:

$$y(k+1) = \frac{y(k)y(k-1)[y(k)+2.5]}{1+y^2(k)+y^2(k-1)} + \sin\left(\frac{2\pi k}{50}\right) + v(k)$$

$$y(0) = 0,$$

$$y(1) = 0,$$

$$1 \leq k \leq 100 \quad (31)$$

501 data points are obtained from (31) and $v(t)$ is a Gaussian noise with variance $\text{Var}(v)=0.25$ that is shown in Figure 5. The number of the used test data is 1001. In order to compare the performance of the proposed method with other approach, the results and the curves are given in Table 7 and in

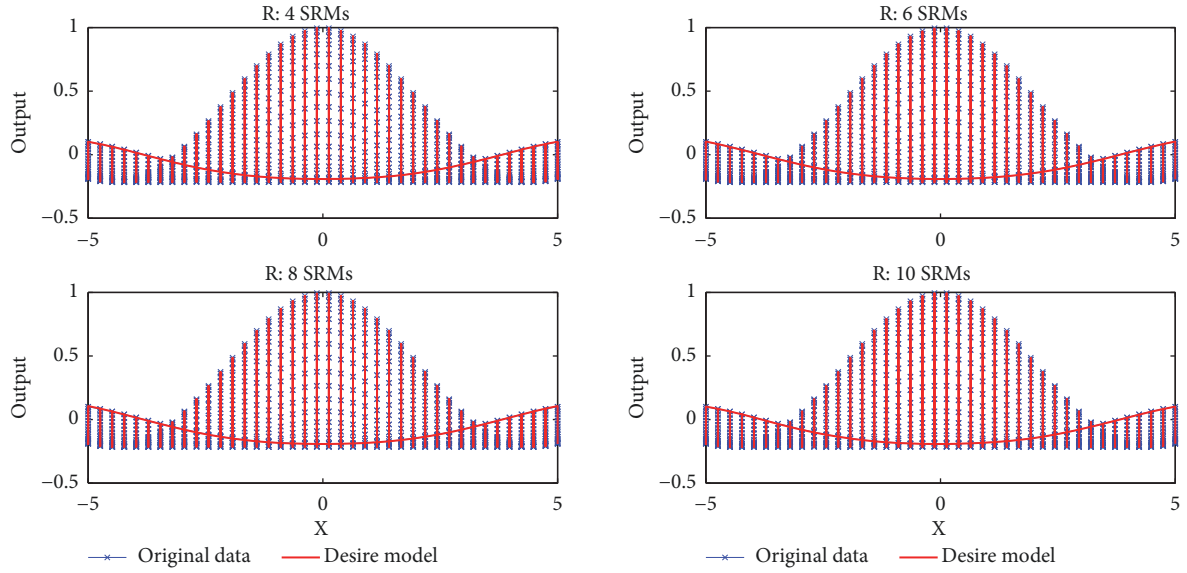


FIGURE 4: Outputs of the proposed FWA-LSSVR approach with $R=4, 6, 8$, and 10 SRMs are shown, respectively.

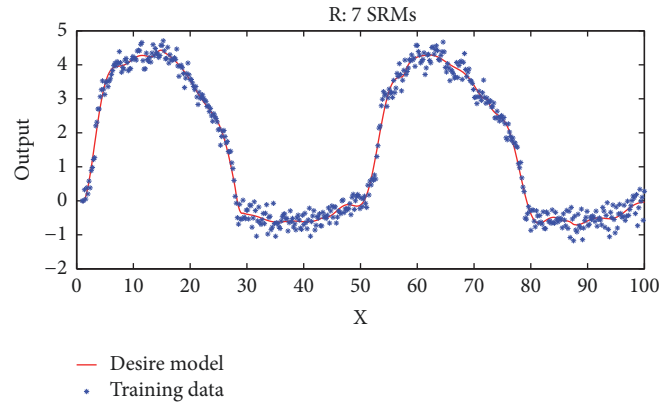


FIGURE 5: Comparison of the proposed model outputs and original output data for Example 3.

Figure 5, respectively. These results show that the proposed method outperforms G-LSSVR, local-LSSVR, and [15], and RMSE in Table 7 indicates the proposed method had the best generalization performance. In addition, the running time of the proposed methods in Table 8 is approximately 10 times shorter comparing with local-LSSVR method at least. In other words, the proposed method leads to a less computational time than local-LSSVR. As shown in Table 8, local-LSSVR needs more computational time. Additionally, the importance of selected different overlap factor λ is also compared by Table 9.

Example 4. In this section, two hundred and ninety-six simulated data generated from a real Box-Jenkins [32] system are applied to the proposed method. These data points consisted of the gas flow rate signal $x(k)$ and the concentration of CO_2 which is described as the output of $y(k)$. Figure 6 shows the training data that include the input signal $x(k)$ and the output signal $y(k)$. To identify the model, we choose $\mathbf{x}_k = [x(k), x(k-1), y(k-1), y(k-2)]$ as

the input variables and $y(k)$ as the output variable. In this example, 5-folds cross-validation is employed to evaluate the performance. According to the cross-validation method, the (training/testing) RMSE and the computational time of the global LSSVR (G-LSSVR) approach, the local-LSSVR (L-LSSVR) approach with $M = 41, 61, 81$, and the proposed approach $R = 4, 8, 12$ SRMs are summarized in Tables 10 and 11. From Tables 10 and 11, there is a little larger RMSE (training) for our technique than that of those approaches, the RMSE (testing) corresponding to our approach is smaller than that of them, and the run time for our technique is smaller than other local modeling approaches but litter bigger than global modeling techniques based on LSSVR. Figure 7 gives the comparisons between the actual output and the predicted output of our techniques. The importance of the selected different overlap factor λ is also compared in Table 12. As shown in Table 12, although the RMSE (training) corresponding to the overlap factor λ of 3.5 is less than that of 2.8, the RMSE (testing) corresponding to the overlap factor λ as 2.8 is less than that of as 3.4. That is to say, generalization

TABLE 7: Comparison results of the proposed method, [15], G-LSSVR and Local-LSSVR with the hyper-parameter set $\{1, 1000\}$, and the overlap factor λ as 1.5 are shown for Example 3.

R: the number of the SRMs	RMSE (Training)		RMSE (Testing)	
	our approach	[15]	our approach	[15]
3-SRMs	0.2523	0.2715	0.2546	0.2832
5-SRMS	0.2531	0.2659	0.2661	0.2762
7-SRMs	0.2287	0.2487	0.2537	0.2645
9-SRMs	0.2298	0.2465	0.2464	0.2583
G-LSSVR	0.2922		0.3185	
Local-LSSVR				
M=21 training points	1.8583		1.8721	
M=41 training points	0.2332		0.3383	
M=61 training points	0.2259		0.3267	
M=81 training points	0.2433		0.3561	

TABLE 8: Comparison results of the proposed method, [15], G-LSSVR and Local-LSSVR with the hyper-parameter set $\{1, 1000\}$, and the overlap factor λ as 1.5 are shown for Example 3.

R:the number of SRMs		3	5	7	9
Proposed approach	L-T(Second)	0.2500	0.2813	0.2656	0.3594
	T-T(Second)	0.3281	1.0938	1.3438	1.5313
G-LSSVR		T-T(Second): 0.1875		L-T (Second): —	
M: Training points		49	81	121	169
Local-LSSVR	L-T(Second)	—	—	—	—
	T-T(Second)	16.5469	16.7813	17.9063	18.1719

TABLE 9: Comparison results of the selected different overlap factor λ for our approach are shown for Example 3, where R represents the number of SRMs, L-T the computational time for constructing all SRMs, and Δ_i the number of data points for each training subset.

The total training data points $\Delta=501$								RMSE	RMSE	L-T
$R=5$	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7	(Training)	(Testing)	(second)
$\lambda=1.1$	106	106	118	118	117	—	—	0.2531	0.0142	0.2813
$\lambda=2.5$	268	183	266	183	267	—	—	0.2756	0.2881	0.3438
$R=7$	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7			
$\lambda=1.1$	83	99	98	97	98	98	83	0.2287	0.2537	0.2656
$\lambda=2.5$	222	147	147	215	222	221	215	0.2818	0.2894	0.4063

TABLE 10: Comparison results of the proposed method, [15], G-LSSVR and Local-LSSVR with the hyper-parameter set $\{25, 1000\}$, and the overlap factor λ as 2.8 are shown for Example 4.

R: the number of the SRMs	RMSE (Training)		RMSE (Testing)	
	our approach	[15]	our approach	[15]
4-SRMs	0.2544	0.1589	0.3304	0.5163
8-SRMS	0.2506	0.1365	0.3321	0.4861
12-SRMs	0.2491	0.1248	0.3338	0.4218
G-LSSVR	0.2480		0.3355	
Local-LSSVR				
M=41 training points	0.3328		0.3805	
M=61 training points	0.3234		0.3666	
M=81 training points	0.3237		0.3366	

TABLE 11: Computational time of the proposed method, G-LSSVR and Local-LSSVR with the hyperparameter set {25, 1000}, and the overlap factor λ as 2.8 are shown for Example 4, where the T-T represents total computational time for building the overall process of the proposed method and L-T represents the computational time for constructing all SRMs.

R:the number of SRMs		4	8	12
Proposed approach	L-T(Second)	1.5938	2.1407	3.3437
	T-T(Second)	2.4375	3.6875	4.8438
G-LSSVR		T-T(Second): 0.8281		L-T (Second): —
M: Training points		41	61	81
Local-LSSVR	L-T(Second)	—	—	—
	T-T(Second)	25.4375	25.7813	26.3906

TABLE 12: Comparison results of the selected different overlap factor λ for our approach are shown for Example 4, where R represents the number of SRMs, L-T the computational time for constructing all SRMs, and Δ_i the number of data points for each training subset.

The total training data points $\Delta=296$, $\lambda=2.8$					RMSE (Training)	RMSE (Testing)	L-T (second)
$R=4$	Δ_1	Δ_2	Δ_3	Δ_4			
CV_1	162	197	194	160	0.2236	0.4429	0.2969
CV_2	155	190	190	161	0.2801	0.2374	0.3438
CV_3	196	191	163	148	0.2883	0.2004	0.3750
CV_4	146	162	196	176	0.2563	0.3282	0.2656
CV_5	197	162	160	194	0.2236	0.4429	0.3125
$(CV_1 + CV_2 + CV_3 + CV_4 + CV_5)/5$					0.2544	0.3304	1.5938
The total training data points $\Delta=296$, $\lambda=3.5$					RMSE (Training)	RMSE (Testing)	L-T (second)
$R=4$	Δ_1	Δ_2	Δ_3	Δ_4			
CV_1	208	193	219	217	0.2151	0.4381	0.4219
CV_2	215	213	197	200	0.2637	0.2728	0.3125
CV_3	218	222	200	192	0.2761	0.2066	0.3906
CV_4	196	183	218	205	0.2465	0.3437	0.3438
CV_5	227	219	208	193	0.2151	0.4381	0.2969
$(CV_1 + CV_2 + CV_3 + CV_4 + CV_5)/5$					0.2433	0.3399	1.7747

performance in relatively smaller value of λ outperforms that of the large value. Additionally, in comparison of the [15], the proposed method only utilizes fewer hyperparameters to construct model, and the overlap factor λ is chosen in relatively smaller value to further reduce more computational time.

5. Conclusion

In this paper, a fuzzy weighted least squares support vector regression (FW-LSSVR) method for nonlinear system modeling have been proposed and illustrated based on the advantages of fuzzy weighted mechanism and some ideas from LSSVR. Considering that each training subset is mutually independent, all SRMs can be constructed simultaneously and our method can largely reduce computational time. As shown in our experimental results, there have better superiorities in calculation time and modeling accuracy for our approach than those approaches such as local or global modeling method. It is noted that, run time for our technique is smaller than other local modeling approaches but litter bigger than global modeling techniques based on LSSVR.

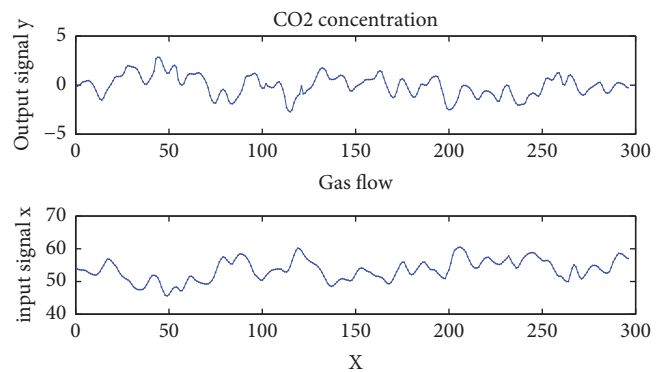


FIGURE 6: The simulated data set obtained from Box and Jenkins is used to validate the proposed method.

Nevertheless, modeling accuracy for our approach has a considerable improvement than other techniques. Furthermore, in comparison with SVR, the proposed method only utilizes fewer hyperparameters to construct model, and the overlap

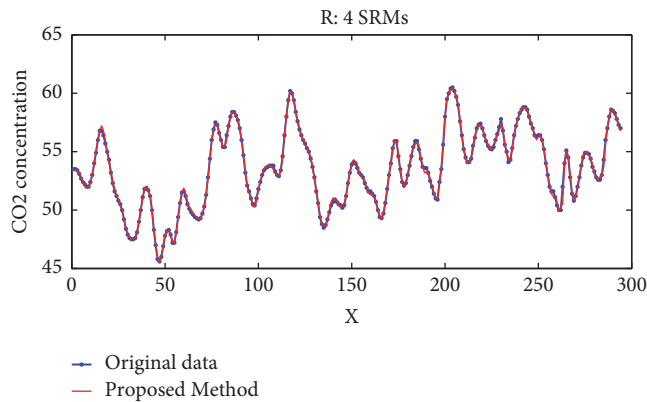


FIGURE 7: Comparison of the proposed model outputs and original output data for Box and Jenkins example.

factor λ is chosen in relatively smaller value than SVR to further reduce more computational time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgments

The research was partially funded by the training program of high level innovative talents of Guizhou ([2017]3, [2017]19), the Guizhou Province Natural Science Foundation in China (KY[2016]018, KY[2016]254), the Science and Technology Research Foundation of Hunan Province (13C333), the Science And Technology Project of Guizhou ([2017]1207, [2018]1179), and the Doctoral Fund of Zunyi Normal University (BS[2015]13, BS[2015]04).

References

- [1] D. Zhang, Z. Zhou, and X. Jia, "Networked fuzzy output feedback control for discrete-time Takagi-Sugeno fuzzy systems with sensor saturation and measurement noise," *Information Sciences*, vol. 457/458, pp. 182–194, 2018.
- [2] M. De Gregorio and M. Giordano, "Background estimation by weightless neural networks," *Pattern Recognition Letters*, vol. 96, pp. 55–65, 2017.
- [3] K. Li, J.-X. Peng, and E.-W. Bai, "A two-stage algorithm for identification of nonlinear dynamic systems," *Automatica*, vol. 42, no. 7, pp. 1189–1197, 2006.
- [4] J. Suykens, L. Lukas, P. Van Dooren, B. De Moor, J. Vandewalle et al., "Least squares support vector machine classifiers: a large scale algorithm in," in *European Conference on Circuit Theory and Design, ECCTD*, vol. 99, pp. 839–842, Citeseer, 1999.
- [5] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [6] T. van Gestel, J. A. K. Suykens, B. Baesens et al., "Benchmarking least squares support vector machine classifiers," *Machine Learning*, vol. 54, no. 1, pp. 5–32, 2004.
- [7] L. Zhang, W.-D. Zhou, P.-C. Chang, J.-W. Yang, and F.-Z. Li, "Iterated time series prediction with multiple support vector regression models," *Neurocomputing*, vol. 99, pp. 411–422, 2013.
- [8] T. Quan, X. Liu, and Q. Liu, "Weighted least squares support vector machine local region method for nonlinear time series prediction," *Applied Soft Computing*, vol. 10, no. 2, pp. 562–566, 2010.
- [9] T. Falck, P. Dreesen, K. de Brabanter, K. Pelckmans, B. de Moor, and J. A. K. Suykens, "Least-squares support vector machines for the identification of wiener-hammerstein systems," *Control Engineering Practice*, vol. 20, no. 11, pp. 1165–1174, 2012.
- [10] L. Bako, G. Mercère, S. Lecoecue, and M. Lovera, "Recursive subspace identification of Hammerstein models based on least squares support vector machines," *IET Control Theory & Applications*, vol. 3, no. 9, pp. 1209–1216, 2009.
- [11] B.-Y. Sun, D.-S. Huang, and H.-T. Fang, "Lidar signal denoising using least-squares support vector machine," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 101–104, 2005.
- [12] B.-Y. Sun, D.-S. Huang, H.-T. Fang, and X.-M. Yang, "A novel robust regression approach of lidar signal based on modified least squares support vector machine," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 5, pp. 715–729, 2005.
- [13] H. B. Zheng, R. J. Liao, S. Grzybowski, and L. J. Yang, "Fault diagnosis of power transformers using multi-class least square support vector machines classifiers with particle swarm optimization," *IET Electric Power Applications*, vol. 5, no. 9, pp. 691–696, 2011.
- [14] K. De Brabanter, J. De Brabanter, J. A. K. Suykens, and B. De Moor, "Approximate confidence and prediction intervals for least squares support vector regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 22, no. 1, pp. 110–120, 2011.
- [15] C.-C. Chuang, "Fuzzy weighted support vector regression with a fuzzy partition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 3, pp. 630–640, 2007.
- [16] A. Ouyang, X. Peng, Y. Liu, L. Fan, and K. Li, "An Efficient Hybrid Algorithm Based on HS and SFLA," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 30, no. 05, p. 1659012, 2016.
- [17] 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 Transactions on Neural Networks and Learning Systems*, vol. 24, no. 2, pp. 207–218, 2013.
- [18] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: an approach to regression analysis by local fitting," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.
- [19] V. Kecman and T. Yang, "Adaptive Local Hyperplane for regression tasks," in *Proceedings of the 2009 International Joint Conference on Neural Networks, IJCNN 2009*, pp. 1566–1570, USA, June 2009.

- [20] A. Ouyang, X. Peng, J. Liu, and A. Sallam, "Hardware/Software Partitioning for Heterogenous MPSoC Considering Communication Overhead," *International Journal of Parallel Programming*, vol. 45, no. 4, pp. 899–922, 2017.
- [21] E. E. Elattar, J. Goulermas, and Q. H. Wu, "Electric load forecasting based on locally weighted support vector regression," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 4, pp. 438–447, 2010.
- [22] H. Yang, K. Huang, I. King, and M. R. Lyu, "Localized support vector regression for time series prediction," *Neurocomputing*, vol. 72, no. 10–12, pp. 2659–2669, 2009.
- [23] H. Jiang and W. He, "Grey relational grade in local support vector regression for financial time series prediction," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2256–2262, 2012.
- [24] F. Wang, "A general learning framework using local and global regularization," *Pattern Recognition*, vol. 43, no. 9, pp. 3120–3129, 2010.
- [25] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [26] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proceedings of the IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, pp. 761–766, San Diego, Calif, USA, January 1979.
- [27] X. Liu, H. Fang, and Z. Chen, "A novel cost function based on decomposing least-square support vector machine for Takagi-Sugeno fuzzy system identification," *IET Control Theory & Applications*, vol. 8, no. 5, pp. 338–347, 2014.
- [28] L. A. Zadeh, "Fuzzy sets," *Information and Computation*, vol. 8, pp. 338–353, 1965.
- [29] J. M. Leski, "TSK-fuzzy modeling based on ϵ -insensitive learning," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 2, pp. 181–193, 2005.
- [30] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 2, no. 3, pp. 267–278, 1994.
- [31] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.
- [32] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, Calif, USA, 1976.

