Neurocomputing 482 (2022) 139–153

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Fuzzy clustering-based neural networks modelling reinforced with the aid of support vectors-based clustering and regularization technique

Hao Huang [a], Sung-Kwun Oh [b,c,*], Chuan-Kun Wu [d,*], Witold Pedrycz [e,f,g]

[a] Department of Computer, The University of Suwon, San 2-2 Wau-ri, Bongdam-eup, Hwaseong-si, Gyeonggi-do 445-743, South Korea
[b] School of Electrical & Electronic Engineering, The University of Suwon, 17 Wauan-gil, Bongdam-eup, Hwaseong-si, Gyeonggi-do 18323, South Korea
[c] Research Center for Big Data and Artificial Intelligence, Linyi University, Linyi 276005, China
[d] School of Information Science and Engineering, Linyi University, 276005 Linyi, China
[e] Department of Electrical & Computer Engineering, University of Alberta, Edmonton T6R 2V4 AB, Canada
[f] Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[g] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

ARTICLE INFO

Article history:
Received 5 February 2021
Revised 22 August 2021
Accepted 14 November 2021
Available online 29 November 2021

Keywords:
FCNNs
Support vectors (SVs)
L_2 norm-regularization
Clustering
Hard C-means
Fuzzy C-means
Statistical analysis

ABSTRACT

In recent years, classical fuzzy clustering-based neural networks (FCNNs) have been successfully applied to regression tasks. The determination of the parameters such as cluster centers of the existing hard c-means (HCM) or fuzzy c-means (FCM), leads to the performance deterioration of the model because of the sensitivity of HCM or FCM to noise and outliers. Moreover, there are also several factors for overfitting and degradation of the robustness of the ensuing model. To solve such problems, two improved clustering techniques and L_2 norm-regularization are considered in the proposed robust fuzzy clustering-based neural networks (RFCNNs) modeling. SVs-based hard c-means (SVs-based HCM) and SVs-based fuzzy c-means (SVs-based FCM) designed with support vectors (SVs) can reduce the interference of uncorrelated data, including noise and outliers, thereby enhancing the main data characteristics effectively, as well as leading to the construction on the improved network model. L_2 norm-regularization can be used to alleviate the degradation of robustness caused by overfitting. In terms of improving the performance of the model through SVs-based HCM or SVs-based FCM, as well as robustness completed through L_2 norm-regularization, the superiority of RFCNNs was verified by experimenting with synthetic data and publicly available data from machine learning datasets.

1. Introduction

In the synergy of fuzzy logic and neural networks, fuzzy neural networks (FNNs) have become one of the most compelling research fields, which mainly refer to the use of neural network structure to realize fuzzy logic inference, therefore, there is no clear physical meaning of weights that can contribute to the interpretation of inference parameters in fuzzy logic in traditional neural networks [1]. FNNs have recently been applied to various areas as one of core technologies of regression, classification, or control [2–3]. Although FNNs have certain advantages, these networks exhibit limitations. In particular, the number of fuzzy sets and the dimensionality of data can seriously affect the computational complexity of FNNs, because the growing number of fuzzy relationships will make calculations very difficult. In order to solve this problem, FCNNs involve various structures such as clustering techniques-based model and fuzzy inference systems, whose impact is analyzed by Oh and Pedrycz with existing clustering techniques and evolutionary algorithms to the fuzzy logic system or model [4–5].

In the conventional FCNNs, the parameters of membership functions standing in the hidden layer are identified by HCM or FCM [6]. The fuzzy relationships can be directly replaced by the membership degree not affected by the number of fuzzy sets and the dimensionality of data, thereby simplifying the computational complexity of FNNs. However, HCM or FCM clustering techniques are based on Euclidean distance [7–11], both of which lead to performance deterioration of FCNNs due to sensitivity to noise and outliers. SVs-based clustering delivers an effective solution to the above problems with the only dataset (SVs-based) being selected from the entire training data to obtain newly determined clusters [12–13]. The preferred model is constructed by realizing the

* Corresponding authors.
E-mail addresses: ohsk@suwon.ac.kr (S.-K. Oh), chuankunwu@lyu.edu.cn (C.-K. Wu).

https://doi.org/10.1016/j.neucom.2021.11.029
0925-2312/

optimal selection of the error tolerance parameter "ε" and its ensuing SVs-selection, with SVs for the effective preprocessing of the training data. As for the learning method, the connection weights of polynomial coefficients are computed with the use of the least square error (LSE) estimation [14–16]. However, the loss error minimizing the sum of squared errors between model output and target output regarded as a quadratic function (e.g., LSE) is not always effective in prediction tasks, because there is no sound tradeoff between variance and bias during the model training [17–18]. To solve these problems, as well as further improve the performance and robustness of this conventional model, we proposed an improved RFCNNs.

In this study, the RFCNNs modeling is proposed by combining both SVs-based clustering techniques and L_2-norm regularization. Of which, SVs-based clustering techniques, such as SVs-based HCM and SVs-based FCM, are designed for coping with performance degradation of FCNNs caused by the interference of uncorrelated data including noise and outliers. For the weak robustness of FCNNs, L_2-norm regularization added as the penalty term to the LSE function can produce a tradeoff between variance and bias by selecting a suitable penalty parameter during the model training, which improves the robustness of the model. In statistics, L_2-norm regularization is an approach to reduce the multicollinearity existing amongst predictor variables by adding a small slack factor to the variables [19–21].

The main motivation of this study can be summarized as follows:

1) To deal with the problem that the performance degradation of the FCNNs caused by the interference of uncorrelated data including noise and outliers due to the use of the conventional HCM or FCM.

2) To deal with the problem that the conventional FCNNs in prediction tasks exhibits limited robustness.

The key issues and advantages of the proposed model are enumerated as follows:

(a) Two improved clustering techniques are proposed, such as SVs-based HCM and SVs-based FCM, which adopted only dataset (SVs-based) selected from the entire training data to newly determined clusters, corresponding to individual nodes of the hidden layer. The cluster centers obtained from SVs-based HCM (or SVs-based FCM) are considered as the connection weights (v_{it}) between hidden and input layers, which can reduce the interference of uncorrelated data including noise and outliers. SVs are selected from the training dataset by satisfying decision conditions guided by ε-insensitive loss function [22–25]. Consequently, the preferred model is constructed by realizing the optimal selection of the error tolerance parameter "ε" and its ensuing SVs-selection, and with SVs for the effective preprocessing of the training data, the effects of SVs-based clustering techniques are analyzed by comparing the existing HCM and FCM.

(b) Considering the robustness of the model, in the conventional FCNNs modeling, there are some factors that will lead to overfitting and degradation of robustness, especially, multicollinearity is one of the factors causing overfitting, which will increase the variance between model output and expected output, as well as resulting in the estimation of unstable coefficients, in which the deviations among coefficients are very large. As a result of these factors, the robustness gradually deteriorates. Additionally, although there is no exact multicollinearity amongst input variables, an appropriate choice of penalty parameter in L_2-norm penalty will result in a reduction of the deviation between the coefficients, known as the contraction estimate [26–28], providing a method for model training through an analysis on the tradeoff between bias and variances for both preventing the degradation of the robustness and reducing the overfitting by excessive training. L_2-norm in regularization can be applicable by adding L_2-norm penalty term to LSE function in this proposed model.

(c) RFCNNs modeling contains three phases: condition, conclusion, and aggregation phases. In the condition phase, input space is divided into subspaces of several level homogeneity by SVs-based clustering. The cluster centers are determined by SVs-based clustering, each of which (subspace) corresponds to individual nodes in the hidden layer. Polynomial weights are dealt with in the conclusion phase. The coefficients of linear polynomial functions activated by the membership degree matrix are adopted as connection weights in the proposed model, which can lead to local regression models located at the condition phase of individual fuzzy rules. In the aggregation phase, output nodes are realized as a sum of products of corresponding polynomial weights and the membership degree. For the learning part, L_2 norm penalty-based LSE function is selected as the objective training function, and coefficients (weights) of the proposed model are optimized by LSE. In the meantime, several important parameters of the proposed model are analyzed during experiments. Compared with the conventional FCNNs, both of the performance and robustness of the proposed RFCNNs have been improved through the analysis of experiment results.

(d) Statistical comparative analysis with other models is performed with Friedman test and Bonferroni-Dunn test, and the superiority of the proposed model is presented.

In the sequel, the main contributions of this study are summarized as follows: RFCNNs modeling is designed by SVs-based clustering and L_2 norm-regularization. 1) Two SVs-based clustering techniques are proposed, such as SVs-based HCM and SVs-based FCM, to reduce the interference of uncorrelated data including noise and outliers, hence improving the performance of RFCNNs. 2) Learning methods realized by adding L_2 norm-regularization can lead to the improvement of the robustness of RFCNNs.

This study is structured as follows: SVs-based clustering techniques are discussed in Section 2. And in Section 3, the architecture of RFCNNs is presented, and then the learning methods realized by L_2 norm-regularization are discussed in Section 4. The approach to constructing the proposed RFCNNs is expressed in Section 5. Experimental results are presented in Section 6. Finally, concluding remarks are drawn in Section 7.

2. Support vectors-based clustering techniques

In this study, SVs-based clustering techniques are employed to reduce the performance degradation caused by conventional clustering techniques-based FCNNs sensitive to uncorrelated data including noise and outliers. In order to select the appropriate data as support vectors in the training dataset, the decision conditions of support vectors should first be given.

At first, the affine function is given as follows (the number of data is n):

$$f(\boldsymbol{x}_j) = \boldsymbol{w}^T\boldsymbol{x}_j + b \tag{1}$$

where $f(\boldsymbol{x}_j) \in \boldsymbol{R}$ is the output value, $\boldsymbol{x}_j \in \boldsymbol{R}^d$ is the input vector, $\boldsymbol{w} \in \boldsymbol{R}^d$ is the weight vector, $b \in \boldsymbol{R}$, $j = 1, 2, ..., n$.

The ε-insensitive loss function is given as follows ($\varepsilon > 0$):

$$|y_j - f(\boldsymbol{x}_j)|_\varepsilon = \begin{cases} 0, if\,|y_j - f(\boldsymbol{x}_j)| \leqslant \varepsilon \\ |y_j - f(\boldsymbol{x}_j)| - \varepsilon, if\,|y_j - f(\boldsymbol{x}_j)| > \varepsilon \end{cases} \tag{2}$$

We introduce slack variables $\xi_j, \xi_j^* \geqslant 0$, and for all data $\boldsymbol{x}_j$, real output y_j, the constraints obtained from (1), (2) can be expressed as:

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}_j + b - y_j \leqslant \varepsilon + \xi_j \\ y_j - \boldsymbol{w}^T\boldsymbol{x}_j - b \leqslant \varepsilon + \xi_j^* \end{cases} \tag{3}$$

Error tolerance parameter "ε" is very important for calculating decision boundaries. Slack variables (ξ_j) are dynamic in nature, which can be adjusted automatically according to the value of "ε". In Fig. 1 (a) and (b), the related description is shown and also explained in detail.

Fig. 1 (a) shows that the data points in the gray area are the correct regression points. However, some data points fall outside that. In order to fully consider all the data points, the slack variable is used to realize the adjustment of dynamic soft boundaries, which is shown in Fig. 1 (b).

According to the theory of support vector regression [29–33], the objective function can be expressed as:

$$\begin{aligned} F = \arg \max_{(\alpha_j,\alpha_j^*,\beta,\beta_j^*)} \min_{(\boldsymbol{w},b,\xi_j,\xi_j^*)} \{ & \tau \sum_{j=1}^{n}\left(\xi_j + \xi_j^*\right) + \tfrac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \\ & + \sum_{j=1}^{n}\alpha_j\left(y_j - \boldsymbol{w}^T\boldsymbol{x}_j - b - \varepsilon - \xi_j\right) \\ & + \sum_{j=1}^{n}\alpha_j^*\left(\boldsymbol{w}^T\boldsymbol{x}_j + b - y_j - \varepsilon - \xi_j^*\right) \\ & + \sum_{j=1}^{n}\left(-\beta_j\xi_j - \beta_j^*\xi_j^*\right)\} \end{aligned} \tag{4}$$

where $\alpha_j, \alpha_j^*, \beta_j, \beta_j^* \geqslant 0$ are the Lagrangian coefficients [34], $\tau(>0)$ controls the error tolerance. (4) should satisfy the following constraints,

$$\begin{cases} \frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{j=1}^{n}\left(\alpha_j - \alpha_j^*\right)\boldsymbol{x}_j = \boldsymbol{0} \\ \frac{\partial L}{\partial b} = \sum_{j=1}^{n}\left(\alpha_j - \alpha_j^*\right) = 0 \\ \frac{\partial L}{\partial \xi_j} = \tau - \alpha_j - \beta_j = 0 \\ \frac{\partial L}{\partial \xi_j^*} = \tau - \alpha_j^* - \beta_j^* = 0 \end{cases} \tag{5}$$

According to (4) and (5), the dual quadratic programming problem is expressed as:

$$\begin{aligned} Q(\alpha_j, \alpha_j^*) = \arg \max_{\left(\alpha_j - \alpha_j^*\right)} \{ & -\tfrac{1}{2}\sum_{j=1}^{n}\sum_{r=1}^{n}(\alpha_j - \alpha_j^*)(\alpha_r - \alpha_r^*)K(\boldsymbol{x}_j, \boldsymbol{x}_r) \\ & + \sum_{j=1}^{n} y_j(\alpha_j - \alpha_j^*) - \varepsilon\sum_{j=1}^{n}\left|\alpha_j - \alpha_j^*\right| \} \end{aligned} \tag{6}$$

Here (6) should satisfy the following constraints,

$$\begin{cases} \sum_{j=1}^{n}\left(\alpha_j - \alpha_j^*\right) = 0 \\ -\tau \leqslant \alpha_j - \alpha_j^* \leqslant \tau \end{cases} \tag{7}$$

The sequential minimum optimization (SMO) algorithm [35–36] is for calculating $\left(\alpha_j - \alpha_j^*\right)$ of (6). Let $\varphi_j = \alpha_j - \alpha_j^*$, and the iteration process can be described in the form below:

$$\begin{aligned} \varphi_2^{new} &= \varphi_2 + \frac{E_1 - E_2 - \varepsilon[sign(\varphi_2) - sign(\varphi_1)]}{K_{11} + K_{22} - 2K_{12}}, \varphi_1^{new} \\ &= \varphi_1^{old} + \varphi_2^{old} - \varphi_2^{new} \end{aligned}$$

where $E_1 = f(\boldsymbol{x}_1) - y_1$, $E_2 = f(\boldsymbol{x}_2) - y_2$, $K_{jr} = e^{-\upsilon\|x_j - x_r\|^2}(j, r = 1, ..., n)$, $sign(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, x < 0 \end{cases}$.

When an iterative process is completed, the new values of multipliers φ_1 and φ_2 satisfying the constraints can be obtained. Then, in the iteration process, when multipliers satisfy all the constraints, the data corresponding to $\varphi_j \neq 0$ are the support vectors, namely, the decision condition of the support vector $\boldsymbol{x}_s$ can be expressed as:

$$\begin{cases} \boldsymbol{x}_s = \boldsymbol{x}_j, \varphi_j \neq 0 \\ \boldsymbol{x}_s \neq \boldsymbol{x}_j, \varphi_j = 0 \end{cases} \tag{9}$$

According to (9), the SVs of all the training data can be determined $SV = \left\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_g \middle| \boldsymbol{x}_s \in \boldsymbol{R}^d, s = 1, 2, ..., g\right\}$, where $\varphi_j = \alpha_j - \alpha_j^* \neq 0$.

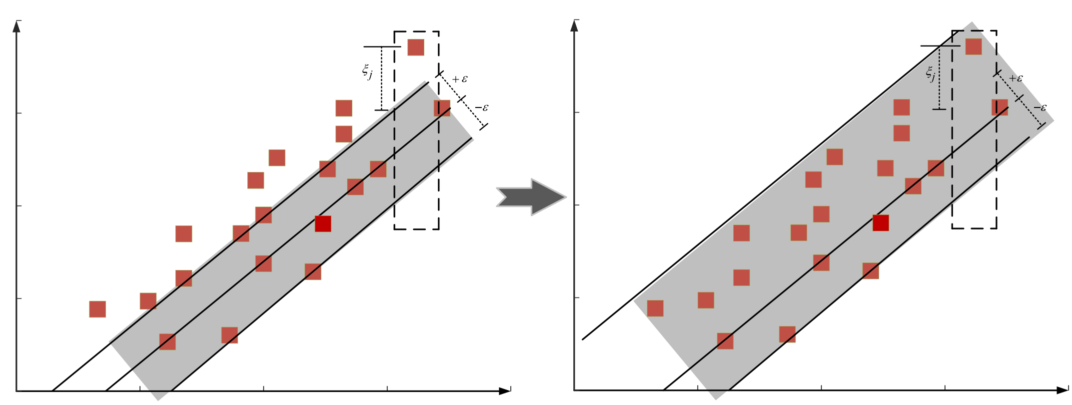
2.1. SVs-based HCM

Given a dataset $SV = \left\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_g | \boldsymbol{x}_s \in \boldsymbol{R}^d, s = 1, 2, ..., g\right\}$, the objective function of SVs-based clustering is to distinguish c subsets in the dataset SV and make similar data $\boldsymbol{x}_s$ located in the same cluster. This function can be defined as:

$$J(\boldsymbol{v}) = \arg \min_{\boldsymbol{u},\boldsymbol{v}} \left\{ \sum_{i=1}^{c}\sum_{s=1}^{g} u_{is}\| \boldsymbol{x}_s - \boldsymbol{v}_i \|^2 \right\} \tag{10}$$

where $u_{is} \in \{0, 1\}$, $\boldsymbol{v} = \left\{\boldsymbol{v}_i | \boldsymbol{v}_i \in \boldsymbol{R}^d, i = 1, 2, ..., c\right\}$, $\boldsymbol{v}_i \in \boldsymbol{v}$ denotes the ith cluster center. $J(\boldsymbol{v})$ represents the sum of squared distances of data to the cluster centers.

If the data x_s belongs to the ith cluster, the element u_{is} should be equal to 1, otherwise, the element takes 0. Once the cluster center $\boldsymbol{v}_i$ has been determined, the value u_{is} can be obtained as follows:



(a) Before the adjustment of decision boundaries (b) After the adjustment of decision boundaries

Fig. 1. Data points and decision boundaries in a regression task.

$$u_{is} = \begin{cases} 1 if \| \boldsymbol{x}_s - \boldsymbol{v}_i \|^2 \leqslant \| \boldsymbol{x}_s - \boldsymbol{v}_r \|^2 \\ 0 else \end{cases} \tag{11}$$

where $i, r = 1, 2, ..., c$ and $s = 1, 2, ..., g$.

The cluster center can be computed as follows:

$$\boldsymbol{v}_i = \frac{\sum_{s=1}^{g} u_{is} \boldsymbol{x}_s}{\sum_{s=1}^{g} u_{is}} \tag{12}$$

2.2. SVs-based FCM

For the same dataset $SV = \left\{ \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_g | \boldsymbol{x}_s \in \boldsymbol{R}^d, s = 1, 2, ..., g \right\}$, the objective function of SVs-based fuzzy clustering can be expressed as follows:

$$J(\boldsymbol{u}, \boldsymbol{v}) = \arg \min_{\boldsymbol{u}, \boldsymbol{v}} \left\{ \sum_{s=1}^{g} \sum_{i=1}^{c} (u_{is})^m \| \boldsymbol{x}_s - \boldsymbol{v}_i \|^2 \right\} \tag{13}$$

where $\boldsymbol{u} = [u_{is}]_{c \times g}$ is the membership matrix, $u_{is} \in [0, 1]$, $\boldsymbol{v} = \left\{ \boldsymbol{v}_i | \boldsymbol{v}_i \in \boldsymbol{R}^d, i = 1, 2, ..., c \right\}$, $\boldsymbol{v}_i \in \boldsymbol{v}$ denotes the ith cluster center, "m" is the fuzzification coefficient. $J(\boldsymbol{u}, \boldsymbol{v})$ is the sum of squared weighted distances between the data and the center of the cluster.

The combination of (13) and constraints can be expressed as:

$$E = \arg \min_{\boldsymbol{u}, \boldsymbol{v}} \left\{ \sum_{s=1}^{g} \sum_{i=1}^{c} (u_{is})^m \| \boldsymbol{x}_s - \boldsymbol{v}_i \|^2 + \sum_{s=1}^{g} \lambda_s \left(1 - \sum_{i=1}^{c} u_{is} \right) \right\} \tag{14}$$

where λ_s is the Lagrange multiplier.

According to (14), the following cluster centers and membership degrees can be calculated as:

$$\boldsymbol{v}_i = \frac{\sum_{s=1}^{g} (u_{is})^m \boldsymbol{x}_s}{\sum_{s=1}^{g} (u_{is})^m} \tag{15}$$

$$u_{is} = \frac{1}{\sum_{r=1}^{c} \left(\frac{\| \boldsymbol{x}_s - \boldsymbol{v}_i \|}{\| \boldsymbol{x}_s - \boldsymbol{v}_r \|} \right)^{2/m-1}} \tag{16}$$

Fig. 2 (a) shows the description of the original dataset. Fig. 2 (b) shows determined clusters from the original dataset (with noise and outliers). The solid line ("-") is the fitted straight line of the original data.

Fig. 3 (a) displays the original dataset. Fig. 3 (b) shows the selected SVs and the uncorrelated data (including noise and outliers). The dashed line ("--") is the decision boundary of the support vector. Fig. 3 (c) shows newly determined clusters from SVs (without noise and outliers). The solid line ("-") is the fitted straight line to the original data.

From the comparison of Fig. 2 (b) and Fig. 3 (c), we can conclude that the newly determined cluster centers from SVs are closer to the fitting straight line.

3. Architecture of robust fuzzy clustering-based neural networks

The proposed RFCNNs is implemented through three processing phases. The condition and conclusion phases are concerned with the formation of fuzzy rules and their ensuing analysis. The aggregation phase is related to a fuzzy inference. In addition, the basic architecture of RFCNNs contained three layers. The connections linking hidden and input layers adopted the cluster centers updated by SVs-based HCM (or SVs-based FCM). As shown in Fig. 4, the prototypes of clusters are represented by the connection weights (v_{ti}). Overall architecture and learning methods of FCNNs and the proposed RFCNNs modeling are shown in Table 1.

First, the prototype can be initialized as follows:

$$v_{ti} = (\max(x_t) - \min(x_t)) \times [0, 1]_i + \min(x_t) \tag{17}$$

where $[0, 1]$ represents a random value between 0 and 1.

3.1. RFCNNs implemented by SVs-based HCM

The values of (17) are updated through iterative learning realized by SVs-based HCM clustering,

$$v_{ti} = \frac{\sum_{s=1}^{g} u_{is} \boldsymbol{x}_{ts}}{\sum_{s=1}^{g} u_{is}} \tag{18}$$

where g is the number of SVs, $t = 1, 2, ..., d$ and $i = 1, 2, ..., c$.

The Euclidean distance "$\| . \|$" between the connection weight v_{ti} and inputs is applied to calculate membership degrees (activation levels),

$$a_{ij}(\boldsymbol{x}) = \frac{1}{\sum_{r=1}^{c} (d_{ij}/d_{rj})^2} \quad d_{ij} = \| \boldsymbol{x}_j - \boldsymbol{v}_i \| \tag{19}$$

where d_{ij} $(j = 1, 2, ..., n)$ is the Euclidean distance, and $u_{ij}(\boldsymbol{x})$ $(j = 1, 2, ..., n)$ is the membership degrees to certain clusters as the membership function (activation function).

Different from the numerical weights used in the weights of traditional neural networks, the proposed RFCNNs adopt polynomial formula including inputs as the connection weights,

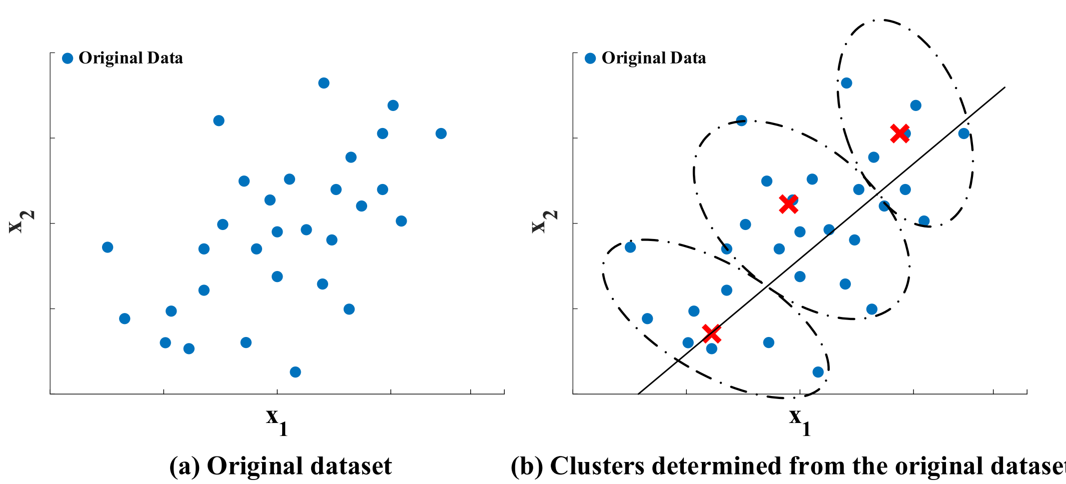


Fig. 2. An example description of FCM.

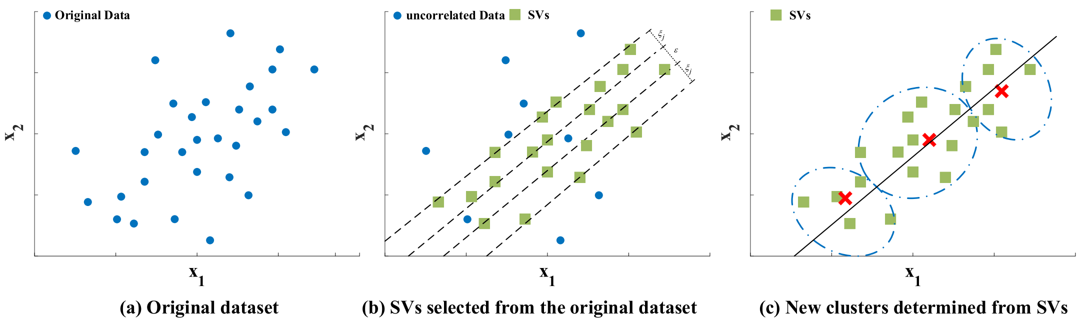


Fig. 3. An example description of support vectors-based FCM.

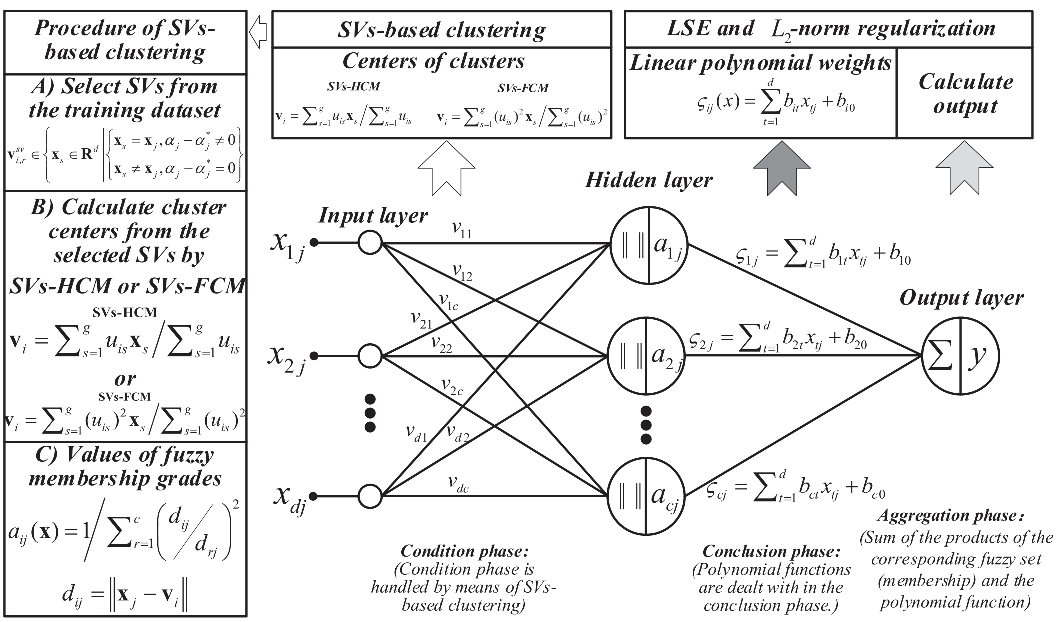


Fig. 4. Overall structure and core algorithmic details of the proposed RFCNNs.

Table 1
Architecture and learning methods of the existing FCNNs and the proposed RFCNNs.

Regression model		This study FCNNs	(Proposed) RBFNNs Model SH	Model SF
Phase of model				
a	Input variables	$x_1, x_2, \ldots, x_n$	$x_1, x_2, \ldots, x_n$	$x_1, x_2, \ldots, x_n$
	Order	Constant: v_{it}	Constant: v_{it}	Constant: v_{it}
	Learning method	FCM	SVs-based HCM	SVs-based FCM
	Objective function	$J = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} \parallel \boldsymbol{x}_j - \boldsymbol{v}_i \parallel^2$	$F = \tau\sum_{j=1}^{n}(\xi_j + \xi_j^{+}) + 1/2w^Tw + \sum_{j=1}^{n}\alpha_j(y_j - w^Tx_j - b - \varepsilon - \xi_j)$ $+\sum_{j=1}^{n}\alpha_j^{*}(w^Tx_j + b - y_j - \varepsilon - \xi_j^{*}) + \sum_{j=1}^{n}(-\beta_j\xi_j - \beta_j^{*}\xi_j^{*})$	
			$J = \sum_{i=1}^{c}\sum_{s=1}^{g} a_{is} \parallel \boldsymbol{x}_s - \boldsymbol{v}_i \parallel^2$	$J = \sum_{i=1}^{c}\sum_{s=1}^{g} a_{is}^{m} \parallel \boldsymbol{x}_s - \boldsymbol{v}_i \parallel^2$
	Output	$u_i = \dfrac{1}{\sum_{r=1}^{c}\left(\frac{d(\boldsymbol{x}_j,\boldsymbol{v}_i)}{d(\boldsymbol{x}_j,\boldsymbol{v}_r)}\right)^{2/(m-1)}}$	$\boldsymbol{v}_{i,r}^{sv} \in \left\{\boldsymbol{x}_s \in \boldsymbol{R}^d \middle\| \begin{cases}\boldsymbol{x}_s = \boldsymbol{x}_j, \alpha_j - \alpha_j^{*} \neq 0\\ \boldsymbol{x}_s \neq \boldsymbol{x}_j, \alpha_j - \alpha_j^{*} = 0\end{cases}\right\}$	
			$a_i = \begin{cases} 1 \text{ if} \parallel \boldsymbol{x}_j - \boldsymbol{v}_i^{sv} \parallel^2 \leqslant \parallel \boldsymbol{x}_j - \boldsymbol{v}_r^{sv} \parallel^2 \\ 0 \text{ else}\end{cases}$	$a_i = 1/\sum_{r=1}^{c}\left(\parallel x_j - v_i^{sv} \parallel / \parallel x_j - v_r^{sv} \parallel\right)^2$
b	Order	Linear: $b_{i0} + \sum_{t=1}^{d} b_{it}x_{tj}$	Linear: $b_{i0} + \sum_{t=1}^{d} b_{it}x_{tj}$	Linear: $b_{i0} + \sum_{t=1}^{d} b_{it}x_{tj}$
	Learning method	LSE	LSE and L_2 norm-regularization	LSE and L_2 norm-regularization
	Cost function	$Loss = \frac{1}{2}(\boldsymbol{Xw} - \boldsymbol{y})^T(\boldsymbol{Xw} - \boldsymbol{y})$	$Loss = \frac{1}{2}(\boldsymbol{Xw} - \boldsymbol{y})^T(\boldsymbol{Xw} - \boldsymbol{y}) + \lambda\boldsymbol{w}^T\boldsymbol{w}$	$Loss = \frac{1}{2}(\boldsymbol{Xw} - \boldsymbol{y})^T(\boldsymbol{Xw} - \boldsymbol{y}) + \lambda\boldsymbol{w}^T\boldsymbol{w}$
	Output	$\boldsymbol{w} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$	$\boldsymbol{w} = \left(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$	$\boldsymbol{w} = \left(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$
c	Output	$z_j = \sum_{i=1}^{c}\sum_{t=1}^{d} u_{ij}(b_{it}x_{ti} + b_{i0})$	$z_j = \sum_{i=1}^{c}\sum_{t=1}^{d} a_{ij}(b_{it}x_{ti} + b_{i0})$	$z_j = \sum_{i=1}^{c}\sum_{t=1}^{d} a_{ij}(b_{it}x_{ti} + b_{i0})$

a: Condition phase; **b**: Conclusion phase; **c**: Aggregation phase; FCM: fuzzy c-means clustering; SVs-based HCM: support vectors-based hard c-means; SVs-based FCM: support vectors-based fuzzy c-means; LSE: least square error estimation; *Model SH*: RFCNNs realized by SVs-based HCM; *Model SF*: RFCNNs realized by SVs-based FCM.

$$\varsigma_{ij}(\boldsymbol{x}) = \sum_{t=1}^{d} b_{it}x_{tj} + b_{i0} \tag{20}$$

The node in the output layer denoted by "$\sum$" is realized as a summation operation of products of membership degrees and their correlated linear polynomial functions. And the output of the proposed model is shown as follows:

$$z_j = \sum_{i=1}^{c} a_{ij}\varsigma_{ij}(\boldsymbol{x}) = \sum_{i=1}^{c}\sum_{t=1}^{d} a_{ij}(b_{it}x_{tj} + b_{i0}) \tag{21}$$

3.2. RFCNNs implemented by SVs-based FCM

Different from the model described above, these values of (17) are updated through interactive learning realized by SVs-based FCM clustering,

$$\boldsymbol{v}_{ti}^{*} = \frac{\sum_{s=1}^{g}(u_{is}^{*})^{2}\boldsymbol{x}_{ts}}{\sum_{s=1}^{g}(u_{is}^{*})^{2}} \tag{22}$$

The membership degrees (activation levels) can be expressed as:

$$a_{ij}^{*}(\boldsymbol{x}) = \frac{1}{\sum_{r=1}^{c}\left(d_{ij}^{*}/d_{rj}^{*}\right)^{2}} d_{ij}^{*} = \| \boldsymbol{x}_j - \boldsymbol{v}_i^{*} \| \tag{23}$$

According to (20), we calculate the output and obtain the following expression:

$$z_j = \sum_{i=1}^{c} a_{ij}^{*}\varsigma_{ij}(\boldsymbol{x}) = \sum_{i=1}^{c}\sum_{t=1}^{d} a_{ij}^{*}(b_{it}x_{tj} + b_{i0}) \tag{24}$$

4. Learning methods of robust fuzzy clustering-based neural networks

The proposed RFCNNs contains two learning mechanisms. The first is SVs-based clustering techniques, such as SVs-based HCM and SVs-based FCM, used to form hidden layers. When forming an aspect of the performance of the regression model, it is important to determine the center of membership function (activation function) corresponding to each node with SVs-based clustering techniques. In addition, there are no specific criteria for selecting the center of membership function (activation function), so SVs-based clustering is effective. The second is LSE to calculate the connection weights between hidden and output layers. L_2-norm in regularization is applied for estimating shrinkage coefficients, which can reduce the degradation of robustness caused by possible overfitting.

In this study, LSE is used to calculate connection weights between hidden and output layers. In order to estimate the connection weights of the proposed model, the learning objective function is expressed as follows:

$$Loss = \arg\min_{\boldsymbol{w}}\left\{\frac{1}{2}[\boldsymbol{Xw}-\boldsymbol{y}]^{T}[\boldsymbol{Xw}-\boldsymbol{y}] + \lambda\boldsymbol{w}^{T}\boldsymbol{w}\right\} \tag{25}$$

where λ is regularization parameter,

$$X = \begin{bmatrix} a_{11} & \cdots & a_{c1} & a_{11}x_{11} & \cdots & a_{c1}x_{11} & \cdots & a_{11}x_{d1} & \cdots & a_{c1}x_{d1} \\ a_{12} & \cdots & a_{c2} & a_{12}x_{12} & \cdots & a_{c2}x_{12} & \cdots & a_{12}x_{d2} & \cdots & a_{c2}x_{d2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1n} & \cdots & a_{cn} & a_{1n}x_{1n} & \cdots & a_{cn}x_{1n} & \cdots & a_{1n}x_{dn} & \cdots & a_{cn}x_{dn} \end{bmatrix}_{n\times(c\cdot(1+d))}$$

$$\boldsymbol{w} = [b_{10} \quad \cdots \quad b_{c0} \quad b_{11} \quad \cdots \quad b_{c1} \quad \cdots \quad b_{1d} \quad \cdots \quad b_{cd}]_{1\times(c\cdot(1+d))}^{T}$$

$$\boldsymbol{y} = [z_1 \quad z_2 \quad \ldots \quad z_n]_{1\times n}^{T}$$

The connection weights $\boldsymbol{w}$ can be obtained from equation (25) as follows:

$$w = \left(X^{T}X + \lambda I\right)^{-1}X^{T}y \tag{26}$$

where

$$\lambda I = \begin{bmatrix} \lambda & 0 & 0 & \cdots & 0 \\ 0 & \lambda & 0 & \cdots & 0 \\ 0 & 0 & \lambda & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & \lambda \end{bmatrix}_{(c\cdot(1+d))\times(c\cdot(1+d))}$$

5. Framework designing of robust fuzzy clustering-based neural networks

Referring to Fig. 5, the framework designing for RFCNNs includes several steps as follows:

(1) Form the training and testing data, and fix the structure of RFCNNs.

The dataset is divided into the training and testing datasets through the ten-fold cross-validation. These structural factors of the proposed RFCNNs, such as the number of clusters (nodes), SV-parameter (ε), and regularization-parameter (λ), are predetermined before training the connection weights of the model.

(2) The connection weights (v_{ti}) between hidden and input layers are updated.

2–1) Number of clusters is determined in the SVs-based clustering process.

Number of clusters should be equal to the number of nodes in the hidden layer. And the number of clusters (nodes) for SVs-based clustering is chosen by thinking about the structure of the data, as well as the complexity and the performance of the model.

2–2) The centers (v_{ti}) of clusters are initialized and updated.

The centers (v_{ti}) are regarded as the connection weights between input and hidden layers. In the hidden layer, the membership function (19) (or (23)) is defined as the activation function and the Euclidean distance between the connection and input as the input of the activation function.

(3) The connection weights ($\boldsymbol{w}$) between output and hidden layers are calculated.

3–1) The partition matrix is computed.

The output of each node in the hidden layer is computed by (19) (or (23)).

3–2) The connection weights between output and hidden layers are calculated by LSE with L_2 norm-regularization.

6. Experimental studies

The experimental studies are included to demonstrate the design of two SVs-based clustering techniques in RFCNNs modeling, to enhance the output performance effectively when FCNNs modeling is used as a comparative model. The robustness of RFCNNs is examined with the testing data adding noises. Moreover, RFCNNs designed with SVs-based clustering and L_2 norm-regularization has better regression performance compared to other models, such as SVR, Random Tree, Linear Regression, IBK, and so on. Statistical analysis is performed with Friedman test and Bonferroni-Dunn test, with the regression performance measured by the root mean squared error (RMSE). The experiments reported here involved in synthetic dataset and Machine Learning datasets. PI and E_PI represent the model performance of training data and testing data. The initial parameter settings of RFCNNs are described in Table 2.

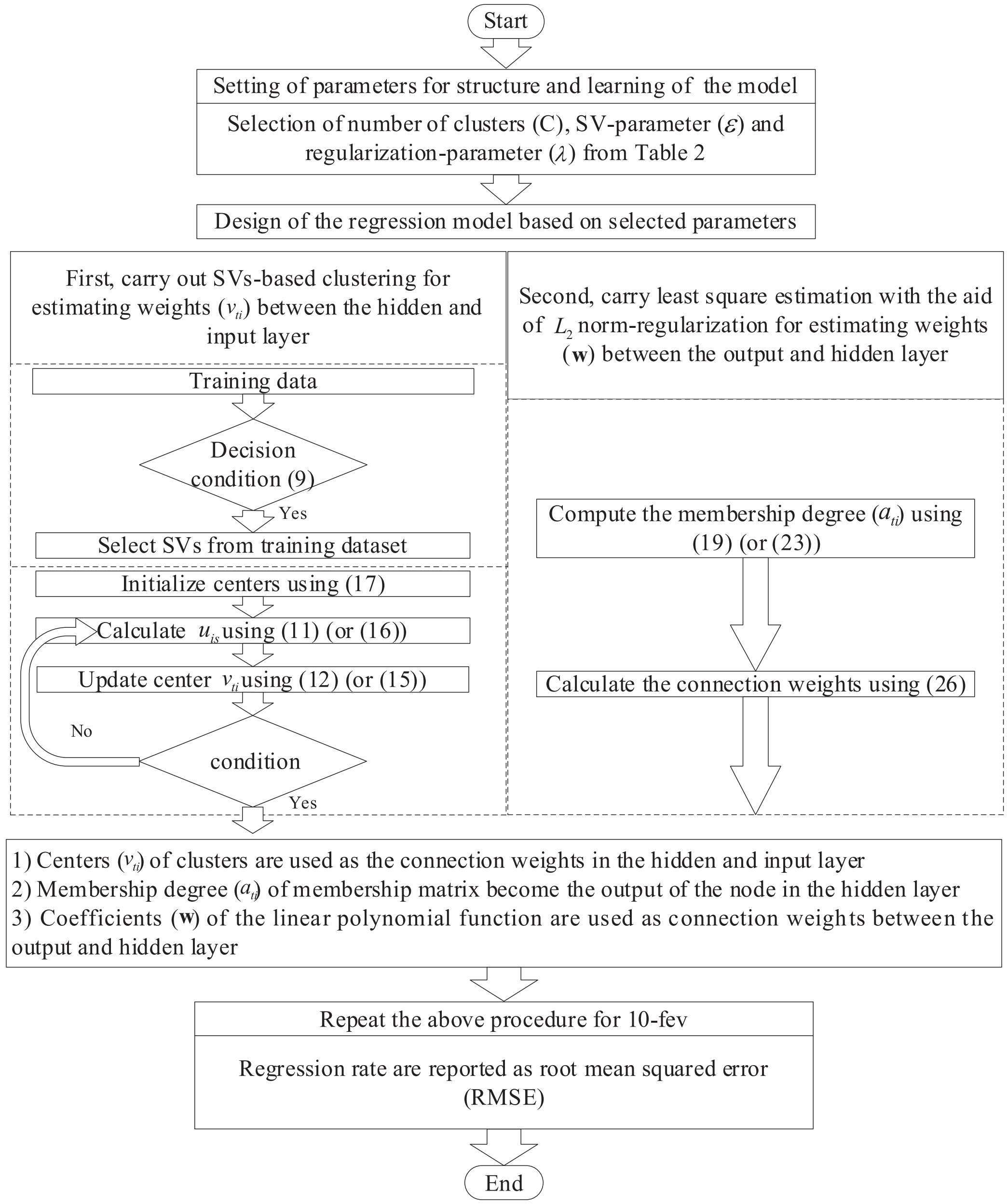


Fig. 5. Overall algorithm flow for each layer of the proposed RFCNNs modeling architecture.

6.1. Synthetic dataset

A two-dimensional synthetic dataset (as shown in Fig. 6) is proposed, which contains 500 data (training data: 450).

As shown in Fig. 7, the shape of membership function is directly affected by the value of fuzzification coefficient "m". In view of this observation, we expect that selecting its value equal to 2 can be a sound choice.

This synthetic dataset is adopted to perform two types of experiments of SVs-based HCM and SVs-based FCM. The description of structure and procedures of the proposed RFCNNs for an example case with this two-dimensional synthetic dataset is shown in Fig. 10. The clustering distribution of SVs-based HCM is shown in Fig. 8. And the clustering distribution of SVs-based FCM is shown in Fig. 9. In Figs. 8 and 9, according to the increase of clusters, the number of SVs extracted from the entire training data is shown by carrying out two types of clustering techniques, such as SVs-HCM and SVs-FCM.

As shown the entire training data in Fig. 8, Fig. 8 (a-1), (b-1), and (c-1) show the different cluster centers corresponding to the number of clusters (3,6,9) calculated by the HCM clustering. Fig. 8 (a-3), (b-3), and (c-3) show the different cluster centers corresponding to the number of clusters (3,6,9) calculated by SVs-based HCM clustering. As we can see from Fig. 8 (a-3), (b-3), and (c-3), the proposed clustering techniques can adjust the cluster centers to reduce the influence of noise and outliers on the clustering results. The uncorrelated training data including noise and outliers are shown by "+" in Fig. 8 (a-2), (b-2), and (c-2). As shown in Table 3, the results of RFCNNs realized by SVs based-HCM clustering are better than that of FCNNs realized by existing HCM clustering.

In entire training data, Fig. 9 (a-1), (b-1), and (c-1) show the different cluster centers corresponding to the number of clusters

Table 2
Initial parameter settings in RFCNNs.

Layer	Parameters	Conventional FCNNs	Proposed RFCNNs
Input	SVs-parameter (ε)	N/A	From 0.001 to 2.000
Hidden	No. of clusters (C)	From 2 to 10	From 2 to 10
	fuzzification coefficient (m)	2	2
	Maximum iteration of clustering	100	100
Output	Regularization parameter (λ)	N/A	From 0.1 to 1
Cross validation	*k*-fcv	10	10

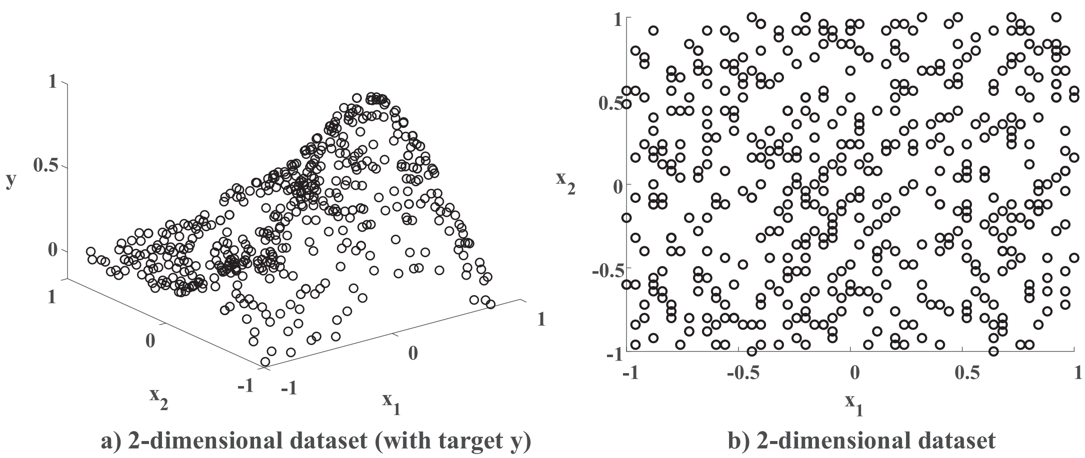


Fig. 6. Distribution of entire Synthetic data.

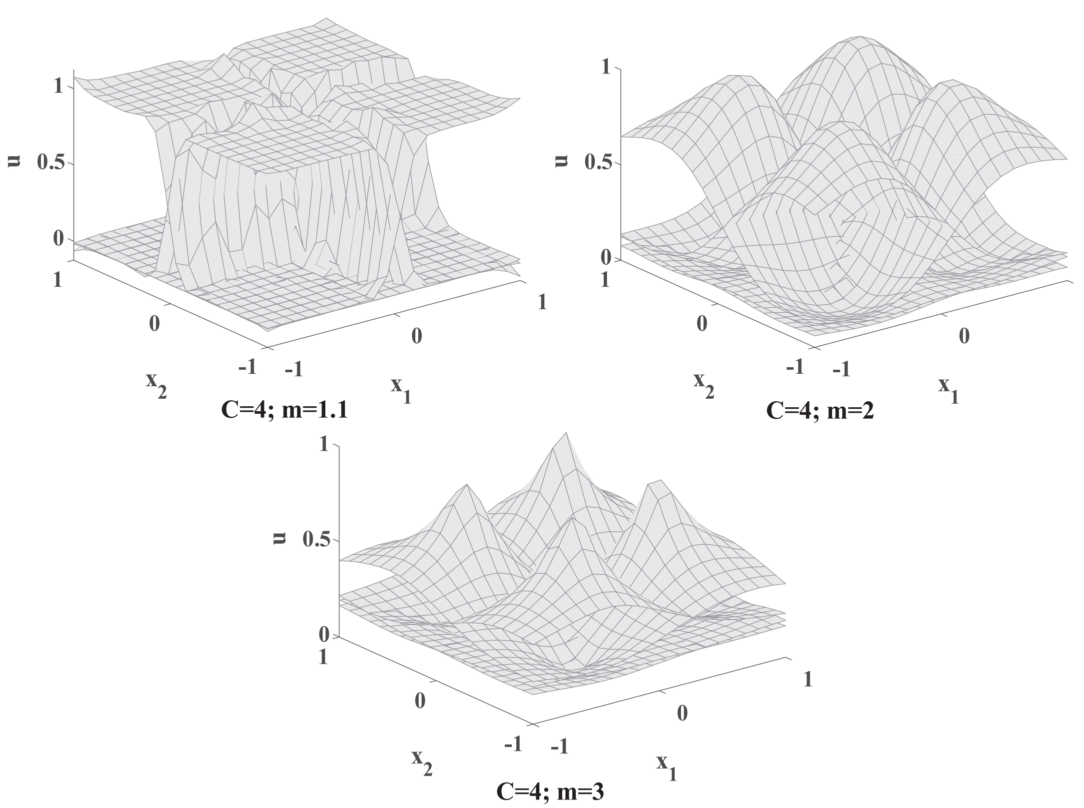


Fig. 7. Shape of membership function obtained for selected values of the fuzzification coefficient.

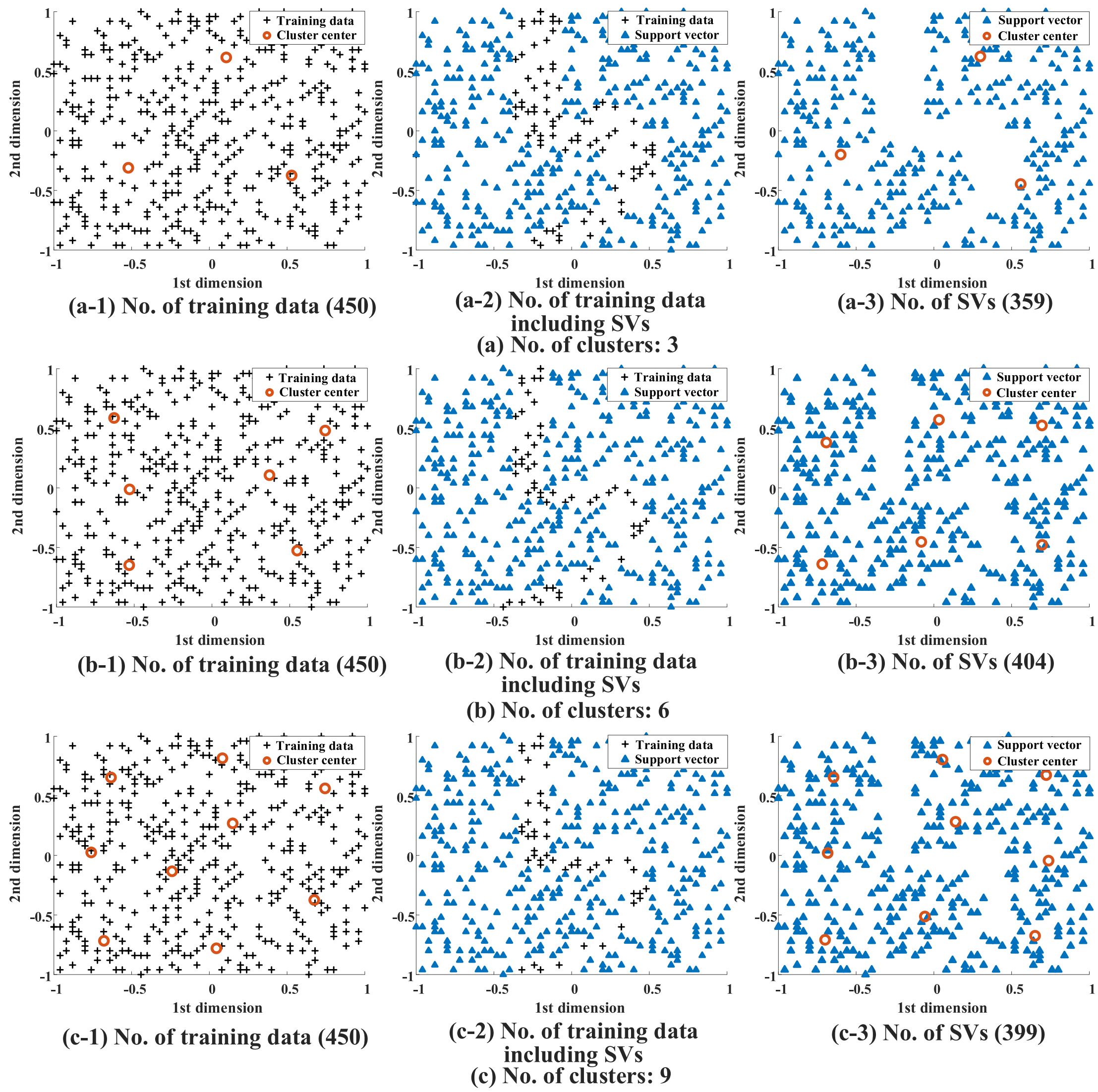


Fig. 8. Distribution of entire synthetic training data, SVs, and cluster centers (No. of clusters, a: 3, b: 6, c: 9); (a-1), (b-1), and (c-1) show cluster centers calculated by HCM; (a-2), (b-2), and (c-2) show SVs extracted by the decision conditions of SVs; (a-3), (b-3), and (c-3) show SVs and cluster centers calculated by SVs-based HCM.

(3,6,9) calculated by FCM clustering, and (a-3), (b-3), and (c-3) show the different cluster centers corresponding to the number of clusters (3,6,9) calculated by SVs-based FCM clustering. As we can see from Fig. 9 (a-3), (b-3), and (c-3), the proposed clustering techniques can adjust the cluster centers to reduce the influence of noise and outliers on the clustering results. The uncorrelated training data including noise and outliers are shown as symbol " +" in Fig. 9 (a-2) (b-2) (c-2). As shown in Table 4, the results of RFCNNs realized by SV- based FCM clustering are better than that of FCNNs realized by the existing FCM clustering.

6.2. Machine learning dataset

In these experiments, we consider several machine learning datasets (https://cml.ics.uci.edu/), we summarized the pertinent details of datasets, such as the number of input variables and the number of data, as shown in Table 5.

The experiment is performed to obtain the results of RFCNNs and to compare with FCNNs about regression performance and robustness. Furthermore, through the Friedman test and Bonferroni-Dunn test, it is shown that RFCNNs also has better performance than any other models.

In this study, the value of the parameter 'ε' is critical. According to the change of the value for the SV-parameter 'ε', we considered several experimental cases for the MPG dataset in RFCNNs designed with SVs-based HCM. (Number of clusters is 6; Number of training data is 353)

Case 1, if ε = 0.2, the number of SVs is 308, and RMSE for testing data is 2.86 ± 0.45.

Case 2, if ε = 0.4, the number of SVs is 264, and RMSE for testing data is 2.94 ± 0.55.

Case 3, if ε = 0.6, the number of SVs is 214, and RMSE for testing data is 3.02 ± 0.65.

Case 4, if ε = 0.8, the number of SVs is 172, and RMSE for testing data is 2.92 ± 0.56.

Case 5, if ε = 1.0, the number of SVs is 125, and RMSE for testing data is 2.71 ± 0.36.

Case 6, if ε = 1.2, the number of SVs is 103, and RMSE for testing data is 2.70 ± 0.36. (Preferred case)

Case 7, if ε = 1.4, the number of SVs is 78, and RMSE for testing data is 3.01 ± 0.97.

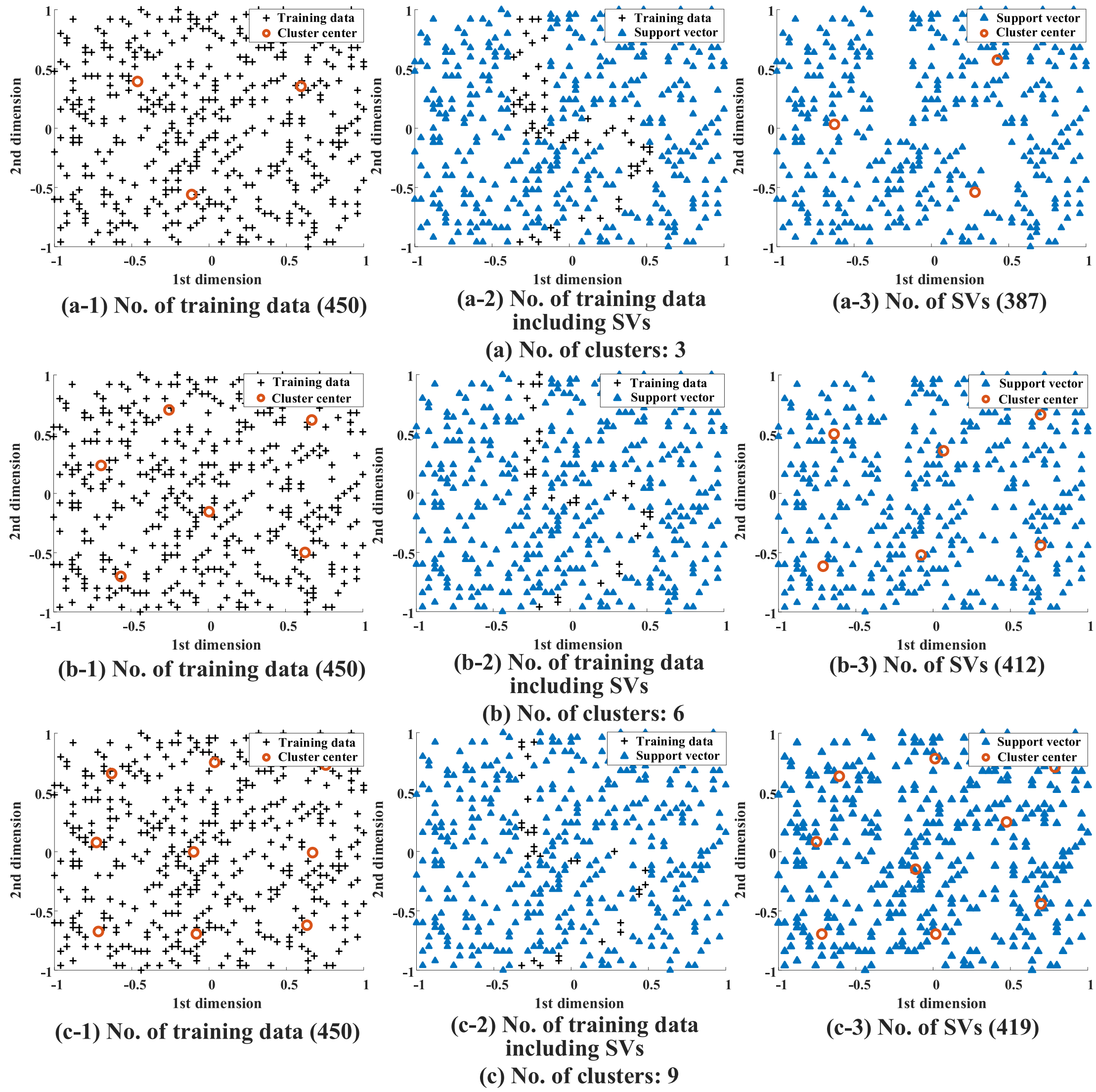


Fig. 9. Distribution of entire Synthetic training data, SVs, and cluster centers (No. of clusters, a: 3, b: 6, c: 9); (a-1), (b-1), and (c-1) show cluster centers calculated by FCM; (a-2), (b-2), and (c-2) show SVs extracted by the decision conditions of SVs; (a-3), (b-3), and (c-3) show SVs and cluster centers calculated by SVs-based FCM.

Table 6, Table 7 and Table 8 show the comparison of regression performance between the existing FCNNs (Model H or Model F) and the proposed RFCNNs (Model SH or Model SF) in the MPG, NOx, and Boston Housing datasets. The experiment results show that the proposed RFCNNs modeling has better regression performance than conventional FCNNs. Table 10 shows the comparison of RMSE for each dataset between the existing FCNNs (Model H or Model F) and the proposed RFCNNs (Model SH or Model SF).

Table 9 lists the optimized parameter settings including C (Number of clusters), ε (SVs-parameter), and λ (regularization parameter).

The robustness of the model is tested by machine learning datasets such as noise. According to Table 11 (a) and (b), better robustness in the presence of noise is obtained for the proposed RFCNNs (Model SH and Model SF).

In the regression model designed through the shrinkage estimators by L_2 norm-regularization, the loss function on the testing dataset is more superb than that without L_2 norm-regularization. Fig. 11 shows the comparison of RMSE between the proposed RFCNNs and conventional FCNNs. Obviously, RFCNNs modeling has better performance when predicting the dataset with added noise.

In this study, the value of the parameter ‘λ’ is critical. According to the change of this regularization parameter ‘λ’, we considered several experimental cases for MPG dataset in RFCNNs designed with SVs-based HCM (or SVs-based FCM) and L_2 norm-regularization. Fig. 12 (a) and (b) show the change for the RMSE according to the increase of the regularization parameter (λ). When “λ = 0”, L_2 norm-regularization does not work, which leads to the high values of RMSE. This indicates that the model shows weak robustness. When “λ = 0.1”, (a) and (b) show the least value of RMSE, thus indicating that the RFCNNs has better robustness than the model without L_2 norm-regularization. Moreover, from the perspective of the influence of different values of “λ” on the value of RMSE, “λ = 0.1” is the more appropriate choice in this study.

For a fair comparison, repeated observations of the same subjects are compared with the use of the Friedman test, to check whether the measured average ranks are significantly different from the mean rank. The numbers in parentheses represent the rank.

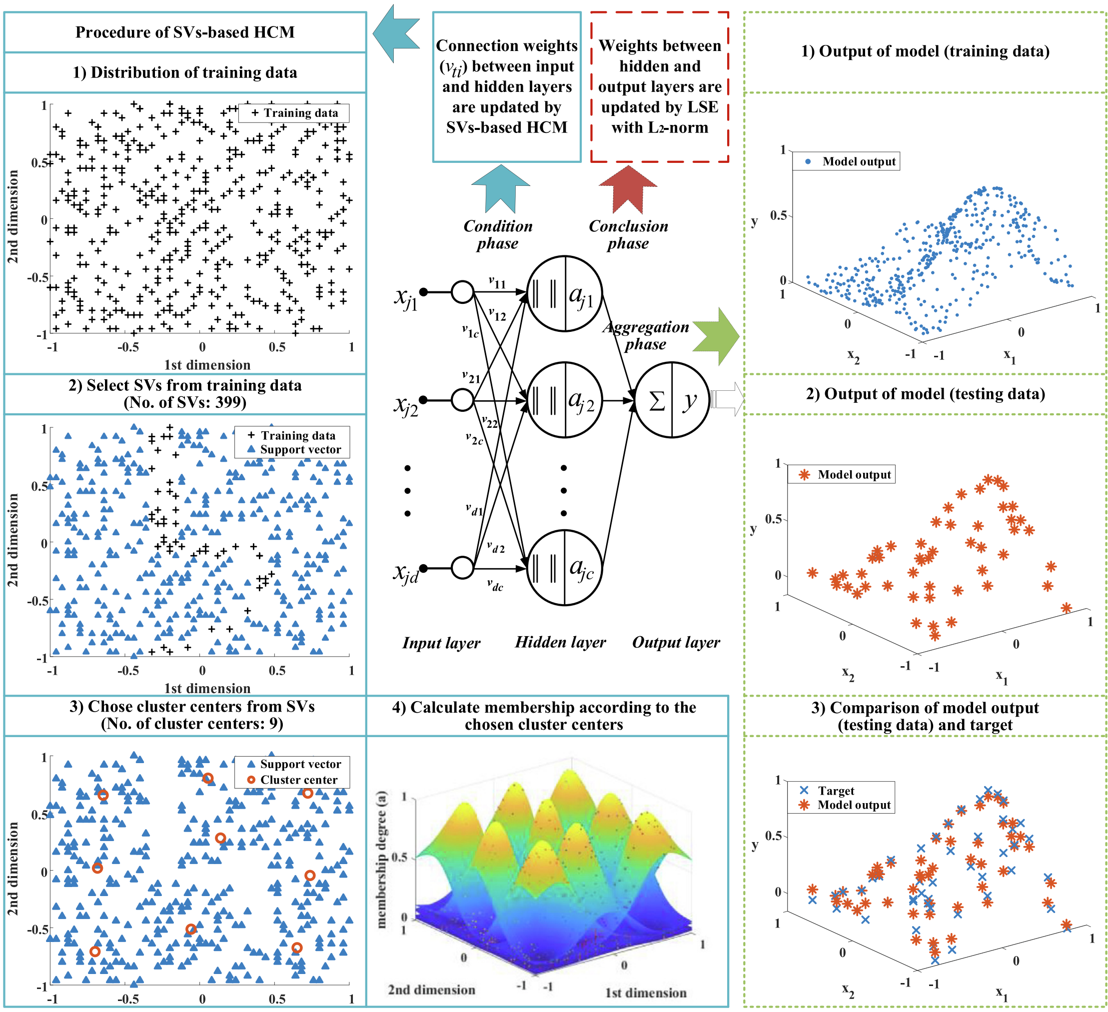


Fig. 10. Description of procedure and functionality of the proposed RFCNNs (designed by SVs-based HCM) for an example case on the two-dimensional synthetic dataset.

Table 3
Simulation results of RFCNNs (realized by SVs-based HCM) and FCNNs (realized by HCM). (ε = 0.05, λ = 0.1).

C	Model-H		Model-SH	
	PI	E_PI	PI	E_PI
3	0.163 ± 0.020	0.167 ± 0.026	0.136 ± 0.019	0.142 ± 0.030
6	0.074 ± 0.010	0.078 ± 0.013	0.072 ± 0.010	0.076 ± 0.011
9	0.055 ± 0.003	0.056 ± 0.010	0.047 ± 0.003	**0.050** ± **0.006**

Model-H: FCNNs realized by HCM; Model-SH: RFCNNs realized by SVs-based HCM.

Table 4
Simulation results of RFCNNs (realized by SVs-based FCM) and FCNNs (realized by FCM). (ε = 0.07, λ = 0.1).

C	Model-F		Model-SF	
	PI	E_PI	PI	E_PI
3	0.161 ± 0.002	0.166 ± 0.020	0.161 ± 0.003	0.162 ± 0.025
6	0.106 ± 0.007	0.113 ± 0.019	0.105 ± 0.007	0.107 ± 0.016
9	0.068 ± 0.006	0.074 ± 0.011	0.066 ± 0.001	**0.069** ± **0.008**

Model-F: FCNNs realized by FCM; Model-SF: RFCNNs realized by SVs-based FCM.

Friedman test assumes that all algorithms should be equivalent, so their ranks are equal. From Friedman test based on the average rank in Table 12, the null- hypothesis is rejected (F_F = 4.338 > F (7,70) for 0.05 = 2.143). Therefore, a post-hoc test is performed to check whether the proposed RFCNNs are statistically superior to other regression models. Among several methods, Bonferroni-Dunn test is considered in the post-hoc test for controlling the family-wise test error rate in the multiple hypothesis testing.

Bonferroni-Dunn test is applicable if only all regression models are compared to the target model without comparison between

Table 5
Machine learning datasets used in the study.

Dataset (ABB.)	Dimensionality	Number of entire data
MPG (MPG)	7	392
NOx (NOX)	5	260
Boston Housing (BOH)	13	509
MIS (MIS)	10	390
Concrete (CON)	8	1030
Friedman (FRI)	5	1200
Wizmir (WIZ)	9	1461
Pyrimidine (PYR)	26	74
Friedman_2 (FR2)	25	250
Winequality_red (WIR)	11	1599
Winequality_white (WIW)	11	4898

Table 6
Simulation results of FCNNs and (proposed) RFCNNs for MPG dataset. (A) Model H (FCNNs realized by HCM) and Model SH (RFCNNs realized by SVs-based HCM) (ε = 1.200, λ = 0.1).

C	Model H		Model SH	
	PI	E_PI	PI	E_PI
2	2.76 ± 0.03	3.18 ± 0.26	2.78 ± 0.03	2.82 ± 0.41
3	2.74 ± 0.04	3.17 ± 0.37	2.72 ± 0.04	2.83 ± 0.40
4	2.66 ± 0.04	3.12 ± 0.34	2.67 ± 0.06	2.77 ± 0.43
5	2.65 ± 0.04	3.10 ± 0.43	2.67 ± 0.05	2.74 ± 0.30
6	2.61 ± 0.06	3.14 ± 0.65	2.65 ± 0.05	**2.70 ± 0.45**
7	2.62 ± 0.03	3.10 ± 0.44	2.58 ± 0.04	2.71 ± 0.48
8	2.60 ± 0.06	3.27 ± 0.58	2.49 ± 0.04	2.81 ± 0.39
C	Model F		Model SF	
	PI	E_PI	PI	E_PI
2	2.77 ± 0.03	3.04 ± 0.24	2.79 ± 0.06	2.87 ± 0.56
3	2.74 ± 0.06	3.00 ± 0.54	2.74 ± 0.04	2.76 ± 0.42
4	2.68 ± 0.05	3.08 ± 0.46	2.71 ± 0.02	2.78 ± 0.23
5	2.66 ± 0.05	3.00 ± 0.46	2.65 ± 0.04	**2.65 ± 0.44**
6	2.65 ± 0.06	2.94 ± 0.47	2.63 ± 0.04	2.69 ± 0.42
7	2.60 ± 0.03	3.11 ± 0.38	2.59 ± 0.05	2.72 ± 0.39
8	2.57 ± 0.07	3.22 ± 0.62	2.59 v 0.07	3.08 ± 0.54

C: Number of clusters (nodes); ε: SVs-parameter; λ: Regularization parameter.
(B) Model F (FCNNs realized by FCM) and Model SF (RFCNNs realized by SVs-based FCM)
(ε = 0.800, λ = 0.1)C: Number of clusters (nodes); ε: SVs-parameter; λ: Regularization parameter.

Table 7
Simulation results of FCNNs and (proposed) RFCNNs for the NOx dataset. Model H (FCNNs realized by HCM) and Model SH (RFCNNs realized by SVs-based HCM) (ε = 0.590, λ = 0.1).

C	Model H		Model SH	
	PI	E_PI	PI	E_PI
2	3.84 ± 0.09	4.03 ± 0.44	3.87 ± 0.08	3.97 ± 0.64
3	3.25 ± 0.18	3.73 ± 1.45	3.16 ± 0.21	3.51 ± 1.59
4	2.41 ± 0.18	2.85 ± 0.77	2.39 ± 0.19	2.77 ± 0.57
5	1.78 ± 0.12	2.13 ± 0.60	1.79 ± 0.14	2.09 ± 0.39
6	1.58 ± 0.16	1.95 ± 0.50	1.59 ± 0.14	**1.70 ± 0.49**
7	1.43 ± 0.13	2.40 ± 1.70	1.40 ± 0.15	1.87 ± 0.44
8	1.10 ± 0.28	2.61 ± 3.55	1.16 ± 0.19	2.44 ± 3.20
C	Model F		Model SF	
	PI	E_PI	PI	E_PI
2	2.70 ± 0.08	4.63 ± 2.91	2.94 ± 0.02	3.43 ± 0.71
3	2.41 ± 0.20	4.51 ± 4.80	2.38 ± 0.25	2.65 ± 2.33
4	1.65 ± 0.13	1.91 ± 0.57	1.67 ± 0.10	1.77 ± 0.38
5	1.31 ± 0.10	1.61 ± 0.51	1.33 ± 0.12	1.51 ± 0.46
6	1.17 ± 0.08	1.40 ± 0.50	1.17 ± 0.11	1.31 ± 0.56
7	1.01 ± 0.05	1.38 ± 0.49	1.02 ± 0.03	1.19 ± 0.61
8	0.91 ± 0.03	1.24 ± 0.53	0.91 ± 0.02	**1.10 ± 0.49**

C: Number of clusters (nodes); ε: SVs-parameter; λ: Regularization parameter.
C: Number of clusters (nodes); ε: SVs-parameter; λ: Regularization parameter.
Model F (FCNNs realized by FCM) and Model SF (RFCNNs realized by SVs-based FCM)
(ε = 0.330, λ = 0.1)

Table 8
Simulation results of FCNNs and (proposed) RFCNNs for the Boston Housing dataset. Model H (FCNNs realized by HCM) and Model SH (RFCNNs realized by SVs-based HCM) (ε = 0.002, λ = 0.3).

C	Model H		Model SH	
	PI	E_PI	PI	E_PI
2	3.45 ± 0.06	4.02 ± 0.50	3.49 ± 0.05	3.74 ± 0.55
3	3.35 ± 0.05	3.87 ± 0.48	3.35 ± 0.06	3.71 ± 0.50
4	3.19 ± 0.06	3.86 ± 0.49	3.20 ± 0.13	3.66 ± 0.67
5	3.09 ± 0.06	3.88 ± 0.52	3.07 ± 0.04	3.51 ± 0.59
6	3.07 ± 0.07	3.84 ± 0.45	2.99 ± 0.06	**3.36 ± 0.60**
7	2.99 ± 0.09	4.28 ± 0.49	2.99 ± 0.09	3.40 ± 0.54
8	2.97 ± 0.07	4.46 ± 0.44	2.93 ± 0.05	3.92 ± 0.56
C	Model F		Model SF	
	PI	E_PI	PI	E_PI
2	3.43 ± 0.06	4.12 ± 0.59	3.46 ± 0.06	3.80 ± 0.52
3	3.27 ± 0.08	4.06 ± 0.69	3.27 ± 0.08	3.71 ± 0.44
4	3.21 ± 0.06	3.93 ± 0.37	3.18 ± 0.06	**3.34 ± 0.48**
5	2.95 ± 0.05	3.88 ± 0.57	2.95 ± 0.06	3.64 ± 0.70
6	2.82 ± 0.06	3.97 ± 0.54	2.78 ± 0.04	3.52 ± 0.53
7	2.75 ± 0.08	4.08 ± 0.57	2.77 ± 0.07	3.67 ± 0.75
8	2.70 ± 0.07	4.13 ± 0.83	2.72 ± 0.06	3.80 ± 0.49

C: Number of clusters (nodes); ε: SVs-parameter; λ: Regularization parameter.
C: Number of clusters (nodes); ε: SVs-parameter; λ: Regularization parameter.

(A) Model F (FCNNs realized by FCM) and Model SF (RFCNNs realized by SVs-based FCM)

(ε = 0.005, λ = 0.3)

Table 9
Parameter settings in RFCNNs modeling.

Dataset type	Model SH			Model SF		
	C	ε	λ	C	ε	λ
MPG	6	1.200	0.1	5	0.800	0.1
NOX	6	0.590	0.1	8	0.330	0.1
BOH	6	0.002	0.3	4	0.005	0.3
MIS	4	0.750	0.1	2	0.690	0.1
CON	10	0.008	0.1	10	0.006	0.1
FRI	7	0.018	0.2	8	0.026	0.2
WIZ	10	0.001	0.1	10	0.001	0.1
PYR	8	0.660	0.1	8	0.665	0.1
FR2	10	0.700	0.2	10	0.800	0.2
WIR	4	0.800	0.3	4	0.960	0.3
WIW	4	1.300	0.3	4	1.150	0.3

Model SH: RFCNNs realized by SVs-based HCM; Model SF: RFCNNs realized by SVs-based FCM.

Table 10
Comparison of RMSE for each dataset between FCNNs and the proposed RFCNNs.

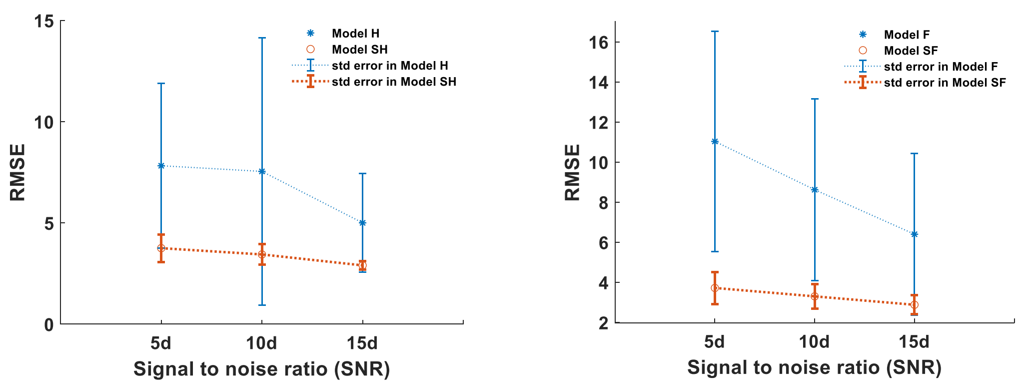
Dataset type	Model H	Model F	Model SH	Model SF
MPG	3.10 ± 0.43	3.00 ± 0.46	2.70 ± 0.45	**2.65 ± 0.44**
NOX	1.95 ± 0.50	1.24 ± 0.53	1.70 ± 0.49	**1.10 ± 0.49**
BOH	3.84 ± 0.45	3.88 ± 0.57	3.36 ± 0.60	**3.34 ± 0.48**
MIS	1.16 ± 0.39	1.40 ± 0.68	0.99 ± 0.25	**0.97 ± 0.31**
CON	8.16 ± 0.48	9.41 ± 0.48	**6.62 ± 0.53**	7.05 ± 0.60
FRI	1.20 ± 0.08	1.89 ± 0.18	**1.16 ± 0.06**	1.33 ± 0.10
WIZ	1.20 ± 0.16	1.23 ± 0.11	**1.11 ± 0.10**	1.11 ± 0.14
PYR	0.16 ± 0.05	0.20 ± 0.09	0.10 ± 0.05	**0.10 ± 0.04**
FR2	1.01 ± 0.17	1.16 ± 0.15	**0.77 ± 0.15**	0.81 ± 0.16
WIR	0.65 ± 0.04	0.65 ± 0.04	0.57 ± 0.05	**0.55 ± 0.04**
WIW	0.74 ± 0.02	**0.76 ± 0.02**	0.66 ± 0.03	**0.65 ± 0.04**

Model H: FCNNs realized by HCM; Model F: FCNNs realized by FCM; Model SH: RFCNNs realized by SVs-based HCM; Model SF: RFCNNs realized by SVs-based FCM.

Table 11
Comparison of RMSE between RFCNNs and FCNNs by adding noise to the testing dataset. (a) Model H and Model SH.

Dataset type	Model H				Model SH			
	5 dB	*10 dB*	*15 dB*	*Without noise*	*5 dB*	*10 dB*	*15 dB*	*Without noise*
MPG	7.82 ± 4.07	7.54 ± 6.60	5.00 ± 2.44	3.10 ± 0.43	3.75 ± 0.68	3.44 ± 0.50	2.90 ± 0.21	2.70 ± 0.45
BOH	19.08 ± 3.08	12.45 ± 1.91	9.04 ± 2.58	3.84 ± 0.45	7.09 ± 0.96	5.15 ± 0.66	4.32 ± 0.44	3.36 ± 0.60
FRI	8.02 ± 0.42	5.37 ± 0.24	3.63 ± 0.20	1.20 ± 0.08	5.44 ± 0.26	4.09 ± 0.21	2.60 ± 0.13	1.16 ± 0.06
WIR	22.87 ± 3.65	12.47 ± 2.01	8.38 ± 1.28	0.65 ± 0.04	1.55 ± 0.13	1.02 ± 0.06	0.77 ± 0.05	0.57 ± 0.05
WIW	42.28 ± 2.34	34.96 ± 2.81	24.14 ± 1.38	0.74 ± 0.02	1.55 ± 0.07	1.04 ± 0.04	0.85 ± 0.03	0.66 ± 0.03
Dataset type	Model F				Model SF			
	5dB	*10dB*	*15dB*	*Without noise*	*5dB*	*10dB*	*15dB*	*Without noise*
MPG	11.03 ± 5.49	8.62 ± 4.54	6.40 ± 4.03	2.94 ± 0.47	3.72 ± 0.80	3.30 ± 0.61	2.88 ± 0.48	2.65 ± 0.44
BOH	20.41 ± 3.57	12.33 ± 3.88	9.11 ± 4.04	3.88 ± 0.57	7.06 ± 0.90	4.97 ± 0.59	4.20 ± 0.51	3.34 ± 0.48
FRI	8.39 ± 0.52	5.48 ± 0.31	3.82 ± 0.24	1.89 ± 0.18	5.27 ± 0.25	3.67 ± 0.20	2.63 ± 0.17	1.33 ± 0.10
WIR	23.94 ± 2.86	14.77 ± 2.70	9.81 ± 1.38	0.66 ± 0.04	1.15 ± 0.07	0.86 ± 0.07	0.71 ± 0.06	0.55 ± 0.04
WIW	43.26 ± 3.24	33.93 ± 1.93	23.45 ± 1.73	0.74 ± 0.01	1.37 ± 0.06	0.97 ± 0.03	0.81 ± 0.04	0.65 ± 0.04

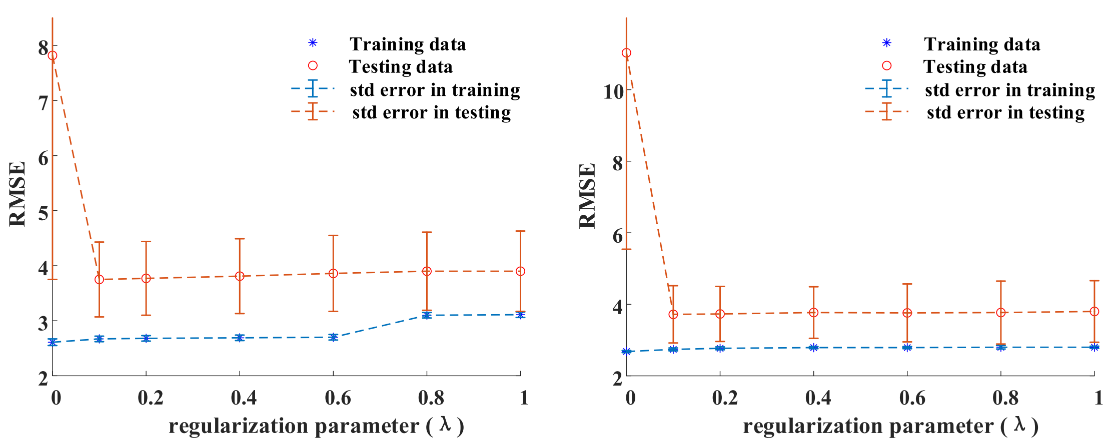
Model F: FCNNs designed without L_2 norm-regularization; Model SF: RFCNNs designed with L_2 norm-regularization.
Model H: FCNNs designed without L_2 norm-regularization; Model SH: RFCNNs designed with L_2 norm-regularization.
(b) Model F and Model SF



(a) Model SH (RFCNNs realized by SVs-based HCM and L_2 norm-regularization) and Model H (FCNNs designed by HCM)

(b) Model SF (RFCNNs realized by SVs-based FCM and L_2 norm-regularization) and Model F (FCNNs designed by FCM)

Fig. 11. Root mean square error (RMSE) of RFCNNs and FCNNs for the MPG testing dataset including Gaussian white noise, according to the increase of SNR.



(a) Model SH (RFCNNs realized by SVs-based HCM and L_2 norm-regularization)

(a) Model SF (RFCNNs realized by SVs-based FCM and L_2 norm-regularization)

Fig. 12. Root mean square error (RMSE) of RFCNNs for the MPG test dataset including Gaussian white noise (5 dB), according to the increase of regularization parameter (λ).

Table 12
Comparison of performance results of the proposed RFCNNs with other models and statistical comparison of regression models over some datasets.

Dataset type	SVR(Gaussian)	Random Tree	Linear Regression	IBK	Multilayer perceptron	RBFNN	REP Tree	Method_1	Method_2
MPG	3.54 ± 0.57 (6|6)	3.89 ± 0.69 (7|7)	3.36 ± 0.45 (5|5)	3.25 ± 0.59 (3|3)	3.19 ± 0.72 (2|2)	4.23 ± 0.61 (8|8)	3.31 ± 0.60 (4|4)	2.70 ± 0.45 (1)	**2.65 ± 0.44** (1)
NOX	4.75 ± 0.94 (7|7)	1.53 ± 1.88 (1|2)	4.33 ± 0.51 (5|5)	4.78 ± 1.34 (8|8)	4.60 ± 1.42 (6|6)	2.74 ± 1.04 (3|3)	2.74 ± 1.60 (4|4)	1.70 ± 0.49 (2)	**1.10 ± 0.49** (1)
BOH	5.49 ± 1.27 (8|8)	4.93 ± 1.30 (7|7)	4.80 ± 1.30 (6|6)	4.61 ± 0.99 (3|3)	4.32 ± 1.17 (2|2)	4.61 ± 1.31 (4|4)	4.64 ± 1.08 (5|5)	3.36 ± 0.60 (1)	**3.34 ± 0.48** (1)
MIS	1.19 ± 0.35 (6|6)	1.35 ± 0.35 (8|8)	1.05 ± 0.23 (2|2)	1.19 ± 0.31 (5|5)	1.26 ± 0.35 (7|7)	1.11 ± 0.45 (3|3)	1.14 ± 0.32 (4|4)	0.99 ± 0.25 (1)	**0.97 ± 0.31** (1)
CON	10.80 ± 0.65 (8|8)	6.67 ± 1.01 (2|1)	10.46 ± 0.73 (6|6)	8.98 ± 0.83 (5|5)	10.70 ± 1.56 (7|7)	7.20 ± 1.28 (3|3)	7.27 ± 0.78 (4|4)	**6.62 ± 0.53** (1)	7.05 ± 0.60 (2)
FRI	2.71 ± 0.26 (5|5)	2.87 ± 0.19 (7|7)	2.69 ± 0.24 (4|4)	2.42 ± 0.14 (2|2)	3.06 ± 0.50 (8|8)	2.43 ± 0.22 (3|3)	2.78 ± 0.19 (6|6)	**1.16 ± 0.06** (1)	1.33 ± 0.10 (1)
WIZ	1.27 ± 0.13 (4|4)	2.11 ± 0.24 (7|7)	1.26 ± 0.13 (3|3)	2.35 ± 0.19 (8|8)	1.35 ± 0.30 (5|5)	1.23 ± 0.43 (2|2)	1.75 ± 0.14 (6|6)	**1.11 ± 0.10** (1)	1.11 ± 0.14 (1)
PYR	0.11 ± 0.07 (6|6)	0.10 ± 0.07 (3.5|3.5)	0.12 ± 0.07 (8|8)	**0.09 ± 0.04** (1|1)	0.11 ± 0.05 (5|5)	0.10 ± 0.07 (3.5|3.5)	0.12 ± 0.06 (7|7)	0.10 ± 0.05 (2)	0.10 ± 0.04 (2)
FR2	0.98 ± 0.12 (4|4)	0.98 ± 0.19 (6|6)	0.89 ± 0.10 (3|3)	1.28 ± 0.17 (8|8)	1.12 ± 0.19 (7|7)	0.98 ± 0.15 (5|5)	**0.66 ± 0.12** (1|1)	0.77 ± 0.15 (2)	0.81 ± 0.16 (2)
WIR	0.67 v 0.04 (2.5|2.5)	0.71 ± 0.05 (5.5|5.5)	0.66 ± 0.04 (2.5|2.5)	0.81 ± 0.05 (8|8)	0.74 ± 0.11 (7|7)	0.71 ± 0.05 (5.5|5.5)	0.67 ± 0.04 (4|4)	0.57 ± 0.05 (1)	**0.55 ± 0.04** (1)
WIW	0.75 ± 0.04 (5|5)	**0.46 ± 0.04** (1|1)	0.73 ± 0.03 (4|4)	0.77 ± 0.04 (6|6)	0.77 ± 0.09 (7|7)	0.78 ± 0.02 (8|8)	0.49 ± 0.04 (2|2)	0.66 ± 0.03 (3)	0.65 ± 0.04 (3)
AR	5.59|5.59	5.00|5.00	4.41|4.41	5.18|5.18	5.73|5.73	4.36|4.36	4.27|4.27	1.45	1.45

AR: Average Rank; Method_1: RFCNNs designed with the aid of SVs-based HCM and L_2 norm-regularization; Method_2: RFCNNs designed with the aid of SVs-based FCM and L_2 norm-regularization. (+|-): "+" is the order in Method_1 and other models (without Method_2); "-" is the order in Method_2 and other models (without Method_1).

Table 13
Difference in the average rank between proposed RFCNNs and other models.

No.	Type of Model	Difference in the average rank between the proposed model and other models							Comparison with CD (CD = 2.81)
		SVR (Gaussian) Model A	Random Tree Model B	Linear Regression Model C	IBK Model D	Multilayer perceptron Model E	RBFNN Model H	REP Tree Model G	
1	Method_1	4.14	3.55	3.05	3.73	4.28	2.91	2.82	(A,1), (B,1), (C,1), (D,1), (E,1), (F,1), (G,1) > CD
2	Method_2	4.14	3.55	2.96	3.73	4.28	2.91	2.82	(A,2), (B,2), (C,2), (D,2), (E,2), (F,2), (G,2) > CD

Method_1: RFCNNs designed with the ± aid of SVs-based HCM and L_2 norm-regularization; Method_2: RFCNNs designed with the aid of SVs-based FCM and L_2 norm-

themselves. The performance of any two models will be significantly different if the corresponding average ranking is at least off the threshold (CD). At $p = 0.05$ (significance level), the CD is equal to 2.81. In Table 13, it can be concluded that the proposed RFCNNs modeling is significantly superior to most of the other comparative regression models used in this study based on the values of the CD.

7. Concluding comments

In this study, a new approach to RFCNNs modeling designed with SVs-based clustering techniques and L_2 norm-regularization is introduced. Two types of clustering techniques are used in this modeling, namely, SVs-based HCM and SVs-based FCM. The cluster centers are determined from the extracted data (SVs) instead of the entire training data, which helps to reduce the interference of uncorrelated data including noise and outliers. The detailed representation of two types of clustering techniques based on SVs is described. Experimental quantification shows the reduced interference coming from irrelevant data as well as the better predictive power of the points presented in the RFCNN modeling.

In the future work, additional optimization mechanisms could be considered to optimize the error tolerance "ε", which can more effectively reduce irrelevant data, including noise and outliers. The proposed techniques could be taken into consideration in classification, as well as enhanced classification problems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (NRF-2021R1F1A1056102) and Shandong Provincial Key Research and Development Program (Major Science and Technological Innovation Project, No.2019JZZY010134).

References

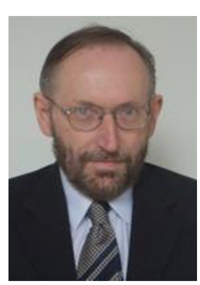
[1] M. Gupta, D. Rao, On the principles of fuzzy neural networks, Fuzzy Set. Syst. 61 (1994) 1–18.
[2] H.-J. Zimmermann, Fuzzy set theory—and its applications, Springer Science & Business Media, 2011.
[3] L.A. Zadeh, Fuzzy sets, Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh, World Scientific (1996) 394–432.

[4] S.-K. Oh, W.-D. Kim, W. Pedrycz, B.-J. Park, Polynomial-based radial basis function neural networks (P-RBF NNs) realized with the aid of particle swarm optimization, Fuzzy Sets Syst. 163 (2011) 54–77.
[5] S.-K. Oh, W.-D. Kim, W. Pedrycz, S.-C. Joo, Design of K-means clustering-based polynomial radial basis function neural networks (pRBF NNs) realized with the aid of particle swarm optimization and differential evolution, Neurocomputing 78 (2012) 121–132.
[6] S.-K. Oh, W.-D. Kim, W. Pedrycz, K. Seo, Fuzzy radial basis function neural networks with information granulation and its parallel genetic optimization, Fuzzy Sets Syst. 237 (2014) 96–117.
[7] G. Gan, C. Ma, J. Wu, Data clustering: theory, algorithms, and applications (SIAM, 2020).
[8] B.A. Pimentel, R.M. de Souza, Multivariate Fuzzy C-Means algorithms with weighting, Neurocomputing 174 (2016) 946–965.
[9] W. Lin, J. Gao, Q. Wang, X. Li, Learning to detect anomaly events in crowd scenes from synthetic data, Neurocomputing 436 (2021) 248–259.
[10] X. Xu, S. Li, T. Liang, T. Sun, Sample selection-based hierarchical extreme learning machine, Neurocomputing 377 (2020) 95–102.
[11] S. Zeng, Z. Wang, R. Huang, L. Chen, D. Feng, A study on multi-kernel intuitionistic fuzzy C-means clustering with multiple attributes, Neurocomputing 335 (2019) 59–71.
[12] X.-L. Xia, M.R. Lyu, T.-M. Lok, G.-B. Huang, Methods of decreasing the number of support vectors via k-mean clustering, International Conference on Intelligent Computing, (Springer2005), pp. 717-726.
[13] Z. Wang, R. Zhang, J. Qi, B. Yuan, Dbsvec: Density-based clustering using support vector expansion, 2019 IEEE 35th International Conference on Data Engineering (ICDE), (IEEE2019), pp. 280-291.
[14] F. Li, L. Jia, Parameter estimation of Hammerstein-Wiener nonlinear system with noise using special test signals, Neurocomputing 344 (2019) 37–48.
[15] P.A. Mastorocostas, J.B. Theocharis, V.S. Petridis, A constrained orthogonal least-squares method for generating TSK fuzzy models: application to short-term load forecasting, Fuzzy Sets Syst. 118 (2001) 215–233.
[16] J.A. Gallego, F.A. González, O. Nasraoui, Robust Kernels for Robust Location Estimation, Neurocomputing 429 (2021) 174–186.
[17] A.E. Hoerl, R.W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67.
[18] M. He, Y. Xiu, Estimation of the complex frequency of a harmonic signal based on a linear least squares method, Geod. Geodyn. 6 (3) (2015) 220–225.
[19] T. Hastie, R. Tibshirani, M. Wainwright, Statistical learning with sparsity: the lasso and generalizations, CRC Press, 2015.
[20] H. Shahsavari, A. Nateghi, Optimal design of probabilistically robust $PI^{\lambda}D^{\mu}$ controller to improve small signal stability of PV integrated power system, J. Franklin Inst. 356 (2019) 7183–7209.
[21] Y. Wang, Y. Li, R. Yang, Sparse adaptive channel estimation based on mixed controlled l2 and lp-norm error criterion, J. Franklin Inst. 354 (2017) 7215–7239.
[22] C. Alzate, J.A. Suykens, Kernel component analysis using an epsilon-insensitive robust loss function, IEEE Trans. Neural Networks 19 (2008) 1583–1598.
[23] H. Tong, M.K. Ng, Calibration of $\epsilon-$ insensitive loss in support vector machines regression, J. Franklin Inst. 356 (2019) 2111–2129.
[24] D. Pan, Z. Jiang, Z. Chen, W. Gui, Y. Xie, C. Yang, Temperature measurement and compensation method of blast furnace molten iron based on infrared computer vision, IEEE Trans. Instrum. Meas. 68 (2018) 3576–3588.
[25] P. Meropi, C. Bikos, Z. George, Outlier detection in skewed data, Simul. Model. Pract. Theory 87 (2018) 191–209.
[26] L.Q. Anh, N. Van Hung, V.M. Tam, Regularized gap functions and error bounds for generalized mixed strong vector quasiequilibrium problems, Computat. Appl. Mathemat. 37 (5) (2018) 5935–5950.
[27] A. Abadie, M. Kasy, Choosing among regularized estimators in empirical economics: The risk of machine learning, Rev. Econ. Stat. 101 (5) (2019) 743–762.
[28] C.M. Le, K. Levin, P.J. Bickel, E. Levina, Comment: Ridge Regression and Regularization of Large Matrices, Technometrics 62 (4) (2020) 443–446.
[29] J. Shawe-Taylor, P.L. Bartlett, R.C. Williamson, M. Anthony, Structural risk minimization over data-dependent hierarchies, IEEE Trans. Inf. Theory 44 (5) (1998) 1926–1940.
[30] V. Vapnik, E. Levin, Y.L. Cun, Measuring the VC-dimension of a learning machine, Neural Comput. 6 (5) (1994) 851–876.
[31] K. Cheng, Z. Lu, Y. Zhou, Y. Shi, Y. Wei, Global sensitivity analysis using support vector regression, Appl. Math. Model. 49 (2017) 587–598.
[32] S.M. Clarke, J.H. Griebsch, T.W. Simpson, Analysis of support vector regression for approximation of complex engineering analyses, (2005).
[33] Z. Xuegong, Introduction to statistical learning theory and support vector machines, Acta Autom. Sin. 26 (2000) 32–42.
[34] G. Di Pillo, S. Lucidi, An augmented Lagrangian function with improved exactness properties, SIAM J. Optim. 12 (2002) 376–406.
[35] S.K. Shevade, S.S. Keerthi, C. Bhattacharyya, K.R.K. Murthy, Improvements to the SMO algorithm for SVM regression, IEEE Trans. Neural Networks 11 (2000) 1188–1193.
[36] S. Hashmi, S.M. Halawani, O.M. Barukab, A. Ahmad, Model trees and sequential minimal optimization based support vector machine models for estimating minimum surface roughness value, Appl. Math. Model. 39 (2015) 1119–2113.

Hao Huang received his M.Sc. degrees in computer science from University of Suwon, Suwon, South Korea. He is working toward the PhD degree at computer science, University of Suwon, Suwon, South Korea. His current research interests include fuzzy logic and fuzzy system, Computational Intelligence, machine learning, and security techniques in Internet of Things.

Sung-Kwun Oh received his B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, Korea, in 1981, 1983, and 1993, respectively. During 1983-1989, he was a Senior Researcher of R&D Lab. of Lucky-Goldstar Industrial Systems Co., Ltd. From 1996 to 1997, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada. He is currently a Professor with the Department of Electrical Engineering, University of Suwon, Suwon, South Korea. His research interests include fuzzy system, fuzzy-neural networks, automation systems, advanced Computational Intelligence, and intelligent control. He currently serves as an Associate Editor of the KIEE Transactions on Systems and Control, International Journal of Fuzzy Logic and Intelligent Systems of the KFIS, and Information Sciences.

Chuan-Kun Wu, currently a Distinguished Professor of School of Information Science, Linyi University. Before joining Linyi University, he was a research professor of State Key Laboratory of Information Security, Chinese Academy of Sciences, and then a research leader at a private company. Chuan-Kun Wu has been a senior member of IEEE since 2000, served as an associate editor of IEEE Communications Letters, as an editorial board member of International Journal of Network Security, served a number of international conferences as an organizing co-chair, or a PC member. He has published over 200 research papers, and has co-authored 6 books. His research interests include cryptography, cryptographic boolean functions, authentication protocols, security techniques in Internet of Things, and industrial control security.



Witold Pedrycz received his M.Sc., Ph.D., D.Sci. all from the Silesian University of Technology, Gliwice, Poland. He is a Professor and Canada Research Chair (CRC) in Computational Intelligence in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is also with the Polish Academy of Sciences, Systems Research Institute, Warsaw, Poland. His research interests encompass Computational Intelligence, fuzzy modeling, knowledge discovery and data mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, granular and relational computing, and Software Engineering. He has published numerous papers in these areas. He is also an author of 15 research monographs. He is also an Editor-in-Chief of Information Sciences (Elsevier), WIREs Data Mining and Knowledge Discovery (Wiley), and Int. J. of Granular Computing (Springer). Witold Pedrycz has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. He served as Editor-in-Chief of IEEE Transactions on Systems Man and Cybernetics Part A and President of IFSA, and currently serves on the Advisory Board of IEEE Transactions on Fuzzy Systems.