Journal of Engineering Science and Technology Review 8 (5) (2015) 2129- 226

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Parsimonious Wavelet Kernel Extreme Learning Machine

Wang Qin[1], Shen Yuantong[2], Kuang Yu[3], Wu Qiang[4*] and Sun Lin[5]

[1]Department of Information Technology, Hainan Medical University, Haikou 571101, China
[2] School of Mathematics and Physics, China University of Geosciences, Wuhan 430074, China
[3] Department of Medical Physics, University of Nevada Las Vegas, Las Vegas, NV 89154, United States
[4] Affiliated Hospital of Hainan Medical University, Haikou 571101, China
[5] Foreign Languages Department, Hainan Medical University, Haikou 571101, China

Received 19 August 2015; Accepted 10 November 2015

Abstract

In this study, a parsimonious scheme for wavelet kernel extreme learning machine (named PWKELM) was introduced by combining wavelet theory and a parsimonious algorithm into kernel extreme learning machine (KELM). In the wavelet analysis, bases that were localized in time and frequency to represent various signals effectively were used. Wavelet kernel extreme learning machine (WELM) maximized its capability to capture the essential features in “frequency-rich” signals. The proposed parsimonious algorithm also incorporated significant wavelet kernel functions via iteration in virtue of Householder matrix, thus producing a sparse solution that eased the computational burden and improved numerical stability. The experimental results achieved from the synthetic dataset and a gas furnace instance demonstrated that the proposed PWKELM is efficient and feasible in terms of improving generalization accuracy and real time performance.

Keywords: Kernel extreme learning machine, Wavelet kernel function, Householder matrix, Sparse solution

1. Introduction

The support vector machine (SVM) proposed by Vapnik solves quadratic regression problems with inequations [1]. Least squares support vector machine (LS-SVM) is a modified version of the standard SVM, and replaces the inequality constraints in addressing quadratic programming [2]. Thus, the training speed of LS-SVM is higher than that of SVM. Unfortunately, LS-SVM loses the sparseness of support vector, thereby degrading the generalization performance of SVM. In view of the sparseness of LS-SVM, a fixed-size LS-SVM was proposed to realize the sparse representation in primal weight space [3]. Upon adding a bias term to the objective function, LS-SVM was solved through forward least squares approximation; thus, a sparse solution is generated in the least square sense [4]. Afterward**,** a sparse algorithm of least squares support vector regression was established based on Householder transformation [5]. This algorithm can effectively sparsify the solution of LS-SVM.

The extreme learning machine(ELM) for single hidden-layer feedforward networks (SLFNs) has garnered significant interests among researchers since Huang et al.[6] first proposed their seminal work on ELM. In this machine, the input weights are chosen at random; the output weights can be analytically determined via the simple generalized inverse operation on the hidden layer output matrices. Empirical studies have indicated that the generalization capability of ELM is comparable to or even better than that of SVMs and its variants [7].Thus, ELM has successfully been applied for classification [8] and regression [9].

Nonetheless, the number of hidden layer nodes, which is an important parameter of ELM, usually should be determined beforehand by users through some time-consuming methods. A special batch variant of ELM, namely, kernel extreme learning machine (KELM) [7], is proposed to avoid the problem of hidden nodes' selection. KELM transposes the explicit activation function into implicit kernel mapping. Unlike LS-SVM, KELM does not constrain Lagrange multipliers α_i ’s, therefore, LS-SVM generates a solution that is suboptimal to ELM. As with the conventional kernel methods, the core problem of KELM is the selection of kernel functions. In consideration of the nonstationary signals, an appropriate kernel function that can accurately capture the underlying information is expected to yield a compact representation. Commonly used kernel functions include Gauss function and polynomial function.

Wavelet analysis technology has recently been widely used to optimize neural networks. This technology is characterized by multiscale interpolation and the localization feature in both frequency and time domains. At present, wavelet kernel extreme learning machine (WKELM) has performed excellently in terms of classification [10].

In theory, the entire training sample in WKELM must be stored in the memory, which may generate redundant or unimportant wavelet kernel functions. A fundamental principle followed in system modeling is the well-recognized Occam’s razor hypothesis: “plurality should not be posited without necessity.” In other words, the simpler a solution is, the more reasonable it is. Sparse models are preferable in engineering applications because the

[*] E-mail address: wuqiang_6545@163.com

ISSN: 1791-2377 © 2015 Kavala Institute of Technology.
All rights reserved.

computational complexity of a model scales with its model complexity. Deng et al. recently proposed a reduced KELM (RKELM) that is capable of locating sparse topology by randomly selecting a subset from a training dataset [11]. However, numerical instability still exists. A fast sparse approximation scheme for KELM (FSA-KELM) [12] was recently introduced to obtain a simple sparse solution; this process begins with a null solution and gradually selects a new hidden node according to some criteria. This procedure is repeated until the stopping criterion is met.

In the current study, a parsimonious algorithm and wavelet technique are developed for KELM on the basis of the aforementioned analysis, the resultant model is referred to as parsimonious wavelet kernel extreme learning machine (PWKELM). Wavelet function benefits from multiscale interpolation and is also suitable for the local analysis and detection of transient signals. As a result, a wavelet expansion representation is compact and is easy to implement. Householder matrix is also used to orthogonalize the linear equation set in WKELM, and significant wavelet kernel functions are recruited iteratively. Thus, a sparse solution is established. Synthetic and real-world data sets are utilized in conducting experiments whose results confirm the effectiveness and feasibility of the proposed PWKELM.

The rest of this paper is organized as follows: ELM and WKELM are introduced in Section 2. The PWKELM algorithm and its detailed procedure are listed as well. The experimental results and analyses are presented in Section 3. Conclusions follow in the final section.

2. Methodology

2.1 ELM

ELM is based on the perceptron model with a single hidden-layer and has a simple three layer structure. This structure is composed of the input, output, and hidden layers. If we randomly assign the input weights and bias values, we need not to adjust the input weights or hidden layer bias throughout the learning process.

For N training samples $\left\{\left(\boldsymbol{x}_i,t_i\right)\right\}_{i=1}^{N},\boldsymbol{x}_i\in\mathbb{R}^n,t_i\in\mathbb{R}$, the SLFN model with $\tilde{N}$ hidden nodes (additive or RBF nodes) can be formulated as

$$\sum_{i=1}^{\tilde{N}}\beta_i g\left(\boldsymbol{w}_i,b_i,\boldsymbol{x}_j\right)=o_j, j=1,\cdots,N \tag{1}$$

where β_i is the output weight connected to the i th hidden layer node, $\boldsymbol{w}_i$ and b_i are learning parameters of hidden layer nodes, $o_j\in\mathbb{R}$ is the output of ELM, and $g(\bullet)$ is the activation function. Eq. (1) can be written compactly as

$$\mathbf{H}\boldsymbol{\beta}=\boldsymbol{o} \tag{2}$$

where

$$\boldsymbol{o}=[o_1,\cdots,o_N]^T,\boldsymbol{\beta}=[\beta_1,\cdots,\beta_{\tilde{N}}]^T,$$

$$\mathbf{H}=\begin{bmatrix} g\left(\boldsymbol{w}_1\cdot\boldsymbol{x}_1+b_1\right)\cdots g\left(\boldsymbol{w}_{\tilde{N}}\cdot\boldsymbol{x}_1+b_{\tilde{N}}\right)\\ \vdots \quad \cdots \quad \vdots \\ g\left(\boldsymbol{w}_1\cdot\boldsymbol{x}_N+b_1\right)\cdots g\left(\boldsymbol{w}_{\tilde{N}}\cdot\boldsymbol{x}_N+b_{\tilde{N}}\right)\end{bmatrix}_{N\times\tilde{N}}.$$

$\mathbf{H}$ is called the output matrix of the hidden layer. If the ELM model with $\tilde{N}$ hidden nodes can approximate N samples with zero error, then it means that there exist β_i such that

$$\sum_{i=1}^{\tilde{N}}\boldsymbol{\beta}_i G\left(\boldsymbol{w}_i,b_i,\boldsymbol{x}_j\right)=t_j, j=1,\cdots,N \tag{3}$$

where t_j is the target value. Eq. (3) can be expressed in the following matrix-vector form

$$\mathbf{H}\boldsymbol{\beta}=\mathbf{T} \tag{4}$$

where $\mathbf{T}=\left[t_1,t_2,\cdots,t_N\right]^T$. The hidden nodes' learning parameters $\boldsymbol{w}_i$ and b_i are randomly generated in the beginning. Thus Eq. (4) then becomes a linear system and the output weights $\boldsymbol{\beta}$ can be computed analytically by finding the minimum norm least squares' solution as follows

$$\boldsymbol{\beta}=\mathbf{H}^{\dagger}\mathbf{T} \tag{5}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of $\mathbf{H}$.

2.2 WKELM

The universal approximation capability of ELM with hidden nodes has been proven. KELM is proposed in response to the problem of selecting the demanding hidden nodes. KELM substitutes the kernel function mapping for the hidden layer mapping $\boldsymbol{h}(\boldsymbol{x})$.

Given a training set $\left\{\left(\boldsymbol{x}_i,t_i\right)\right\}_{i=1}^{N},\boldsymbol{x}_i\in\mathbb{R}^n,t_i\in\mathbb{R}$, the original optimization problem of KELM can be expressed as

$$\begin{aligned} &\min \quad L_{ELM}=\frac{1}{2}\|\boldsymbol{\beta}\|^2+\frac{C}{2}\xi_i^2 \\ &s.t. \quad \boldsymbol{h}\left(\boldsymbol{x}_i\right)\boldsymbol{\beta}=t_i-\xi_i,\ i=1,\cdots,N \end{aligned} \tag{6}$$

The corresponding Lagrangian dual problem can be formatted as

$$L_{ELM}=\frac{1}{2}\|\boldsymbol{\beta}\|^2+\frac{C}{2}\sum_{i=1}^{N}\xi_i^2-\sum_{i=1}^{N}\alpha_i\left(\boldsymbol{h}\left(\boldsymbol{x}_i\right)\boldsymbol{\beta}-t_i+\xi_i\right) \tag{7}$$

The KKT optimality conditions of Eq. (7) are as follows

$$\boldsymbol{\beta}=\mathbf{H}^T\boldsymbol{\alpha} \tag{8}$$

$$\left(\frac{\boldsymbol{I}}{\boldsymbol{C}}+\mathbf{H}\mathbf{H}^T\right)\boldsymbol{\alpha}=\mathbf{T} \tag{9}$$

The feature mapping $\boldsymbol{h}(\boldsymbol{x})$ can be unknown, but the corresponding kernel is usually given, the kernel matrix of ELM can be defined as follow

$$\boldsymbol{K}_{ELM}=\mathbf{H}\mathbf{H}^{\mathrm{T}}:\boldsymbol{K}_{ELM(i,j)}=\boldsymbol{h}\left(\boldsymbol{x}_i\right)\bullet\boldsymbol{h}\left(\boldsymbol{x}_j\right)=\boldsymbol{K}_{ij} \tag{10}$$

Consequently, the output function of KELM can be written as

$$
\begin{aligned}
f(\boldsymbol{x})&=\boldsymbol{h}(\boldsymbol{x})\mathbf{H}^{\mathrm{T}}\left(\frac{\mathbf{I}}{\mathbf{C}}+\mathbf{H}\mathbf{H}^{\mathrm{T}}\right)^{-1}\mathbf{T}\\
&=\begin{bmatrix} k(\boldsymbol{x},\boldsymbol{x}_1)\\ \mathrm{M}\\ k(\boldsymbol{x},\boldsymbol{x}_N)\end{bmatrix}^{\mathrm{T}}\left(\frac{\mathbf{I}}{C}+\boldsymbol{K}_{ELM}\right)^{-1}\mathbf{T}
\end{aligned}
\qquad (11)
$$

The wavelet is a finite-length waveform that is characterized as follows: (1) either time domain or approximation is compactly supported; And (2) the advantages of wavelet analysis are integrated into the time and frequency domains. The time domain characteristics of wavelet transform can be expressed with wavelet functions that are translated from a wavelet kernel function. Meanwhile, we apply different wavelet functions to approximate the original signals. Wavelet analysis is a relatively new signal processing tool that is widely applied by many researchers in power systems due to its excellent characteristics. Thus, the combination of the wavelet analysis technique and KELM is significant.

If a function satisfies the Mercer theorem, then it is a kernel function. The function can also be used as a kernel function of the KELM. Many wavelet functions satisfy the translation-invariant kernel conditions, including the Mexican hat wavelet kernel function [13]. In this study, WKELM is utilized for regression.

2.3 PWKELM

Given a wavelet kernel function $k(\boldsymbol{x},\boldsymbol{y})$, the function $k(\boldsymbol{x},\boldsymbol{x}_i)$ corresponds to a wavelet kernel function for each training sample $\boldsymbol{x}_i$. A set of wavelet kernel functions $V=\{k(\boldsymbol{x},\boldsymbol{x}_i)|i=1,2,\cdots,N\}$ is called a dictionary. PWKELM is a sequential forward greedy algorithm that selects a new wavelet kernel function at each iteration until some stopping criteria are satisfied. Two key components of PWKELM must be solved: the recruitment of the wavelet kernel functions and the solution of subproblem. PWKELM recruits a new wavelet kernel function $k(\boldsymbol{x},\boldsymbol{x}_p)$ from the set $\{k(\boldsymbol{x},\boldsymbol{x}_i)|i\in Q\}$ according to some criteria by starting with an empty index set $S=\varnothing$ and a full index set $Q=\{1,\cdots,N\}$. Then, the index p is removed from Q and added to S. At the $(n-1)$ th iteration, the number of wavelet kernel functions in the set $\{k(\boldsymbol{x},\boldsymbol{x}_i)|i\in S\}$ is presumably $n-1$. Eq. (9) is rearranged as

$$\bar{\boldsymbol{K}}_S\boldsymbol{\alpha}_S^{*(n-1)}=\mathbf{T} \qquad (12)$$

where $\bar{\boldsymbol{K}}_S=[\bar{\boldsymbol{k}}_i,\cdots,\bar{\boldsymbol{k}}_j](i,j\in S)$ with $\bar{\boldsymbol{k}}_i=\boldsymbol{k}_i+\boldsymbol{e}_i/C$, $\bar{\boldsymbol{k}}_i$ is the i th column vector of $\bar{\boldsymbol{K}}$, and $\boldsymbol{\alpha}_S^{*(n-1)}$ is a subvector consisting of elements that are confined to the index set S at the $(n-1)$ th iteration. Eq. (12) is an overdetermined linear equation set; its solution is simply equivalent to finding the optimal solution to the following problem

$$\min_{\boldsymbol{\alpha}_S^{(n-1)}}\left\{G_S^{(n-1)}=\left\|\bar{\boldsymbol{K}}_S\boldsymbol{\alpha}_S^{(n-1)}-\mathbf{T}\right\|\right\} \qquad (13)$$

The optimal solution of Eq. (13) can be analytically determined as follows

$$\boldsymbol{\alpha}_S^{*(n-1)}=\left(\bar{\boldsymbol{K}}_S^{\,T}\bar{\boldsymbol{K}}_S\right)^{-1}\bar{\boldsymbol{K}}_S^{\,T}\mathbf{T} \qquad (14)$$

Thus, we can get the minimizer of (13) with

$$G_S^{*(n-1)}=\left\|\bar{\boldsymbol{K}}_S\boldsymbol{\alpha}_S^{*(n-1)}-\mathbf{T}\right\| \qquad (15)$$

Eq. (13) indicates that the larger the $G_S^{*(n-1)}$ value is, the worse the approximate effectiveness becomes. If $k(\boldsymbol{x},\boldsymbol{x}_p)$ is recruited as the new wavelet kernel function at the n th iteration, then $G_{S\cup\{p\}}^{*(n)}$ can be obtained by following by the similar lines

$$G_{S\cup\{p\}}^{*(n)}=\left\|\begin{bmatrix}\bar{\boldsymbol{K}}_S & \bar{\boldsymbol{k}}_p\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_S^{*(n)}\\ \alpha_p^{*}\end{bmatrix}-\mathbf{T}\right\| \qquad (16)$$

Consequently, $G_{S\cup\{p\}}^{*(n)}\le G_S^{*(n-1)}$, and $\Delta G_p^{(n)}\ge 0$. Thus, the criterion of recruiting the next wavelet kernel function at the n th iteration is obtained as follows

$$p=\arg\max_{i\in Q}\left\|\Delta G_i^{(n)}\right\| \qquad (17)$$

where $Q=\{1,\ldots,N\}\setminus S$. Eq. (17) indicates that at each iteration, the wavelet kernel function introducing the most significant decrease on G is recruited as the new kernel function. Eq. (13) can be solved by using Eq. (14) to calculate G and ΔG at each iteration. The use of Eq. (14) incurs two potential risks: on the one hand, the reliability and robustness of calculating $\left(\bar{\boldsymbol{K}}_S^{\,T}\bar{\boldsymbol{K}}_S\right)^{-1}$ is closely related to its condition number, defined as

$$\kappa\left(\bar{\boldsymbol{K}}_S^{\,T}\bar{\boldsymbol{K}}_S\right)=\kappa^2\left(\bar{\boldsymbol{K}}_S\right)=\frac{\mu_{\max}^2}{\mu_{\min}^2} \qquad (18)$$

where $\mu_{\max}$ and $\mu_{\min}$ represent the maximum and minimum nonzero singular values of $\bar{\boldsymbol{K}}_S$, respectively. In general, the larger the condition number is, the less stable the numerically calculating result is. The condition number easily incurs roundoff errors if it is excessively large. On the other hand, the recursive strategy is more difficult to use for recruiting new wavelet kernel function at the n th iteration. Hence, an improved method of computing ΔG is determined.

Based on knowledge regrading matrix transformation, matrix $\bar{\boldsymbol{K}}_S$ can be decomposed as $\bar{\boldsymbol{K}}_S=\boldsymbol{Q}_{N\times N}^{(n-1)}\begin{bmatrix}\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}\\ \mathbf{0}_{(N+1-n)\times(n-1)}\end{bmatrix}$ through QR decomposition [14]. In the present study, $\boldsymbol{Q}_{N\blacklozenge N}^{(n-1)}$ is an orthogonal matrix that satisfies $\left(\boldsymbol{Q}_{N\times N}^{(n-1)}\right)^T\boldsymbol{Q}_{N\times N}^{(n-1)}=\boldsymbol{Q}_{N\times N}^{(n-1)}\left(\boldsymbol{Q}_{N\times N}^{(n-1)}\right)^T$ and $\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}$ is an upper triangular matrix with the same rank as $\bar{\boldsymbol{K}}_S$. Thus,

$$\begin{aligned}G_S^{(n-1)}&=\left\|\bar{\boldsymbol{K}}_S\boldsymbol{\alpha}_S{}^{(n-1)}-\mathbf{T}\right\|=\left\|\left(\boldsymbol{Q}_{N\times N}^{(n-1)}\right)^T\bar{\boldsymbol{K}}_S\boldsymbol{\alpha}_S{}^{(n-1)}-\left(\boldsymbol{Q}_{N\times N}^{(n-1)}\right)^T\mathbf{T}\right\|\\&=\left\{\left\|\begin{bmatrix}\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}\\ \mathbf{0}_{(N+1-n)\times(n-1)}\end{bmatrix}\boldsymbol{\alpha}_S{}^{(n-1)}-\begin{bmatrix}\widehat{\mathbf{T}}_{(n-1)\times1}^{(n-1)}\\ \tilde{\mathbf{T}}_{(N-n+1)\times1}^{(n-1)}\end{bmatrix}\right\|\right\}\\&=\left\|\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}\boldsymbol{\alpha}_S{}^{(n-1)}-\widehat{\mathbf{T}}_{n\times1}^{(n-1)}\right\|+\left\|\tilde{\mathbf{T}}_{(N-n+1)\times1}^{(n-1)}\right\|\end{aligned}\tag{19}$$

Given that $\left\|\tilde{\mathbf{T}}_{(N-n+1)\times1}^{(n-1)}\right\|$ is constant, the matrix is an upper triangular matrix and $\boldsymbol{\alpha}_S{}^{*(n-1)}$ can easily be obtained by solving

$$\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}\boldsymbol{\alpha}_S{}^{(n-1)}=\widehat{\mathbf{T}}_{n\times1}^{(n-1)}\tag{20}$$

For Eq. (14), $\kappa\left(\bar{\boldsymbol{K}}_S{}^T\bar{\boldsymbol{K}}_S\right)=\mu_{\max}^2/\mu_{\min}^2$; however, $\kappa\left(\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}\right)=\kappa\left(\bar{\boldsymbol{K}}_S\right)=\mu_{\max}/\mu_{\min}$ in Eq. (20). Generally, $\kappa>1$, which indicates that the optimal solution of Eq. (13) derived from Eq. (20) is more stable numerically than that from Eq. (14). In addition,

$$G_S^{*(n-1)}=\left\|\tilde{\mathbf{T}}_{(N-n+1)\times1}^{(n-1)}\right\|\tag{21}$$

In the following iteration, if $k\left(\boldsymbol{x},\boldsymbol{x}_p\right)$ is recruited as the new wavelet kernel function, then $G_{S\cup\{p\}}^{*(n)}$ is obtained through a lower-cost strategy rather than from scratch through Eq. (19).

Let $\left(\boldsymbol{Q}^{(n-1)}\right)^T\begin{bmatrix}\widehat{\boldsymbol{K}}_{(n-1)\times(N-n+1)}^{(n-1)}\\ \tilde{\boldsymbol{K}}_{(N+1-n)\times(N-n+1)}^{(n-1)}\end{bmatrix}=\bar{\boldsymbol{K}}_Q$. Householder matrix [15] is applied as follows

$$\begin{aligned}G_{S\cup\{p\}}^{(n)}&=\left\|\begin{bmatrix}I_{(n-1)\times(n-1)}&\mathbf{0}\\ \mathbf{0}&\boldsymbol{H}_p^{(n)}\end{bmatrix}\left(\boldsymbol{Q}_{N\times N}^{(n-1)}\right)^T\left(\left[\bar{\boldsymbol{K}}_S\ \bar{\boldsymbol{k}}_p\right]\begin{bmatrix}\boldsymbol{\alpha}_S{}^{(n)}\\ \alpha_p\end{bmatrix}-\mathbf{T}\right)\right\|=\left\|\begin{bmatrix}I_{(n-1)\times(n-1)}&\mathbf{0}\\ \mathbf{0}&\boldsymbol{H}_p^{(n)}\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}&\widehat{\boldsymbol{k}}_p^{(n-1)}\\ \mathbf{0}_{(N+1-n)\times(n-1)}&\tilde{\boldsymbol{k}}_p^{(n-1)}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_S{}^{(n)}\\ \alpha_p\end{bmatrix}-\begin{bmatrix}\widehat{\mathbf{T}}_{(n-1)\times1}^{(n-1)}\\ \tilde{\mathbf{T}}_{(N-n+1)\times1}^{(n-1)}\end{bmatrix}\right)\right\|\\&=\left\|\left(\begin{bmatrix}\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}&\widehat{\boldsymbol{k}}_p^{(n-1)}\\ \mathbf{0}_{1\times(n-1)}&-\operatorname{sign}\left(\tilde{\boldsymbol{k}}_p^{(n-1)}\right)_1\left\|\tilde{\boldsymbol{k}}_p^{(n-1)}\right\|\\ \mathbf{0}_{(N-n)\times(n-1)}&\mathbf{0}_{(N-n)\times1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_S{}^{(n)}\\ \alpha_p\end{bmatrix}-\begin{bmatrix}\widehat{\mathbf{T}}_{(n-1)\times1}^{(n-1)}\\ \widehat{\mathrm{T}}_p^{(n)}\\ \tilde{\mathbf{T}}_{(N-n)\times1}^{(n)}\end{bmatrix}\right)\right\|=\left\|\left(\begin{bmatrix}\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}&\widehat{\boldsymbol{k}}_p^{(n-1)}\\ \mathbf{0}_{1\times(n-1)}&-\operatorname{sign}\left(\tilde{\boldsymbol{k}}_p^{(n-1)}\right)_1\left\|\tilde{\boldsymbol{k}}_p^{(n-1)}\right\|\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_S{}^{(n)}\\ \alpha_p\end{bmatrix}-\begin{bmatrix}\widehat{\mathbf{T}}_{(n-1)\times1}^{(n-1)}\\ \widehat{\mathrm{T}}_p^{(n)}\end{bmatrix}\right)\right\|+\left\|\tilde{\mathbf{T}}_{(N-n)\times1}^{(n)}\right\|\end{aligned}\tag{22}$$

where $\widehat{\boldsymbol{k}}_p^{(n-1)}$ and $\tilde{\boldsymbol{k}}_p^{(n-1)}$ are derived from $\widehat{\boldsymbol{K}}_{(n-1)\times(N-n+1)}^{(n-1)}$ and $\tilde{\boldsymbol{K}}_{(N+1-n)\times(N-n+1)}^{(n-1)}$ corresponding to the index p. $\boldsymbol{H}_p^{(n)}$ is derived as

$$\boldsymbol{H}_p^{(n)}==\boldsymbol{I}-2\frac{\boldsymbol{v}_p^{(n)}\left(\boldsymbol{v}_p^{(n)}\right)^T}{\left(\boldsymbol{v}_p^{(n)}\right)^T\boldsymbol{v}_p^{(n)}}\tag{23}$$

where

$$\boldsymbol{v}_p^{(n)}=\tilde{\boldsymbol{k}}_p^{(n-1)}+sign(\left(\tilde{\boldsymbol{k}}_p^{(n-1)}\right)_1)\left\|\tilde{\boldsymbol{k}}_p^{(n-1)}\right\|\boldsymbol{e}_1\tag{24}$$

$\boldsymbol{R}_{n\times n}^{(n)}$, together with $\widehat{\mathbf{T}}_{n\blacklozenge1}^{(n-1)}$, is updated as

$$\boldsymbol{R}_{n\times n}^{(n)}=\begin{bmatrix}\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}&\widehat{\boldsymbol{k}}_p^{(n-1)}\\ \mathbf{0}_{1\times(n-1)}&-\operatorname{sign}\left(\tilde{\boldsymbol{k}}_p^{(n-1)}\right)_1\left\|\tilde{\boldsymbol{k}}_p^{(n-1)}\right\|\end{bmatrix},\quad \widehat{\mathbf{T}}_{n\times1}^{(n)}=\begin{bmatrix}\widehat{\mathbf{T}}_{(n-1)\times1}^{(n-1)}\\ \widehat{\mathrm{T}}_p^{(n)}\end{bmatrix}\tag{25}$$

Thus, the optimal solution at the n th iteration is determined with

$$\begin{bmatrix}\boldsymbol{R}_{(n-1)\times(n-1)}^{(n-1)}&\widehat{\boldsymbol{k}}_p^{(n-1)}\\ \mathbf{0}_{1\times(n-1)}&-\operatorname{sign}\left(\tilde{\boldsymbol{k}}_p^{(n-1)}\right)_1\left\|\tilde{\boldsymbol{k}}_p^{(n-1)}\right\|\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_S{}^{(n)}\\ \alpha_p\end{bmatrix}=\begin{bmatrix}\widehat{\mathbf{T}}_{(n-1)\times1}^{(n-1)}\\ \widehat{\mathrm{T}}_p^{(n)}\end{bmatrix}\tag{26}$$

The following is obtained as well

$$G_{S\cup\{p\}}^{*(n)}=\left\|\tilde{\mathbf{T}}_{(N-n)\times1}^{(n)}\right\|\tag{27}$$

Hence, according to Eqs. (21) , (22), and (27), the following is obtained

$$\begin{aligned}\Delta G_p^{(n)} &= \left\|\tilde{\mathbf{T}}_{(N-n+1)\times 1}^{(n-1)}\right\| - \left\|\tilde{\mathbf{T}}_{(N-n)\times 1}^{(n)}\right\| \\ &= \left\|\boldsymbol{H}_p^{(n)}\tilde{\mathbf{T}}_{(N-n+1)\times 1}^{(n-1)}\right\| - \left\|\tilde{\mathbf{T}}_{(N-n)\times 1}^{(n)}\right\| = \left\|\widehat{\mathrm{T}}_p^{(n)}\right\|\end{aligned} \tag{28}$$

The criterion considered in recruiting the next wavelet kernel function is

$$p = \arg\max_{i\in Q}\left\|\widehat{\mathrm{T}}_i^{(n)}\right\| \tag{29}$$

According to Eq. (22), $\widehat{\mathrm{T}}_p^{(n)}$ is easily gained explicitly as

$$\begin{aligned}\widehat{\mathrm{T}}_i^{(n)} &= \left(\boldsymbol{H}_p^{(n)}\tilde{\mathbf{T}}_{(N-n+1)\times 1}^{(n-1)}\right)_1 \\ &= \left(\tilde{\mathbf{T}}_{(N-n+1)\times 1}^{(n-1)}\right)_1 - 2\frac{\left(\boldsymbol{v}_i^{(n)}\right)_1\left(\boldsymbol{v}_i^{(n)}\right)^T\tilde{\mathbf{T}}_{(N-n+1)\times 1}^{(n-1)}}{\left(\boldsymbol{v}_i^{(n)}\right)^T\boldsymbol{v}_i^{(n)}}\end{aligned} \tag{30}$$

where $(\cdot)_1$ represents the extraction of the first element of a given column vector. $\boldsymbol{H}_p^{(n)}$ should be constructed when the new wavelet kernel function $k\left(\boldsymbol{x},\boldsymbol{x}_p\right)$ is determined; subsequently, Eq. (25) and the index sets $S \leftarrow S \cup \{p\}$, and $Q \leftarrow Q \backslash \{p\}$ are updated. Then, $\widehat{\boldsymbol{K}}_{(n)\times(N-n-1)}^{(n)}$ and $\tilde{\boldsymbol{K}}_{(N-n)\times(N-n-1)}^{(n)}$ can be refreshed as

$$\begin{bmatrix}\widehat{\boldsymbol{K}}_{(n)\times(N-n)}^{(n)} \\ \tilde{\boldsymbol{K}}_{(N-n)\times(N-n)}^{(n-1)}\end{bmatrix} = \begin{bmatrix} I_{(n-1)\times(n-1)} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_p^{(n)}\end{bmatrix}\begin{bmatrix}\widehat{\boldsymbol{K}}_{(n-1)\times(N-n+1)}^{(n-1)} \backslash \widehat{\boldsymbol{k}}_p^{(n-1)} \\ \tilde{\boldsymbol{K}}_{(N+1-n)\times(N-n+1)}^{(n-1)} \backslash \tilde{\boldsymbol{k}}_p^{(n-1)}\end{bmatrix} \tag{31}$$

In this study, the value of index n starts from 1. Thus, we must first initialize PWKELM. Let

$$\bar{\boldsymbol{k}}_i = \boldsymbol{Q}_i\boldsymbol{r}_i\left(i = 1,\cdots,N\right) \tag{31}$$

where $\bar{\boldsymbol{k}}_i$ is obtained from $\bar{\boldsymbol{K}} = \boldsymbol{K}_{ELM} + \boldsymbol{I}/C$ that corresponds to the index i . $\boldsymbol{Q}_i$ is obtained via QR decomposition. According to Eq. (19), $\tilde{\mathbf{T}}_i^{(0)}$ is derived as

$$\tilde{\mathbf{T}}_i^{(0)} = \boldsymbol{Q}_i\mathbf{T} \backslash \widehat{\mathbf{T}}_i^{(0)} \tag{32}$$

Thus, the first wavelet kernel function is selected as follows

$$p = \arg\min_{i\in Q}\left\|\tilde{\mathbf{T}}_i^{(0)}\right\| \tag{33}$$

In summary, the flowchart of PWKELM is depicted as follows

Step 1) Initializations

- Obtain the training data set $\left\{\left(\boldsymbol{x}_i, t_i\right)\right\}_{i=1}^N$.
- Choose the appropriate regularization and wavelet kernel parameters.
- Let $S = \varnothing$ and $Q = \left\{1,\cdots,N\right\}$, $n = 1$.
- Calculate $\tilde{\mathbf{T}}_i^{(0)}$ according to (32).
- Recruit the first $k\left(\boldsymbol{x},\boldsymbol{x}_p\right)$ according to (33).
- Let $S \leftarrow S \cup \{p\}$, $Q \leftarrow Q \backslash \{p\}$, $n = 2$.
- Let $\boldsymbol{H}^{(1)} = \boldsymbol{Q}_p$, $\boldsymbol{R}^{(1)} = \left(\boldsymbol{r}_p\right)_1$, calculate $\widehat{\boldsymbol{K}}_{1\times N}^{(1)}$ and $\tilde{\boldsymbol{K}}_{(N-1)\times N}^{(1)}$ via $\begin{bmatrix}\widehat{\boldsymbol{K}}_{1\times N}^{(1)} \\ \tilde{\boldsymbol{K}}_{(N-1)\times N}^{(1)}\end{bmatrix} = \boldsymbol{H}^{(1)}\left(\boldsymbol{K}_{ELM} + \boldsymbol{I}/C\right)$, compute $\widehat{\mathbf{T}}_{1\times 1}^{(1)}$ and $\tilde{\mathbf{T}}_{(N-1)\times 1}^{(1)}$ according to $\begin{bmatrix}\widehat{\mathbf{T}}_{1\times 1}^{(1)} \\ \tilde{\mathbf{T}}_{(N-1)\times 1}^{(1)}\end{bmatrix} = \boldsymbol{H}^{(1)}\mathbf{T}$.
- Choose a positive M or a small $\varepsilon > 0$.

Step 2) If $n > M$ or $\dfrac{\sum_{i=0}^{n}\left(\widehat{\mathrm{T}}_p^{(i)}\right)^2}{\left\|\mathbf{T}\right\|} < \varepsilon$

Step 3) Go to step 11

Step 4) Else

Step 5) Calculate $\boldsymbol{v}_i^{(n)}$ and $\widehat{\mathrm{T}}_i^{(n)}$ according to (24) and (30) respectively.

Step 6) Choose the next $k\left(\boldsymbol{x},\boldsymbol{x}_p\right)$ according to (29).

Step 7) Update $\boldsymbol{R}^{(n)}$ and $\widehat{\mathbf{T}}^{(n)}$ according to (25).

Step 8) Update $\widehat{\boldsymbol{K}}_{(n)\times(N-n)}^{(n)}$ and $\tilde{\boldsymbol{K}}_{(N-n)\times(N-n)}^{(n)}$ according to (31).

Step 9) Let $S \leftarrow S \cup \{p\}$, $Q \leftarrow Q \backslash \{p\}$, $n = n+1$, go to step 2.

Step 10) End if

Step 11) Solve $\boldsymbol{R}^{(n)}\boldsymbol{\alpha}_S^{(n)} = \widehat{\mathbf{T}}_{n\times 1}^{(n)}$.

Step 12) Output $\boldsymbol{f}\left(\boldsymbol{x}\right) = \begin{bmatrix} k\left(\boldsymbol{x},\boldsymbol{x}_i\right) \\ M \\ k\left(\boldsymbol{x},\boldsymbol{x}_l\right)\end{bmatrix}_{(i,\cdots,l\in S)}^T \boldsymbol{\alpha}_S^{(n)}$.

3. Result Analysis and Discussion

In an attempt to highlight the efficacy of our proposed PWKELM， we conduct experiments with a synthetic dataset and a real-world example, i.e., a gas furnace instance. The proposed algorithm is also compared with other algorithms, such as LS-SVM, RKELM, and FSA-ELM. The Sigmoid function is used in ELM. We tune the regularization and width parameters in kernel function via 10-fold cross validation. Because wavelet kernel function has the capability to capture the local behavior of signals both in frequency and time, Mexican hat wavelet kernel function, without loss of generality, is utilized in the experiment. For convenient comparison, one performance index, namely, root mean squared error (RMSE), is defined as a derivation measurement between the target and the predictive values. In an effort to search for the average performance rather than the best one, 30 trials are conducted in each dataset for every algorithm. The average training time and the average RMSE are computed for both training and testing datasets.

3.1 Single-variable synthetic data

In this experiment, we approximate the following single-variable function

$$f(x) = x\sin(4\pi x)e^{1-x^2} + 2x^2\tanh(10x)\cos(2\pi x), x \in [0,1] \quad (34)$$

We uniformly generate 200 training sets of size 400. In a bid to make the regression problem “real”, normal distribution noise $(0,0.1^2)$ is added to all training samples, while testing data remain noise-free. The optimal regularization and kernel parameters are chosen from the set $\{20,30,\cdots,150\}\times\{0.01,0.02,\cdots,0.2\}$ via cross validation. WELM is compared with ELM, LS-SVM and ELM with RBF kernel. The experimental precisions are presented in Table 1. The time requirement of WKELM is comparable with those of ELM and RBF KELM but is slightly shorter than that of the RBF LS-SVM. The accuracy of WKELM, which is comparable with other algorithms on training samples, is the highest on testing samples. Wavelet kernel function can match well the original series on different scales. The generalization capability of ELM is clearly comparable with or is even better than LS-SVM.

Tab. 1. Comparison of different algorithms

Algorithm	Parameters	Times (s)		Training	Testing
	(C,σ^2)	Training	Testing	RMSE	RMSE
WKELM	$(60,0.04)$	0.0335	0.0094	0.0896	0.0124
RBF KELM	$(100,0.03)$	0.0336	0.0089	0.0895	0.0141
ELM	_	0.0287	0.0134	0.0893	0.0159
RBF LS-SVM	$(70,0.04)$	0.0381	0.0125	0.0900	0.0164

Figs.1 to 3 illustrate the experimental results for different sparse algorithms given only 10 wavelet kernel functions. PWKELM with only 10 wavelet kernel functions can attain an excellent generalization performance similar to WKELM, but RKELM and FSA-ELM lose approximation capability to some extent. On the premise of the same sparsity ratio of kernel functions, PWKELM proves to be superior in regression accuracy to RKELM and FSA-ELM, that is, new kernel functions must be recruited for other sparse algorithms to achieve the same generalization performance as PWKELM. Therefore, PWKELM obtains the best sparseness among these algorithms; this finding signifies that PWKELM can enhance computational efficiency in the testing phase, thereby improving its real time performance. With regard to computational stability, roundoff errors and numerical instability are easily incurred by calculating $\boldsymbol{\alpha}_S^{(n)}$ with $\left(\boldsymbol{K}_S{}^T\boldsymbol{K}_S\right)^{-1}$ directly in RKELM, but Householder matrix introduced in this study eliminates the bottleneck.

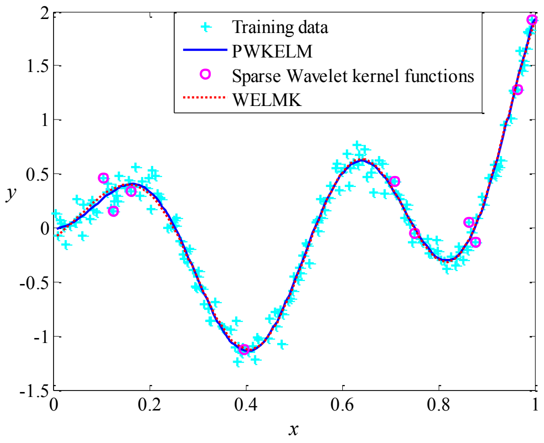


Fig.1. Simulation results of PWKELM

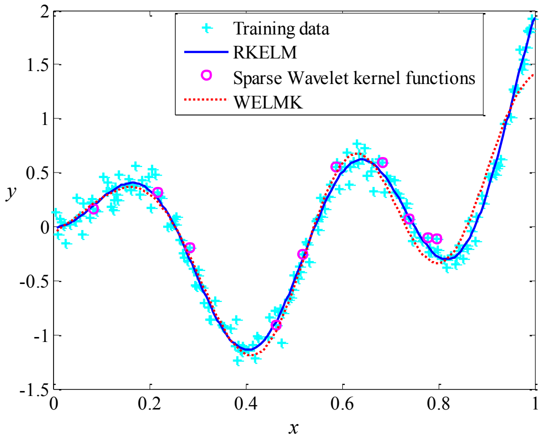


Fig.2. Simulation results of RKELM

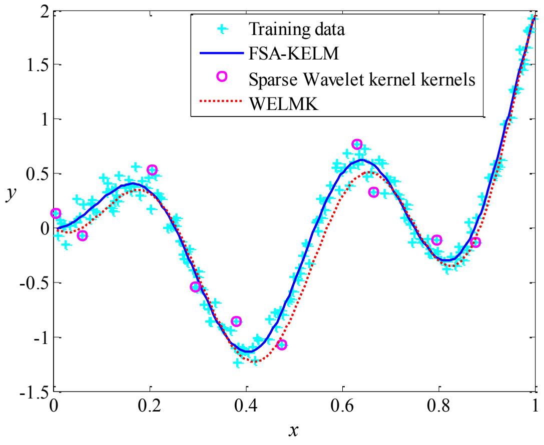


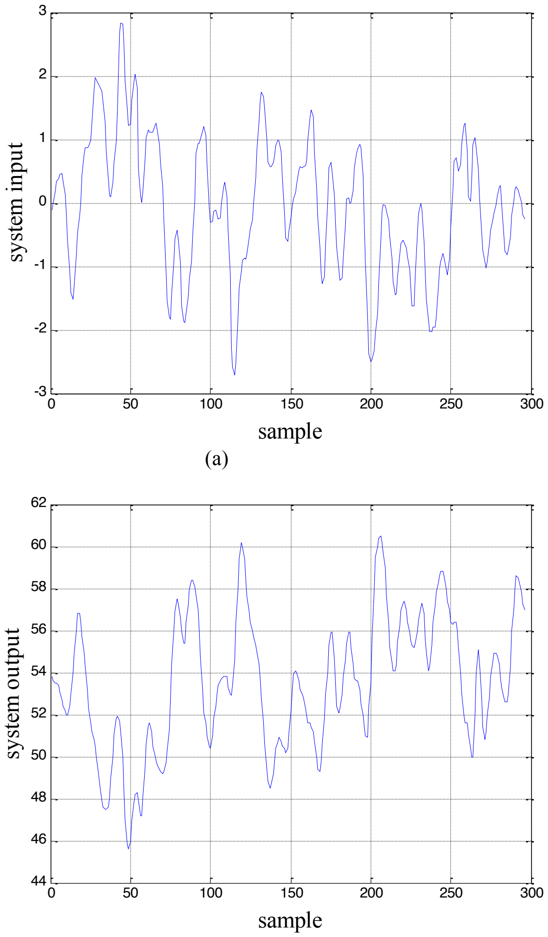
Fig.3. Simulation results of FSA-ELM

3.2 Gas furnace example

The gas furnace data set is a time series containing 296 pairs of input-output points as depicted in Fig. 3, where the input v_k is the coded input gas feed rate and the output y_k represents the CO$_2$ concentration from the gas furnace [16]. A total of 293 new data points $\{(\boldsymbol{x}_k, y_k)\}$ are derived from these pairs and constructed with $\boldsymbol{x}_k$ given as

$$\boldsymbol{x}_k = \left[y_{k\text{-}1}, y_{k\text{-}2}, y_{k\text{-}3}, v_{k\text{-}1}, \mathrm{v}_{k\text{-}2}, v_{k\text{-}3}\right]^T \quad (35)$$

Fig. 3 indicates that the gas furnace sample series fluctuates sharply, and the rear part is different from the front. Hence, the even-number pairs of $\{(\boldsymbol{x}_k, y_k)\}$ are used for training, whereas the odd-numbered pairs for testing. As a result, the training sets contain 146 data points, while the training sets have 147. The optimal regularization and kernel parameters are derived from the set $\{2^{-5}, 2^{-4}, \ldots, 2^{15}\}$ using cross validation, and the experimental results are presented in Table 2. Fig.4 exhibits the tendency of the performance index RMSE against the number of wavelet kernel functions.



(a)

(b)

Fig. 3. Gas furnace: (a) input v_k and (b) output y_k .

Tabla 2. Comparison of different kernels

Algorithm	Parameters (C,σ^2)	Times (s) Training	Testing	Training RMSE	Testing RMSE
WKELM	$(2^{15},2^{14})$	0.0210	0.0084	0.2650	0.2417
RBF KELM	$(2^{10},2^{9})$	0.0212	0.0103	0.2623	0.2540
Wavelet LS-SVM	$(2^{12},2^{14})$	0.0282	0.0085	0.2704	0.2595
RBF LS-SVM	$(2^{10},2^{12})$	0.0334	0.0096	0.2855	0.2861

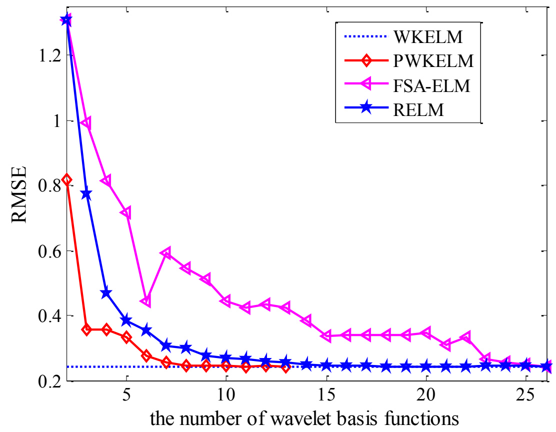


Fig.4. RMSE vs.the number of wavelet kernel functions on gas furnace instance

Table 2 implies that WKELM is relatively more accurate in prediction and efficient in calculating. ELM usually reports a generalization performance on regression that is superior to that of LS-SVM. WKELM gains advantages in different wavelet functions that approximate the details of numerous series. Fig4 illustrates that RMSE decreases with an increase in the number of wavelet kernel functions for every algorithm. The dashed line generated by WKELM is regarded as the benchmark line; the other results are terminated when they reach this line. PWKELM reaches the benchmark line first, which suggests that PWKELM requires the least number of wavelet kernel functions to attain almost the same generalization performance compared with the other sparse algorithms do. That is, the use of PWKELM serves to provide WKELM with a much sparser solution without sacrificing the generalization performance. The testing time is directly proportional to the number of kernel functions; thus, a sparser solution indicates the less testing time. A decreased testing time suggests improved real time performance. In sum, the experiment based on a gas furnace example demonstrates the feasibility and efficacy of the proposed PWKELM.

4. Conclusions

In this study, the use of the wavelet kernel function and a parsimonious algorithm in KELM, namely PWKELM, is proposed and investigated. KELM with wavelet kernel can more effectively capture essential features in “frequency-rich” signals than other kernel functions can. WKELM also loses the solution sparseness, thereby deteriorating the real time and increasing the computational burden; hence, the solution of WKELM must be sparsified. PWKELM recruits important wavelet kernel functions in the original dictionary according to some criterion successively; therefore, this algorithm contributes to effective generalization by excluding redundant wavelet kernel functions. Householder matrix is utilized during the process of iteration, thereby commendably circumventing the ill-conditioned subproblem and improving numerical stability. In a bid to confirm the effectiveness and feasibility of the proposed algorithm, we conduct numerous experiments, including a synthetic dataset and a gas furnace example. Based on the experiments, KELM with wavelet kernel performs better than RBF kernel or LS-SVM with wavelet kernel does, thereby indicating the superiority of ELM and wavelet technique. Moreover, PWKELM is superior to other parsimonious methods, such as RKELM and FSA-ELM, in terms of the number of wavelet kernel functions under nearly the same generalization performance. That is to say, PWKELM is able to generate a more parsimonious WKELM without impairing the generalization and identification accuracy. This outcome is paramount for the environments strictly demanding the computational efficiency in engineering applications.

Acknowledgements
The study was supported by the National Natural Science Foundation of China (Grant Nos. 11301120).

References

1. Vapnik V. N., "The Nature of Statistical Learning Theory". New York: Springer-Verlag, 1995.
2. Suykens J. A. K., Vandewalle J., "Least squares support vector machine classifiers". *Neural Processing Letters*, 9(3), 1999, pp.293–300.
3. Mall R., Suykens J.A.K., "Sparse reductions for fixed-size least squares support vector machines on large scale". Advances in Knowledge Discovery and Data Mining. *Lecture Notes in Computer Science*, 2013, pp.161–173.
4. Xia X. L., Jiao W., Li K., Irwin G., "A novel sparse least squares support vector machines". *Mathematical Problems in Engineering*, 2013(1), 2013, pp.681-703.
5. Zhao Y. P., Li B., Li Y. B., Wang K. K., "Householder transformation based sparse least squares support vector regression". *Neurocomputing*, 161, 2015, pp.243-253.
6. Huang G. B., Zhu Q. Y., Siew C. K., "Extreme learning machine: Theory and applications". *Neurocomputing*, 70(1), 2006, pp.489-501.
7. Huang G. B., Zhou H., Ding X., Zhang R., "Extreme learning machine for regression and multiclass classification". *IEEE Transactions on Systems, Man and Cybernetics part B, Cybernetics*, 42 (2), 2012, pp.513-529.
8. Man Z., Lee K., Wang D., Cao Z., Khoo S., "Robust single-hidden layer feedforward network-based pattern classifier". *IEEE Transactions on Neural Networks & Learning Systems*, 23(12), 2012, pp.1974-1986.
9. Grigorievskiy A., Miche Y., Ventelä A. M., Séverin E., Lendasse A., "Long-term time series prediction using OP-ELM". *Neural Networks*, 51(3), 2014, pp.50-56.
10. Wang J., Guo C. L., "Wavelet kernel extreme learning classifier". *Microelectron &Computer*, 10, 2013, pp.73-76+80.
11. Deng W. Y., Zheng Q. H., Zhang K., "Reduced kernel extreme learning machine". *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, Milkow, Poland: Springer International Publishing, 2013, pp.63-69.
12. Li X. D., Mao W. J, Jiang W., "Fast sparse approximation of extreme learning machine". *Neurocomputing*, 128(5), 2014, pp.96-103
13. Ding S. F., Wu F. L., Shi Z. Z., "Wavelet twin support vector machine". *Neural Computing & Applications*, 25(6), 2014, pp.1241-1247.
14. Zhang X., "Matrix Analysis and Applications".Tsinghua University Press, Beijing, China, 2004.
15. Dubrulle A. A., "Householder transformations revisited". *Siam Journal on Matrix Analysis & Applications*, 22(1), 2000, pp.33-40.
16. S. Chen, X. Hong, C. J. Harris. "Particle swarm optimization aided orthogonal forward regression for unified data modeling". *IEEE Transaction on Evolutionary Computation*, 14(4), 2010, pp.477-499.