Contents lists available at ScienceDirect

Neurocomputing

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

Confidence-weighted extreme learning machine for regression problems

Zhigen Shang [*], Jianqiang He

Department of Automation, Yancheng Institute of Technology, Yancheng 224003, Jiangsu, China

ARTICLE INFO

Article history:
Received 15 February 2014
Received in revised form
9 May 2014
Accepted 2 July 2014
Communicated by G.-B. Huang
Available online 14 July 2014

Keywords:
Extreme learning machine
Relative entropy
Gaussian margin machine
Regression

ABSTRACT

Based on Gaussian margin machine (GMM) and extreme learning machine (ELM), confidence-weighted ELM (CW-ELM) is proposed to provide point forecasts and confidence intervals. CW-ELM maintains a multivariate normal distribution over the output weight vector. It is applied to seek the least informative distribution from those that keep the targets within the forecast confidence intervals. For simplicity, the covariance matrix is assumed to be diagonal. The simplified problem of CW-ELM is approximately solved by using Leave-One-Out-Incremental ELM (LOO-IELM) and the interior point method. Our experimental results on both synthetic and real-world regression datasets demonstrate that CW-ELM has better performance than Bayesian ELM and Gaussian process regression.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Extreme learning machine (ELM), which is an efficient learning algorithm for single-hidden layer feedforward neural networks (SLFNs), has been recently proposed by Huang et al. [1]. ELM randomly initializes the parameters of the hidden layer and the weights of the output layer are analytically computed by using Moore–Penrose generalized inverse. Thus, ELM obtains an extremely low computational time. However, the generalization ability of ELM is influenced by changing the number of hidden neurons. Like other similar models based on feedforward neural networks, ELM also needs to address the problem of determining the optimal number of hidden neurons.

Many approaches have been proposed to determine the most suitable structure of ELM. Deconstructive methods (or pruning methods) are one type of algorithms to solve this problem. Rong et al. proposed a pruned ELM (P-ELM) for classification problems [2]. Miche et al. developed a method named optimally pruned ELM (OP-ELM) in [3] and its improvement in [4]. Deconstructive methods, however, in general are inefficient since the most of time they are dealing with a network that is larger than necessary.

There are also some researchers managing to solve the problem based on constructive methods (or growing methods). Huang and Chen presented the incremental ELM (I-ELM) [5] and its modifications [6–7], which are examples of constructive methods. Wang et al. proposed an algorithm for architecture selection of ELM based on localized generalization error model [8]. But in these methods, the expected training accuracy or the maximum number of hidden neurons needs to be set in advance. Yu et al. proposed a method called Leave-One-Out-Incremental ELM (LOO-IELM) in which the LOO error is directly calculated by the PRESS statistics [9]. LOO-IELM adds hidden nodes one-by-one and stops automatically based on the stop criteria. However, the PRESS has the problem of numerical instabilities because of the use of a pseudo-inverse in the calculation. Fortunately, the Tikhonov-regularized PRESS can eliminate this problem [4].

In many regression problems, it is advantageous to have both point forecasts and confidence intervals (CIs). To derive CIs, neural networks are usually combined with other methods, such as bootstrap methods [10] and Bayesian methods [11]. Bootstrap methods are nonparametric approaches of statistical inference based on re-sampling. Their high computational cost makes them less attractive. Bayesian methods for neural networks have been researched intensively in recent years due to their efficiency and effectiveness [12]. For example, Bayesian neural network was employed for rainfall–runoff modeling in [13] and for short time load forecasting in [14]. It is worth noting that Emilio et al. presented a Bayesian approach to extreme learning machine and proposed Bayesian ELM (BELM) in which a normal distribution was introduced on the output weight vector [15]. Compared with ELM, BELM has the advantages of allowing regularization automatically and producing point forecasts and CIs simultaneously. However, BELM lacks proper adaptability to complex noise (e.g., heteroscedastic noise) since an isotropic normal distribution is

[*] Corresponding author.
E-mail address: zgshang@ycit.edu.cn (Z. Shang).

http://dx.doi.org/10.1016/j.neucom.2014.07.009
0925-2312/© 2014 Elsevier B.V. All rights reserved.

used. Moreover, Gaussian process regression (GPR) can provide both point forecasts and CIs simultaneously [16]. The hyperparameters of GPR are estimated by maximizing the likelihood of the samples. Like BELM, GPR also lacks the adaptability to complex noise. Gaussian margin machine (GMM) [17] offers another method for obtaining CIs. GMM, originally proposed for linear classification problems, assumes that the weight vector follows a multivariate normal distribution, and aims to seek the least informative distribution that classifies each training sample with a high probability. The probability that a sample belongs to a certain class is automatically provided by GMM.

The present study proposes confidence-weighted extreme learning machine (CW-ELM) for regression problems by combining GMM and ELM. The output weight vector of CW-ELM follows a multivariate normal distribution. The method aims to seek the least informative distribution from those that keep the targets within the forecast CIs. The covariance matrix of the normal distribution is taken to be diagonal for simplicity, and the simplified problem is approximately solved by two steps. The first step is to implement LOO-IELM and to substitute the results into the simplified problem. That is, the hidden layer parameters of LOO-IELM are used as those of CW-ELM, and its corresponding output weight vector is set to be the mean vector of the normal distribution. The second step is to solve the final problem by using the interior point method [18]. It should be noted that, in LOO-IELM in this study, the Tikhonov-regularized PRESS is applied instead of the PRESS. Like BELM and GPR, CW-ELM offers the CIs automatically. The diagonal covariance matrix used in CW-ELM is more complex than the isotropic one adopted in BELM. Thus, CW-ELM may have proper adaptability to complex noise.

The rest of this paper is organized as follows: Gaussian margin machine is introduced in Section 2, which is followed by Section 3 describing some preliminaries of LOO-ELM and BELM. In Section 4, CW-ELM is proposed. Our CW-ELM is evaluated using synthetic and real-world regression datasets, and CW-ELM is compared with BELM and GPR in Section 5. Section 6 draws the final conclusions.

2. Gaussian margin machine

Suppose the samples $\{(\boldsymbol{x}_i, o_i)\}_{i=1}^{l}$, where $\boldsymbol{x}_i \in \mathbf{R}^m$ is a column vector and $o_i \in \{-1, 1\}$ is a scalar output. The weight vector $\boldsymbol{w}_1$ of the linear classifier is assumed to follow a normal (Gaussian) distribution $N_m(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, where $\boldsymbol{\mu}_1 \in \mathbf{R}^m$ is a column vector and $\boldsymbol{\Sigma}_1 \in \mathbf{R}^{m \times m}$ is a definite matrix. For the sample $\boldsymbol{x}_i$, we get

$$\boldsymbol{x}_i^T \boldsymbol{w}_1 \sim N(\boldsymbol{x}_i^T \boldsymbol{\mu}_1, \boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i), \tag{1}$$

where T means matrix transposition.

The linear classifier is required to correctly classify the sample $\boldsymbol{x}_i$ with a high probability, that is

$$\Pr(o_i \boldsymbol{x}_i^T \boldsymbol{w}_1 \geq 0) \geq \delta, \tag{2}$$

where $\delta \in (0.5, \quad 1]$ is a confidence parameter.

Combining Eqs. (1) and (2) yields

$$\Pr\left(\frac{o_i \boldsymbol{x}_i^T \boldsymbol{w}_1 - o_i \boldsymbol{x}_i^T \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i}} \leq \frac{-o_i {\boldsymbol{x}_i}^T \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i}} \right) \leq 1 - \delta. \tag{3}$$

GMM is designed to seek the least informative distribution that will classify the training samples with high probability, which is implemented by seeking a distribution with minimum relative entropy with respect to an isotropic distribution $N_m(\boldsymbol{0}, a\boldsymbol{I}_m)$, where a is a prior parameter. The optimization problem of GMM can be expressed as

$$\begin{aligned} &\min_{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1} \; D_{KL}(N_m(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \| N_m(\boldsymbol{0}, a\boldsymbol{I}_m)) \\ &s.t. \quad \Pr\left(\frac{o_i {\boldsymbol{x}_i}^T \boldsymbol{w}_1 - o_i \boldsymbol{x}_i^T \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i}} \leq \frac{-o_i \boldsymbol{x}_i^T \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i}} \right) \leq 1 - \delta, \\ &\boldsymbol{\Sigma}_1 > 0, \quad i = 1, \cdots, l, \end{aligned} \tag{4}$$

where D_{KL} stands for the relative entropy of $N_m(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $N_m(\boldsymbol{0}, a\boldsymbol{I}_m)$, which can be calculated by

$$\frac{1}{2} \ln \det(a\boldsymbol{I}_m \boldsymbol{\Sigma}_1^{-1}) + \frac{1}{2} tr\left((a\boldsymbol{I}_m)^{-1} (\boldsymbol{\mu}_1 \boldsymbol{\mu}_1^T + \boldsymbol{\Sigma}_1 - a\boldsymbol{I}_m) \right). \tag{5}$$

By disregarding the constant terms of objective function and transforming the constraints of Eq. (4), the problem can be reformulated as

$$\begin{aligned} &\min_{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1} \tfrac{1}{2}(-\ln \det \boldsymbol{\Sigma}_1 + \tfrac{1}{a} tr(\boldsymbol{\Sigma}_1) + \tfrac{1}{a} \|\boldsymbol{\mu}_1\|^2) \\ &\quad s.t. \quad o_i \boldsymbol{x}_i^T \boldsymbol{\mu}_1 \geq \boldsymbol{\Phi}^{-1}(\delta) \sqrt{\boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i} \\ &\quad \boldsymbol{\Sigma}_1 > 0, \quad i = 1, \cdots, l, \end{aligned} \tag{6}$$

where $\boldsymbol{\Phi}^{-1}$ denotes the inverse cumulative distribution function of a standard normal distribution.

The two-sided PAC-Bayesian theorem ensures that GMM is of desirable generalization performance, and the proof process is presented by [17].

3. Preliminaries

ELM can be considered as universal approximation, and has been applied in many fields. This section will briefly describe ELM, LOO-IELM and BELM.

3.1. Extreme learning machine

Let $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^{l}$ be a sample set where $\boldsymbol{x}_i \in \mathbf{R}^m$ is the ith input vector and $t_i \in \mathbf{R}$ is its corresponding target. In ELM, the hidden layer parameters are randomly initialized. ELM is mathematically modeled by

$$y_i = \boldsymbol{g}(\boldsymbol{x}_i)^T \boldsymbol{\mu}_2, \tag{7}$$

where $\boldsymbol{g}(\boldsymbol{x}_i) \in \mathbf{R}^p$ is the output vector of the hidden layer, $\boldsymbol{\mu}_2 \in \mathbf{R}^p$ is the output weight vector, and p is the number of the hidden neurons. ELM computes the output weight vector $\boldsymbol{\mu}_2 = \boldsymbol{H}^\dagger \boldsymbol{t}$ by Moore–Penrose generalized inverse, where $\boldsymbol{H} = [g(\boldsymbol{x}_1), \ldots, g(\boldsymbol{x}_l)]^T$, and $\boldsymbol{t} = [t_1, \ldots, t_l]^T$. Thus, ELM has an extremely low computational time. However, Moore–Penrose generalized inverse leads ELM to suffer from the overfitting problem. Regularization is one of methods to improve the generalization performance of ELM. Regularized ELM aims to minimize not only the training error but also the norm of output weights. The regularized methods include Lasso, Tikhonov, and elastic net [9]. In this study, Tikhonov regularization is used, and is described as

$$\begin{aligned} &\min_{\boldsymbol{\mu}_2, \boldsymbol{\xi}} \tfrac{1}{2} \left(C \sum_{i=1}^{l} \xi_i^2 + \|\boldsymbol{\mu}_2\|^2 \right) \\ &s.t. \quad \boldsymbol{g}(\boldsymbol{x}_i)^T \boldsymbol{\mu}_2 - t_i = \xi_i, \quad i = 1, \ldots, l, \end{aligned} \tag{8}$$

where C is the regularization parameter.

Using the KKT conditions, the output weight vector $\boldsymbol{\mu}_2$ can be analytically computed as

$$\boldsymbol{\mu}_2 = \left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}^T \boldsymbol{H} \right)^{-1} \boldsymbol{H}^T \boldsymbol{t}. \tag{9}$$

In addition to the regularized methods, many other approaches have been proposed to improve the generalization performance of ELM. For example, Zhai et al. proposed a dynamic ensemble ELM based on sample entropy [19] and developed an approach of fusion of extreme learning machine with fuzzy integral [20], and Chacko et al. presented a combination method using wavelet energy feature and ELM [21]. A survey paper on the ELM method has been published, aiming to introduce ELM's historical development, newest advances, and main advantages and disadvantages [22].

3.2. Leave-One-Out-Incremental ELM

The LOO error is used in LOO-IELM to select the best neurons from a cluster of n random neurons. The PRESS formula can be used to directly provide the LOO error, and is expressed as

$$MSE^{PRESS} = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{t_i - \boldsymbol{H}_i(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{t}}{1-\boldsymbol{H}_i(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}_i^T}\right)^2, \tag{10}$$

where $\boldsymbol{H}_i$ is the ith row of $\boldsymbol{H}$. The main procedures of LOO-IELM are shown in Fig. 1. More details of LOO-IELM are provided in [9].

However, The PRESS has the problem of numerical instabilities since a pseudo-inverse is used in Eq. (10). Fortunately, the Tikhonov regularization can be applied to address this problem. The Tikhonov regularized PRESS formula is described as

$$MSE^{TR-PRESS} = \frac{1}{l}\sum_{i=1}^{l}\left(\frac{t_i - \boldsymbol{H}_i((I/C)+\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{t}}{1-\boldsymbol{H}_i((I/C)+\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}_i^T}\right)^2. \tag{11}$$

3.3. Bayesian ELM

Bayesisan ELM optimizes the output weight vector based on Bayesian linear regression. The model describes the relationship by

$$y = \mathbf{g}(\boldsymbol{x})^T\boldsymbol{w}_2 + \varepsilon, \tag{12}$$

where $\varepsilon \sim N(0,\rho^2)$. Thus, we have

$$y \sim N(\mathbf{g}(\boldsymbol{x})^T\boldsymbol{w}_2, \rho^2). \tag{13}$$

The output weight vector $\boldsymbol{w}_2$ is committed to be an isotropic normal distribution $N(\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$. The posterior distribution is recognized as a normal distribution with a mean value $\boldsymbol{m}$ and a covariance matrix $\boldsymbol{S}$ defined as

$$\boldsymbol{m} = \frac{1}{\rho^2}\boldsymbol{S}\boldsymbol{H}^T t, \tag{14}$$

$$\boldsymbol{S} = (\alpha^{-1}\boldsymbol{I} + \rho^{-2}\boldsymbol{H}^T\boldsymbol{H})^{-1}. \tag{15}$$

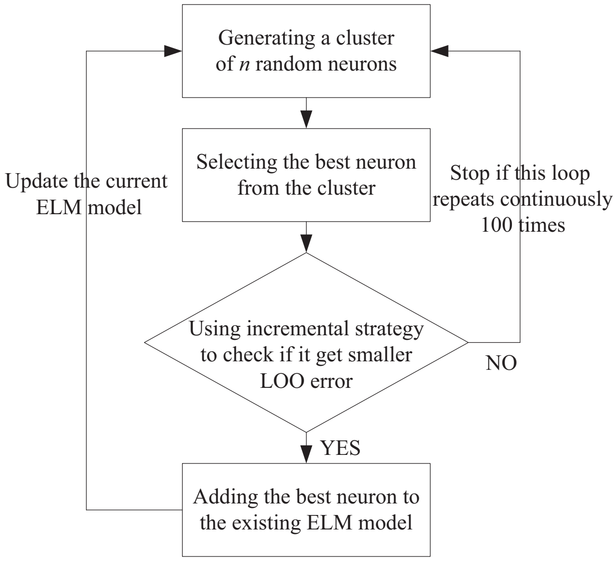
Generating a cluster of n random neurons
Selecting the best neuron from the cluster
Using incremental strategy to check if it get smaller LOO error
YES
NO
Adding the best neuron to the existing ELM model
Update the current ELM model
Stop if this loop repeats continuously 100 times

Fig. 1. The main procedures of LOO-IELM.

Parameters from the posterior distribution are optimized using the ML-II [23] or Evidence Procedure [24]. This process is an iterative one, and Eqs. (14)–(18) are calculated in each iteration

$$\zeta = p - \alpha tr(\boldsymbol{S}), \tag{16}$$

$$\alpha = \frac{\zeta}{\boldsymbol{m}^T\boldsymbol{m}}, \tag{17}$$

$$\rho^2 = \frac{\sum_{i=1}^{l}(t_i - \mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{m})^2}{l-\zeta}. \tag{18}$$

The iterative process is stopped when the difference of the 2-norm of $\boldsymbol{m}$ between successive iterations is less than a given value. For a test observation $\boldsymbol{x}^*$, the point forecast is $\mathbf{g}(\boldsymbol{x}^*)^T\boldsymbol{m}$, and the forecast confidence interval is

$$[\mathbf{g}(\boldsymbol{x}^*)^T\boldsymbol{m} - \eta\nu(\boldsymbol{x}^*), \mathbf{g}(\boldsymbol{x}^*)^T\boldsymbol{m} + \eta\nu(\boldsymbol{x}^*)] \tag{19}$$

where $\eta > 0$ and $\nu(\boldsymbol{x}^*) = \sqrt{\mathbf{g}(\boldsymbol{x}^*)^T\boldsymbol{S}\mathbf{g}(\boldsymbol{x}^*) + \rho^2}$. $\eta\nu(\boldsymbol{x}^*)$ is termed as the forecast uncertainty.

4. Confidence-weighted extreme learning machine

4.1. Formulation of CW-ELM

Like BELM, a distribution is maintained over alternative weight vectors by assuming $\boldsymbol{w} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\mu} \in \mathbf{R}^p$ and the positive definite covariance matrix $\boldsymbol{\Sigma} \in \mathbf{R}^{p\times p}$. Let y_i be equal to $\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{w}$. Then, we obtain

$$y_i \sim N(\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\mu}, \mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\Sigma}g(\boldsymbol{x}_i)). \tag{20}$$

CW-ELM aims to seek the least informative distribution from those that keep the targets within the forecast confidence intervals. Thus, we have the following constraints:

$$\begin{aligned} &\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\mu} - \eta\sqrt{\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\Sigma}g(\boldsymbol{x}_i)} \le t_i, \\ &\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\mu} + \eta\sqrt{\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\Sigma}g(\boldsymbol{x}_i)} \ge t_i, \quad i = 1, \dots, l. \end{aligned} \tag{21}$$

Using Eqs. (5) and (21) and disregarding the constant terms of the objective function, we obtain

$$\begin{aligned} &\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \ -\frac{1}{2}\ln\det\boldsymbol{\Sigma} + \frac{1}{2a}\boldsymbol{\mu}^T\boldsymbol{\mu} + \frac{1}{2a}tr(\boldsymbol{\Sigma}) \\ &s.t. \quad \mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\mu} - \eta\sqrt{\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\Sigma}g(\boldsymbol{x}_i)} \le t_i, \\ &\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\mu} + \eta\sqrt{\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\Sigma}g(\boldsymbol{x}_i)} \ge t_i, \quad i = 1, \dots, l. \end{aligned} \tag{22}$$

Eq. (22) is complicated and hard to solve. Thus, simplifying Eq. (22) is taken into account. Here, the covariance matrix $\boldsymbol{\Sigma}$ is assumed to be a diagonal one. Let $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}$ wherein $\boldsymbol{\Lambda} = diag(\lambda_1, \dots, \lambda_p)$ and $\lambda_i > 0$, $i = 1, \dots, p$. Then, Eq. (22) is simplified as

$$\begin{aligned} &\min_{\boldsymbol{\mu},\boldsymbol{\Lambda}} \ -\frac{1}{2}\sum_{i=1}^{p}\ln\lambda_i + \frac{1}{2a}\boldsymbol{\mu}^T\boldsymbol{\mu} + \frac{1}{2a}\sum_{i=1}^{p}\lambda_i \\ &s.t. \quad \mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\mu} - \eta\sqrt{\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\Lambda}g(\boldsymbol{x}_i)} \le t_i, \\ &\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\mu} + \eta\sqrt{\mathbf{g}(\boldsymbol{x}_i)^T\boldsymbol{\Lambda}g(\boldsymbol{x}_i)} \ge t_i, \quad i = 1, 2, \dots, l. \end{aligned} \tag{23}$$

4.2. Solution of the simplified problem

For computational efficiency, we apply the implementation results of LOO-IELM and the interior point method to seek the approximate solution of Eq. (23). Suppose the hidden layer output matrix $\overline{\boldsymbol{H}} = [\overline{g}(\boldsymbol{x}_1), \ldots, \overline{g}(\boldsymbol{x}_l)]^T$ and the corresponding output weight vector $\overline{\boldsymbol{\mu}}$ stand for the implementation results of LOO-IELM. Substituting $\overline{\boldsymbol{H}}$ and $\overline{\boldsymbol{\mu}}$ into Eq. (23) and ignoring the term $(1/2a)\boldsymbol{\mu}^T\boldsymbol{\mu}$ in the objective function, we rewrite Eq. (23) as

$$\begin{aligned} \min_{\Lambda} \quad & -\tfrac{1}{2}\sum_{i=1}^{p} \ln \lambda_i + \frac{1}{2a_i}\sum_{i=1}^{p} \lambda_i \\ s.t. \quad & \overline{\mathbf{g}}(\boldsymbol{x}_i)^T \Lambda \overline{\mathbf{g}}(\boldsymbol{x}_i) \geq \frac{1}{\eta^2}\left|t_i - \overline{\mathbf{g}}(\boldsymbol{x}_i)^T \overline{\boldsymbol{\mu}}\right|^2, \quad i = 1, \ldots, l. \end{aligned} \tag{24}$$

Let $\boldsymbol{G}$ be $\overline{\boldsymbol{H}} \odot \overline{\boldsymbol{H}}$, where $\odot$ denotes the element-wise product. Eq. (24) is transformed into

$$\begin{aligned} \min_{\Lambda} \quad & -\tfrac{1}{2}\sum_{i=1}^{p} \ln \lambda_i + \frac{1}{2a_i}\sum_{i=1}^{p} \lambda_i \\ s.t. \quad & G\boldsymbol{\lambda} \geq \overline{\boldsymbol{\xi}}. \end{aligned} \tag{25}$$

where $\overline{\boldsymbol{\xi}} = [\overline{\xi}_1, \ldots \overline{\xi}_p]^T$, $\overline{\xi}_i = \frac{1}{\eta^2}\left|t_i - \overline{\mathbf{g}}(\boldsymbol{x}_i)^T \overline{\boldsymbol{\mu}}\right|^2$, and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_p]^T$.

The solution of Eq. (25) can be calculated by iterative approaches, such as the active set method and the interior point method. In this study, the interior point method is chosen. Let $\overline{\Lambda}$ be the solution of Eq. (25). Since the weight vector is assumed to follow a normal distribution, CW-ELM yields the forecast result with the form of a normal distribution. For a test observation $\boldsymbol{x}^*$, the point forecast is $\mathbf{g}(\boldsymbol{x}^*)^T\overline{\boldsymbol{\mu}}$. And, the forecast confidence interval is calculated as

$$[\mathbf{g}(\boldsymbol{x}^*)^T\overline{\boldsymbol{\mu}} - \eta\sigma(\boldsymbol{x}^*), \mathbf{g}(\boldsymbol{x}^*)^T\overline{\boldsymbol{\mu}} + \eta\sigma(\boldsymbol{x}^*)], \tag{26}$$

where $\sigma(\boldsymbol{x}^*) = \sqrt{\mathbf{g}(\boldsymbol{x}^*)^T\overline{\Lambda}g(\boldsymbol{x}^*)}$. The forecast uncertainty of CW-ELM is $\eta\sigma(\boldsymbol{x}^*)$.

The algorithm for approximately solving problem (23) is described in the following:

Step 1: Make independent samples $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^{l}$.
Step 2: Select the activation function, and implement Tikhonov regularized LOO-IELM.
Step 3: Substitute $\overline{\boldsymbol{H}}$ and $\overline{\boldsymbol{\mu}}$ into Eq. (25).
Step 4: The solution of Eq. (25), which is denoted as $\overline{\Lambda}$, is calculated by the interior point algorithm. Given a test observation $\boldsymbol{x}^*$, the point forecast is $\mathbf{g}(\boldsymbol{x}^*)^T\overline{\boldsymbol{\mu}}$, and the forecast confidence interval is calculated as Eq. (26).

Summing up, CW-ELM has the following advantages:

(1) The accuracy of point forecasts of CW-ELM is guaranteed by LOO-IELM. It is worth noting that any improved ELM (e.g., optimally pruned ELM [3], Weighted ELM [25], and Robust ELM [26]) can be used instead of LOO-IELM.
(2) The standard deviation $\sigma(\boldsymbol{x}^*)$ of CW-ELM contains more free parameters than that of BELM, indicating its better adaptability to complex noise (e.g., heteroscedastic noise).

5. Experiments

In this section, the experiments on synthetic and real-world regression datasets were made to prove the effectiveness of the proposed CW-ELM model. The models were built using MATLAB 7.7. The experiments were conducted on a computer with a Win7 32 bit OS running on 3.1-GHz Intel Core i5-3450 with 4 GB RAM.

The hyper-parameter C in LOO-IELM has a great influence on the model performance. The proper value of C was selected from $\{2^{-20}, 2^{-18}, \ldots, 2^{20}\}$, and was determined by minimizing the LOO error. The Gaussian function was used as the activation function. In CW-ELM, the value of η is determined based on the confidence level at which the forecast interval includes the target. To make the confidence level higher than 95%, η should be greater than 1.96 computed by $\Phi^{-1}(1-(1-0.95)/2)$. In this study, $\eta = 1.96$. The value of a was selected from $\{2^{-20}, 2^{-19}, \ldots, 2^{0}\}$, and the average probability density of the whole training set (averaged over 20 times) was taken as the selection criterion. The hyper-parameter p in BELM was chosen from the set $\{25, 35, \ldots, 195\}$, and was determined by 5-fold cross-validation. To compare different models, the statistics, including RMSE, confidence interval coverage, and average length of CIs, were taken into account.

5.1. Synthetic regression datasets

Two synthetic datasets were used here. The first example was to model the sinc function, which is defined by

$$\sin \mathbf{c}(x) = \begin{cases} \frac{\sin(x)}{x} & x \neq 0, \\ 1 & x = 0. \end{cases} \tag{27}$$

A training set and testing set, which included 200 and 100 instances respectively, were created with x_i sampled uniformly from $[-3\pi, 3\pi]$. Moreover, random noise drawn uniformly from $[-0.2, 0.2]$ was added to both sets. The parameter of the LOO-IELM model for the sinc function was selected as $C = 2^{20}$. The value of a in CW-ELM was 2^{-13}.

Figs. 2–4 show the testing results of three models, demonstrating that for BELM, there are three points that fall outside of their corresponding CIs, while for GPR and CW-ELM, there is only one point that is not included in its corresponding CI. Three models were implemented 20 times. The statistics are shown in Table 1. Although CW-ELM needs more computational time than BELM, it performs better in other statistics. And, CW-ELM outperforms GPR.

The second example was to model data with heteroscedastic noise. The dataset was produced by

$$y = 0.2 \;\sin(2\pi x) + 0.2x^2 + 0.3 + (0.1x^2 + 0.05)e, \tag{28}$$

where x and noise e were drawn uniformly at random from [0, 1] and $[-1, 1]$, respectively. The training set and testing set respectively include 200 and 100 instances. For LOO-IELM, the optimal parameter turned out to be $C = 2^{16}$. The value of a in CW-ELM was chosen as $a = 2^{-20}$.

Figs. 5–7 display the testing results of three models. CW-ELM has no point that is outside its corresponding CI. The forecast uncertainties are shown in Fig. 8, which demonstrates that

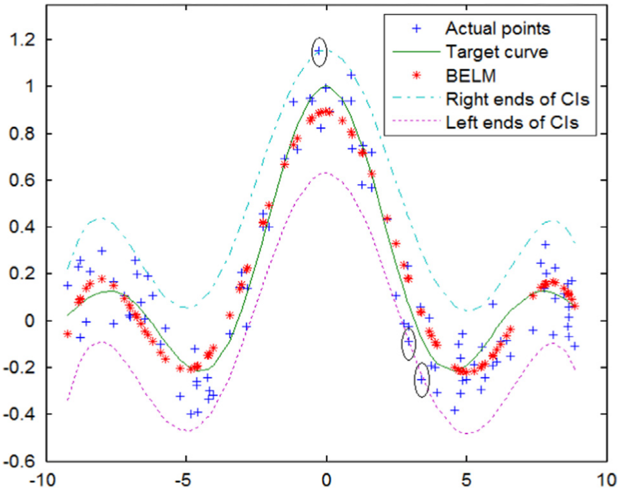

Fig. 2. Testing results of BELM on sinc dataset.

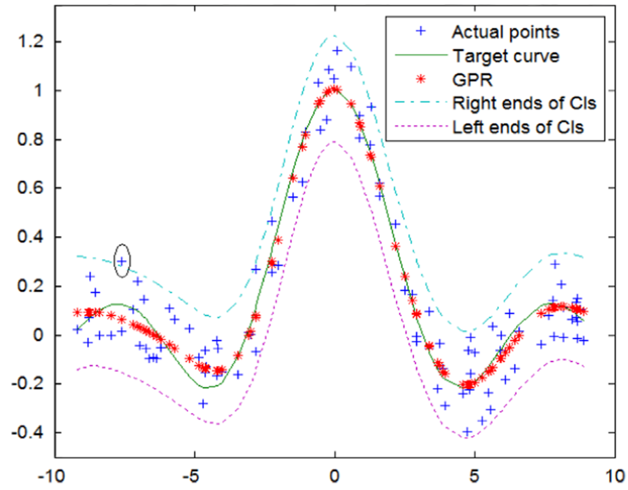

Fig. 3. Testing results of GPR on sinc dataset.

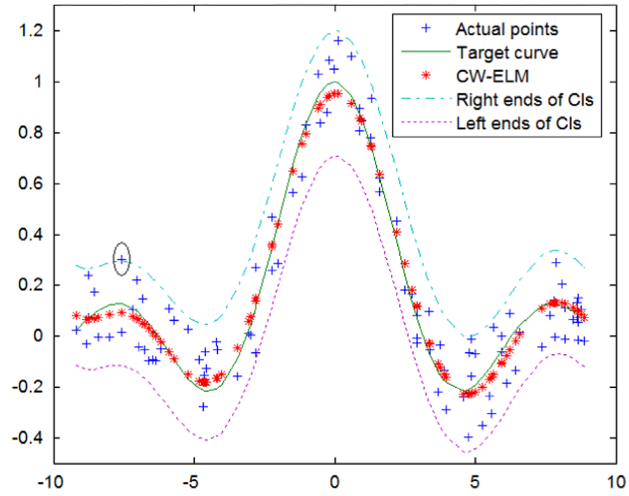

Fig. 4. Testing results of CW-ELM on sinc dataset.

Table 1
Testing results of three models on two synthetic datasets.

Data sets	Models	Average testing time (s)	RMSE	CI coverage (%)	Average length of CIs
Sinc	BELM	0.704	0.131	96.800	0.503
	GPR	2.143	0.114	99.000	0.447
	CW-ELM	0.725	0.110	99.500	0.446
Heteroscedastic	BELM	0.268	0.062	97.250	0.228
	GPR	2.007	0.057	95.000	0.227
	CW-ELM	0.294	0.057	100.000	0.221

CW-ELM captures the heteroscedastic characteristics of the data. The statistics indicate that CW-ELM has the shortest average length of CIs and the highest CI coverage.

5.2. Real-world regression datasets

Nine real datasets were used here to verify the effectiveness of CW-ELM. For the first example, we used product design time dataset that has been used in [27,28]. The second dataset was Silverman's motorcycle dataset [29], which contains

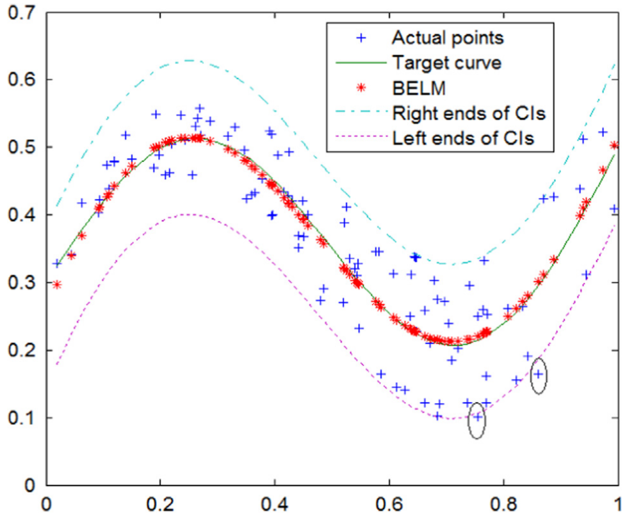

Fig. 5. Testing results of BELM on heteroscedastic dataset.

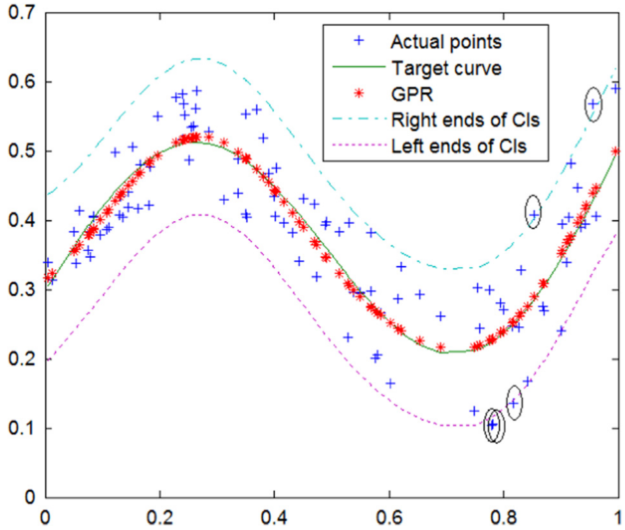

Fig. 6. Testing results of GPR on heteroscedastic dataset.

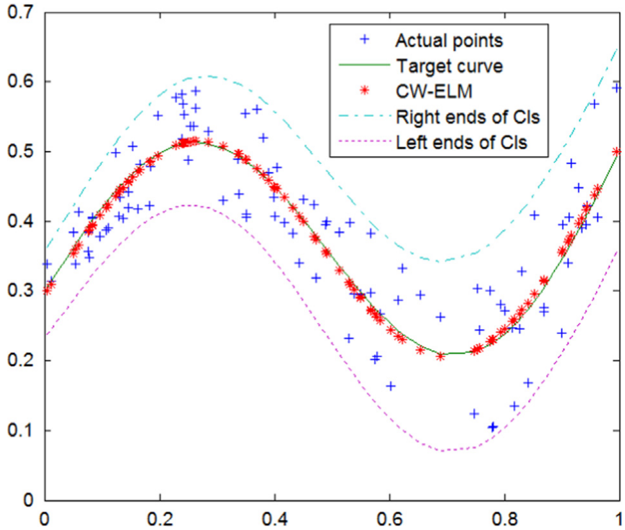

Fig. 7. Testing results of CW-ELM on heteroscedastic dataset.

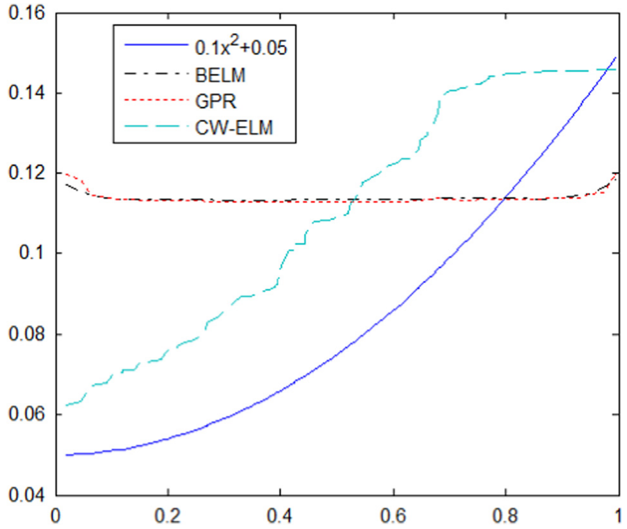

Fig. 8. Forecast uncertainty of three models on heteroscedastic dataset.

Table 2
Information about selected real datasets.

Data sets	Number of variables	Number of training samples	Number of testing samples
Product	6	60	12
Motorcycle	1	120	13
Slump	7	80	23
Boston Housing	13	400	106
Machine CPU	6	180	29
Forest Fires	12	450	67
Energy Efficiency Efficency	7	600	168
Airfoil Self-Noise Self-Noise	5	1200	303
Yacht Hydrodynamics	6	250	58

heteroscedastic noise. The other seven datasets were all from the UCI repository [30]. The basic information of the selected real datasets is shown in Table 2. The data in the training set was normalized to be within [0,1], and the testing data was also normalized using the same parameters used for the training set.

The first dataset has six time factors, the first three of which are expressed as linguistic variables and the last three as numerical ones. The linguistic variables, VL, *L*, *M*, *H* and VH, were transformed into the crisp values: 0.1, 0.25, 0.5, 0.75 and 0.95, respectively. CW-ELM was trained with 60 instances, and the remaining instances were left for testing. For the product design time dataset, the optimal parameter of LOO-IELM was $C=2^{16}$, and the value of a in CW-ELM was chosen as $a=2^{-15}$. The statistics indicate that compared with BELM, CW-ELM has a shorter average length of CIs and a better CI coverage. Moreover, CW-ELM obtains comparable performance with GPR.

For the second dataset, 13 instances were randomly selected for testing and the remaining instances were used for training. The optimal parameter of LOO-IELM was chosen as $C=2^{20}$. The value of a in CW-ELM was 2^{-20}. The statistics show that CW-ELM has a shorter average length of CIs and a better CI coverage as compared to BELM (Table 3).

The other seven datasets were selected from the UCI repository. The statistics indicate that compared with BELM, CW-ELM has a smaller RMSE, a shorter average length of CIs and a better CI coverage. And, CW-ELM has better performance than GPR with respect to RMSE.

Table 3
Testing results of three models on selected real datasets.

Data sets	Models	Average testing time (s)	RMSE	CI coverage (%)	Average length of CIs
Product	BELM	0.118	0.030	85.417	0.088
	GPR	1.787	0.029	91.655	0.066
	CW-ELM	0.122	0.027	91.667	0.070
Motorcycle	BELM	0.390	0.164	89.230	0.602
	GPR	1.044	0.126	92.304	0.439
	CW-ELM	0.416	0.127	100.000	0.575
Slump	BELM	0.471	0.028	86.957	0.086
	GPR	2.149	0.035	91.304	0.115
	CW-ELM	0.498	0.025	91.304	0.084
Boston Housing	BELM	0.625	0.187	68.870	0.381
	GPR	9.500	0.236	61.321	0.423
	CW-ELM	1.778	0.227	82.075	0.445
Machine CPU	BELM	0.407	0.047	75.862	0.123
	GPR	2.178	0.050	89.655	0.109
	CW-ELM	1.008	0.038	97.732	0.114
Forest Fires	BELM	0.889	0.084	98.507	0.210
	GPR	7.849	0.084	98.506	0.208
	CW-ELM	0.681	0.083	99.102	0.234
Energy Efficiency	BELM	4.813	0.016	94.047	0.053
	GPR	6.499	0.012	96.429	0.049
	CW-ELM	3.574	0.013	94.313	0.051
Airfoil Self-Noise	BELM	3.491	0.138	89.439	0.481
	GPR	48.406	0.146	94.057	0.637
	CW-ELM	2.950	0.139	91.874	0.502
Yacht Hydrodynamics	BELM	1.388	0.010	96.552	0.038
	GPR	3.107	0.008	94. 880	0.024
	CW-ELM	1.274	0.008	95.462	0.031

6. Conclusions

This paper has presented CW-ELM by combining GMM and ELM. The output weight vector of CW-ELM follows a multivariate normal distribution. The method aims to minimize the relative entropy with respect to a fixed isotropic distribution when the target values are within the forecast CIs. For the sake of simplicity, the covariance matrix is assumed to be diagonal. The implementation results of LOO-IELM are substituted into the simplified problem, and the interior point algorithm is then applied for obtaining an approximate solution.

Experiments on synthetic and real-world regression datasets are conducted for convincing evaluation. Results from them have verified that the proper accuracy of point forecasts is guaranteed by LOO-IELM, and that, compared with BELM, CW-ELM promises better performance in terms of RMSE and forecast CIs. Moreover, CW-ELM outperforms GPR with respect to RMSE, and shows its good adaptability to heteroscedastic noise.

However, CW-ELM is only applied to the data without outliers. How to make CW-ELM applicable to the data with outliers would be a desired topic worthy of future study.

Acknowledgments

This work is supported by research fund of Key Laboratory for Advanced Technology in Environmental Protection of Jiangsu Province (No. AE201121). We thank the reviewers for their valuable comments and suggestions.

References

[1] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[2] H.J. Rong, Y.S. Ong, A.H. Tan, Z. Zhu, A fast pruned-extreme learning machine for classification problem, Neurocomputing 72 (2008) 359–366.

[3] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, Opelm: optimally-pruned extreme learning machine, IEEE Trans. Neural Netw. 21 (2010) 158–162.
[4] Y. Miche, M. van Heeswijk, P. Bas, O. Simula, A. Lendasse, Trop-elm: a double-regularized elm using Lars and Tikhonov regularization, Neurocomputing 74 (2011) 2413–2421.
[5] G.B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (2007) 3056–3062.
[6] G.B. Huang, M.B. Li, L. Chen, C.K. Siew, Incremental extreme learning machine with fully complex hidden nodes, Neurocomputing 71 (2008) 576–583.
[7] G. Feng, G.B. Huang, Q. Lin, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, IEEE Trans. Neural Netw. 20 (8) (2009) 1352–1357.
[8] X. Wang, Q. Shao, M. Qi, J. Zhai, Architecture selection for networks trained with extreme learning machine using localized generalization error model, Neurocomputing 102 (2013) 1–9.
[9] Q. Yu, Y. Miche, E. Severin, A. Lendasse, Bankruptcy prediction using extreme learning machine and financial expertise, Neurocomputing 128 (2014) 296–302.
[10] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall, New York, 1993.
[11] P. Congdon, Bayesian Statistical Modelling, Wiley, New York, 2007.
[12] C. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, New York, 2006.
[13] M.S. Khan, P. Coulibaly, Bayesian neural network for rainfall–runoff modeling, Water Resour. Res. 42 (7) (2006) W07409.
[14] P. Lauret, E. Fock, R.N. Randrianarivony, J.F. Manicom-Ramsamy, Bayesian neural network approach to short time load forecasting, Energy Convers. Manag. 49 (5) (2008) 1156–1166.
[15] S.O. Emilio, G.S. Juan, D.M. José, V.F. Joan, M. Marcelino, R.M. José, J.S. Antonio, BELM: Bayesian extreme learning machine, IEEE Trans. Neural Netw. 22 (3) (2011) 505–509.
[16] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, Massachusetts, 2006.
[17] K. Crammer, M. Mohri, F. Pereira, Gaussian margin machines, in: Proceedings of 12th International Conference on Artificial Intelligence Statistics, vol. 5, 2009, pp. 105–112.
[18] R.H. Byrd, J.C. Gilbert, J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, Math. Program. 89 (1) (2000) 149–185.
[19] J. Zhai, H. Xu, X. Wang, Dynamic ensemble extreme learning machine based on sample entropy, Soft Comput. 16 (9) (2012) 1493–1502.
[20] J. Zhai, H. Xu, Y. Li, Fusion of extreme learning machine with fuzzy integral, Int. J. Uncertain. Fuzziness Knowl. Based Syst. 21 (2013) 23–34.
[21] B. Chacko, V. Krishnan, G. Raju, P. Anto, Handwritten character recognition using wavelet energy and extreme learning machine, Int. J. Mach. Learn. Cybern. 3 (2) (2012) 149–161.
[22] G.B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. 2 (2011) 107–122.
[23] J.O. Berger, Statistical Decision Theory and Bayesian Analysis, Springer-Verlag, New York, 1985.
[24] D.J.C. MacKay, Probable networks and plausible predictions—a review of practical Bayesian methods for supervised neural networks, Network: Comput. Neural 6 (3) (1995) 469–505.
[25] W.W. Zong, G.B. Huang, Y.Q. Chen, Weighted extreme learning machine for imbalance learning, Neurocomputing 101 (2013) 229–242.
[26] P. Horata, S. Chiewchanwattana, K. Sunat, Robust extreme learning machine, Neurocomputing 102 (2013) 31–44.
[27] D. Xu, H.S. Yan, An intelligent estimation method for product design time, Int. J. Adv. Manuf. Technol. 30 (7–8) (2006) 601–613.
[28] H.S. Yan, D. Xu, An approach to estimating product design time based on fuzzy ν-support vector machine, IEEE Trans. Neural Netw. 18 (3) (2007) 721–731.
[29] B.W. Silverman, Some aspects of the spline smoothing approach to nonparametric regression curve fitting, J. R. Stat. Soc. 47 (1985) 1–52.
[30] A. Frank, A. Asuncion, UCI Machine Learning Repository, 2011, ⟨http://archive.ics.uci.edu/ml⟩.

Zhigen Shang was born in 1979. He received the M.S. degree and Ph.D. degree in control theory and engineering from Shanghai University in 2006, and from Southeast University in 2013, respectively. Currently, he is a lecturer in the Department of Automation at Yancheng Institute of Technology. His research interests include intelligent algorithm, and applications of forecasting technology.

Jianqiang He received the B.S. degree in automatic control from Wuhan University of Technology, Wuhan, China, in 1986 and the M.S. degree in automatic control theory and application from Southeast University, Nanjing, in 2000. Since August 2008, he has been a professor with the Department of Automation, Yancheng Institute of Technology. His research interests include neural networks and fuzzy systems.