Ensemble Stochastic Configuration Networks for Estimating Prediction Intervals: A Simultaneous Robust Training Algorithm and Its Application

Jun Lu, Jinliang Ding, *Senior Member, IEEE*, Xuewu Dai, *Member, IEEE*, and Tianyou Chai, *Fellow, IEEE*

***Abstract*— Obtaining accurate point prediction of industrial processes' key variables is challenging due to the outliers and noise that are common in industrial data. Hence the prediction intervals (PIs) have been widely adopted to quantify the uncertainty related to the point prediction. In order to improve the prediction accuracy and quantify the level of uncertainty associated with the point prediction, this article estimates the PIs by using ensemble stochastic configuration networks (SCNs) and bootstrap method. The estimated PIs can guarantee both the modeling stability and computational efficiency. To encourage the cooperation among the base SCNs and improve the robustness of the ensemble SCNs when the training data are contaminated with noise and outliers, a simultaneous robust training method of the ensemble SCNs is developed based on the Bayesian ridge regression and M-estimate. Moreover, the hyperparameters of the assumed distributions over noise and output weights of the ensemble SCNs are estimated by the expectation–maximization (EM) algorithm, which can result in the optimal PIs and better prediction accuracy. Finally, the performance of the proposed approach is evaluated on three benchmark data sets and a real-world data set collected from a refinery. The experimental results demonstrate that the proposed approach exhibits better performance in terms of the quality of PIs, prediction accuracy, and robustness.**

***Index Terms*— Bayesian ridge regression, bootstrap, ensemble stochastic configuration networks, M-estimate, prediction intervals, simultaneous robust training.**

I. Introduction

ACCURATE prediction of key variables of industrial processes is critical for managers and engineers to make the right decisions to optimize manufacturing and production. At present, the commonly used modeling methods to predict these key variables are mostly based on point prediction. These point-prediction-based methods only provide potential varying trends of key variables to managers and engineers for decision-making. However, the performance of point prediction degrades significantly when noise and outliers presented in the observed data set. Unfortunately, the industrial data usually contain outliers due to the measurement errors and human mistakes, which would decrease the prediction accuracy [1]. Moreover, the point-prediction-based method does not provide any indication of the prediction accuracy and cannot characterize the reliability of the prediction results [2]. In the actual decision-making process, the uncertainty related to the point prediction should be taken into account [3].

In recent years, the prediction interval (PI) has received more and more attention due to its advantage of providing both the point prediction and the corresponding potential uncertainty together. The essence of PIs is an estimate of an interval in which a future value of the expected output variable will fall with a certain confidence level (CL) [4]. The common algorithms of estimating PIs include the Bayesian [5], bootstrap [6], delta [7], mean–variance estimation (MVE) [8], and the lower–upper bound estimation (LUBE) [9]. The LUBE method does not need assumption on the data distribution [10]. However, the evolutionary algorithms that are adopted to train the neural networks are time-consuming. In addition, the LUBE-based PIs cannot provide point prediction value [11]. In the delta technique, the noise is assumed to be homogeneous, which is not true in numerous real-world applications [3]. The Bayesian method uses Bayes' theory to train the neural networks, which can avoid overfitting [12], [13], however, it suffers from low prediction accuracy with small sample size. The performance of the Bayesian method relies on prior knowledge. Although the MVE technique requires less computational cost, it only considers the errors caused by the noise without taking the error caused by the model mismatch into account. Hence, in real-world applications, it will result in unreliable PIs. All the aforementioned Bayesian, delta, and MVE are single-neural network-based methods and their quality of PIs and the accuracy of point prediction are limited. Hence, it is difficult to give guarantee of modeling stability of these methods.

It is well known that PIs based on ensemble neural networks and bootstrap method are good at solving the above-mentioned problems, and it is also worth studying these problems by using the ensemble neural networks and bootstrap. Furthermore, the bootstrap method can also reduce the influence of randomness caused by neural networks [14], [15].

Manuscript received January 15, 2019; revised October 18, 2019; accepted January 11, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61988101, Grant 61525302, Grant 61590922, Grant 61773068, and Grant 61733005, in part by the National Key Research and Development Program of China under Grant 2018YFB1701104, and in part by the Xingliao Plan of Liaoning Province under Grant XLYC1808001. *(Corresponding author: Jinliang Ding.)*

The authors are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: junluchn@hotmail.com; jlding@mail.neu.edu.cn; daixuewu@mail.neu.edu.cn; tychai@mail.neu.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2967816

The bootstrap method has the advantage of easy implementation and simple calculation; therefore, it has been successfully applied in the fields of wind-power-generation forecasts [16], [17], traffic noise estimation [18], electricity price prediction [19], and so on. However, the drawback of bootstrap method is also obvious, namely, the high computational cost when processing large data sets due to the complex ensemble and the slow training process of backpropagation neural networks (BPNNs) [20]. In addition, the effectiveness of the ensemble neural networks used in bootstrap is influenced by the base neural network in the ensemble, and it is critical for the training method to consider the performance of base neural networks [21]. There are three approaches to training ensemble neural networks: sequential training, independent training, and simultaneous training [22]. In the original bootstrap method of estimating PIs, all the base neural networks in the ensemble are trained independently because of the simplicity and ease of implementation of the independent training. However, this approach seriously limits the performance of the ensemble neural networks since the base neural networks are trained independently. As reported in the literature, simultaneous training is the most effective method of training the ensemble neural networks, owing to that the parameters of all the base neural networks are adjusted simultaneously and cooperate with each other [23]–[25]. Due to the low computational efficiency and the complex training process, there are few simultaneous training methods of the bootstrap-based ensemble neural networks. Latterly, some improved bootstrap methods have been studied. One example is the relevance vector machine that is introduced as the base components and trained by a parallel algorithm [26]. However, the essence of parallel algorithm is to use computer hardware to accelerate the training process, instead of training the base components in the ensemble simultaneously. Recently, the stochastic configuration networks (SCNs) have demonstrated excellent performance in the field of data modeling because of the universal approximation property [27], [28]. Compared with traditional neural networks, the simple structure and fast learning speed of SCNs can reduce the computational cost [27]. Therefore, SCN is a suitable candidate for estimating PIs based on the bootstrap and ensemble neural networks.

Moreover, in real-world applications, the data collected from the industrial processes are almost contaminated with outliers due to adverse interferences and noise [29], [30]. When the outliers presented in the training data, the aforementioned methods are not robust, which would lead to questionable PIs with low prediction accuracy. Therefore, it is extremely useful to develop robust methods which are insensitive to noise and outliers for estimating PIs.

In order to address the challenges of slow convergence of traditional BPNNs, the low accuracy of single-neural network-based PIs, and the high computation costs of original bootstrap methods, the PIs based on the ensemble SCNs and bootstrap method are proposed in this article, which aims at quantifying the reliability and potential uncertainty associated with the point prediction and ensuring the computational efficiency. To improve robustness of the proposed algorithm and encourage the cooperation among the base SCNs, the simultaneous robust training method based on the Bayesian ridge regression and M-estimate is developed. Meanwhile, the hyperparameters of the assumed distributions are estimated by the expectation-maximization (EM) algorithm. To examine the performance of the estimated PIs, experiments are carried out based on three benchmark data sets and a real-world data set collected from a refinery. The experimental results demonstrate the superiority of the proposed method of estimating PIs.

The rest of this article is organized as follows. Section II briefly introduces the bootstrap-method-based PIs and SCNs. Section III presents the proposed method of estimating PIs based on the bootstrap method and ensemble SCNs. The simultaneous robust training method based on the Bayesian ridge regression and M-estimate is also described in detail. EM-algorithm-based hyperparameters optimization is also given in Section III. Section IV presents the experimental results on three benchmark data sets and a real-world data set collected from a refinery. This article is concluded in Section V.

II. Preliminaries

This section briefly introduces the bootstrap-method-based PIs and the SCNs.

A. Estimation of PIs by Bootstrap Method and Ensemble Neural Networks

In regression tasks, the relation between the output variable y and the input variable $\boldsymbol{x}$ can be represented as follows:

$$y = t + \varepsilon = f(\boldsymbol{x}, \theta) + \varepsilon \tag{1}$$

where y is the observed value, t is the actual value, and $f(\boldsymbol{x}, \theta)$, which is parameterized by θ, denotes the true mapping relation between the input variable and the output variable. Generally, ε is assumed to be of the Gaussian random noise with a zero expectation and variance σ_ε^2 [24]. ε leads to the deviation of the observed value y from the actual value t.

If the neural network is used to build the prediction model, then the prediction output $\hat{y}$ of a trained neural network is an estimation of $f(\boldsymbol{x}, \theta)$

$$\hat{y} = \hat{f}(\boldsymbol{x}, \vartheta) \tag{2}$$

where ϑ denotes the parameters of neural network.

Then, the prediction error can be computed as follows:

$$y - \hat{y} = t - \hat{y} + \varepsilon = [f(\boldsymbol{x}, \theta) - \hat{f}(\boldsymbol{x}, \vartheta)] + \varepsilon. \tag{3}$$

Suppose that $(t - \hat{y})$ and ε, i.e., the two items on the right-hand side of (3) are independent, the variance σ_y^2 of the prediction value can be expressed as follows [15]:

$$\sigma_y^2 = \sigma_{\hat{y}}^2 + \sigma_\varepsilon^2 \tag{4}$$

where $\sigma_{\hat{y}}^2$ denotes the variance caused by model mismatch.

Bootstrap is a data resampling method based on statistical inference and has been widely used to estimate PIs [6]. In the bootstrap method, K subsample data sets $D_k = \{(\boldsymbol{x}_k^n, y_k^n), n = 1, \ldots, N\}$, $k = 1, \ldots, K$, are uniformly resampled with replacement from the original data set $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$. Bootstrap method calculates the prediction result and the

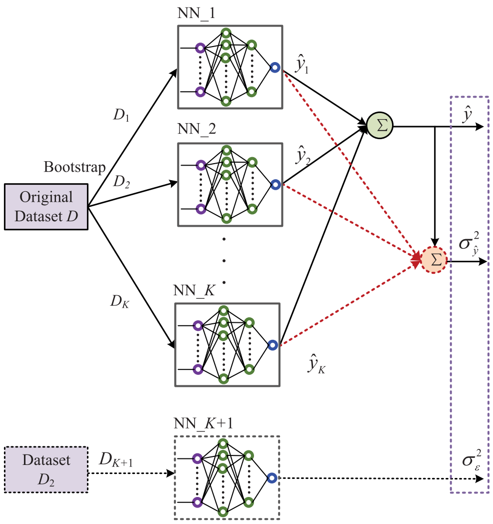


Fig. 1. Structure of PIs based on bootstrap and ensemble neural networks.

variance $\sigma^2_{\hat{y}}$ of model mismatch by building K neural networks using the K subsample data. The structure of PIs based on ensemble neural networks and bootstrap is shown in Fig. 1.

According to Fig. 1, it can be seen that for the input variable $\boldsymbol{x}$, the corresponding prediction $\hat{y}$ is the average of prediction outputs of the K neural networks as follows:

$$\hat{y} = \frac{1}{K}\sum_{k=1}^{K}\hat{y}_k. \tag{5}$$

If the prediction of a trained neural network is assumed to be unbiased, the variance $\sigma^2_{\hat{y}}$ caused by model mismatch can be computed by the variance of outputs of the K neural networks [15] as follows:

$$\sigma^2_{\hat{y}} = \frac{1}{K-1}\sum_{k=1}^{K}(\hat{y}_k - \hat{y})^2. \tag{6}$$

In general, an extra $(K+1)$th neural network is usually trained by another data set, as shown in Fig. 1, to model the variance σ^2_{ε} of the random noise, and the details refer to [15]. Once $\hat{y}$, $\sigma^2_{\hat{y}}$, and σ^2_{ε} are obtained, the PI with $(1-\alpha)\%$ CL, which is denoted by CL $= (1-\alpha)\%$, can be estimated [15]

$$\left[\hat{y} - t_{1-\alpha/2}(K)\sqrt{\sigma^2_{\hat{y}} + \sigma^2_{\varepsilon}},\ \hat{y} + t_{1-\alpha/2}(K)\sqrt{\sigma^2_{\hat{y}} + \sigma^2_{\varepsilon}}\right] \tag{7}$$

where $t_{1-\alpha/2}(K)$ is the t-distribution with $(1-\alpha/2)$ quantile and K degrees of freedom.

B. Stochastic Configuration Networks

SCN [27] is the most advanced constructive single-hidden-layer feedforward neural network. Its input weights and hidden layer biases are randomly generated in a varying uniform distribution symmetric scope $[-\mu, \mu]$, $\mu > 0$. Then, these randomly generated parameters are selected in light of a supervisory mechanism [27]. The structure of SCN used in this article is the same as the random vector functional link networks (RVFLs) [31], namely, the input layer is directly connected to the output layer. Given a data set $D = (\boldsymbol{x}, \boldsymbol{y}) = \{(\boldsymbol{x}_i, y_i) \in R^d \times R\}_{i=1}^{N}$, the output vector of the SCN with $L-1$ hidden nodes can be written as follows:

$$\boldsymbol{O}_{L-1}(\boldsymbol{x}; \boldsymbol{\beta}) = \boldsymbol{H}(\boldsymbol{x})\boldsymbol{\beta} \tag{8}$$

where $L = 1, 2, \ldots,$ and $\boldsymbol{H}(\boldsymbol{x})$ denotes a matrix combining the input data of d dimensionalities with the output matrix of hidden layer, which is abbreviated to $\boldsymbol{H}$ in the rest of this article. It can be written as follows:

$$\begin{cases}\boldsymbol{H} = [\boldsymbol{h}^T(\boldsymbol{x}_1), \ldots, \boldsymbol{h}^T(\boldsymbol{x}_i), \ldots, \boldsymbol{h}^T(\boldsymbol{x}_N)]^T \\ \boldsymbol{h}(\boldsymbol{x}_i) = \big[\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{id}, g_1\big(\boldsymbol{w}_1^T \cdot \boldsymbol{x}_i + b_1\big) \\ \qquad \ldots, g_{L-1}\big(\boldsymbol{w}_{L-1}^T \cdot \boldsymbol{x}_i + b_{L-1}\big)\big]\end{cases} \tag{9}$$

$\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{L-1+d}]^T$ denote the output weights, and $\boldsymbol{w}_l \in R^d$ and $b_l \in R$, $l = 1, \ldots, L-1$, denote the input weights and the hidden layer bias of the lth hidden node, respectively. $g(\cdot)$ is the activation function, and a commonly used activation function is the sigmoid function $g(x) = 1/(1+\exp(-x))$. The superscript T denotes the matrix transpose.

The output weights $\boldsymbol{\beta}$ can be computed by the least-squares algorithm [27], [28]

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta}\|_2^2 = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{y} \tag{10}$$

where$\|\cdot\|_2$ denotes the Euclidean norm and $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ is the vector of output variable.

When the SCN does not reach the predefined terminating conditions, a new hidden node is generated. The output matrix of the newly added Lth hidden node is written as follows:

$$\boldsymbol{G}_L = \big[g_L\big(\boldsymbol{w}_L^T \cdot \boldsymbol{x}_1 + b_L\big), \ldots, g_L\big(\boldsymbol{w}_L^T \cdot \boldsymbol{x}_N + b_L\big)\big]^T. \tag{11}$$

According to the theorem of SCNs proposed in [27], the randomly generated parameters ($\boldsymbol{w}_L$ and b_L) should be subject to the following supervisory mechanism to ensure the universal approximation property of SCNs:

$$\zeta = \frac{\langle \boldsymbol{e}_{L-1}^T, \boldsymbol{G}_L\rangle^2}{\langle \boldsymbol{G}_L^T, \boldsymbol{G}_L\rangle} - (1 - r - \gamma_L) \times \langle \boldsymbol{e}_{L-1}^T, \boldsymbol{e}_{L-1}\rangle > 0 \tag{12}$$

where $\boldsymbol{e}_{L-1} = \boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta}^*$ denotes the vector of training errors of SCN with $L-1$ hidden nodes, $0 < r < 1$, $0 < \gamma_L < 1 - r$, $\lim_{L\to\infty}\gamma_L = 0$, and $\langle\cdot,\cdot\rangle$ denotes the scalar product.

The output matrix of the hidden layer of the SCN is $[\boldsymbol{H}, \boldsymbol{G}_L]$, and the output weights $\boldsymbol{\beta}$ are evaluated by using (10). The generation of new hidden nodes continues until some predefined terminating conditions are reached, and the random parameters ($\boldsymbol{w}$ and b) of the new hidden node are determined based on the supervisory mechanism (12). More details about the SCNs can be found in [27].

Remark 1: According to the algorithms of SCNs described in [27] and [28], one should notice that $T_{\max}$ new hidden nodes $\{g_L^1(\boldsymbol{w}_L^1, b_L^1), \ldots, g_L^{T_{\max}}(\boldsymbol{w}_L^{T_{\max}}, b_L^{T_{\max}})\}$ are produced, and their random parameters ($\boldsymbol{w}_L$ and b_L) are generated in the varying range $[-\mu_j, \mu_j]$, $j = 1, \ldots, J$. The hidden nodes that satisfy the supervisory mechanism $\zeta > 0$ are selected as the candidates, and finally, the node with the largest ζ is chosen as the newly added one.

III. Simultaneous Robust Training of Bootstrap Ensemble SCNs for Estimation of PIs

This section begins with an introduction to the proposed approach of estimating PIs. Then, a simultaneous robust

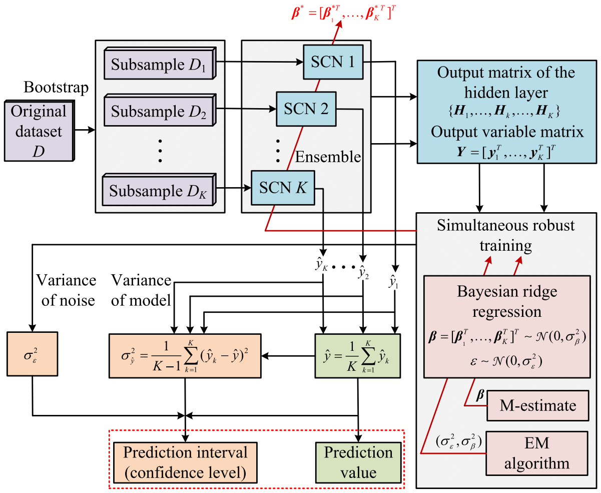

$\beta^* = [\beta_1^{*T}, \ldots, \beta_K^{*T}]^T$
Bootstrap
Original dataset D
Subsample D_1
Subsample D_2
Subsample D_K
SCN 1
SCN 2
SCN K
Ensemble
Output matrix of the hidden layer $\{H_1, \ldots, H_k, \ldots, H_K\}$
Output variable matrix $Y = [y_1^T, \ldots, y_K^T]^T$
Simultaneous robust training
Bayesian ridge regression
$\beta = [\beta_1^T, \ldots, \beta_K^T]^T \sim \mathcal{N}(0, \sigma_\beta^2)$
$\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$
β
M-estimate
$(\sigma_\varepsilon^2, \sigma_\beta^2)$
EM algorithm
$\hat{y}_K \cdots \hat{y}_2$
$\hat{y}_1$
Variance of noise
Variance of model
σ_ε^2
$\sigma_{\hat{y}}^2 = \frac{1}{K-1}\sum_{k=1}^{K}(\hat{y}_k - \hat{y})^2$
$\hat{y} = \frac{1}{K}\sum_{k=1}^{K}\hat{y}_k$
Prediction interval (confidence level)
Prediction value

Fig. 2. Structure of the proposed method of estimating PIs.

training algorithm of ensemble SCNs based on the Bayesian ridge regression and M-estimate is described in detail, and the EM-algorithm-based hyperparameter estimation is derived. Finally, the estimated PIs are given.

A. Proposed Approach of Estimating PIs

According to the previous analysis, it can be found that in the bootstrap and ensemble neural networks for PI estimation, the base neural networks are trained independently. This is because of the high computational cost of simultaneous training for the ensemble. While in the training stage of SCNs, once the configuration process completed, the input weights and hidden layer biases are fixed, and only the output weights need to be updated. This strategy can be used to mitigate the problem of high computational cost. However, in real-world applications, since the output weights of SCNs are computed by the least-squares method, such a method may suffer from overfitting as well as noise and outliers sensitive [30]. It is known that the Bayesian-framework-based training approach can effectively improve the generalization capability of neural networks and avoid overfitting [12]. Unfortunately, traditional Bayesian method is not robust to noise and outliers. In order to address this issue, a training method based on the M-estimate and Bayesian framework is proposed to improve the robustness.

In addition, under some assumptions, the Bayesian framework can also automatically infer the hyperparameters of the distributions of the random noise and output weights of the ensemble SCNs. Considering the aforementioned advantages, this article proposes a method of estimating PIs based on the bootstrap method and ensemble SCNs. Meanwhile, the simultaneous robust training approach of the ensemble SCNs is developed by using the Bayesian ridge regression and M-estimate. The structure of the proposed method of estimating PIs is shown in Fig. 2.

According to Fig. 2, it can be seen that K subdata sets are produced by using the bootstrap method, and K base SCNs are generated based on the K subdata sets. Then, the ensemble SCNs with size K is built based on the proposed simultaneous robust training method, and the hyperparameters are estimated by the EM algorithm. The prediction value and the variance caused by model mismatch are calculated by using (5) and (6). Finally, by combining the variance of model mismatch and the variance of noise, which is inferred by the Bayesian framework, the PIs can be estimated by (7).

B. Simultaneous Robust Training for Bootstrap Ensemble SCNs Based on the Bayesian Ridge Regression and M-Estimate

Given an original data set $D = \{(\boldsymbol{x}_i, y_i) \in R^d \times R\}_{i=1}^{N}$, K subtraining data sets $\{D_k\}_{k=1}^{K}$ can be obtained by resampling from D, where $D_k = (\boldsymbol{X}_k, \boldsymbol{y}_k) = \{(\boldsymbol{x}_k^n, y_k^n)\}_{n=1}^{N}$. The process of building bootstrap ensemble SCNs is divided into two steps.

1) Based on the K subtraining data sets, K base SCNs with the same structure are built independently by using the SC-III algorithm proposed in [27].
2) The input weights and biases of the base SCN built in the first step are kept fixed; only the output weights of the ensemble SCNs are retrained in light of the ensemble strategy.

According to the solution process of SCN described in Section II-B and the above-mentioned discussion on the ensemble SCNs, one can see that the training process of ensemble SCNs is equivalent to solving a large-scale linear regression problem. Therefore, the ensemble SCNs can be built based on the proposed simultaneous robust training method. For the kth SCN in the ensemble and the nth sample in the subtraining data set D_k, we can derive that

$$y_k^n = \hat{y}_k^n + \varepsilon_k^n = \boldsymbol{h}\left(\boldsymbol{x}_k^n\right)\boldsymbol{\beta}_k + \varepsilon_k^n \tag{13}$$

where $\varepsilon_k^n \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, $\mathcal{N}(\cdot)$ denotes the Gaussian distribution, and $\boldsymbol{\beta}_k = [\beta_k^1, \ldots, \beta_k^{L+d}]^T$ are the output weights of the kth SCN with L hidden nodes.

Then, the probability density function (PDF) of the observed output variable can be written as follows:

$$\begin{aligned} p\left(y_k^n|\boldsymbol{x}_k^n, \boldsymbol{\beta}_k, \sigma_\varepsilon^2\right) &= \mathcal{N}\left(y_k^n|\boldsymbol{h}(\boldsymbol{x}_k^n)\boldsymbol{\beta}_k, \sigma_\varepsilon^2\right) \\ &= \frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}}\exp\left\{-\frac{\left(y_k^n - \boldsymbol{h}(\boldsymbol{x}_k^n)\boldsymbol{\beta}_k\right)^2}{2\sigma_\varepsilon^2}\right\}. \end{aligned} \tag{14}$$

Generally speaking, all the observed samples are considered to be independent identically distributed (i.i.d.); thus, we can obtain the following likelihood function:

$$\begin{aligned} p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma_\varepsilon^2) &= p\left(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K|\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K, \boldsymbol{\beta}, \sigma_\varepsilon^2\right) \\ &= \prod_{k=1}^{K}\prod_{n=1}^{N} p\left(y_k^n|\boldsymbol{x}_k^n, \boldsymbol{\beta}_k, \sigma_\varepsilon^2\right) \\ &= \frac{1}{\left(\sqrt{2\pi\sigma_\varepsilon^2}\right)^{K\times N}} \\ &\quad \cdot \exp\left\{-\sum_{k=1}^{K}\sum_{n=1}^{N}\frac{\left(y_k^n - \boldsymbol{h}(\boldsymbol{x}_k^n)\boldsymbol{\beta}_k\right)^2}{2\sigma_\varepsilon^2}\right\} \end{aligned} \tag{15}$$

where $\boldsymbol{X} = [\boldsymbol{X}_1^T, \boldsymbol{X}_2^T, \ldots, \boldsymbol{X}_K^T]^T$, $\boldsymbol{Y} = [\boldsymbol{y}_1^T, \boldsymbol{y}_2^T, \ldots, \boldsymbol{y}_K^T]^T$, and $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, \ldots, \boldsymbol{\beta}_K^T]^T$.

In general, if we have little prior knowledge about the output weights of the ensemble neural networks, then the prior of the output weights is assumed to be of the Gaussian distribution with a large variance, and the output weights of each neural networks are assumed to be independent [24]. Accordingly, the output weights of the ensemble SCNs are also assumed to follow the Gaussian distribution, and then, the prior of $\boldsymbol{\beta}$ of the ensemble SCNs can be formulated as follows:

$$\begin{aligned} p\left(\boldsymbol{\beta}|\sigma_\beta^2\right) &= p\left(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K|\sigma_\beta^2\right) \\ &= \prod_{k=1}^{K} p\left(\boldsymbol{\beta}_k|\sigma_\beta^2\right) \\ &= \prod_{k=1}^{K} \frac{1}{\left(\sqrt{2\pi\sigma_\beta^2}\right)^{L+d}} \exp\left\{-\frac{1}{2\sigma_\beta^2}\|\boldsymbol{\beta}_k\|_2^2\right\} \\ &= \frac{1}{\left(\sqrt{2\pi\sigma_\beta^2}\right)^{K\times(L+d)}} \exp\left\{-\frac{1}{2\sigma_\beta^2}\sum_{k=1}^{K}\|\boldsymbol{\beta}_k\|_2^2\right\}. \end{aligned} \quad (16)$$

Remark 2: In the process of building bootstrap ensemble SCNs, the input weights and biases of all the base SCNs are randomly generated based on the supervisory mechanism proposed in [27] and kept fixed. Therefore, the training process of bootstrap ensemble SCNs is equivalent to solving a large-scale linear system that can be mathematically described as $\mathbb{H}\boldsymbol{\beta} = \boldsymbol{Y}$, where $\mathbb{H} = blkdiag[\boldsymbol{H}_1, \ldots, \boldsymbol{H}_K]$ and $blkdiag[\cdot]$ represents the operator of constructing block diagonal matrix. Since the input weights and biases are kept fixed, accordingly, $\mathbb{H}$ is a constant matrix. Therefore, there is a linear mapping relation between $\boldsymbol{\beta}$ and $\boldsymbol{Y}$. Consequently, according to the rule of transformations of continuous probability densities described in [32], the PDFs of $\boldsymbol{\beta}$ and $\boldsymbol{Y}$ are related by the following expression:

$$f_{\boldsymbol{\beta}}(\beta) = |\det(\mathbb{H})| \cdot f_{\boldsymbol{Y}}(\mathbb{H}\beta) \quad (17)$$

where $f_{\boldsymbol{\beta}}(\cdot)$ and $f_{\boldsymbol{Y}}(\cdot)$ are the PDFs of $\boldsymbol{\beta}$ and $\boldsymbol{Y}$, respectively. $\det(\cdot)$ denotes the determinant. Since the observed variable $\boldsymbol{Y}$ is assumed to be i.i.d. and follows the Gaussian distribution, according to (17) and linear algebra calculations, it can be concluded that $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$ are independent of each other.

Based on Bayes' theorem, if the likelihood function and the prior are known, for the kth SCN and D_k, the posterior of $\boldsymbol{\beta}_k$ can be written as follows:

$$p\left(\boldsymbol{\beta}_k|\boldsymbol{y}_k, \boldsymbol{X}_k, \sigma_\varepsilon^2, \sigma_\beta^2\right) = \frac{p\left(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\beta}_k, \sigma_\varepsilon^2\right)p\left(\boldsymbol{\beta}_k|\sigma_\beta^2\right)}{p\left(\boldsymbol{y}_k|\boldsymbol{X}_k, \sigma_\varepsilon^2, \sigma_\beta^2\right)} \quad (18)$$

where $p(\boldsymbol{y}_k|\boldsymbol{X}_k, \sigma_\varepsilon^2, \sigma_\beta^2)$ is the marginal likelihood also viewed as the normalization constant

$$p\left(\boldsymbol{y}_k|\boldsymbol{X}_k, \sigma_\varepsilon^2, \sigma_\beta^2\right) = \int p\left(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\beta}_k, \sigma_\varepsilon^2\right)p\left(\boldsymbol{\beta}_k|\sigma_\beta^2\right)d\boldsymbol{\beta}_k. \quad (19)$$

According to the conclusion described in [33], the posterior of $\boldsymbol{\beta}_k$ is still Gaussian

$$p\left(\boldsymbol{\beta}_k|\boldsymbol{y}_k, \boldsymbol{X}_k, \sigma_\varepsilon^2, \sigma_\beta^2\right) = \mathcal{N}\left(\frac{1}{\sigma_\varepsilon^2}\boldsymbol{\Lambda}_k\boldsymbol{H}_k^T\boldsymbol{y}_k, \boldsymbol{\Lambda}_k\right) \quad (20)$$

where $\boldsymbol{y}_k = [y_k^1, \ldots, y_k^N]^T$, $\boldsymbol{H}_k = [\boldsymbol{h}^T(\boldsymbol{x}_k^1), \ldots, \boldsymbol{h}^T(\boldsymbol{x}_k^N)]^T$, and

$$\boldsymbol{\Lambda}_k^{-1} = \frac{1}{\sigma_\varepsilon^2}\boldsymbol{H}_k^T\boldsymbol{H}_k + \frac{1}{\sigma_\beta^2}\boldsymbol{I}. \quad (21)$$

For the ensemble SCNs and the total training data sets $\{D_1, D_2, \ldots, D_K\}$, according to Bayes' theorem, we can obtain the posterior of $\boldsymbol{\beta}$ of the ensemble SCNs

$$p\left(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\right) = \frac{p\left(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma_\varepsilon^2\right)p\left(\boldsymbol{\beta}|\sigma_\beta^2\right)}{p\left(\boldsymbol{Y}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\right)} \quad (22)$$

where $p(\boldsymbol{Y}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2)$ is the marginal likelihood, which is defined as the same as $p(\boldsymbol{y}_k|\boldsymbol{X}_k, \sigma_\varepsilon^2, \sigma_\beta^2)$. Then, we can rewrite the posterior of $\boldsymbol{\beta}$ of the ensemble SCNs as follows:

$$\begin{aligned} &p\left(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\right) \\ &\propto p\left(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma_\varepsilon^2\right)p\left(\boldsymbol{\beta}|\sigma_\beta^2\right) \\ &\propto \frac{1}{\left(\sqrt{2\pi\sigma_\varepsilon^2}\right)^{K\times N}\left(\sqrt{2\pi\sigma_\beta^2}\right)^{K\times(L+d)}} \\ &\quad\cdot \exp\left\{-\frac{1}{2\sigma_\varepsilon^2}\sum_{k=1}^{K}\sum_{n=1}^{N}\left(y_k^n - \boldsymbol{h}\left(\boldsymbol{x}_k^n\right)\boldsymbol{\beta}_k\right)^2 \right. \\ &\qquad\left. - \frac{1}{2\sigma_\beta^2}\sum_{k=1}^{K}\|\boldsymbol{\beta}_k\|_2^2\right\} \end{aligned} \quad (23)$$

where $a \propto b$ denotes that a is proportionate to b. For the convenience of calculation, we generally take the logarithm of the posterior of output weights of the ensemble SCNs, and it can be written as follows:

$$\begin{aligned} &\ln p\left(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\right) \\ &= -\frac{1}{2\sigma_\varepsilon^2}\sum_{k=1}^{K}\sum_{n=1}^{N}\left(y_k^n - \boldsymbol{h}\left(\boldsymbol{x}_k^n\right)\boldsymbol{\beta}_k\right)^2 \\ &\quad - \frac{1}{2\sigma_\beta^2}\sum_{k=1}^{K}\|\boldsymbol{\beta}_k\|_2^2 + c \\ &= -\frac{1}{2\sigma_\varepsilon^2}\sum_{k=1}^{K}\|\boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{\beta}_k\|_2^2 - \frac{1}{2\sigma_\beta^2}\sum_{k=1}^{K}\|\boldsymbol{\beta}_k\|_2^2 + c \\ &= -\frac{1}{2\sigma_\varepsilon^2}\|\boldsymbol{Y} - \mathbb{H}\boldsymbol{\beta}\|_2^2 - \frac{1}{2\sigma_\beta^2}\|\boldsymbol{\beta}\|_2^2 + c \end{aligned} \quad (24)$$

where c denotes a constant which is independent of $\boldsymbol{\beta}$.

Subsequently, the output weights $\boldsymbol{\beta}^*$ of the ensemble SCNs can be obtained by maximizing the logarithm of the posterior distribution (24), which is also called the maximum *a posteriori* (MAP) estimation

$$\begin{aligned} \boldsymbol{\beta}^* &= \underset{\boldsymbol{\beta}\in R^{K\times(L+d)}}{\arg\max}\left\{\ln p\left(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\right)\right\} \\ &= \underset{\boldsymbol{\beta}\in R^{K\times(L+d)}}{\arg\min}\left\{\frac{1}{2\sigma_\varepsilon^2}\|\boldsymbol{Y} - \mathbb{H}\boldsymbol{\beta}\|_2^2 + \frac{1}{2\sigma_\beta^2}\|\boldsymbol{\beta}\|_2^2\right\}. \end{aligned} \quad (25)$$

Since the noise in (13) is assumed to be of the Gaussian distribution, the above-mentioned MAP estimation is equivalent to the regularized least-squares method with the regularization coefficient $\lambda = \sigma_\varepsilon^2/\sigma_\beta^2$ [13]

$$\boldsymbol{\beta}^* = \left[\mathbb{H}^T\mathbb{H} + \left(\sigma_\varepsilon^2/\sigma_\beta^2\right)\boldsymbol{I}\right]^{-1}\mathbb{H}^T\boldsymbol{Y}. \tag{26}$$

It is well known that the neural networks obtained by the regularized least-squares algorithm are not robust to noise and outliers [29]. However, there are always noise and outliers in the real-world industrial data. The M-estimate is a most popular robust estimation method [34]. Therefore, by combining the Bayesian ridge regression and M-estimate, the robust ensemble SCNs is proposed in this article, and the corresponding cost function is given by

$$T(\boldsymbol{\beta}) = \sum_{k=1}^{K}\sum_{n=1}^{N}\rho\left(\frac{y_k^n - \boldsymbol{h}\left(\boldsymbol{x}_k^n\right)\boldsymbol{\beta}_k}{s}\right) + \frac{\sigma_\varepsilon^2}{2\sigma_\beta^2}\sum_{k=1}^{K}\|\boldsymbol{\beta}_k\|_2^2 \tag{27}$$

where $\rho(\cdot)$ denotes the function of robust criterion and $s = 1.4826 \times med_{k,n}(|r_k^n - med_{k,n}(r_k^n)|)$ denotes the robust scale estimator [29], [34]. Here, $med(\cdot)$ denotes the median operator, and $r_k^n = y_k^n - \boldsymbol{h}(\boldsymbol{x}_k^n)\boldsymbol{\beta}_k$ is the residual error of the nth sample in the kth subsample data set.

Then, we can calculate the output weights of the ensemble SCNs by solving the following optimization problem:

$$\begin{aligned}\boldsymbol{\beta}^* &= \underset{\boldsymbol{\beta}\in R^{K\times(L+d)}}{\arg\min}\; T(\boldsymbol{\beta})\\ &= \underset{\boldsymbol{\beta}\in R^{K\times(L+d)}}{\arg\min}\sum_{k=1}^{K}\sum_{n=1}^{N}\rho\left(\frac{r_k^n}{s}\right) + \frac{\sigma_\varepsilon^2}{2\sigma_\beta^2}\sum_{k=1}^{K}\|\boldsymbol{\beta}_k\|_2^2. \end{aligned}\tag{28}$$

Let $\partial T(\boldsymbol{\beta})/\partial\boldsymbol{\beta}_k = 0$, we can obtain

$$\begin{aligned}-\frac{1}{s}&\sum_{k=1}^{K}\sum_{n=1}^{N}\boldsymbol{h}^T\left(\boldsymbol{x}_k^n\right)\psi\left(r_k^n/s\right) + \frac{\sigma_\varepsilon^2}{\sigma_\beta^2}\sum_{k=1}^{K}\boldsymbol{\beta}_k\\ &= -\frac{1}{s^2}\sum_{k=1}^{K}\sum_{n=1}^{N}\boldsymbol{h}^T\left(\boldsymbol{x}_k^n\right)w\left(r_k^n/s\right)\\ &\quad\times r_k^n + \frac{\sigma_\varepsilon^2}{\sigma_\beta^2}\sum_{k=1}^{K}\boldsymbol{\beta}_k = 0\end{aligned}\tag{29}$$

where

$$\begin{cases}\psi\left(r_k^n/s\right) \overset{\Delta}{=} \rho'\left(r_k^n/s\right)\\ w\left(r_k^n/s\right) \overset{\Delta}{=} \psi\left(r_k^n/s\right)/\left(r_k^n/s\right).\end{cases}\tag{30}$$

The selection of $\rho(\cdot)$ should guarantee that the $\psi(\cdot)$ is an odd function and $\psi(x) \geq 0$ for $x \geq 0$ [34]. The Cauchy function $\rho(x) = (\sigma^2/2)\ln(1 + (x/\sigma)^2)$ is selected in this article, where $\sigma = 2.3849$ is suggested in [35].

For the ensemble SCNs, (29) can be written in matrix form

$$\left[\mathbb{H}^T\boldsymbol{W}\mathbb{H} + s^2\left(\sigma_\varepsilon^2/\sigma_\beta^2\right)\boldsymbol{I}\right]\boldsymbol{\beta} = \mathbb{H}^T\boldsymbol{W}\boldsymbol{Y} \tag{31}$$

where $\boldsymbol{W} = diag\{w(r_k^n/s)\}_{k=1:K}^{n=1:N}$ is the weight matrix and $diag\{\cdot\}$ is the operator of constructing diagonal matrix.

Then, the iterative formula of output weights $\boldsymbol{\beta}^{(m+1)}$ of the ensemble SCNs can be calculated by the iteratively reweighted regularized least-squares

$$\boldsymbol{\beta}^{(m+1)} = \left[\mathbb{H}^T\boldsymbol{W}^{(m)}\mathbb{H} + s^2\left(\sigma_\varepsilon^2/\sigma_\beta^2\right)\boldsymbol{I}\right]^{-1}\mathbb{H}^T\boldsymbol{W}^{(m)}\boldsymbol{Y} \tag{32}$$

where m denotes the number of iterations.

The terminating condition of the iterative process is selected as $||\boldsymbol{\beta}^{(m+1)} - \boldsymbol{\beta}^{(m)}|| < \kappa$, where κ is a small positive number.

In accordance with the above-mentioned analysis and discussion, the implementation step of the bootstrap ensemble SCNs based on the proposed simultaneous robust training method is summarized in Algorithm 1.

Algorithm 1 Bootstrap Ensemble SCNs Using the Proposed Simultaneous Robust Training Method

```
Input: Data set $D = \{(\boldsymbol{x}_i, y_i) \in R^d \times R\}_{i=1}^N$.
Output: $\boldsymbol{\beta}$.
Initialization: Set the initial hyperparameters in (32);
Generate $K$ subsample data sets from data set $D$;
Generate $K$ base SCNs and build the ensemble SCNs;
Calculate the initial output weights $\boldsymbol{\beta}$ using (26);
while $||\boldsymbol{\beta}^{(m+1)} - \boldsymbol{\beta}^{(m)}|| \geq \kappa$ do
  Calculate the residual error vector;
  Obtain the robust estimator $s$ and weight matrix $W^{(m)}$;
  Compute output weights $\boldsymbol{\beta}^{(m+1)}$ using (32);
  Update the terminating condition;
end while
Return $\boldsymbol{\beta}$ of the bootstrap ensemble SCNs.
```

C. Estimation of Hyperparameters Using EM Algorithm

In order to obtain optimal output weights of the ensemble SCNs, hyperparameters σ_ε^2 and σ_β^2 should be chosen sensibly. Accordingly, hyperparameters σ_ε^2 and σ_β^2 are optimized by the EM algorithm [36].

From the marginal likelihood $p(\boldsymbol{Y}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2)$, we can see that the output weights $\boldsymbol{\beta}$ are marginalized out. Thus, the output weights $\boldsymbol{\beta}$ can be treated as the latent variables, and the EM algorithm can be implemented to optimize the marginal likelihood function. Since the output weights $\boldsymbol{\beta}$ are treated as latent variables, the data set $\{\boldsymbol{Y}, \boldsymbol{\beta}\}$ is referred to the complete data. Given the initial hyperparameters σ_ε^2 and σ_β^2, one can obtain the output weights $\boldsymbol{\beta}$ by using Algorithm 1. Then, in the expectation step (E step) of the EM algorithm, we compute the expectation of the logarithm of the complete-data marginal likelihood function. In the maximization step (M step) of the EM algorithm, we reestimate σ_ε^2 and σ_β^2 by maximizing the expectation obtained in the E step.

By computing the marginal likelihood function and taking its logarithm, the following expression can be obtained:

$$\ln p\left(\boldsymbol{Y}, \boldsymbol{\beta}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\right) = \ln p\left(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma_\varepsilon^2\right) + \ln p\left(\boldsymbol{\beta}|\sigma_\beta^2\right). \tag{33}$$

The σ_ε^2 and σ_β^2 can be obtained by maximizing the logarithm of the complete-data marginal likelihood function

$$\sigma_{\varepsilon*}^2, \sigma_{\beta*}^2 = \underset{\sigma_\varepsilon^2, \sigma_\beta^2}{\arg\max}\left\{\ln p\left(\boldsymbol{Y}, \boldsymbol{\beta}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\right)\right\}. \tag{34}$$

Then, the EM algorithm is used to solve the above-mentioned optimization problem to obtain the optimal hyperparameters. In the E step, by taking expectation of

$\ln p(\boldsymbol{Y}, \boldsymbol{\beta}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2)$ with respect to $\boldsymbol{\beta}$, we can derive

$$\begin{aligned}\mathbb{E}\big[\ln p\big(\boldsymbol{Y}, \boldsymbol{\beta}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\big)\upsilon\big]
&= -\frac{1}{2\sigma_\varepsilon^2}\sum_{k=1}^{K}\sum_{n=1}^{N}\mathbb{E}\big[\big(y_k^n - \boldsymbol{h}(\boldsymbol{x}_k^n)\boldsymbol{\beta}_k\big)^2\big]\\
&\quad + \frac{K \times N}{2}\ln\frac{1}{2\pi\sigma_\varepsilon^2}\\
&\quad + \frac{K \times (L+d)}{2}\ln\frac{1}{2\pi\sigma_\beta^2}\\
&\quad - \frac{1}{2\sigma_\beta^2}\sum_{k=1}^{K}\mathbb{E}\big[\|\boldsymbol{\beta}_k\|_2^2\big] \qquad (35)\end{aligned}$$

where $\mathbb{E}[\cdot]$ denotes the statistical expectation operator.

In the M step, the derivative of $\mathbb{E}[\ln p(\boldsymbol{Y}, \boldsymbol{\beta}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2)]$ with respect to σ_β^2 is set to zero

$$\begin{aligned}&\frac{\partial}{\partial(\sigma_\beta^2)}\mathbb{E}\big[\ln p\big(\boldsymbol{Y}, \boldsymbol{\beta}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\big)\big]\\
&\quad = -\frac{K \times (L+d)}{2}\sigma_\beta^2 + \frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\big[\|\boldsymbol{\beta}_k\|_2^2\big] = 0. \qquad (36)\end{aligned}$$

Then, the reestimation of σ_β^2 is

$$\begin{aligned}\sigma_\beta^2 &= \frac{\sum_{k=1}^{K}\mathbb{E}\big[\|\boldsymbol{\beta}_k\|_2^2\big]}{K \times (L+d)} = \frac{\sum_{k=1}^{K}\big[\|\boldsymbol{\beta}_k^*\|_2^2 + \mathrm{Tr}(\boldsymbol{\Lambda}_k)\big]}{K \times (L+d)}\\
&= \frac{\|\boldsymbol{\beta}^*\|_2^2 + \sum_{k=1}^{K}\mathrm{Tr}(\boldsymbol{\Lambda}_k)}{K \times (L+d)} \qquad (37)\end{aligned}$$

where $\boldsymbol{\beta}_k^*$ represents the current output weights of the kth base SCN. $\boldsymbol{\beta}^*$ represents the current output weights of the ensemble SCNs, and $\mathrm{Tr}(\cdot)$ denotes the trace operator for matrix.

The derivative of $\mathbb{E}[\ln p(\boldsymbol{Y}, \boldsymbol{\beta}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2)]$ with respect to σ_ε^2 is set to zero, and we can obtain

$$\begin{aligned}&\frac{\partial}{\partial(\sigma_\varepsilon^2)}\mathbb{E}\big[\ln p\big(\boldsymbol{Y}, \boldsymbol{\beta}|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\big)\big]\\
&= -\frac{K \times N}{2}\sigma_\varepsilon^2 + \frac{1}{2}\sum_{k=1}^{K}\sum_{n=1}^{N}\mathbb{E}\big[\big(y_k^n - \boldsymbol{h}(\boldsymbol{x}_k^n)\boldsymbol{\beta}_k\big)^2\big]\\
&= -\frac{K \times N}{2}\sigma_\varepsilon^2 + \frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\big[\|\boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{\beta}_k\|_2^2\big]\\
&= -\frac{K \times N}{2}\sigma_\varepsilon^2\\
&\quad + \frac{1}{2}\sum_{k=1}^{K}\big[\big\|\boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{\beta}_k^*\big\|_2^2 + \mathrm{Tr}\big(\boldsymbol{H}_k^T\boldsymbol{H}_k\boldsymbol{\Lambda}_k\big)\big] = 0. \qquad (38)\end{aligned}$$

Then, the reestimation of σ_ε^2 is

$$\begin{aligned}\sigma_\varepsilon^2 &= \frac{\sum_{k=1}^{K}\big[\|\boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{\beta}_k^*\|_2^2 + \mathrm{Tr}(\boldsymbol{H}_k^T\boldsymbol{H}_k\boldsymbol{\Lambda}_k)\big]}{K \times N}\\
&= \frac{\|\boldsymbol{Y} - \mathbb{H}\boldsymbol{\beta}^*\|_2^2 + \sum_{k=1}^{K}\mathrm{Tr}\big(\boldsymbol{H}_k^T\boldsymbol{H}_k\boldsymbol{\Lambda}_k\big)}{K \times N}. \qquad (39)\end{aligned}$$

```
Algorithm 2 Estimation of PIs Using the Proposed Method
Input: The training data and testing input data $\boldsymbol{x}^*$.
Output: $\hat{y}^*$, $L(\boldsymbol{x}^*)$ and $U(\boldsymbol{x}^*)$.
 1: Initialization: Set the ensemble size $K$, the predefined CL,
    the number of hidden nodes $L$ of base SCN, the varying
    scope of random parameters $\Upsilon = [-\mu_j, \mu_j]_{j=1}^{J}$, the maximum
    random configuration times $T_{max}$ and the parameters
    involved in Algorithm 1;
 2: Generate $K$ subsample data sets from the training data set
    using bootstrap method;
 3: Build $K$ base SCNs by using SC-III algorithm proposed
    in [27] based on the $K$ data sets generated in step 2;
 4: Establish ensemble SCNs;
 5: while terminating condition (40) is not reached do
 6:    Build the robust ensemble SCNs based on Algorithm 1
       (step 5−step 11);
 7:    Estimate $\sigma_\beta^2$ and $\sigma_\varepsilon^2$ using (37) and (39);
 8:    Update the hyperparameters in (32);
 9:    Renew the terminating condition (40);
10: end while
11: Output $\boldsymbol{\beta}^* = [\boldsymbol{\beta}_1^{*T}, \ldots, \boldsymbol{\beta}_K^{*T}]^T$ of the ensemble SCNs and
    the hyperparameters $\sigma_{\varepsilon^*}^2$ and $\sigma_{\beta^*}^2$;
12: Input the testing data $\boldsymbol{x}^*$;
13: Calculate $\hat{y}^*$ using (41);
14: Compute $L(\boldsymbol{x}^*)$ and $U(\boldsymbol{x}^*)$ using (42).
```

The proposed simultaneous robust training process of the ensemble SCNs based on the Bayesian ridge regression and M-estimate is summarized as follows. First, the initial hyperparameters are assigned. Second, we use Algorithm 1 to compute output weights of the robust ensemble SCNs. Third, the hyperparameters are reestimated based on the EM algorithm by using the output weights obtained in the second step. The second and third steps are repeated until some terminating conditions are reached. In this article, the terminating condition is selected as follows:

$$\left|\frac{\mathbb{E}\big[\ln p\big(\boldsymbol{Y}, \boldsymbol{\beta}_{\text{new}}^*|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\big)\big]}{\mathbb{E}\big[\ln p\big(\boldsymbol{Y}, \boldsymbol{\beta}_{\text{old}}^*|\boldsymbol{X}, \sigma_\varepsilon^2, \sigma_\beta^2\big)\big]} - 1\right| < \tau \qquad (40)$$

where $\boldsymbol{\beta}_{\text{new}}^*$ and $\boldsymbol{\beta}_{\text{old}}^*$ denote the current and previous output weights of the ensemble SCNs, respectively. $|\cdot|$ is the operator of computing absolute value, and τ is a small positive number, which is set to $(1e-6)$ in this article.

D. Estimation of PIs

Once the output weights $\boldsymbol{\beta}^* = [\boldsymbol{\beta}_1^{*T}, \boldsymbol{\beta}_2^{*T}, \ldots, \boldsymbol{\beta}_K^{*T}]^T$ of the ensemble SCNs are obtained, for the new input variable $\boldsymbol{x}^*$, we can obtain the point prediction value of the corresponding output as follows:

$$\hat{y}^* = \frac{1}{K}\sum_{k=1}^{K}\hat{y}_k^* = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{H}_k(\boldsymbol{x}^*)\boldsymbol{\beta}_k^*. \qquad (41)$$

As mentioned in Section II-A, another neural network is needed to estimate the variance of the noise. This process is time-consuming due to the indirectly training of an extra neural network based on a constructive data set [3], [24]. However, the hyperparameter $\sigma_{\varepsilon^*}^2$ can be used to estimate the variance of uncertainty, which is caused by the intrinsic noise [24]. The variance of uncertainty caused by model mismatch is calculated by using (6). Then, the lower bound $L(\boldsymbol{x}^*)$ and the upper bound $U(\boldsymbol{x}^*)$ of the PI with CL = $(1-\alpha)\%$ are given as follows:

$$\begin{cases} L(\boldsymbol{x}^*) = \hat{y}^* - t_{1-\alpha/2}(K)\sqrt{\sigma_{\hat{y}^*}^2 + \sigma_{\varepsilon^*}^2} \\ U(\boldsymbol{x}^*) = \hat{y}^* + t_{1-\alpha/2}(K)\sqrt{\sigma_{\hat{y}^*}^2 + \sigma_{\varepsilon^*}^2}. \end{cases} \tag{42}$$

According to the above-mentioned description and analysis, the implementation procedure of the proposed method of estimating PIs is summarized in Algorithm 2.

IV. Experimental Results and Analysis

In this section, the performance of the proposed algorithm is evaluated on three benchmark data sets and a real-world data set collected from a refinery in Southern China. The proposed algorithm is compared with several representative algorithms of estimating PIs including: 1) the single-neural network-based algorithms: Bayesian [5], delta [7], and MVE [8] and 2) the ensemble neural networks-based algorithms: the negative-correlation-learning-based ensemble RVFL (NCL-E-RVFL) [2] and the optimized bootstrap method (OPT-bootstrap) [15]. The experiment is repeated 50 times, and the average value of the 50 experimental results is reported. All the algorithms in the experiment are programed in MATLAB and run on a computer with a 3.4-GHz CPU.

The root-mean-squared error (RMSE), the mean absolute percentage error (MAPE), and the Nash–Sutcliffe coefficient (NSC) are adopted to evaluate the prediction accuracy of each method, and a large NSC indicates high prediction accuracy. The PI coverage probability (PICP), the normalized mean PI width (NMPIW), and the Winkler score (W-score) are introduced to evaluate the performance of the PIs. The PIs with high quality should have small NMPIW while the PICP is larger than the predefined CL, and the narrower PIs have the smaller absolute value of W-score [10]. These indices are defined as follows:

$$\begin{cases} \text{RMSE} = \sqrt{\dfrac{1}{N_t}\sum\limits_{i=1}^{N_t}(y_i - \hat{y}_i)^2} \\ \text{MAPE} = 100\% \times \dfrac{1}{N_t}\sum\limits_{i=1}^{N_t}\left|\dfrac{(y_i - \hat{y}_i)}{y_i}\right| \\ \text{NSC} = 1 - \dfrac{\sum\limits_{i=1}^{N_t}(\hat{y}_i - y_i)^2}{\sum\limits_{i=1}^{N_t}(\bar{y} - y_i)^2} \end{cases} \tag{43}$$

where $\hat{y}_i$ is the prediction value, y_i is the observed value, and $\bar{y} = \sum_{i=1}^{N_t} y_i/N_t$

$$\begin{cases} \text{PICP} = \dfrac{1}{N_t}\sum\limits_{i=1}^{N_t} B_i \\ \text{NMPIW} = \dfrac{1}{RN_t}\sum\limits_{i=1}^{N_t}[U(\boldsymbol{x}_i) - L(\boldsymbol{x}_i)] \end{cases} \tag{44}$$

where N_t is the number of testing data, B_i denotes a Boolean variable, if $y_i \in [L(\boldsymbol{x}_i), U(\boldsymbol{x}_i)]$, $B_i = 1$, otherwise $B_i = 0$, and R denotes the range of observed output values

$$\text{W-score} = \frac{1}{N_t}\sum_{i=1}^{N_t} S_i \tag{45}$$

where

$$S_i = \begin{cases} -2\alpha \times \Delta_i - 4[L(\boldsymbol{x}_i) - y_i], & \text{if } \ y_i < L(\boldsymbol{x}_i) \\ -2\alpha \times \Delta_i, & \text{if } \ y_i \in [L(\boldsymbol{x}_i), U(\boldsymbol{x}_i)] \\ -2\alpha \times \Delta_i - 4[y_i - U(\boldsymbol{x}_i)], & \text{if } \ y_i > U(\boldsymbol{x}_i) \end{cases} \tag{46}$$

where $\Delta_i = U(\boldsymbol{x}_i) - L(\boldsymbol{x}_i)$ and $\alpha = 1 - \text{CL}$.

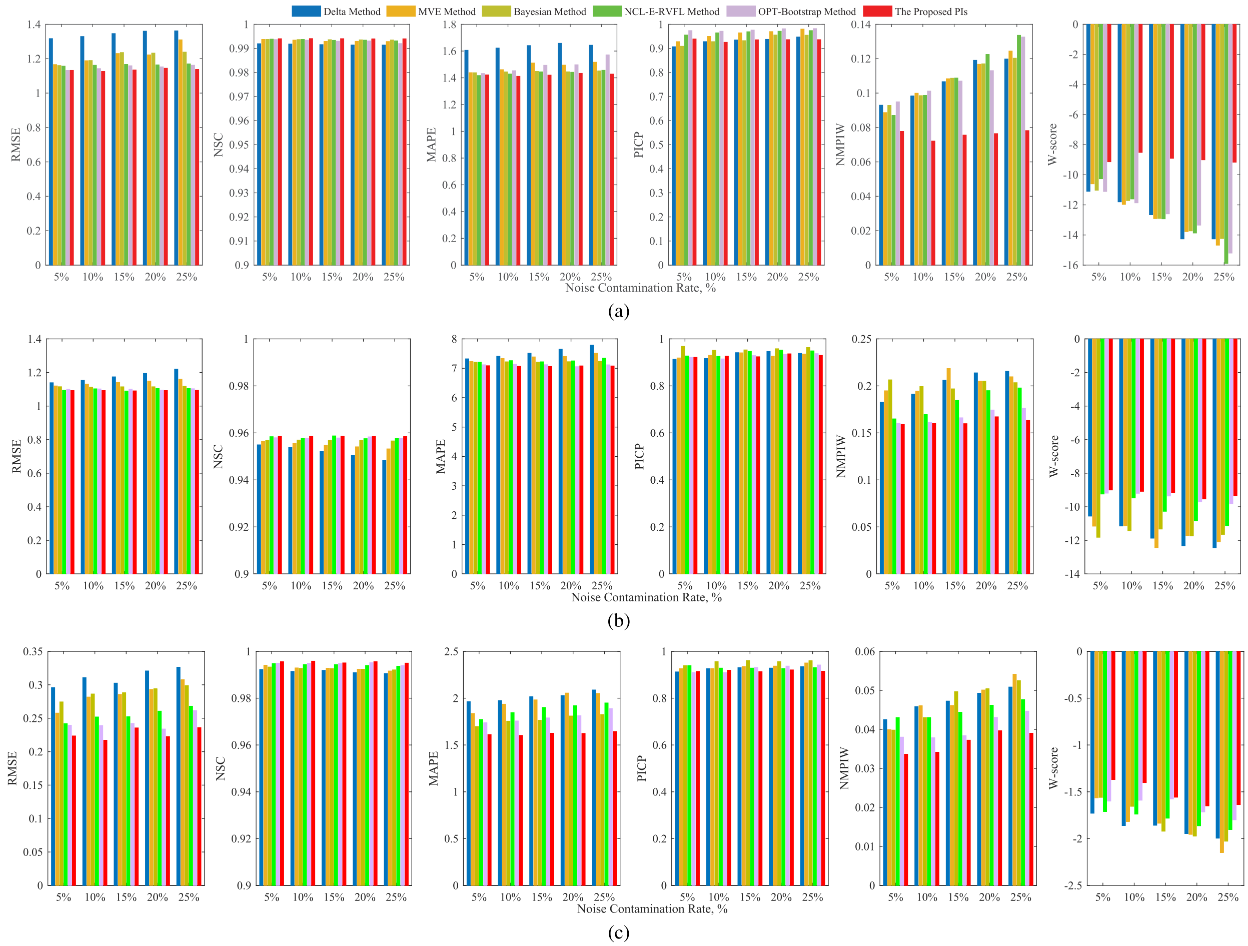
A. Case Studies on Benchmark Data Sets

Three benchmark data sets are collected from KEEL[1]: Wizmir (DB1), Friedman (DB2), and Treasury (DB3). Each data set is divided into three parts: 60% of the total samples are used as the training data, 20% of the total samples are used as the validation data, and the remaining 20% of the total samples are used as the testing data.

1) Parameter Setting: In this article, the predefined CL of the estimated PIs is set to 90%, i.e., CL $= (1-\alpha)\% = 90\%$. In the proposed method of estimating PIs, the input weights and hidden biases of base SCN in the ensemble are automatically selected in the varying set $[-\mu, \mu]$, $\mu = 1, 2, 4, 8$, based on the supervisory mechanism (12), the maximum random configuration time is $T_{\max} = 50$, and all base SCNs have the same number of hidden nodes. Therefore, the parameters of the ensemble SCNs to be optimized include the number of hidden nodes L and the size of ensemble K. The exhaustive linear search is an effective method to optimize the parameters in the ensemble [25]; therefore, it is also adopted to select the parameters L and K. The parameter L is searched in [10, 80] with ten steps. According to the empirical rule stated in [15], it is known that in real-world applications, if one wants to estimate the statistical bias and variance based on bootstrap, there should be about 100 groups of bootstrap subsample data sets. If the number of bootstrap data sets exceeds 100, there is little improvement in the estimated performance. In order to estimate the PIs with high quality, in this study, the size K of ensemble SCNs, namely, the number of subsample data sets, is searched in [75, 110] with five steps. The initial values of the hyperparameters are set to $\sigma_\varepsilon^2 = 1$ and $\sigma_\beta^2 = 0.5$. All the parameters are determined according to the results on the validation data, the parameter K is set to 80, and the parameter L is set to 50, 40, and 60 for DB1, DB2, and DB3, respectively.

2) Comparative Experiments: The evaluation indices of the proposed PIs and other representative methods on the normal benchmark data sets are listed in Tables I and II. As shown in Table I, the RMSE and MAPE of the proposed method are smaller than that of other methods, and the NSC is the largest among all the methods. Therefore, the proposed method

[1] KEEL: http://www.keel.es/



(a)

(b)

(c)

Fig. 3. Average evaluation indices with a different ξ of each method on the three benchmark data sets. (a) DB1. (b) DB2. (c) DB3.

TABLE I
PREDICTION ACCURACY OF EACH METHOD ON NORMAL DATA SET

Dataset	Method	RMSE	NSC	MAPE
DB1	Bayesian method	1.14096	0.99402	1.42500
	Delta method	1.29382	0.99230	1.57604
	MVE method	1.14552	0.99388	1.42674
	NCL-E-RVFL method	1.12735	0.99409	1.41423
	OPT-Bootstrap method	1.11964	0.99418	1.41458
	The proposed PIs	1.11260	0.99422	1.36204
DB2	Bayesian method	1.11048	0.95716	7.18476
	Delta method	1.10426	0.95764	7.10180
	MVE method	1.10626	0.95748	7.13730
	NCL-E-RVFL method	1.09318	0.95846	7.20250
	OPT-Bootstrap method	1.09894	0.95808	7.10942
	The proposed PIs	1.09100	0.95862	7.07682
DB3	Bayesian method	0.24920	0.99450	1.76116
	Delta method	0.25004	0.99444	1.78204
	MVE method	0.22866	0.99536	1.70420
	NCL-E-RVFL method	0.23002	0.99530	1.69902
	OPT-Bootstrap method	0.21420	0.99592	1.69280
	The proposed PIs	0.21352	0.99598	1.68230

TABLE II
EVALUATION INDICES OF PIs OF EACH METHOD ON NORMAL DATA SET

Dataset	Method	PICP	NMPIW	W-score
DB1	Bayesian method	90.958	0.07611	−9.2947
	Delta method	91.026	0.10230	−12.2041
	MVE method	91.370	0.08150	−9.7673
	NCL-E-RVFL method	92.400	0.07186	−8.6113
	OPT-Bootstrap method	93.082	0.07184	−8.5565
	The proposed PIs	92.602	0.07100	−8.3614
DB2	Bayesian method	95.332	0.19738	−11.3407
	Delta method	90.918	0.19224	−11.1702
	MVE method	90.250	0.17782	−10.2470
	NCL-E-RVFL method	91.250	0.15954	−8.9606
	OPT-Bootstrap method	91.000	0.15474	−8.7127
	The proposed PIs	92.334	0.15348	−8.6953
DB3	Bayesian method	92.000	0.03402	−1.4434
	Delta method	91.712	0.03676	−1.4741
	MVE method	90.668	0.03468	−1.4251
	NCL-E-RVFL method	92.284	0.03446	−1.4006
	OPT-Bootstrap method	90.382	0.03478	−1.4032
	The proposed PIs	90.878	0.03292	−1.3840

possesses a higher prediction accuracy on each normal data set. It can be observed from Table II that the proposed method has a reasonable PICP since its PICP is greater than the predefined CL = 90%, while the NMPIW and the absolute value of W-score are exactly smaller than that of other methods. This indicates that the proposed method can estimate narrower PIs and maintain the appropriate coverage probability on the three benchmark data sets.

To further evaluate the performance of the proposed method, $\xi\%$ ($\xi = 5, 10, 15, 20, 25$) of the total training samples is randomly chosen, and the noise, which is produced as $y \times \text{rand}(0, 1) \times [-25\%, 25\%]$, is introduced into these chosen training samples, where $\text{rand}(0, 1)$ denotes a random number in $(0, 1)$. The average evaluation indices with different ξ of each method are shown in Fig. 3. According to Fig. 3, one can see that when the noise contamination rate is increased, the proposed method can maintain smaller RMSE and MAPE and larger NSC than that of other methods on all the three data sets; this demonstrates that the prediction accuracy of the other comparative methods decreases more rapidly than that of the proposed method with the increase of noise contamination rate. The PICPs with different ξ of the proposed method are slightly small but still greater than the predefined CL (90%); the NMPIW and absolute value of W-score are exactly smaller than that of other representative methods with the increasing ξ. These results indicate that the proposed method can construct PIs with narrower interval width, better prediction accuracy, and the appropriate coverage probability, which can hold the predefined CL.

Remark 3: A large NMPIW can lead to a high PICP, but the PIs with extremely large interval width (i.e., large NMPIW) convey no information about the actual targets. Therefore, the optimal PIs should have small NMPIW, and the PICP should not be less than the predefined CL [9].

B. Prediction of Total Nitrogen in Crude Oil

In this experiment, the real-world data set is collected from the fast evaluation system for the physicochemical properties of crude oil in a refinery in Southern China. The input of the estimated PIs is the nuclear magnetic resonance (NMR) hydrogen spectrum data $\boldsymbol{x} \in R^{700}$. Total nitrogen content is an important physicochemical property for evaluating the crude oil, which exists in forms of different organic compounds. The existence of nitrogen in crude oil can lead to a series of problems, such as catalyst damage and reduction of storage stability. The nitrogen can also result in the darkened oil and the formations of colloids and precipitate in storage and transportation. Hence, fast evaluation of total nitrogen in crude oil is of great significance for increasing the economic benefits of refineries. Therefore, the total nitrogen content in crude oil is selected as the modeling output in this study.

A total of 863 sets of real-world data between May 2016 and October 2017 are collected. The 863 groups of NMR hydrogen spectra are shown in Fig. 4, where the "x-axis" represents the relative chemical shift, and the "z-axis" indicates the intensity of the hydrogen absorption peak at the corresponding chemical shift. The corresponding total nitrogen content of crude oil was collected from the laboratory of the refinery. The high dimensionality of NMR spectrum data may cause the problem of overfitting and increase the computational cost. The features of NMR spectrum data can be effectively extracted by the principal component analysis (PCA) [37]. Hence, the PCA is adopted to reduce the dimensionality of NMR spectrum data, and the number of principal components with a 99% cumulative percent variance contribution rate (CPVCR) is chosen.

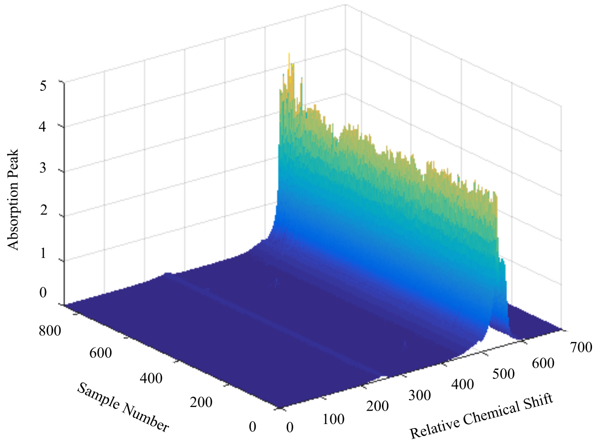


Fig. 4. 863 groups of the NMR hydrogen spectra.

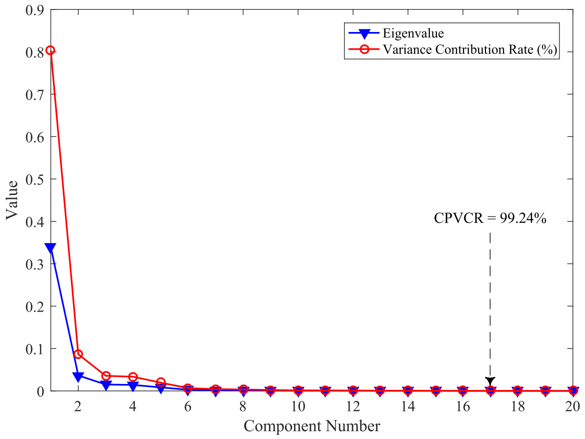


Fig. 5. Eigenvalue and variance contribution rate of each component.

By using PCA, the eigenvalue and variance contribution rate can be computed, as shown in Fig. 5, where only the first 20 terms are plotted for better visualization. From Fig. 5, we can conclude that the CPVCR of the first 17 terms is 99.24%. This means that the first 17 terms contain most of the information about the original input data. Accordingly, 17 principal components are chosen as the inputs of the PIs. Then, the data set is split into three parts, of which 733 groups are used as the training data, 55 groups are used as the validation data, and the remaining 75 groups are used for performance evaluation.

1) Parameter Selection: The predefined CL of the estimated PIs is set to 95%, i.e., $\text{CL} = (1 - \alpha)\% = 95\%$. The original data set is the training data set. The random parameters (input weights and biases) of the base SCNs in the ensemble are adaptively selected in the varying set $[-\mu, \mu]$, $\mu = 0.5, 1, 2, 4$, and the maximum random configuration time is $T_{\max} = 50$. The exhaustive linear search is also adopted to determine the number of hidden nodes L and the ensemble size K. The parameter L is searched in $[10, 100]$ with ten steps, and K is searched in $[70, 115]$ with five steps. The initial values of hyperparameters are set to $\sigma_{\varepsilon}^{2} = 2$ and $\sigma_{\beta}^{2} = 5$. The influences of different L and K on the results of the validation

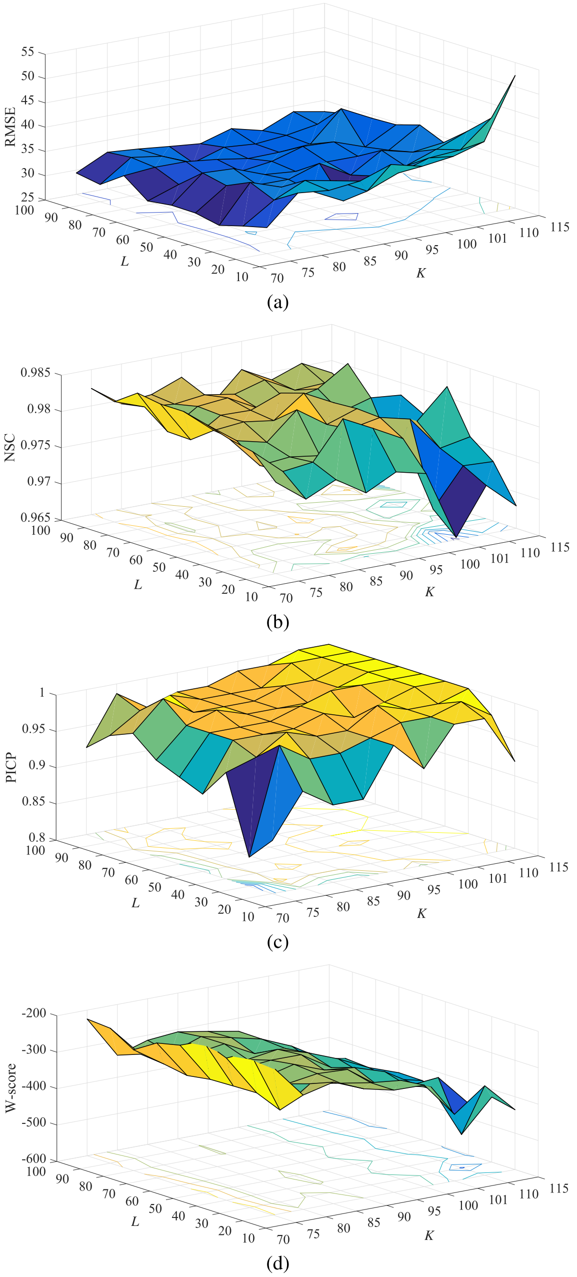


Fig. 6. Influence of different L and K on the performance of PIs. (a) RMSE. (b) NSC. (c) PICP. (d) W-score.

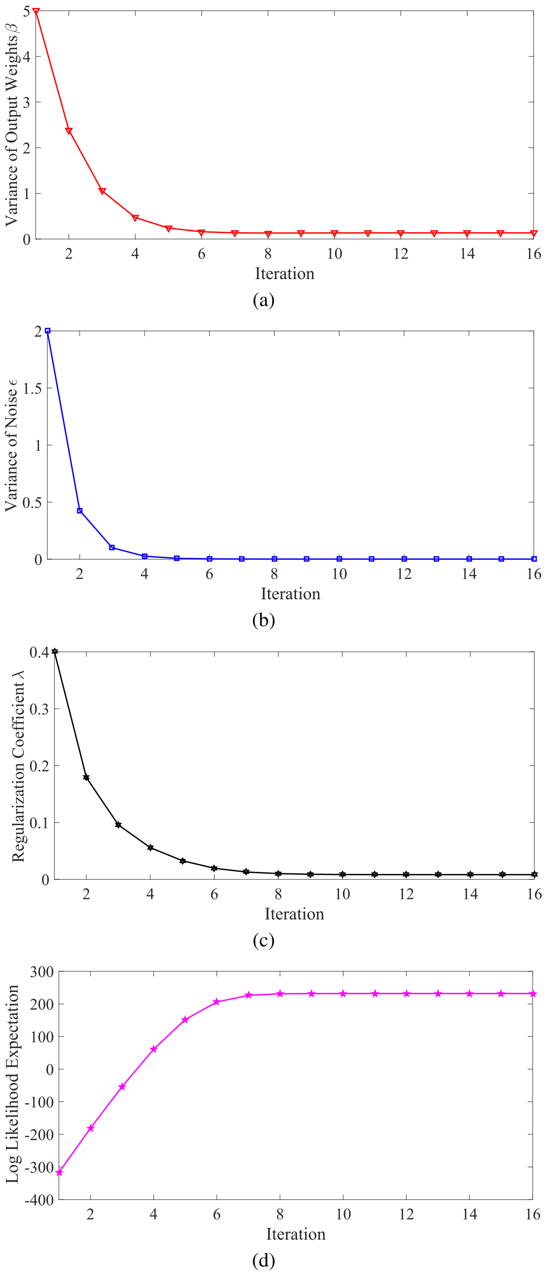


Fig. 7. Convergence process. (a) σ_β^2. (b) σ_ε^2. (c) λ. (d) Log likelihood expectation of the complete data.

data set are shown in Fig. 6. It can be seen that with the increase of the parameters L and K, the greater PICP can be obtained. However, the sharpness and prediction accuracy of the estimated PIs decrease. Therefore, in view of the coverage probability, interval width, and prediction accuracy of the estimated PIs, the parameters L and K are set to $L = 70$ and $K = 90$, respectively.

2) Comparison of Experimental Results: The convergences of hyperparameters in the training process of ensemble SCNs, which includes the variance σ_β^2 of output weights, the variance σ_ε^2 of noise, and the regularization coefficient $\lambda = \sigma_\varepsilon^2/\sigma_\beta^2$ are shown in Fig. 7. As shown in Fig. 7(a), (b), and (d), the optimal hyperparameters σ_ε^2 and σ_β^2 can be obtained after several iterations. From Fig. 7(c), it can be observed that the optimal regularization coefficient can be obtained by iterative optimization. This means that the optimal output weights of the ensemble SCNs can be achieved.

To better illustrate the superiority of the proposed method, Table III gives the PICPs, NMPIWs, and W-scores of the proposed method and the existing methods, which are applied to quantify the quality of PIs. From Table III, we can observe that the PICP of the proposed method is greater than that of the existing methods, while the NMPIW and the absolute value of the W-score are smaller. According to these results, we can conclude that the estimated PIs of the proposed method have higher coverage probability and narrower shape, which indicates the effectiveness of the proposed method.

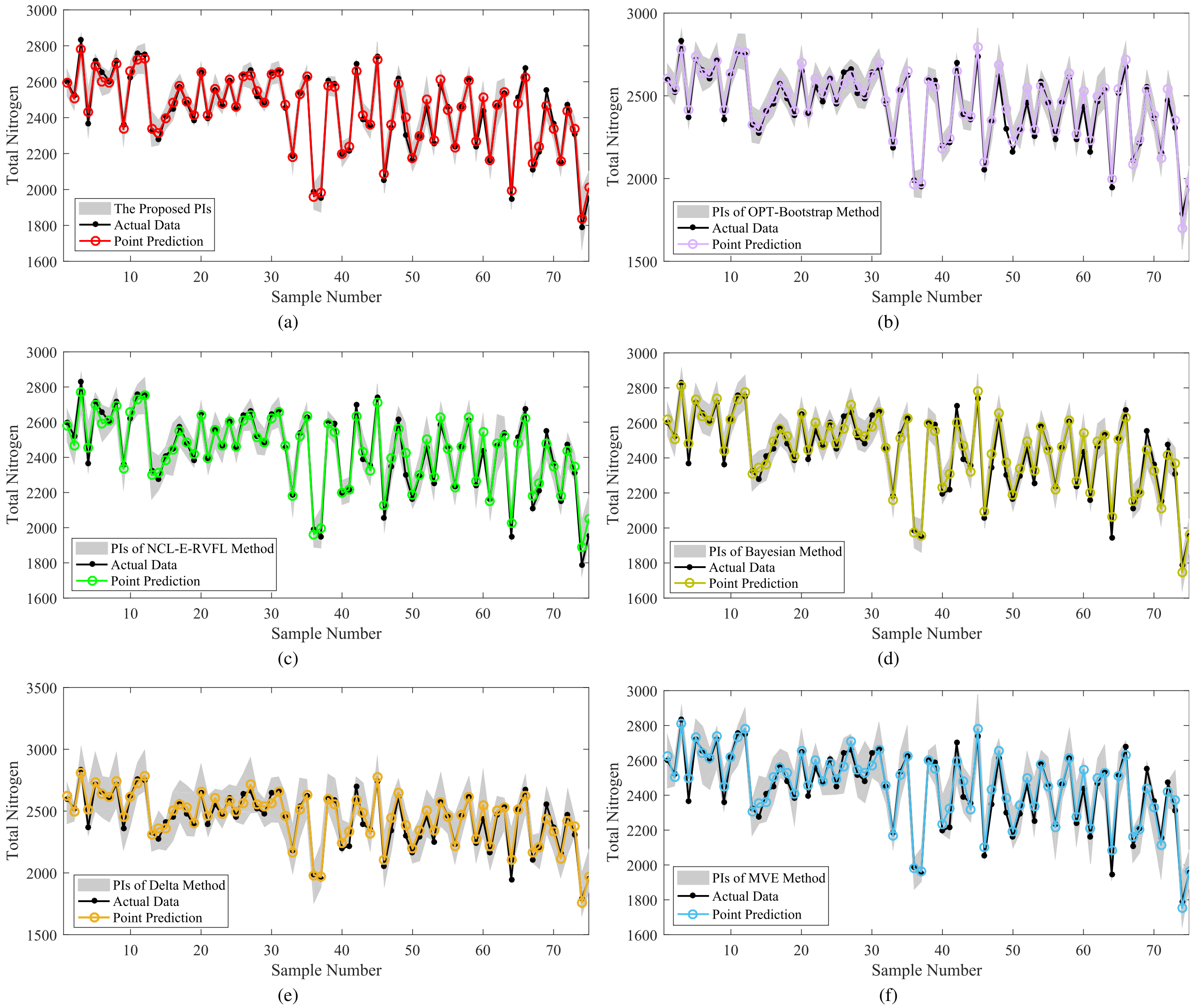


Fig. 8. Estimated PIs and prediction results of each method. (a) Proposed method. (b) OPT-bootstrap method. (c) NCL-E-RVFL method. (d) Bayesian method. (e) Delta method. (f) MVE method.

TABLE III
EVALUATION INDICES OF PIs USING EACH METHOD

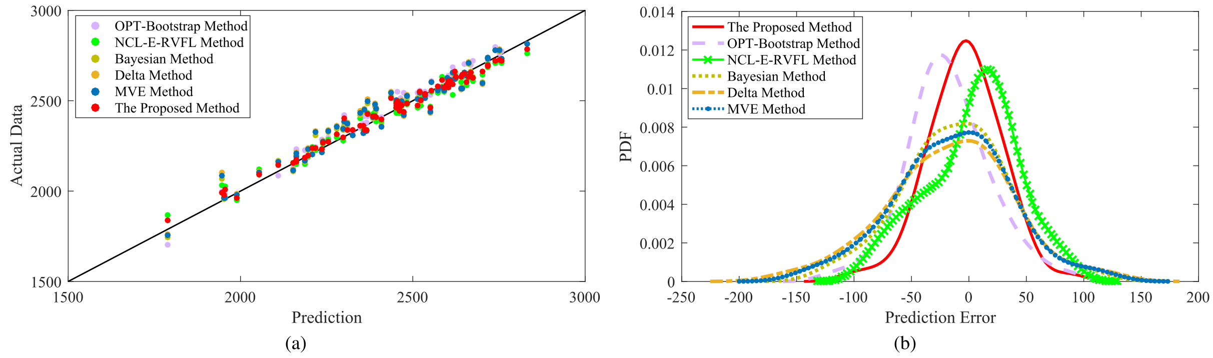
Method	PICP	NMPIW	W-score
Bayesian method	92.000	0.1798	−353.976
Delta method	94.667	0.3101	−619.092
MVE method	96.000	0.2203	−440.568
NCL-E-RVFL method	96.534	0.1623	−336.199
OPT-Bootstrap method	97.866	0.1561	−326.161
The proposed PIs	98.667	0.1447	−314.301

TABLE IV
PREDICTION ACCURACY COMPARISON OF EACH METHOD

Method	RMSE	MAPE	NSC
Bayesian method	46.836	1.5413	0.9552
Delta method	55.485	1.7951	0.9371
MVE method	51.465	1.6789	0.9459
NCL-E-RVFL method	40.735	1.2914	0.9601
OPT-Bootstrap method	36.682	1.1767	0.9694
The proposed PIs	29.278	0.9989	0.9798

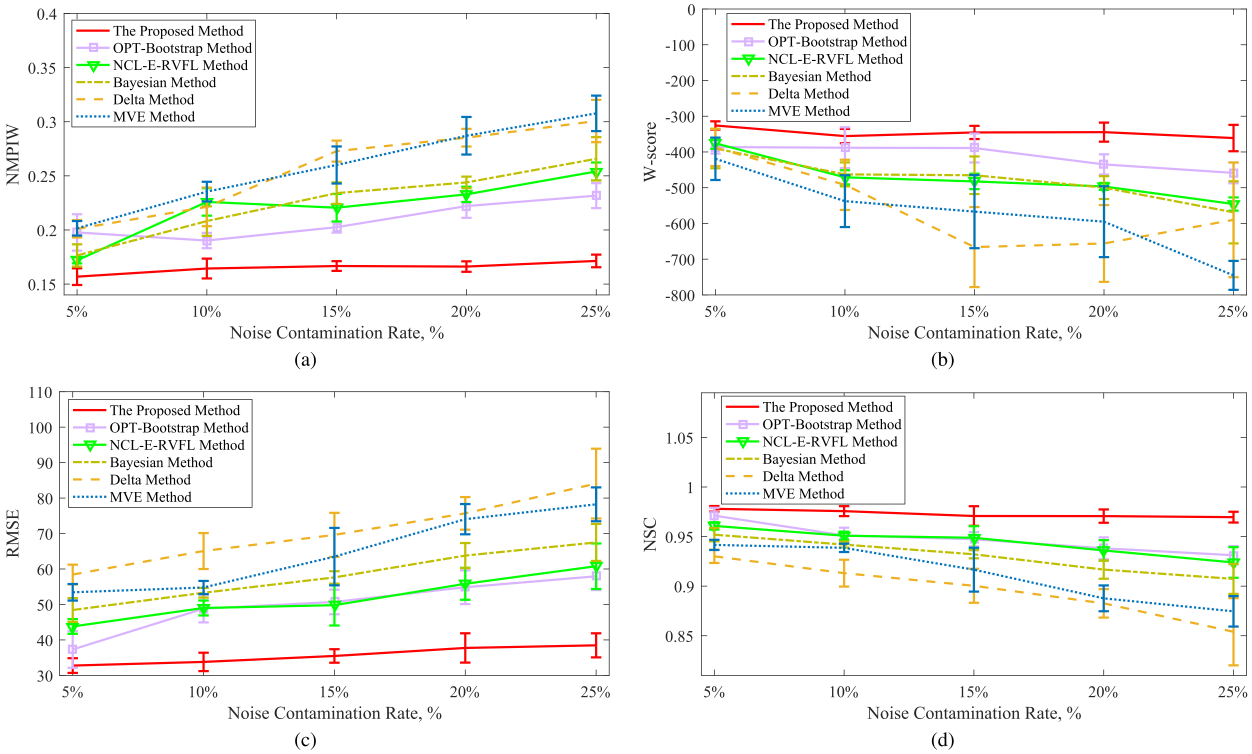
The estimated PIs and point prediction results of the proposed method and the other five comparative methods are given in Fig. 8. It can be seen from Fig. 8(a) that the estimated PIs based on the proposed method can cover all other output variables besides one point, which indicates the high reliability of the estimated PIs of the proposed method. One can also see that the point prediction of the proposed PIs can fit the actual output variables better. It can also capture the trend of the output variables, which exhibits good prediction accuracy. According to Fig. 8(b)–(f), we can notice that the number of output variables that are not covered by the estimated PIs of the five existing methods is greater than that of the proposed method. This result indicates that the estimated PIs of the proposed method outperform that of the existing methods in terms of coverage probability.

The RMSEs, MAPEs, and NSCs of the proposed method and the existing methods are given in Table IV for evaluating the prediction accuracy. The NSC takes the value in $[-\infty, 1]$, and the greater value of NSC means that the prediction model has higher prediction accuracy. From Table IV, it can be



(a) (b)

Fig. 9. Scatter diagram of prediction results and the PDF of the prediction errors of each method. (a) Scatter diagram. (b) PDF.



(a) (b)

(c) (d)

Fig. 10. Performance of each method with a different ξ. (a) NMPIW. (b) W-score. (c) RMSE. (d) NSC.

observed that the RMSE and MAPE of the proposed method are smaller than that of the existing methods. On the contrary, the NSC is greater than that of the existing methods. Therefore, the proposed method has better accuracy than the existing methods. The scatter plots of the actual values and prediction values and the PDF of the prediction error are shown in Fig. 9. According to Fig. 9(a), we can see that prediction values of the proposed method are closer to the actual values than that of the other five comparative methods. Moreover, it can be also concluded from Fig. 9(b) that the PDF of prediction errors of the proposed method exhibits a narrower impulse shape around zero; this demonstrates that the mean value of prediction errors is zero in the perspective of probability.

To further verify the robustness of the proposed method, the noise, which is generated in a manner similar to that in the previous experiment in Section IV-A, is introduced into the training data. The variations of mean values and standard deviations of the NMPIW, W-score, RMSE, and NSC with different noise contamination rates of each method are exhibited in Fig. 10, and the mean values of PICP with different noise contamination rate ξ of each method are listed in Table V. According to the comparisons of NMPIW

TABLE V

Mean Values of PICP With a Different ξ of Each Method

Method	ξ				
	5	10	15	20	25
Bayesian method	94.13	95.20	96.80	95.20	94.32
Delta method	93.06	93.06	96.53	94.66	93.33
MVE method	94.93	97.86	96.00	95.46	95.46
NCL-E-RVFL method	96.80	96.00	95.47	96.53	96.80
OPT-Bootstrap method	96.53	96.27	96.00	97.60	95.45
The proposed PIs	97.06	97.77	97.06	97.77	97.46

TABLE VI
COMPUTATIONAL EFFICIENCY COMPARISON OF EACH METHOD

Method	Training time (s)	Testing time (s)
Bayesian method	0.3876 ± 0.0097	0.0066 ± 0.0003
Delta method	0.5585 ± 0.0105	0.0073 ± 0.0010
MVE method	0.2749 ± 0.0051	0.0057 ± 0.0004
NCL-E-RVFL method	12.3214 ± 1.5013	0.0976 ± 0.0123
OPT-Bootstrap method	310.8807 ± 21.8524	0.0979 ± 0.0272
The proposed PIs	40.4074 ± 3.3331	0.0983 ± 0.0120

and W-score shown in Fig. 10(a) and (b) as well as the comparisons of RMSE and NSC shown in Fig. 10(c) and (d), one can find that the estimated PIs of the proposed method can maintain narrower shape and higher prediction accuracy with the increase in the noise contamination rate. It can also be seen from Fig. 10 that when the noise contamination rate is increased, the prediction accuracy and the sharpness of the estimated PIs of the existing methods decrease more rapidly than that of the proposed method. It can be observed from Table V that although the noise is introduced, the proposed method and the existing methods can maintain a large coverage probability. Only in the case of $\xi = 10$, the MVE method outperforms the proposed method. It can be seen from Fig. 10(a) that the interval width of the MVE method is larger than that of the proposed method. As mentioned in Remark 3, a wider interval may lead to a higher PICP. Therefore, the case that the PICP of MVE method is larger than that of the proposed algorithm can occur. Therefore, it can be concluded that the estimated PIs of the proposed method are more robust than the PIs that are estimated by using the existing methods.

The computational efficiency (training time and testing time) of each method is listed in Table VI. One can see from Table VI that the iterative optimization process of the proposed method slows the training process. The training time of the method based on ensemble neural networks is obviously longer than that of the method based on the single-neural network. The modified firefly algorithm is adopted in the OPT-bootstrap method, which leads to the expensive computational cost. The testing time of the proposed method is slightly longer than that of the method based on the single-neural network, but the testing time is still acceptable in the real-world case carried out in this article.

From the above-mentioned comparison of the experimental results on the real-world data set, we can see that compared with the representative methods of estimating PIs, the proposed method can estimate PIs with higher coverage probability, better prediction accuracy, and narrower interval width. It can effectively quantify the potential uncertainties associated with the targets. The proposed method is also robust to the noise and outliers to a certain extent. Moreover, the proposed method can also guarantee the computational efficiency and meet the requirement of application in the refinery. The experimental results also demonstrate that the PIs estimated by the proposed method can provide reliable and useful information for the decision-making process of crude oil refining.

V. CONCLUSION

This article presents a novel method of estimating PIs by using ensemble SCNs and bootstrap method. A simultaneous robust training algorithm of the ensemble SCNs based on the Bayesian ridge regression and M-estimate is developed, which can achieve the cooperation among the base SCNs and obtain considerable robustness. The EM algorithm is derived to estimate the hyperparameters of some assumed prior distributions; hence, the estimated PIs with better prediction accuracy and higher reliability are obtained. Moreover, the experiments are carried out using three benchmark data sets and a real-world data set collected from a refinery. The experimental results illustrate that the proposed method can estimate PIs with high quality and guarantee both certain robustness and computational efficiency. The experimental results on the real-world data set also demonstrate that the proposed method is suitable for the application in the fast evaluation of physicochemical properties of crude oil, and it also can be beneficial to the decision-making process of crude oil refining.

ACKNOWLEDGMENT

The authors would like to thank Prof. D. Wang from La Trobe University for the helpful suggestions on this article.

REFERENCES

[1] W. Dai, Q. Liu, and T. Chai, "Particle size estimate of grinding processes using random vector functional link networks with improved robustness," *Neurocomputing*, vol. 169, pp. 361–372, Dec. 2015.

[2] B. Miskony and D. Wang, "A randomized algorithm for prediction interval using RVFL networks ensemble," in *Proc. Int. Conf. Neural Inf. Process.*, Nov. 2017, pp. 51–60.

[3] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1341–1356, Sep. 2011.

[4] D. L. Shrestha and D. P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," *Neural Netw.*, vol. 19, no. 2, pp. 225–235, Mar. 2006.

[5] D. J. C. Mackay, "The evidence framework applied to classification networks," *Neural Comput.*, vol. 4, no. 5, pp. 720–736, Sep. 1992.

[6] T. Heskes, "Practical confidence and prediction intervals," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 176–182.

[7] T. Lu and M. Viljanen, "Prediction of indoor temperature and relative humidity using neural network models: Model comparison," *Neural Comput. Appl.*, vol. 18, no. 4, pp. 345–357, May 2009.

[8] E. Mazloumi, G. Rose, G. Currie, and S. Moridpour, "Prediction intervals to account for uncertainties in neural network predictions: Methodology and application in bus travel time prediction," *Eng. Appl. Artif. Intell.*, vol. 24, no. 3, pp. 534–542, Apr. 2011.

[9] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 337–346, Mar. 2011.

[10] N. A. Shrivastava, A. Khosravi, and B. K. Panigrahi, "Prediction interval estimation of electricity prices using PSO-tuned support vector machines," *IEEE Trans. Ind. Inf.*, vol. 11, no. 2, pp. 322–331, Apr. 2015.

[11] J. Lu and J. Ding, "Construction of prediction intervals for carbon residual of crude oil based on deep stochastic configuration networks," *Inf. Sci.*, vol. 486, pp. 119–132, Jun. 2019.

[12] S. Scardapane, D. Wang, and A. Uncini, "Bayesian random vector functional-link networks for robust data modeling," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 2049–2059, Jul. 2018.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[14] S. Lee, M. Bolic, V. Z. Groza, H. R. Dajani, and S. Rajan, "Confidence interval estimation for oscillometric blood pressure measurements using bootstrap approaches," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 10, pp. 3405–3415, Oct. 2011.

[15] A. Khosravi, S. Nahavandi, D. Srinivasan, and R. Khosravi, "Constructing optimal prediction intervals by using neural networks and bootstrap method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1810–1815, Aug. 2015.

[16] A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction intervals for short-term wind farm power generation forecasts," *IEEE Trans. Sustain. Energy*, vol. 4, no. 3, pp. 602–610, Jul. 2013.

[17] H. Quan, D. Srinivasan, and A. Khosravi, "Incorporating wind power forecast uncertainties into stochastic unit commitment using neural network-based prediction intervals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2123–2135, Sep. 2015.

[18] C. Liguori, A. Ruggiero, P. Sommella, and D. Russo, "Choosing bootstrap method for the estimation of the uncertainty of traffic noise measurements," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 5, pp. 869–878, May 2017.

[19] A. Khosravi, S. Nahavandi, and D. Creighton, "Quantifying uncertainties of neural network-based electricity price forecasts," *Appl. Energy*, vol. 112, pp. 120–129, Dec. 2013.

[20] C. Lian, Z. Zeng, W. Yao, H. Tang, and C. L. P. Chen, "Landslide displacement prediction with uncertainty based on neural networks with random hidden weights," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2683–2695, Dec. 2016.

[21] Y. Liu and X. Yao, "Simultaneous training of negatively correlated neural networks in an ensemble," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 6, pp. 716–725, Dec. 1999.

[22] M. Islam, X. Yao, and K. Murase, "A constructive algorithm for training cooperative neural network ensembles," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 820–834, Jul. 2003.

[23] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Netw.*, vol. 12, no. 10, pp. 1399–1404, Dec. 1999.

[24] C. Sheng, J. Zhao, W. Wang, and H. Leung, "Prediction intervals for a noisy nonlinear time series based on a bootstrapping reservoir computing network ensemble," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 7, pp. 1036–1048, Jul. 2013.

[25] M. Alhamdoosh and D. Wang, "Fast decorrelated neural network ensembles with random weights," *Inf. Sci.*, vol. 264, pp. 104–117, Apr. 2014.

[26] Y. Xu, C. Mi, Q.-X. Zhu, J.-Y. Gao, and Y.-L. He, "An effective high-quality prediction intervals construction method based on parallel bootstrapped RVM for complex chemical processes," *Chemometrics Intell. Lab. Syst.*, vol. 171, pp. 161–169, Dec. 2017.

[27] D. Wang and M. Li, "Stochastic configuration networks: Fundamentals and algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3466–3479, Oct. 2017.

[28] D. Wang and C. Cui, "Stochastic configuration networks ensemble for large-scale data analytics," *Inf. Sci.*, vol. 417, pp. 55–71, Jul. 2017.

[29] P. Zhou, Y. Lv, H. Wang, and T. Chai, "Data-driven robust RVFLNs modeling of a blast furnace iron-making process using cauchy distribution weighted M-estimation," *IEEE Trans. Ind. Electron.*, vol. 64, no. 9, pp. 7141–7151, Sep. 2017.

[30] D. Wang and M. Li, "Robust stochastic configuration networks with kernel density estimation for uncertain data regression," *Inf. Sci.*, hboxvols. 412–413, pp. 210–222, Oct. 2017.

[31] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov.1995.

[32] W. Härdle and L. Simar, *Applied Multivariate Statistical Analysis*. New York, NY, USA: Springer, 2007.

[33] C. Rasmussen and C. K. I. Williams, *Gaussian Process for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[34] R. A. Maronna, D. R. Martin, and V. J. Yohai, *Robust Statistics: Theory Methods*. New York, NY, USA: Wiley, 2006.

[35] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image Vis. Comput.*, vol. 15, no. 1, pp. 59–76, Jan. 1997.

[36] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.

[37] A. Masili, S. Puligheddu, L. Sassu, P. Scano, and A. Lai, "Prediction of physical–chemical properties of crude oils by ^{1}H NMR analysis of neat samples and chemometrics," *Magn. Reson. Chem.*, vol. 50, no. 11, pp. 729–738, Nov. 2012.

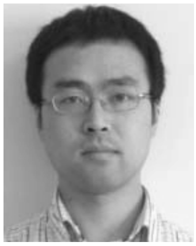
Jun Lu received the B.S. degree from the Shenyang University of Technology, Shenyang, China, in 2012, and the M.S. degree from Northeastern University, Shenyang, in 2014, where he is currently pursuing the Ph.D. degree in control theory and control engineering with the State Key Laboratory of Synthetical Automation for Process Industries.

His current research interests include randomized-learning-algorithms-based neural networks, robust modeling techniques, the Bayesian learning, and their applications in complex industrial processes.

Jinliang Ding (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2012.

He is currently a Professor with the State Key Laboratory of Synthetical Automation for Process Industry, Northeastern University. He has authored or coauthored over 100 refereed journal articles and refereed articles at international conferences. He is also the inventor or a co-inventor of 17 patents. His current research interests include modeling, plant-wide control, and optimization for the complex industrial systems, stochastic distribution control, and multiobjective evolutionary algorithms and its applications.

Dr. Ding was a recipient of the Young Scholars Science and Technology Award of China in 2016 and the National Science Fund for Distinguished Young Scholars in 2015. He received the National Technological Invention Award in 2013 and three First-Prize of Science and Technology Awards of the Ministry of Education in 2006, 2012, and 2018, respectively. One of his articles published in *Control Engineering Practice* was selected for the Best Paper Award of the period 2011–2013.



Xuewu Dai (Member, IEEE) received the B.Eng. degree in electronic engineering and the M.Sc. degree in computer science from Southwest University, Chongqing, China, and the Ph.D. degree from The University of Manchester, Manchester, U.K.

He is currently a Professor with the State Key Laboratory of Synthetical Automation for Process Industry, Northeastern University, Shenyang, China. His interests include robust state estimation and condition monitoring of industrial systems, wireless sensor actuator networks, and the Industrial Internet of Things. He is also interested in network control systems and train rescheduling.

Tianyou Chai (Fellow, IEEE) received the Ph.D. degree in control theory and engineering from Northeastern University, Shenyang, China, in 1985.

He is currently a Professor with the State Key Laboratory of Synthetical Automation for Process Industry, Northeastern University, Shenyang. He has authored more than 120 peer-reviewed international journal articles and around 224 international conference papers. His current research interests include adaptive control, intelligent decoupling control, and the development of control technologies with applications to various industrial processes.

Dr. Chai is also a member of the Chinese Academy of Engineering, an Academician of the International Eurasian Academy of Sciences, and a fellow of the International Federation of Automatic Control. He is also a Distinguished Visiting Fellow of the Royal Academy of Engineering, U.K., and an Invitation Fellow of the Japan Society for the Promotion of Science. He received the 2002 Technological Science Progress Award from the Ho Leung Ho Lee Foundation, the 2007 Industry Award for Excellence in Transitional Control Research from the IEEE Control Systems Society, and the 2010 Yang Jia-Chi Science and Technology Award from the Chinese Association of Automation.