Mixed-Distribution-Based Robust Stochastic Configuration Networks for Prediction Interval Construction

Jun Lu and Jinliang Ding, *Senior Member, IEEE*

***Abstract*—It is challenging to develop point prediction models with high accuracy due to that outliers and noise are commonly present in the real-world data. In this context, this article proposes a novel robust stochastic configuration network (SCN) and uses the bootstrap ensemble strategy to construct prediction intervals (PIs). Since the output weights of the original SCN are computed by the least-squares method, which is sensitive to noise with an unknown distribution or outliers, a robust SCN based on a mixture of the Gaussian and Laplace distributions (MoGL-SCN) in the Bayesian framework is proposed. The mixed distributions can effectively characterize the complex distributions of the real-world data, and their heavy-tailed properties can improve the robustness of SCNs. Furthermore, there are no analytical solutions available to obtain the network parameters due to the assumption on the mixed distributions, hence, the parameters of the MoGL-SCN are estimated by the expectation–maximization algorithm. In addition, considering the uncertainties caused by both the model mismatch and noise in the real-world data, a bootstrap ensemble strategy using MoGL-SCN is designed to construct the PIs. The experimental results on two benchmark datasets and a real-world dataset demonstrate the effectiveness of the proposed method in terms of the quality of PIs, prediction accuracy, and robustness.**

***Index Terms*—Bootstrap, expectation-maximization (EM) algorithm, mixed distributions, prediction intervals, robust modeling, stochastic configuration networks.**

I. Introduction

PREDICTING the key variables in industrial processes is crucial for managers and engineers to make appropriate decisions, and the data-driven prediction models for the key process variables have been widely developed [1], [2]. So far, many data-driven prediction models have been applied to the industrial processes, such as the debutanizer column [3] and mineral grinding process [4]. Among the data-driven methods, one of the single hidden layer feed-forward networks (SLFNs), namely, the random vector functional-link (RVFL) network [5] has drawn increasing attention and achieved satisfactory application performance [6]. But determining the ranges of the input weights and biases of RVFL is challenging [7]. To solve this problem, an innovative randomized learner model, termed stochastic configuration networks (SCNs), was proposed in [8]. The input weights and biases of SCNs are generated in varying ranges and determined by a data-dependent supervisory mechanism [8], and then, these randomly generated parameters are kept fixed. Hence, compared with the traditional neural networks, the simple structure and fast learning speed of SCNs can reduce the computational cost. The supervisory mechanism suggested in [8] can effectively avoid producing junk nodes and guarantee the universal approximation property of SCNs. In addition, the inequality supervisory mechanism of the SCN for the selection of random parameters can exactly improve the prediction performance [8]. Therefore, the SCN and its variants have been successfully applied in the field of data modeling with promising performance [9], [10].

However, in the real-world applications, most data are collected in noisy environments, therefore, outliers are commonly present owing to the influence of different types of noise. If a training dataset is contaminated with unknown noise or outliers, the accuracy and reliability of the resulting model will deteriorate [11]. Recently, data-driven robust modeling methods have become increasingly popular. M-estimation is a commonly used robust technique that can eliminate the influence of noise or outliers on the modeling performance by constructing robust cost functions [12], and it has been successfully used to build robust back-propagation neural networks (BPNNs) [13] and robust self-organizing maps (SOMs) [14]. However, the BP-based algorithms suffer from the problem of parameter initialization and also have some drawbacks such as slow convergence and convergence to local optima [15]. To solve these problems, the robust RVFLs based on the M-estimation and kernel density estimation (KDE) have been studied and successfully used in the blast furnace iron-making process [16] and grinding process [17]. Moreover, the KDE method has also been implemented to build

Manuscript received June 15, 2019; revised September 29, 2019 and October 31, 2019; accepted November 5, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61988101, Grant 61590922, Grant 61525302, 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. Paper no. TII-19-2527. *(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).

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/TII.2019.2954351

the robust SCNs and the resulting robust SCNs have obtained the satisfied performance in the industrial applications [18].

Additionally, neural networks in the Bayesian framework have been extensively studied, such as the multilayer perceptron networks [19] and the reservoir computing networks [20], in which the noise is assumed to be of the Gaussian distribution. It is well known that the Gaussian distribution is not robust to outliers. The Laplace distribution is insensitive to noise and outliers due to its heavy-tailed property [21], [22]. Therefore, the echo state networks and the neural networks with random weights in the Bayesian framework have also been studied, in which the noise and outliers are assumed to follow a single Laplace distribution [23], [24]. However, in numerous real-world applications, the distribution of noise or outliers may be more complex due to the uncertain and heterogeneous environments [25]. As a result, no single distribution may be appropriate. Therefore, the assumption on a specific distribution of noise or outlier may lead to weak robustness and low prediction accuracy and inhibit optimal modeling performance. Compared with using a specific distribution, the mixture of different types of distributions can provide a better characterization of the complex statistical distribution of noise or outliers, and the heavy-tailed properties of mixed distributions can improve the robustness of the resulting model.

Robust data modeling techniques can alleviate some uncertainties caused by noise or outliers. Furthermore, the uncertainty from mismatching parameters of the models should also be taken into account. Moreover, the uncertainties caused by the real-world data and model mismatch can lead to unacceptable prediction performance if the point prediction occurs without performing a quantitative reliability analysis of the prediction errors [26], [27]. Fortunately, the prediction intervals (PIs) can overcome the deficiencies of the traditional point prediction methods by considering the uncertainties caused by both the real-world data and the model mismatch [28]. The PIs have been well used in the real-world applications such as the wind power generation process [29], the traffic noise measurement [30], and the prediction of gas flow in the blast furnace [31]. In the methods of constructing PIs, the bootstrap strategy is the most suitable candidate due to that it can construct reliable PIs and reduce the influence of model mismatch, it also has the advantage of easy implementation [32], [33]. Hence, the bootstrap strategy is preferable for constructing PIs.

In this article, we aim to develop a novel robust estimation approach with SCNs to improve the prediction performance under a related assumption on noise distribution, resulting in a robust SCN model based on a mixture of Gaussian and Laplace distributions (MoGL-SCN) for constructing PIs. The heavy-tailed properties of the mixed Gaussian and Laplace distributions can improve the robustness of the model and alleviate the influence of noise and outliers on the modeling performance. Moreover, due to the assumption on the mixed distributions, the parameters of the MoGL-SCN have no analytical solutions, therefore, the expectation-maximization (EM)-algorithm-based parameter estimation is derived. Furthermore, to quantify the reliability and uncertainty of the point prediction results, the PIs are developed using the bootstrap ensemble MoGL-SCN (termed BE-MoGL-SCN). The performance of the proposed method is evaluated on two benchmark datasets and a real-world dataset. The experimental results indicate the effectiveness of the proposed method.

The rest of this article is organized as follows. Section II briefly introduces the SCNs, the properties of the Laplace distribution. Section III presents the proposed MoGL-SCN, the EM-algorithm-based parameter estimation of the MoGL-SCN and the PIs based on bootstrap ensemble MoGL-SCN. Sections IV and V give the experimental results on two benchmark datasets and a real-world dataset, respectively. Finally, Section VI concludes this article.

II. Preliminaries

This section briefly introduces the SCN concept [8] and some properties of the Laplace distribution.

A. Stochastic Configuration Networks

Assume that a set of data $\boldsymbol{D} = \{\boldsymbol{X}, \boldsymbol{y}\} = \{(\boldsymbol{x}_n, y_n) \in R^d \times R\}_{n=1}^N$ is given. An SCN with $P-1$ hidden nodes can be described as follows:

$$f_{P-1}(\boldsymbol{X};\boldsymbol{\beta}) = \sum_{n=1}^{N}\sum_{p=1}^{P-1} \beta_p g_p(\boldsymbol{w}_p^T \boldsymbol{x}_n + b_p) = \boldsymbol{H}(\boldsymbol{X})\boldsymbol{\beta} \quad (1)$$

where $P = 1, \ldots, p = 1, \ldots, P-1$, $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_{P-1}]^T$ denotes the output weights, $\boldsymbol{w}_p \in R^d$ and $b_p \in R$ are the input weights and bias of the pth hidden node, respectively, and $g(\cdot)$ denotes an activation function. The output matrix $\boldsymbol{H}(\boldsymbol{X})$ of the hidden layer is defined as follows:

$$\begin{cases} \boldsymbol{H}(\boldsymbol{X}) = & [\boldsymbol{h}^T(\boldsymbol{x}_1), \ldots, \boldsymbol{h}^T(\boldsymbol{x}_n), \ldots, \boldsymbol{h}^T(\boldsymbol{x}_N)]^T \\ \boldsymbol{h}(\boldsymbol{x}_n) = & [g_1(\boldsymbol{w}_1^T \cdot \boldsymbol{x}_n + b_1), \\ & \quad \ldots, g_{P-1}(\boldsymbol{w}_{P-1}^T \cdot \boldsymbol{x}_n + b_{P-1})] \end{cases} \quad (2)$$

where the superscript T denotes the matrix transpose.

Then, $\boldsymbol{\beta}$ can be obtained by the least-squares method [8], [10]

$$\begin{aligned} \boldsymbol{\beta}^* &= \arg\min_{\boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{H}(\boldsymbol{X})\boldsymbol{\beta}\|_2^2 \\ &= [\boldsymbol{H}(\boldsymbol{X})^T \boldsymbol{H}(\boldsymbol{X})]^{-1} \boldsymbol{H}(\boldsymbol{X})^T \boldsymbol{y} \end{aligned} \quad (3)$$

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

If the SCN with $P-1$ hidden nodes does not meet the termination criterion, a new hidden node should be produced and its output is expressed as follows:

$$\boldsymbol{G}_P(\boldsymbol{X}) = [g_P(\boldsymbol{w}_P^T \boldsymbol{x}_1 + b_P), \ldots, g_P(\boldsymbol{w}_P^T \boldsymbol{x}_N + b_P)]^T. \quad (4)$$

The input weights $\boldsymbol{w}_P$ and bias b_P of the new hidden node should satisfy the following supervisory mechanism:

$$\begin{aligned} \zeta = & \left\langle \boldsymbol{e}_{P-1}^T, \boldsymbol{G}_P(\boldsymbol{X}) \right\rangle^2 / \left\langle \boldsymbol{G}_P^T(\boldsymbol{X}), \boldsymbol{G}_P(\boldsymbol{X}) \right\rangle \\ & - (1 - r - \rho_P) \times \left\langle \boldsymbol{e}_{P-1}^T, \boldsymbol{e}_{P-1} \right\rangle > 0 \end{aligned} \quad (5)$$

where $\boldsymbol{e}_{P-1} = \boldsymbol{y} - \boldsymbol{H}(\boldsymbol{X})\boldsymbol{\beta}^*$ represents the residual error vector of the SCN with $P-1$ hidden nodes, $0 < \rho_P < 1 - r$, $0 < r < 1$, $\lim_{P \to \infty} \rho_P = 0$, and $\langle \cdot, \cdot \rangle$ is the scalar product.

The new hidden nodes are generated until some relevant termination criteria (i.e., the predefined error tolerance or the maximum number of hidden nodes) are met. More details about the SCNs can be found in [8].

Remark 1: According to the SCN algorithm, $T_{\max}$ new hidden nodes are produced and the input weights and biases are assigned in ranges of $[-\lambda_j, \lambda_j]$, $j = 1, \ldots, J$. The node with the largest ζ is chosen as the newly added one [8], [10].

B. Properties of the Laplace Distribution

The basic probability density function (PDF) of the Laplace distribution can be written as follows:

$$\mathcal{L}(x|\mu,\eta) = \frac{1}{\sqrt{2\eta^2}} \exp\left(-\frac{\sqrt{2}\,|x-\mu|}{\eta}\right) \tag{6}$$

where x denotes a random variable, and μ and $\eta > 0$ represent the location and scale parameters, respectively.

A random variable that follows a Laplace distribution can be represented as a mixture of random variables that follow a normal distribution and a distribution related to the exponential distribution [21], [22]. A random variable v is introduced that follows a distribution related to the exponential distribution, and its PDF is defined as follows:

$$\mathfrak{g}(v) = \frac{1}{v^3} \exp\left(-\frac{1}{2v^2}\right). \tag{7}$$

If v is given, then the conditional distribution of x is a normal distribution [21], [22]

$$\mathcal{N}(x|v,\mu,\eta) = \frac{v}{\sqrt{\pi\eta^2}} \exp\left[-\frac{v^2(x-\mu)^2}{\eta^2}\right]. \tag{8}$$

As described in [21] and [22], by introducing v, we can obtain the following PDF of the joint distribution:

$$\begin{aligned}\mathcal{L}(x,v|\mu,\eta) &= \mathcal{N}(x|v,\mu,\eta)\cdot\mathfrak{g}(v)\\ &= \frac{1}{v^2\sqrt{\pi\eta^2}} \exp\left[-\frac{v^2(x-\mu)^2}{\eta^2} - \frac{1}{2v^2}\right].\end{aligned} \tag{9}$$

III. Prediction Intervals Based on the BE-MoGL-SCN

This section details the proposed MoGL-SCN framework including the process of parameter estimation and prediction intervals construction.

A. Robust SCN Based on the Mixture of Gaussian and Laplace Distributions

It is well known that the data collected from the real-world applications are uncertain and may be influenced by unknown noise or outliers. Consequently, a robust SCN based on the mixture of Gaussian and Laplace distributions is presented, and the structure is shown in Fig. 1.

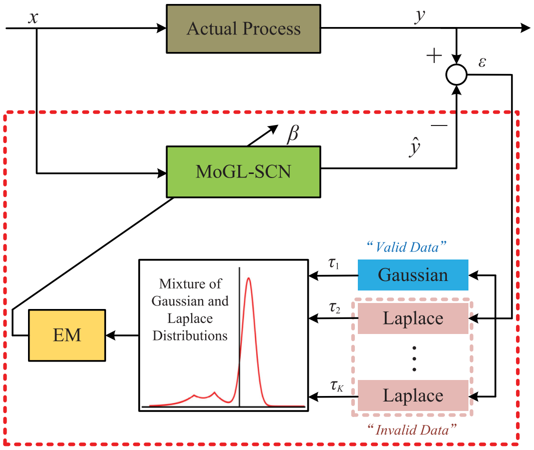


Fig. 1. Structure of the proposed method of constructing PIs.

Given a dataset $\boldsymbol{D} = \{\boldsymbol{X}, \boldsymbol{y}\} = \{(\boldsymbol{x}_n, y_n) \in R^d \times R\}_{n=1}^N$, for an SCN with P hidden nodes and the nth sample, according to Fig. 1, we can derive the following equation:

$$y_n = \hat{y}_n + \varepsilon_n = f_P(\boldsymbol{x}_n; \boldsymbol{\beta}) + \varepsilon_n = \boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta} + \varepsilon_n \tag{10}$$

where ε_n is the random noise and $\hat{y}_n$ is the predicted value. To improve the robustness of the SCN, the noise ε_n is assumed to follow a mixture of the Gaussian distribution $\mathcal{N}(\varepsilon|0,\sigma_{\mathcal{G}}^2)$ and the $K-1$ Laplace distributions $\mathcal{L}(\varepsilon|0,\sigma_{\mathcal{L};k})$ [with K ($K \geq 2$) components] with the appropriate mixing coefficients, namely

$$p(\varepsilon) = \tau_1\mathcal{N}(\varepsilon|0,\sigma_{\mathcal{G}}^2) + \sum_{k=2}^{K} \tau_k \mathcal{L}(\varepsilon|0,\sigma_{\mathcal{L};k}) \tag{11}$$

where $k = 1, \ldots K$, $\boldsymbol{\Gamma} = \{\tau_1, \tau_2, \ldots, \tau_K\}$ are the mixing coefficients, $\tau_k \geq 0$, $\sum_{k=1}^K \tau_k = 1$, and $\boldsymbol{\Sigma} = \{\sigma_{\mathcal{G}}^2, \sigma_{\mathcal{L};2}, \ldots, \sigma_{\mathcal{L};K}\}$.

And then, the PDF of the mixed distributions of y_n can be expressed as follows:

$$\begin{aligned}&p(y_n|\boldsymbol{x}_n,\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma})\\ &\quad= \tau_1\mathcal{N}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{G}}^2) + \sum_{k=2}^{K} \tau_k \mathcal{L}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{L};k}).\end{aligned} \tag{12}$$

The first term on the right side of (12) explains the normal data, which are called the "*valid data*." The second term is used to explain the data with unknown noise or outliers, which are called the "*invalid data*."

By introducing the variable $\boldsymbol{v} = \{v_n\}_{n=1}^N$ associated with the exponential distribution, one can obtain a new dataset $\boldsymbol{S} = \{\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{v}\} = \{\boldsymbol{x}_n, y_n, v_n\}_{n=1}^N$. Then, the joint PDF of y_n and v_n can be rewritten as follows:

$$\begin{aligned}&p(y_n, v_n|\boldsymbol{x}_n,\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma})\\ &\quad= \tau_1\mathcal{N}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{G}}^2) + \sum_{k=2}^{K} \tau_k \mathcal{L}(y_n, v_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{L};k})\end{aligned} \tag{13}$$

where $\mathcal{L}(y_n, v_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{L};k})$ can be computed by (9).

Assume that all samples are drawn independently, then the following likelihood function can be obtained:

$$\begin{aligned} p(\boldsymbol{y},\boldsymbol{v}|\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma}) &= \prod_{n=1}^{N} p(y_n,v_n|\boldsymbol{x}_n,\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma}) \\ &= \prod_{n=1}^{N} \{\tau_1\mathcal{N}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{G}}^2) \\ &\quad + \sum_{k=2}^{K}\tau_k\mathcal{L}(y_n,v_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{L};k})\}. \end{aligned} \tag{14}$$

In the parameter solution process, compared with the maximum likelihood estimation, the maximum a posterior (MAP) estimation can effectively avoid the singular problem [34]. Therefore, the MAP estimation is adopted to optimize the parameters. Generally, if there is little empirical knowledge about the output weights, then the prior of the output weights is assumed to follow a Gaussian distribution [34]. Then, the prior of output weights can be formulated as follows:

$$p(\boldsymbol{\beta}|\sigma_{\boldsymbol{\beta}}^2) = \frac{1}{(2\pi\sigma_{\boldsymbol{\beta}}^2)^{P/2}}\exp\left(-\frac{\|\boldsymbol{\beta}\|^2}{2\sigma_{\boldsymbol{\beta}}^2}\right). \tag{15}$$

According to Bayes' theorem, the posterior distribution of the output weights $\boldsymbol{\beta}$ of the MoGL-SCN can be expressed by the following formula:

$$p(\boldsymbol{\beta}|\boldsymbol{S},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2) \propto p(\boldsymbol{y},\boldsymbol{v}|\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma})\cdot p(\boldsymbol{\beta}|\sigma_{\boldsymbol{\beta}}^2) \tag{16}$$

Then, we take the logarithm of $p(\boldsymbol{\beta}|\boldsymbol{S},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2)$:

$$\begin{aligned} \ln p(\boldsymbol{\beta}|\boldsymbol{S},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2) &= \ln p(\boldsymbol{y},\boldsymbol{v}|\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma}) \\ &\quad + \ln p(\boldsymbol{\beta}|\sigma_{\boldsymbol{\beta}}^2) + c \end{aligned} \tag{17}$$

where c is a constant.

Therefore, the output weights $\boldsymbol{\beta}$ and the hyperparameters $\boldsymbol{\Gamma}$, $\boldsymbol{\Sigma}$, and $\sigma_{\boldsymbol{\beta}}^2$ can be obtained by maximizing $\ln p(\boldsymbol{\beta}|\boldsymbol{S},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2)$ in the MAP estimation

$$\{\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2\}^* = \underset{\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2}{\arg\max}\{\ln p(\boldsymbol{\beta}|\boldsymbol{S},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2)\}. \tag{18}$$

Nevertheless, due to the assumption on the mixed distributions, there are no analytical solutions to the aforementioned problem. The EM algorithm [35] can solve the optimization problem (18). To implement the EM algorithm, we should introduce the latent variable $\boldsymbol{z}_n = \{z_{kn}\}_{k=1}^K$, where $z_{kn} = 1$ if y_n is from the kth component, otherwise, $z_{kn} = 0$. Then, the prior distribution of $\boldsymbol{z}_n$ is written as follows:

$$p(\boldsymbol{z}_n) = \prod_{k=1}^{K}\tau_k^{z_{kn}}. \tag{19}$$

For the complete data $(\boldsymbol{x}_n, y_n, v_n, \boldsymbol{z}_n)$, the following joint PDF can be obtained:

$$\begin{aligned} &p(y_n,v_n,\boldsymbol{z}_n|\boldsymbol{x}_n,\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma}) \\ &\quad = p(y_n,v_n|\boldsymbol{x}_n,\boldsymbol{z}_n,\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma})p(\boldsymbol{z}_n) \\ &\quad = \left[\tau_1\mathcal{N}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{G}}^2)\right]^{z_{1n}} \\ &\qquad \cdot\prod_{k=2}^{K}\left[\tau_k\mathcal{L}(y_n,v_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{L};k})\right]^{z_{kn}}. \end{aligned} \tag{20}$$

Given a complete dataset $\boldsymbol{T} = \{\boldsymbol{X},\boldsymbol{y},\boldsymbol{v},\boldsymbol{Z}\}$, $\boldsymbol{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$, the likelihood function (14) can be reexpressed as follows:

$$\begin{aligned} p(\boldsymbol{y},\boldsymbol{v},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma}) &= \prod_{n=1}^{N} p(y_n,v_n,\boldsymbol{z}_n|\boldsymbol{x}_n,\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma}) \\ &= \prod_{n=1}^{N}\Big\{\left[\tau_1\mathcal{N}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{G}}^2)\right]^{z_{1n}} \\ &\quad \cdot\prod_{k=2}^{K}\left[\tau_k\mathcal{L}(y_n,v_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta},\sigma_{\mathcal{L};k})\right]^{z_{kn}}\Big\}. \end{aligned} \tag{21}$$

The logarithm of the posterior distribution of the complete dataset can be written as follows:

$$\begin{aligned} \ln p(\boldsymbol{\beta}|\boldsymbol{T},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2) &= \ln p(\boldsymbol{y},\boldsymbol{v},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma}) \\ &\quad + \ln p(\boldsymbol{\beta}|\sigma_{\boldsymbol{\beta}}^2) + c. \end{aligned} \tag{22}$$

Then, by combining (22) with (15) and (21), one can obtain the following expression:

$$\begin{aligned} \ln p(\boldsymbol{\beta}|\boldsymbol{T},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2) &= \sum_{n=1}^{N}\ln p(y_n,v_n,\boldsymbol{z}_n|\boldsymbol{x}_n,\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma}) \\ &\quad + \ln p(\boldsymbol{\beta}|\sigma_{\boldsymbol{\beta}}^2) + c \\ &= \sum_{n=1}^{N} z_{1n}\left[\ln\tau_1 - \frac{\ln\sigma_{\mathcal{G}}^2}{2} - \frac{(y_n - \boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta})^2}{2\sigma_{\mathcal{G}}^2}\right] \\ &\quad + \sum_{n=1}^{N}\sum_{k=2}^{K} z_{kn}\left(\ln\tau_k - \ln v_n^2 - \frac{\ln\sigma_{\mathcal{L};k}^2}{2}\right) \\ &\quad + \sum_{n=1}^{N}\sum_{k=2}^{K} z_{kn}\left[-\frac{v_n^2(y_n - \boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta})^2}{\sigma_{\mathcal{L};k}^2} - \frac{1}{2v_n^2}\right] \\ &\quad - \frac{P}{2}\ln\sigma_{\boldsymbol{\beta}}^2 - \frac{1}{2\sigma_{\boldsymbol{\beta}}^2}\|\boldsymbol{\beta}\|^2 + c. \end{aligned} \tag{23}$$

In the expectation step (E-step) of the EM algorithm, given the dataset $\boldsymbol{D}$, by taking the conditional expectation of the logarithm of the posterior distribution $\ln p(\boldsymbol{\beta}|\boldsymbol{T},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2)$ of the complete dataset and omitting the terms that are not associated with the parameters $\{\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2\}$, we can obtain the following formula:

$$\begin{aligned} &\mathbb{E}[\ln p(\boldsymbol{\beta}|\boldsymbol{T},\boldsymbol{\Gamma},\boldsymbol{\Sigma},\sigma_{\boldsymbol{\beta}}^2)|\boldsymbol{D}] \\ &\quad = \sum_{n=1}^{N}\mathbb{E}[z_{1n}|(\boldsymbol{x}_n,y_n)]\left(\ln\tau_1 - \frac{\ln\sigma_{\mathcal{G}}^2}{2}\right) \\ &\quad - \sum_{n=1}^{N}\mathbb{E}[z_{1n}|(\boldsymbol{x}_n,y_n)]\left[\frac{1}{2\sigma_{\mathcal{G}}^2}(y_n - \boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta})^2\right] \\ &\quad + \sum_{n=1}^{N}\sum_{k=2}^{K}\mathbb{E}[z_{kn}|(\boldsymbol{x}_n,y_n)]\left(\ln\tau_k - \frac{\ln\sigma_{\mathcal{L};k}^2}{2}\right) \\ &\quad - \sum_{n=1}^{N}\sum_{k=2}^{K}\Big\{\mathbb{E}[z_{kn}|(\boldsymbol{x}_n,y_n)] \end{aligned}$$

$$\cdot \frac{\mathbb{E}[v_n^2|(\boldsymbol{x}_n, y_n)](y_n - \boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta})^2}{\sigma_{\mathcal{L};k}^2}\Bigg\}$$
$$- \frac{P}{2}\ln\sigma_\beta^2 - \frac{1}{2\sigma_\beta^2}\|\boldsymbol{\beta}\|^2 + c_1 \qquad (24)$$

where $\mathbb{E}(\cdot)$ denotes the expectation operator and c_1 denotes a constant that is independent of parameters $\{\boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2\}$.

For $k = 1$, we can obtain the following relation:

$$\gamma_{1n} \triangleq \mathbb{E}[z_{1n}|(\boldsymbol{x}_n, y_n)]$$
$$= \frac{\tau_1\mathcal{N}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta}, \sigma_{\mathcal{G}}^2)}{\tau_1\mathcal{N}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta}, \sigma_{\mathcal{G}}^2) + \sum_{k=2}^K \tau_k\mathcal{L}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta}, \sigma_{\mathcal{L};k})}. \qquad (25)$$

For $k \geq 2$, we can obtain

$$\gamma_{kn} \triangleq \mathbb{E}[z_{kn}|(\boldsymbol{x}_n, y_n)]$$
$$= \frac{\tau_k\mathcal{L}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta}, \sigma_{\mathcal{L};k})}{\tau_1\mathcal{N}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta}, \sigma_{\mathcal{G}}^2) + \sum_{k=2}^K \tau_k\mathcal{L}(y_n|\boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta}, \sigma_{\mathcal{L};k})}. \qquad (26)$$

And $\mathbb{E}[v_n^2|(\boldsymbol{x}_n, y_n)]$ is calculated as follows:

$$\chi_{kn} \triangleq \mathbb{E}[v_n^2|(\boldsymbol{x}_n, y_n)] = \frac{\sigma_{\mathcal{L};k}}{\sqrt{2}\,|y_n - \boldsymbol{h}(\boldsymbol{x}_n)\boldsymbol{\beta}|}. \qquad (27)$$

More information concerning the calculation process of χ_{kn} can be found in [21].

Subsequently, in the maximization step (M-step) of the EM algorithm, we maximize $\mathbb{E}[\ln p(\boldsymbol{\beta}|\boldsymbol{T}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2)|\boldsymbol{D}]$ with respect to $\{\boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2\}$ as follows:

$$\{\boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2\}^* = \underset{\boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2}{\arg\max}\, \mathbb{E}[\ln p(\boldsymbol{\beta}|\boldsymbol{T}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2)|\boldsymbol{D}]. \qquad (28)$$

Let $\partial\mathbb{E}[\ln p(\boldsymbol{\beta}|\boldsymbol{T}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2)|\boldsymbol{D}]/\partial\{\boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2\} = 0$, we can obtain the following iterative formulas:

$$\tau_k^{(q+1)} = \frac{tr(\boldsymbol{\gamma}_{kn}^{(q)})}{N} \qquad (29)$$

where $\boldsymbol{\gamma}_{kn}^{(q)} = \mathrm{diag}\{\gamma_{kn}^{(q)}\}_{n=1}^N$, $\mathrm{tr}(\cdot)$ denotes the trace operator, and q denotes the iteration number of the EM algorithm.

$$\sigma_{\mathcal{G}}^{2\,(q+1)} = \frac{\left\|\boldsymbol{\theta}^{(q)} \cdot \left(\boldsymbol{y} - \boldsymbol{H}(\boldsymbol{X})\boldsymbol{\beta}^{(q)}\right)\right\|^2}{tr(\boldsymbol{\gamma}_{1n}^{(q)})} \qquad (30)$$

where $\boldsymbol{\theta}^{(q)} = [\theta_1^{(q)}, \ldots, \theta_n^{(q)}, \ldots, \theta_N^{(q)}]$ and $\theta_n^{(q)} = \sqrt{\gamma_{1n}^{(q)}}$.

$$\sigma_{\mathcal{L};k}^{2\,(q+1)} = \frac{\left\|\boldsymbol{v}_k^{(q)} \cdot \left(\boldsymbol{y} - \boldsymbol{H}(\boldsymbol{X})\boldsymbol{\beta}^{(q)}\right)\right\|^2}{tr\left(\boldsymbol{\gamma}_{kn}^{(q)}\right)} \qquad (31)$$

where $\boldsymbol{v}_k^{(q)} = [v_{1\,k}^{(q)}, \ldots, v_{Nk}^{(q)}]$ and $v_{nk}^{(q)} = \sqrt{2\sum_{n=1}^N \gamma_{kn}^{(q)}\chi_{kn}^{(q)}}$.

Then, we can obtain the estimation of $\sigma_{\mathcal{L};k}^{(q+1)}$ as

$$\sigma_{\mathcal{L};k}^{(q+1)} = \frac{\left|\boldsymbol{v}_k^{(q)} \cdot \left(\boldsymbol{y} - \boldsymbol{H}(\boldsymbol{X})\boldsymbol{\beta}^{(q)}\right)\right|}{\sqrt{tr\left(\boldsymbol{\gamma}_{kn}^{(q)}\right)}}. \qquad (32)$$

The estimation of σ_β^2 is computed as follows:

$$\sigma_\beta^{2\,(q+1)} = \frac{\left\|\boldsymbol{\beta}^{(q)}\right\|^2}{P}. \qquad (33)$$

The output weights of the MoGL-SCN can be calculated using the *iteratively reweighted regularized least-squares* method as

$$\boldsymbol{\beta}^{(q+1)} = \left[\boldsymbol{H}^T(\boldsymbol{X})\Psi^{(q+1)}\boldsymbol{H}(\boldsymbol{X}) + \sigma_{\mathcal{G}}^{2\,(q+1)}\boldsymbol{I}_P\right]^{-1}$$
$$\cdot \left[\boldsymbol{H}^T(\boldsymbol{X})\Psi^{(q+1)}\boldsymbol{y}\right] \qquad (34)$$

where $\boldsymbol{I}_P$ denotes an identity matrix with P dimensions and $\Psi^{(q+1)} = \mathrm{diag}\{\psi_n^{(q+1)}\}_{n=1}^N$ denotes the penalty weight matrix and its element $\psi_n^{(q+1)}$ is computed as follows:

$$\psi_n^{(q+1)} = \sigma_\beta^{2\,(q+1)}\gamma_{1n}^{(q+1)}$$
$$+ 2\sigma_{\mathcal{G}}^{2\,(q+1)}\sigma_\beta^{2(q+1)}\sum_{k=2}^K \frac{\chi_{kn}^{(q+1)}\gamma_{kn}^{(q+1)}}{\sigma_{\mathcal{L};k}^{2\,(q+1)}}. \qquad (35)$$

The training process of the proposed MoGL-SCN is summarized as follows. First, the initial hyperparameters are assigned, and the SCN is built using the SC-III algorithm [8] to obtain the random parameters and the initial output weights. Second, the hyperparameters and the output weights are iteratively reestimated using the EM algorithm. The termination condition is selected as follows:

$$\left|\frac{\mathbb{E}[\ln p(\boldsymbol{\beta}^{(q+1)}|\boldsymbol{T}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2)|\boldsymbol{D}]}{\mathbb{E}[\ln p(\boldsymbol{\beta}^{(q)}|\boldsymbol{T}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2)|\boldsymbol{D}]} - 1\right| < \kappa \qquad (36)$$

where κ equals a small positive number, which is set to $1e - 6$ in this article. Based on the aforementioned description, the implementation of the MoGL-SCN is summarized in Algorithm 1.

B. Construction of Prediction Intervals

The structure of PIs based on the proposed bootstrap ensemble MoGL-SCNs is shown in Fig. 2. First, M subdatasets $\boldsymbol{D}_m = \{(\boldsymbol{x}_{m,i}, y_{m,i})\}_{i=1}^N$, where $m = 1, \ldots, M$, are uniformly resampled from the original dataset $\boldsymbol{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$. Then, the point prediction value $\hat{y}$ and the variance $\sigma_{\hat{y}}^2$ associated with model mismatch are estimated by building M MoGL-SCNs using the M subdatasets.

According to Fig. 2, it can be seen that the point prediction value of the PI is estimated by the average of the prediction outputs of the M MoGL-SCNs as

$$\hat{y} = \frac{1}{M}\sum_{m=1}^M \hat{y}_m. \qquad (37)$$

As described in [28], the variance $\sigma_{\hat{y}}^2$ caused by the model mismatch can be computed by the variance of the prediction

Algorithm 1: MoGL-SCN.

```
Input: The dataset
$\boldsymbol{D} = \{\boldsymbol{X}, \boldsymbol{y}\} = \{(\boldsymbol{x}_n, y_n) \in R^d \times R\}_{n=1}^{N}$.
Output: $\boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}$ and $\sigma_\beta^2$.
 1:  Initialization: Set $P$ as the maximum number of
     hidden nodes of the SCN, $T_{\max}$ as the maximum
     configuration time, and $\boldsymbol{e}_0$ as the error tolerance. Select
     the scopes of input weights and biases of hidden nodes
     $\Upsilon = [-\lambda_j, \lambda_j]_{j=1}^{J}$. Set $K$ as the number of mixed
     components. Initialize the parameters $\{\boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2\}$.
 2:  Build the SCN using the SC-III proposed in [8].
 3:  Set the output weights obtained from step 1 as the
     initial output weights of the MoGL-SCN.
 4:  while termination condition (36) is not reached do
 5:    Calculate $\ln p(\boldsymbol{\beta}|\boldsymbol{T}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2)$ using (23).
 6:    E-step: calculate $\gamma_{1n}$, $\gamma_{kn}$ and $\chi_{kn}$ using (25)–(27).
 7:    M-step: update $\{\boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2\}$ using (29)–(35).
 8:    Renew the termination condition (36).
 9:  end while
10:  Obtain the optimal $\{\boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_\beta^2\}$ of the MoGL-SCN.
```

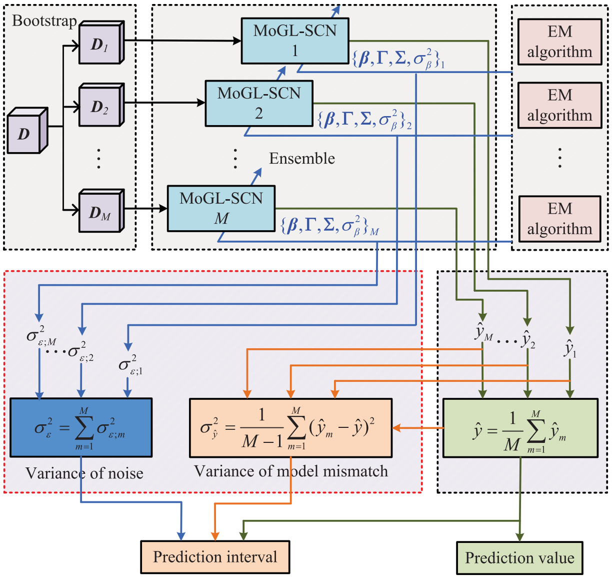

Bootstrap
$\boldsymbol{D}$
$\boldsymbol{D}_1$
$\boldsymbol{D}_2$
$\boldsymbol{D}_M$
MoGL-SCN 1
MoGL-SCN 2
MoGL-SCN M
Ensemble
$\{\boldsymbol{\beta}, \Gamma, \Sigma, \sigma_\beta^2\}_1$
$\{\boldsymbol{\beta}, \Gamma, \Sigma, \sigma_\beta^2\}_2$
$\{\boldsymbol{\beta}, \Gamma, \Sigma, \sigma_\beta^2\}_M$
EM algorithm
EM algorithm
EM algorithm
$\sigma_{\varepsilon;M}^2 \cdots \sigma_{\varepsilon;2}^2$
$\sigma_{\varepsilon;1}^2$
$\hat{y}_M \cdots \hat{y}_2$
$\hat{y}_1$
$\sigma_\varepsilon^2 = \sum_{m=1}^{M} \sigma_{\varepsilon;m}^2$
Variance of noise
$\sigma_{\hat{y}}^2 = \frac{1}{M-1} \sum_{m=1}^{M} (\hat{y}_m - \hat{y})^2$
Variance of model mismatch
$\hat{y} = \frac{1}{M} \sum_{m=1}^{M} \hat{y}_m$
Prediction interval
Prediction value

Fig. 2. Structure of the proposed BE-MoGL-SCN of constructing PIs.

outputs of the M MoGL-SCNs as

$$\sigma_{\hat{y}}^2 = \frac{1}{M-1} \sum_{m=1}^{M} (\hat{y}_m - \hat{y})^2. \tag{38}$$

In general, an extra $(M+1)$th neural network is usually built to model the variance of noise [28]. However, in the proposed method, the variance of the uncertainty caused by the intrinsic noise is estimated by the hyperparameter $\boldsymbol{\Sigma}$, namely, for $1 \leq m \leq M$, we can derive the following expression:

$$\sigma_{\varepsilon;m}^2 = \tau_{1,m}^2 \sigma_{\mathcal{G};m}^2 + \sum_{k=2}^{K} \tau_{k,m}^2 \sigma_{\mathcal{L};k,m}^2. \tag{39}$$

Algorithm 2: Construction of PIs Using BE-MoGL-SCN.

```
Input: The training dataset and the testing input data $\boldsymbol{x}$.
Output: $\hat{y}$, $L(\boldsymbol{x})$ and $U(\boldsymbol{x})$.
1:  Initialization: Set $M$ as the ensemble size and $CL$ as
    the predefined confidence level, and initialize the
    parameters in Algorithm 1.
2:  Generate $M$ subdatasets from the training dataset
    using the bootstrap method.
3:  Build $M$ base MoGL-SCNs based on Algorithm 1
    (step 2–step 9).
4:  Output $\boldsymbol{\beta}_1^*, \ldots, \boldsymbol{\beta}_M^*$ from the $M$ base MoGL-SCNs
    and the optimal hyperparameters.
5:  Input the testing data $\boldsymbol{x}$.
6:  Compute $\hat{y}$, $L(\boldsymbol{x})$ and $U(\boldsymbol{x})$ using (37)–(40).
```

Then, according to the definition in [28] and [32], after $\hat{y}$, $\sigma_{\hat{y}}^2$, and $\sigma_{\varepsilon;m}^2$ are obtained, the PI with confidence level (CL) $(1-\alpha)\%$ can be constructed as follows:

$$\begin{cases} L(\boldsymbol{x}) = \hat{y} - t_{1-\alpha/2}(M) \sqrt{\sigma_{\hat{y}}^2 + \sum_{m=1}^{M} \sigma_{\varepsilon;m}^2} \\ U(\boldsymbol{x}) = \hat{y} + t_{1-\alpha/2}(M) \sqrt{\sigma_{\hat{y}}^2 + \sum_{m=1}^{M} \sigma_{\varepsilon;m}^2} \end{cases} \tag{40}$$

where $t_{1-\alpha/2}(M)$ is the cumulative t-distribution with $(1-\alpha/2)$ quantiles and M degrees of freedom, and $L(\boldsymbol{x})$ denotes the lower bound and $U(\boldsymbol{x})$ denotes the upper bound of the constructed PI, respectively.

In accordance with the structure shown in Fig. 2 and the aforementioned analysis, the implementation step of the PIs constructed by the BE-MoGL-SCN is summarized in Algorithm 2.

IV. Case Studies on Benchmark Datasets

In this section, the effectiveness of the constructed PIs based on the BE-MoGL-SCN is evaluated on two benchmark datasets from KEEL:[1] Friedman (DB1) and Treasury (DB2).

Three other state-of-the-art robust randomized neural networks: M-RVFL [16], RR-RVFL [17], and RSC-KDE [18], are implemented in the bootstrap ensemble strategy to construct PIs, termed BE-RR-RVFL, BE-M-RVFL, and BE-RSC-KDE, respectively, and two novel ensemble neural network-based methods of constructing PIs: the negative correlation-learning-based ensemble RVFL (NCL-E-RVFL) [27] and the optimized bootstrap method (OPT-Bootstrap) [33], are compared with the proposed BE-MoGL-SCN.

All experiments are repeated 50 times and the average value of the 50 experiments is reported. The root-mean-squared error (RMSE) and Nash Sutcliffe coefficient (NSC) are adopted to evaluate the prediction accuracy of each method, and a large NSC indicates the high prediction accuracy. The prediction interval coverage probability (PICP) and normalized mean prediction

[1] [Online]. Available: http://www.keel.es/.

TABLE I
PERFORMANCE COMPARISON OF EACH METHOD ON NORMAL DATASET

Dateset	Method	RMSE	NSC	PICP	NMPIW
DB1	BE-MoGL-SCN	1.0920	0.9585	0.9117	0.1537
	BE-RSC-KDE	1.1063	0.9575	0.9125	0.1578
	BE-M-RVFL	1.1097	0.9572	0.9083	0.1551
	BE-RR-RVFL	1.1076	0.9573	0.9188	0.1543
	NCL-E-RVFL	1.0932	0.9584	0.9125	0.1595
	OPT-Bootstrap	1.0989	0.9581	0.9100	0.1547
DB2	BE-MoGL-SCN	0.2139	0.9960	0.9095	0.0329
	BE-RSC-KDE	0.2174	0.9958	0.9047	00350
	BE-M-RVFL	0.2287	0.9954	0.9133	0.0337
	BE-RR-RVFL	0.2220	0.9956	0.9190	0.0342
	NCL-E-RVFL	0.2300	0.9953	0.9228	0.0345
	OPT-Bootstrap	0.2342	0.9950	0.9038	0.0348

interval width (NMPIW) [28], [32] are introduced to evaluate the performance of the PIs. The PIs with high quality should have the large PICP and small NMPIW.

A. Parameter Setting

In this article, each benchmark dataset 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% are used as the testing data. The predefined confidence level is set to $CL = (1-\alpha) = 90\%$, and the ensemble size of all methods is set to $M = 30$. The number of hidden nodes of the base MoGL-SCN in the ensemble is set to $P = 40$ and $P = 60$ for DB1 and DB2, respectively. The input weights and biases are selected in range of $[-\lambda, \lambda]$, where $\lambda = 0.5$, 1, and 3, based on the supervisory mechanism (5). The maximum random configuration time is set to $T_{\max} = 200$. The number of components K in the mixed distributions and the initial values of hyperparameters $\{\mathbf{\Gamma}, \mathbf{\Sigma}, \sigma_\beta^2\}$ are set to $K = 3$ (a mixture of one Gaussian and two Laplace distributions) and $\tau_1 = 0.8$, $\{\tau_k\}_{k=2}^{k=3} = 0.1$, $\sigma_{\mathcal{G}}^2 = 0.15$, $\{\sigma_{\mathcal{L};k}\}_{k=2}^{k=3} = 0.1$, and $\sigma_\beta^2 = 0.15$. All the aforementioned parameters are determined by the results on the validation dataset.

B. Comparative Experiments

Table I gives the prediction performance of all the methods on DB1 and DB2. The comparisons in Table I indicate that BE-MoGL-SCN has the smallest RMSE and largest NSC, which suggests that the prediction accuracy of the BE-MoGL-SCN is better than that of the other five methods on both DB1 and DB2 without adding noise. The PICP of the BE-MoGL-SCN is relatively small but still larger than the predefined $CL = 90\%$, and the interval width is narrower than those of the other five methods on both DB1 and DB2. These results indicate that the PIs of the BE-MoGL-SCN are narrow but appropriate and can maintain an acceptable coverage probability.

To demonstrate the robustness of the proposed method, we randomly select $\xi\%$, where $\xi = \{10, 15, 20, 25, 30\}$, of the complete training dataset and add sparse random noise that is produced as $y \times \text{rand}(0,1) \times [-50\%, 50\%]$, where rand(0, 1) denotes a uniformly distributed number in (0,1). Figs. 3 and 4 illustrate the variations in the average values and standard deviations of the RMSEs, NSCs, PICPs, and NMPIWs with different ξ of each method on DB1 and DB2. As shown in Fig. 3, the RMSE of the BE-MoGL-SCN is smaller than that of the other five methods with respect to different ξ on DB1, and the NSC is the largest among those of the six methods. This finding demonstrates that BE-MoGL-SCN has better generalization capability than the other models. Compared with the other five methods, the BE-MoGL-SCN can maintain an acceptable PICP and small NMPIW with increasing noise contamination rate ξ. Therefore, the BE-MoGL-SCN can construct PIs with higher quality than

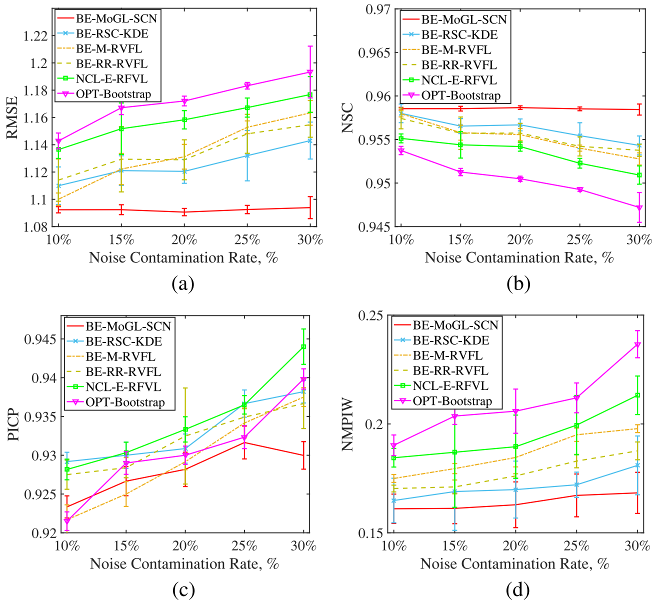


Fig. 3. Prediction performance of each method with different ξ on DB1. (a) RMSE. (b) NSC. (c) PICP. (d) NMPIW.

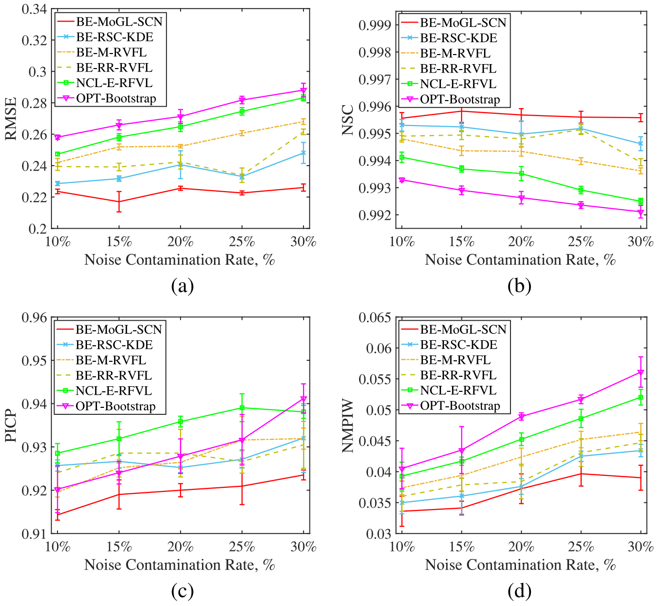


Fig. 4. Prediction performance of each method with different ξ on DB2. (a) RMSE. (b) NSC. (c) PICP. (d) NMPIW.

those of the other five methods. For DB2, from the comparisons in Fig. 4, we can observe that BE-MoGL-SCN outperforms the other five methods in terms of the prediction accuracy as ξ increases and yields a reasonable PICP and small NMPIW. Compared with those of the other five methods, the PIs of the BE-MoGL-SCN are more effective. Moreover, we can see from Figs. 3 and 4 that the interval widths (NMPIWs) of the five comparative methods are larger than that of the proposed method. Therefore, the cases that the PICPs of the proposed method are smaller than that of some other comparative methods can occur. The experimental results on the two benchmark datasets demonstrate the advantage of the proposed BE-MoGL-SCN.

Remark 2: There is a direct relationship between the interval width and the coverage probability of the PIs. In general, a large NMPIW will lead to a high PICP, but the PIs with extremely large interval widths convey no information about the actual targets [9]. Hence, in the real-world applications, the optimal PIs should have small NMPIW, and the PICP should not be less than the predefined confidence level [28].

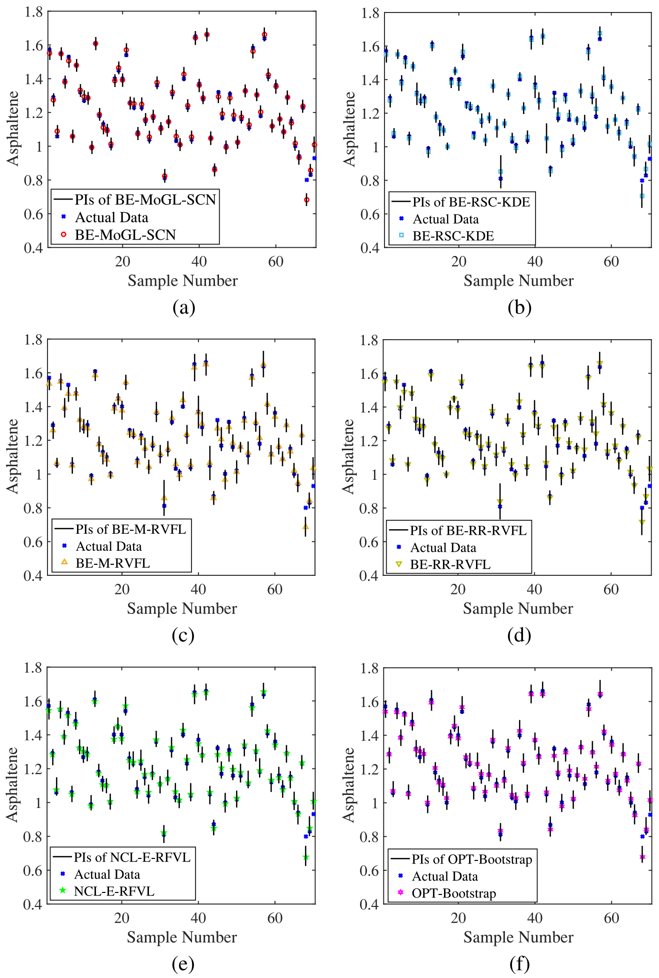
V. Prediction of Asphaltene in Crude Oil

In this section, a real-world dataset collected from a refinery is used to verify the performance of the proposed method.

The real-world dataset was collected from the fast evaluation system for the physicochemical properties of crude oil in a refinery in China. The input features are nuclear magnetic resonance (NMR) hydrogen spectrum data $\boldsymbol{x} \in R^{700}$. As an important physicochemical property of crude oil, the asphaltene consists of highly concentrated poly aromatics, and these components often result in the blockage and corrosion of pipelines and equipments, which can lead to a significant decrease in production profits. Hence, the fast evaluation of asphaltene in crude oil is of great significance for increasing the economic benefits of refineries. Therefore, we select the asphaltene in crude oil as the modeling output. The dataset consists of 863 sets of NMR hydrogen spectrum data and the corresponding asphaltene content data collected between May 2016 and October 2017. However, the high dimensionality of NMR spectrum data will lead to the high computational cost, which severely affects the real-time application of the method. Principal component analysis (PCA) can effectively extract the features of the NMR spectra [36]. Therefore, PCA is adopted to perform the dimensionality reduction. First, the NMR spectrum data are normalized. Then, by using PCA, the principal components with a 99% cumulative percent variance contribution rate are chosen as the inputs.

A. Parameter Selection

In this experiment, the dataset is divided into three parts: the training dataset (743 groups), the validation dataset (50 groups), and the testing dataset (70 groups). The predefined CL is set to 95%, the bootstrap ensemble size of all methods is set to $M = 25$, and the number of hidden nodes of the base MoGL-SCN in the ensemble is set to $P = 60$. The random parameters of the MoGL-SCN are automatically assigned in the range of $[-\lambda, \lambda]$, where $\lambda = 0.2, 0.5, 1, 3$, and 5. The random configuration time is set to 200. The number of mixed components is set to $K = 4$, with one Gaussian and three Laplace distributions. The initial values of the hyperparameters $\{\boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \sigma_{\beta}^2\}$ are set to $\tau_1 = 0.7$, $\{\tau_k\}_{k=2}^{k=4} = 0.1$, $\sigma_{\mathcal{G}}^2 = 0.2$, $\{\sigma_{\mathcal{L};k}\}_{k=2}^{k=4} = 0.1$, and $\sigma_{\beta}^2 = 0.15$, respectively. All the aforementioned parameters are determined from results on validation dataset.



(a) (b) (c) (d) (e) (f)

Fig. 5. PIs and point prediction of each method on the normal dataset. (a) BE-MoGL-SCN. (b) BE-RSC-KDE. (c) BE-M-RVFL. (d) BE-RR-RVFL. (e) NCL-E-RFVL. (f) OPT-Bootstrap.

B. Comparison and Discussion

The constructed PIs and the point prediction results of the BE-MoGL-SCN and the other five comparative methods on the normal dataset are shown in Fig. 5. As shown in Fig. 5, one can see that the point prediction outputs of the BE-MoGL-SCN can fit the actual data better than can the outputs of the other five models, hence, the proposed method has small prediction errors, high prediction accuracy, and satisfactory generalization capability on the normal dataset. Furthermore, the constructed PIs of the BE-MoGL-SCN have a small interval width and acceptable PICP, which is larger than the predefined confidence level (95%), so the constructed PIs are suitable for the decision-making processes in crude oil refining.

To better illustrate the superiority of the BE-MoGL-SCN, the scatter diagram of the point prediction results and the PDF

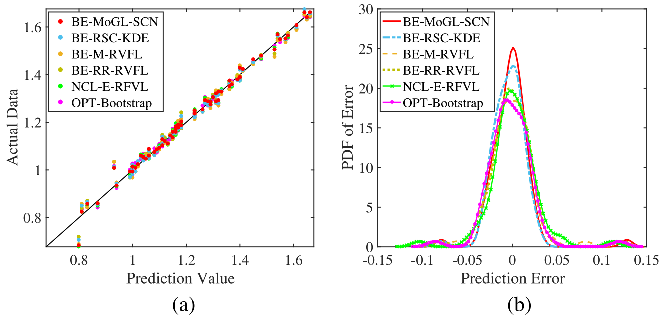


Fig. 6. Point prediction results of each method. (a) Scatter diagram of the point prediction. (b) PDF of the prediction errors.

TABLE II
PERFORMANCE COMPARISON ON NORMAL DATASET

Method	RMSE	NSC	PICP	NMPIW
BE-MoGL-SCN	0.02250	0.98842	0.97143	0.08534
BE-RSC-KDE	0.02329	0.98753	0.95714	0.10109
BE-M-RVFL	0.02407	0.98676	0.97572	0.11412
BE-RR-RVFL	0.02394	0.98688	0.96325	0.11873
NCL-E-RFVL	0.02425	0.98658	0.98284	0.11352
OPT-Bootstrap	0.02467	0.98607	0.98571	0.11445

of the prediction errors of each method on the normal dataset are shown in Fig. 6, and the average values of the RMSEs, NSCs, PICPs, and NMPIWs of the BE-MoGL-SCN and the other five methods on the normal dataset are listed in Table II. As shown in Fig. 6(a), we can see that compared with the other five comparative algorithms, the point prediction results of the BE-MoGL-SCN are much closer to the actual targets. According to Fig. 6(b), one can see that the PDF of prediction errors of the BE-MoGL-SCN emerges a narrower spiking shape around zero. This indicates that from the perspective of probability, the mean value of the prediction errors of the BE-MoGL-SCN is zero. It can also be seen from Table II that the BE-MoGL-SCN has the smallest RMSE and the largest NSC. It can be concluded that compared with the other methods, the proposed BE-MoGL-SCN yields better prediction accuracy. And it also shows that the PICP of the BE-MoGL-SCN is smaller than that of the BE-M-RVFL, NCL-E-RFVL, and OPT-Bootstrap but larger than the predefined confidence level (95%), and the NMPIW is narrower than that of the other five methods, confirming that the PIs constructed by the BE-MoGL-SCN can reflect the important information associated with the actual targets.

The computational efficiency including the averages and standard deviations of the training time and testing time of each method is given in Table III. It is shown in Table III that the incremental approach for building SCNs and the stochastic configuration process of random parameters slow the training process of SCNs, so the training times of the BE-MoGL-SCN and BE-RSC-KDE are longer than those of the BE-RR-RFVL, BE-M-RFVL, and NCL-E-RFVL. And the evolutionary optimization algorithm is implemented in the OPT-Bootstrap method, which results in the most expensive computational cost. The testing time of the BE-MoGL-SCN is slightly longer than that of the other five methods but still acceptable.

TABLE III
COMPUTATIONAL EFFICIENCY COMPARISON OF EACH METHOD

Method	Training time (s)	Testing time (s)
BE-MoGL-SCN	5.5880 ± 0.5078	0.1129 ± 0.0269
BE-RSC-KDE	6.2030 ± 0.8929	0.0275 ± 0.0214
BE-M-RVFL	4.4191 ± 0.3722	0.0540 ± 0.0135
BE-RR-RVFL	4.7590 ± 0.1456	0.0755 ± 0.0151
NCL-E-RFVL	4.7548 ± 1.0957	0.0677 ± 0.0132
OPT-Bootstrap	209.1687 ± 11.2485	0.0965 ± 0.0235

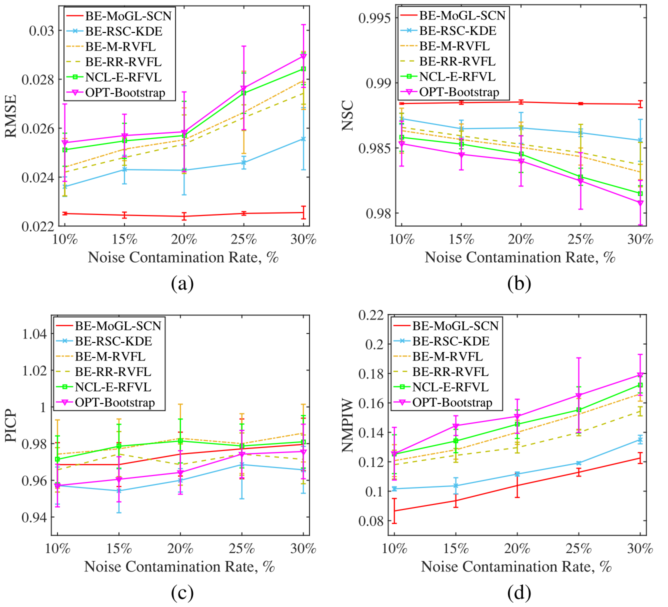


Fig. 7. Prediction performance of each method with different ξ. (a) RMSE. (b) NSC. (c) PICP. (d) NMPIW.

To evaluate the robustness of the proposed BE-MoGL-SCN with respect to different noise contamination rates, sparse random noise, which is generated in a manner similar to that in the previous experiment in Section IV, is introduced into the training data. The variations in the RMSEs, NSCs, PICPs, and NMPIWs of each method with respect to different ξ ($\xi = 10, 15, 20, 25, 30$) are depicted in Fig. 7. According to the comparisons of the RMSE and NSC shown in Fig. 7, as ξ increases, the RMSE of the BE-MoGL-SCN slightly increases and the NSC slightly decreases, so the BE-MoGL-SCN can maintain a high prediction accuracy, suggesting that BE-MoGL-SCN is minimally affected by noise. The prediction accuracy of the other five methods rapidly decreases in comparison. Moreover, as ξ increases, the PICPs of all methods are still larger than $CL = 95\%$. The PICP of the BE-MoGL-SCN is significantly larger than that of BE-RSC-KDE and OPT-Bootstrap. And the NMPIW of the BE-MoGL-SCN is smaller than that of the other five comparative methods. Therefore, the BE-MoGL-SCN is superior to the other five methods in terms of robustness with respect to different noise contamination rates.

In reality, for the real-world applications, we usually care about the worst-case performance instead of the statistical average. Therefore, we report the corresponding worst results of the 50 experiments of each method with respect to different noise

TABLE IV
WORST-CASE PERFORMANCE OF EACH METHOD WITH DIFFERENT ξ

ξ	Method	RMSE	NSC	PICP	NMPIW
0	BE-MoGL-SCN	0.0228	0.9882	0.9571	0.0961
	BE-RSC-KDE	0.0243	0.9865	0.9429	0.1189
	BE-M-RVFL	0.0264	0.9841	0.9571	0.1215
	BE-RR-RVFL	0.0258	0.9848	0.9571	0.1219
	NCL-E-RFVL	0.0260	0.9845	0.9571	0.1221
	OPT-Bootstrap	0.0265	0.9840	0.9571	0.1208
10	BE-MoGL-SCN	0.0229	0.9881	0.9571	0.0971
	BE-RSC-KDE	0.0239	0.9868	0.9429	0.1033
	BE-M-RVFL	0.0258	0.9848	0.9571	0.1259
	BE-RR-RVFL	0.0253	0.9854	0.9571	0.1276
	NCL-E-RFVL	0.0257	0.9847	0.9571	0.1323
	OPT-Bootstrap	0.0255	0.9849	0.9571	0.1376
15	BE-MoGL-SCN	0.0227	0.9883	0.9571	0.0990
	BE-RSC-KDE	0.0249	0.9858	0.9429	0.1126
	BE-M-RVFL	0.0264	0.9841	0.9571	0.1347
	BE-RR-RVFL	0.0251	0.9856	0.9571	0.1298
	NCL-E-RFVL	0.0267	0.9839	0.9571	0.1389
	OPT-Bootstrap	0.0252	0.9855	0.9571	0.1425
20	BE-MoGL-SCN	0.0229	0.9881	0.9571	0.1182
	BE-RSC-KDE	0.0259	0.9846	0.9571	0.1137
	BE-M-RVFL	0.0281	0.9819	0.9571	0.1492
	BE-RR-RVFL	0.0272	0.9830	0.9571	0.1332
	NCL-E-RFVL	0.0284	0.9812	0.9571	0.1342
	OPT-Bootstrap	0.0276	0.9828	0.9571	0.1487
25	BE-MoGL-SCN	0.0231	0.9879	0.9571	0.1172
	BE-RSC-KDE	0.0249	0.9858	0.9571	0.1201
	BE-M-RVFL	0.0297	0.9798	0.9571	0.1579
	BE-RR-RVFL	0.0282	0.9818	0.9714	0.1425
	NCL-E-RFVL	0.0299	0.9795	0.9714	0.1533
	OPT-Bootstrap	0.0289	0.9810	0.9571	0.1667
30	BE-MoGL-SCN	0.0233	0.9878	0.9714	0.1284
	BE-RSC-KDE	0.0293	0.9804	0.9571	0.1401
	BE-M-RVFL	0.0306	0.9785	0.9714	0.1731
	BE-RR-RVFL	0.0292	0.9805	0.9571	0.1568
	NCL-E-RFVL	0.0293	0.9803	0.9571	0.1773
	OPT-Bootstrap	0.0311	0.9778	0.9714	0.1928

contamination rate (ξ) in Table IV. As given in Table IV, on both the normal dataset ($\xi = 0$) and the dataset with introduced noise, the worst prediction accuracy of the BE-MoGL-SCN is better than that of the other five methods, and the interval width of the BE-MoGL-SCN is narrower than that of the other five methods, except in the case of $\xi = 20$. Besides, the coverage probability of the BE-MoGL-SCN is greater than that of the other five methods, except in the case of $\xi = 25$. Therefore, we can conclude that the proposed method can construct PIs with high quality and effectively eliminate the effect of noise or outliers on the modeling performance.

VI. CONCLUSION

This article presented a novel robust SCN based on a mixture of the Gaussian and Laplace distributions to solve the low prediction accuracy problem associated with the presence of noise or outliers with unknown distributions in the real-world data. Moreover, the parameter solution process based on the EM algorithm of the proposed robust SCN was derived. Furthermore, the bootstrap ensemble strategy was adopted to construct the PIs and quantify the uncertainties caused by both model mismatch and noise in the real-world data, and the proposed robust SCN was applied as the base component in the ensemble. The proposed method of constructing PIs was tested on two benchmark datasets and a real-world dataset collected from a refinery. Compared with other methods, the proposed method could construct PIs with higher reliability and prediction accuracy and could also guarantee high computational efficiency. In additional, the experimental results demonstrated that the proposed method exhibits excellent robustness with respect to different noise contamination rates. The experimental results using the real-world dataset suggested that the proposed method was suitable for applications in the refinery.

It was worth noting that the missing data phenomenon was also a common issue that was frequently present in the real-world applications, and the large number of irregularly missing data could result in biased estimation of model parameters that may cause the uncertainty of the prediction result [2]. In this article, we mainly considered the uncertainty related to the prediction results of the model when dealing with the real-world data contaminated with noise or outliers. The issue of missing data was not taken into account in the proposed method. But, in the future, one can attempt to address the issue of missing data by applying the proposed method of constructing PIs with semisupervised techniques and imputation-based methods.

REFERENCES

[1] P. Kadlec, R. Grbić, and B. Gabrys, "Review of adaptation mechanisms for data-driven soft sensors," *Comput. Chem. Eng.*, vol. 35, no. 1, pp. 1–24, Jan. 2011.

[2] J. Zhu, Z. Ge, Z. Song, and F. Gao, "Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data," *Annu. Rev. Control*, vol. 46, pp. 107–133, Oct. 2018.

[3] X. Yuan, Y. Wang, C. Yang, Z. Ge, Z. Song, and W. Gui, "Weighted linear dynamic system for feature representation and soft sensor application in nonlinear dynamic industrial processes," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1508–1517, Feb. 2018.

[4] 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.

[5] 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.

[6] W. Cui *et al.*, "Received signal strength based indoor positioning using a random vector functional link network," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 1846–1855, May 2018.

[7] M. Li and D. Wang, "Insights into randomized algorithms for neural networks: Practical issues and common pitfalls," *Inf. Sci.*, vol. 382–383, pp. 170–178, Mar. 2017.

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

[9] 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.

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

[11] P. Zhou, D. Guo, H. Wang, and T. Chai, "Data-driven robust M-LS-SVR-based NARX modeling for estimation and control of molten iron quality indices in blast furnace ironmaking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4007–4021, Sep. 2018.

[12] L. Huang, H. Wang, and A. Zheng, "The M-estimator for functional linear regression model," *Statist. Probab. Lett.*, vol. 88, pp. 165–173, May. 2014.

[13] C. Chuang, S. Su, and C. Hsiao, "The annealing robust backpropagation (ARBP) learning algorithm," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1067–1077, Sep. 2000.

[14] E. López-Rubio, E. J. Palomo, and E. Domínguez, "Robust self-organization with M-estimators," *Neurocomputing*, vol. 151, pp. 408–423, Mar. 2015.

[15] W. Cao, X. Wang, Z. Ming, and J. Gao, "A review on neural networks with random weights," *Neurocomputing*, vol. 275, pp. 278–287, Jan. 2018.

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

[17] W. Dai, Q. Chen, F. Chu, X. Ma, and T. Chai, "Robust regularized random vector functional link network and its industrial application," *IEEE Access*, vol. 5, pp. 16 162–16 172, Aug. 2017.

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

[19] T. Auld, A. W. Moore, and S. F. Gull, "Bayesian neural networks for internet traffic classification," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 223–239, Jan. 2007.

[20] 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.

[21] R. F. Phillips, "Least absolute deviations estimation via the EM algorithm," *Statist. Comput.*, vol. 12, no. 3, pp. 281–285, Jul. 2002.

[22] W. Song, W. Yao, and Y. Xing, "Robust mixture regression model fitting by Laplace distribution," *Comput. Statist. Data Anal.*, vol. 71, pp. 128–137, Mar. 2014.

[23] D. Li, M. Han, and J. Wang, "Chaotic time series prediction based on a novel robust echo state network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 787–799, May 2012.

[24] F. Cao, H. Ye, and D. Wang, "A probabilistic learning algorithm for robust modeling using neural networks with random weights," *Inf. Sci.*, vol. 313, pp. 62–78, Aug. 2015.

[25] D. Meng and F. De La Torre, "Robust matrix factorization with unknown noise," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1337–1344.

[26] M. A. Hosen, A. Khosravi, S. Nahavandi, and D. Creighton, "Improving the quality of prediction intervals through optimal aggregation," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4420–4429, Jul. 2015.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] Z. Lv, J. Zhao, Y. Liu, and W. Wang, "Use of a quantile regression based echo state network ensemble for construction of prediction intervals of gas flow in a blast furnace," *Control Eng. Pract.*, vol. 46, pp. 94–104, Jan. 2016.

[32] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications," *IEEE Access*, vol. 6, pp. 36 218–36 234, Jul. 2018.

[33] 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.

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

[35] 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.

[36] 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, Sep. 2012.

Jun Lu received the B.S. degree from the Shenyang University of Technology, Shenyang, China, in 2012, and the M.S. degree in 2014 from Northeastern University, Shenyang, where since 2016, he has been working toward the Ph.D. degree in control theory and engineering with the State Key Laboratory of Synthetical Automation for Process Industries.

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

Jinliang Ding (M'09–SM'14) 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 more than 100 refereed journal papers and refereed papers at international conferences. He is also the inventor or coinventor 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 application.

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