Article:
Zhang, J, Li, Y, Xiao, W et al. (1 more author) (2020) Robust Extreme Learning Machine for Modeling with Unknown Noise. Journal of the Franklin Institute, 357 (14). pp. 9885-9908. ISSN 0016-0032

https://doi.org/10.1016/j.jfranklin.2020.06.027

Robust Extreme Learning Machine for Modeling with Unknown Noise

Jie Zhang[a], Yanjiao Li[b], Wendong Xiao[c,*], Zhiqiang Zhang[d]

[a]School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
[b]School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China
[c]School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
[d]School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, UK

Abstract

Extreme learning machine (ELM) is an emerging machine learning technique for training single hidden layer feedforward networks (SLFNs). During the training phase, ELM model can be created by simultaneously minimizing the modeling errors and norm of the output weights. Usually, squared loss is widely utilized in the objective function of ELMs, which is theoretically optimal for the Gaussian error distribution. However, in practice, data collected from uncertain and heterogeneous environments trivially result in unknown noise, which may be very complex and cannot be described well using any single distribution. In order to tackle this issue, in this paper, a robust ELM (R-ELM) is proposed for improving the modeling capability and robustness with Gaussian and non-Gaussian noise. In R-ELM, a modified objective function is constructed to fit the noise using mixture of Gaussian (MoG) to approximate any continuous distribution. In addition, the corresponding solution for the new objective function is developed based on expectation maximization (EM) algorithm. Comprehensive experiments, both on selected benchmark datasets and real world applications, demonstrate that the proposed R-ELM has better robustness and generalization performance than state-of-the-art machine learning approaches.

Keywords: Extreme learning machine, non-Gaussian noise, mixture of Gaussian, expectation maximization algorithm

1. Introduction

In the past decade, extreme learning machine (ELM), as a type of generalized single hidden layer feedforward networks (SLFNs), has been intensively studied both in theory and applications [1]. Unlike the traditional gradient based training approaches for SLFNs, which are easy to trap in the local minimum and time-consuming, the hidden layer parameters of ELM are assigned randomly without iterative tuning, and then it only needs to solve the least-square problem [2, 3]. Accordingly, ELM has much faster learning speed and is easier to implement than state-of-the-art machine learning approaches. Theoretically, Huang et al. [4] have proven the universal approximation of ELM. In addition, ELM has been extended to online learning [5, 6], structure optimization [7, 8], ensemble learning [9, 10], imbalance learning [11, 12], representation learning [13, 14, 15], as well as residual learning [16, 17], etc. For real world applications, ELM has been implemented to landmark recognition [18, 19], industrial production [20, 21], and wireless localization [22, 23], etc.

As mentioned above, ELM is becoming an increasingly significant research topic in the machine learning field, but majority of the existed ELMs assume that the data utilized for modeling are pure without noise and outliers, or with Gaussian error distribution. However, data uncertainty is inevitable in practical scenarios due to sampling errors, measurement errors, and modeling errors, etc., which may lead to noise subject to unknown distributions. It means that noise of the real world applications should be more complex, which may follow Gaussian distribution, Laplace distribution, or mixed distributions. In addition, the performance of the data-driven predictor will degrade seriously if

*Corresponding author
Email address: wdxiao@ustb.edu.cn (Wendong Xiao)

the data are chaotic or too noisy. Therefore, ELMs without considering the effects of uncertainties may be not sufficient. There are usually two ways for strengthening the modeling capability of ELM in uncertain scenarios, including outlier detection and removing, and modifying objective function. For example, FIR-ELM was proposed to reduce the input disturbance by removing some undesired signal components through the FIR filtering [24]. He et al. [25] designed a hierarchical ELM to deal with high-dimensional noisy data, in which some groups of subnets were proposed for simultaneously reducing data dimension and filtering noise. However, the aforementioned outlier detection based ELMs may identify pure data as outliers, which easily break the original data structure and cause information loss. Another set of solutions is to enhance the robustness of the data-driven predictor by modifying the objective function of ELM. Specifically, second order cone programming, widely utilized in robust convex optimization problems, was introduced into ELM, but the computational burden of the novel ELM was relatively heavy [26]. Lu et al. [27] rewrote the objective function of ELM and proposed a probabilistic regularized ELM (PR-ELM) by incorporating the distribution information of modeling error into the modeling process, in which both the modeling error mean and the modeling variance were included in the modified objective function. The experimental results indicated that the proposed PR-ELM had a well-fitting performance and was more robust to noise. Although, ELMs with modified objective functions can achieve satisfactory performance in several tasks, the squared loss is still utilized in most of them, which may not guarantee that those ELMs can achieve the optimal solutions if the noise follows non-Gaussian distribution.

In order to tackle this issue, in this paper, a robust ELM (R-ELM) is proposed for improving the modeling capability and robustness of ELM in dealing with tasks with Gaussian and non-Gaussian noise. Different from the existed ELMs, which minimize the output weights and modeling errors with the assumption that noise follows Gaussian distribution, a new objective function of R-ELM is constructed, in which the characteristic of noise is described using mixture of Gaussian (MoG) for approximating the feature mapping between the inputs and the outputs. In addition, expectation maximization (EM) algorithm is implemented for estimating the parameters in the proposed R-ELM. The main contributions can be summarized as the following aspects:

1) A robust objective function is developed based on MoG for enhancing the modeling capability with complex and unknown noise. Specifically, the squared loss of the modeling errors in the original objective function of ELM is replaced by MoG. Thus, the modified objective function enables R-ELM to be more robust due to the excellent capability of MoG for approximating any continuous noise distribution.

2) Considering the analytical solutions of the parameters in the modified objective function of R-ELM cannot be calculated directly, EM algorithm is implemented to help obtain the optimal parameters.

3) Comprehensive experiments have been conducted, the corresponding experimental results indicate that R-ELM outperforms state-of-the-art machine learning approaches on both selected benchmark datasets and real world applications.

The paper is organized as follows: Section 2 presents the ELM theory. The details of the proposed R-ELM are shown in Section 3, including the limitations of modeling with Gaussian noise, motivations of improving the modeling capability of ELM with unknown noise, modified objective function of R-ELM, and the corresponding solving process. Experimental results and further analysis on selected benchmark datasets are reported in Section 4, followed by the performance verification on two real world applications in Section 5. Finally, discussions, and conclusions and future works are respectively given in Section 6 and Section 7.

2. ELM Theory

In this section, a brief introduction of ELM theory is firstly given to facilitate the understanding of the following sections.

ELM was proposed for training the SLFNs with a three-layer structure, including: input layer, hidden layer, and output layer (see Fig. 1). Different from state-of-the-art machine learning approaches, its hidden layer parameters are generated randomly without iterative tuning, reducing the learning problem to that of estimating the optimal output weights $\boldsymbol{\beta}$ for a given dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\tilde{N}} \subset R^n \times R^m$.

Accordingly, ELM can be treated as a linear combination of L activation functions:

$$f_{ELM}(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{1}$$

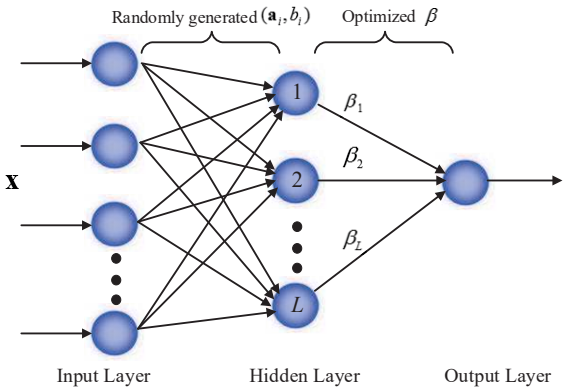


Figure 1: Basic structure of ELM

where L denotes the number of hidden nodes, and $\mathbf{h}(\mathbf{x})$ denotes the mapped feature vector, respectively.

The above equation can be rewritten in compact matrix form as

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

where $\mathbf{H}$ refers to the hidden layer output matrix:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_{\tilde{N}}) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_{\tilde{N}}) & \cdots & h_L(\mathbf{x}_{\tilde{N}}) \end{bmatrix} \tag{3}$$

and $\mathbf{Y}$ refers to the training data target matrix:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_{\tilde{N}}^T \end{bmatrix} = \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & \ddots & \vdots \\ y_{\tilde{N}1} & \cdots & y_{\tilde{N}m} \end{bmatrix} \tag{4}$$

The objective function of ELM is to simultaneously minimize the norm of the output weights and the modeling error:

$$\begin{aligned} &Min: \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C\sum_{i=1}^{\tilde{N}}\boldsymbol{\xi}_i^2 \\ &s.t., \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{y}_i - \boldsymbol{\xi}_i, i = 1, \ldots, \tilde{N} \end{aligned} \tag{5}$$

where C is a regularization coefficient for strengthening the generalization performance, and $\boldsymbol{\xi}_i$ represents the model noise, respectively.

According to the Karush-Kuhn-Tucker (KKT) theorem, we can estimate the output weights by

$$\tilde{\boldsymbol{\beta}} = \begin{cases} \mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{Y}, & \tilde{N} < L \\ \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y}, & \tilde{N} > L \end{cases} \tag{6}$$

where $\tilde{\boldsymbol{\beta}}$ represents the estimated value of $\boldsymbol{\beta}$, and $\mathbf{I}$ represents the unit matrix.

In summary, ELM can be described by Algorithm 1.

Algorithm 1 ELM

Input: Given a training dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\tilde{N}} \subset R^n \times R^m$, an activation function $g(\cdot)$, and hidden nodes number L.

1: Randomly assign hidden layer parameters $\mathbf{a}_i$ and b_i, $i = 1, \ldots, L$;
2: Calculate the hidden layer output matrix $\mathbf{H}$;
3: Calculate the estimated output weights $\tilde{\boldsymbol{\beta}}$ using (6).

3. Robust Modeling with Unknown Noise

In this section, the details of the proposed R-ELM will be given, including limitations of modeling with Gaussian noise, motivation of improving the modeling capability of ELM with unknown noise, objective function of R-ELM, and the corresponding solving process.

3.1. Limitations of Modeling with Gaussian Noise

For a given learning problem, our goal is to create a predictor which can map the input to the output well. Data-driven predictor is usually constructed depending on a corresponding training dataset, but based on the assumptions that the input data are not corrupted with noise, and errors are only confined to the output, or even there is noise, its effect is usually does not fully considered in the learning formulation, such as the objective function of ELM in (5). However, in practice, the above assumptions cannot hold, because sampling errors, measurement errors, and modeling errors result in the complex and unknown noise, and may seriously degrade the performance of the created predictor. Some robust predictors were proposed based on the assumption that the noise follows Gaussian distribution, but it also may follow other unknown distributions, such as Laplace distribution, Beta distribution, and even mixture of several kinds of distributions. Thus, the above assumptions are not reliable in such cases [28, 29].

Considering an arbitrary target function, when we build a predictor for it, we have

$$\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\xi} \tag{7}$$

where $f(\cdot)$ is the created model.

Thus, (2) should be modified as

$$\mathbf{H}\boldsymbol{\beta} + \boldsymbol{\xi} = \mathbf{Y} \tag{8}$$

Assuming that $\boldsymbol{\xi} \sim N(0, \sigma^2)$, we have the following probability density function:

$$P(\boldsymbol{\xi}_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\boldsymbol{\xi}_i^2}{2\sigma^2}\right) \tag{9}$$

The corresponding likelihood function should be

$$P(\boldsymbol{\xi}) = \prod_{i=1}^{\tilde{N}} P(\boldsymbol{\xi}_i) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{\tilde{N}} \exp\left(-\frac{\sum_{i=1}^{\tilde{N}} \boldsymbol{\xi}_i^2}{2\sigma^2}\right) \tag{10}$$

By replacing $\boldsymbol{\xi}_i$ with $\mathbf{y}_i$, (10) can be rewritten as

$$P(\mathbf{y}|\boldsymbol{\beta}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{\tilde{N}} \exp\left[-\frac{\sum_{i=1}^{\tilde{N}} (\mathbf{y}_i - \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta})^2}{2\sigma^2}\right] \tag{11}$$

Accordingly, the log style of the above likelihood function can be represented as

$$L(\mathbf{y}|\boldsymbol{\beta}) = -\frac{\tilde{N}}{2} \log\left(2\pi\sigma^2\right) - \left(\frac{\sum_{i=1}^{\tilde{N}} (\mathbf{y}_i - \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta})^2}{2\sigma^2}\right) \tag{12}$$

By maximizing (12), we can obtain the optimal $\boldsymbol{\beta}$:

$$\tilde{\boldsymbol{\beta}} = \arg\max L\left(\mathbf{y}|\boldsymbol{\beta}\right) = \arg\min \sum_{i=1}^{\tilde{N}} \left(\mathbf{y}_i - \mathbf{h}\left(\mathbf{x}_i\right)\boldsymbol{\beta}\right)^2 \tag{13}$$

However, if the original Gaussian noise is changed as other kinds of noise distributions, we cannot obtain the optimal $\boldsymbol{\beta}$ through (13). Therefore, if a predictor is created based on the assumption that the noise follows a specific distribution, such as Gaussian distribution or Laplace distribution, it cannot guarantee to obtain the satisfactory performance.

3.2. Motivation of Improving Modeling Capability with Unknown Noise

According to (8), the estimated output weights of ELM without the regularization coefficient can be expressed as

$$\begin{aligned}
\tilde{\boldsymbol{\beta}}^{\Psi} &= \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y} \\
&= \frac{\sum_{i=1}^{L}\mathbf{H}_i^T\left(\mathbf{H}_i\beta_i^{\Psi} + \boldsymbol{\xi}_i\right)}{\sum_{i=1}^{L}\mathbf{H}_i^T\mathbf{H}_{\mathbf{i}}} \\
&= \boldsymbol{\beta}^{\Psi} + \frac{\sum_{i=1}^{L}\mathbf{H}_i^T\boldsymbol{\xi}_i}{\sum_{i=1}^{L}\mathbf{H}_i^T\mathbf{H}_i}
\end{aligned} \tag{14}$$

The estimated output weights of ELM with the regularization coefficient should be

$$\begin{aligned}
\tilde{\boldsymbol{\beta}} &= \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y} \\
&= \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1}\left(\mathbf{H}^T\mathbf{H}\right)\left[\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y}\right] \\
&= \left\{\left(\mathbf{H}^T\mathbf{H}\right)\left[\frac{\left(\mathbf{H}^T\mathbf{H}\right)^{-1}}{C} + \mathbf{I}\right]\right\}^{-1}\left(\mathbf{H}^T\mathbf{H}\right)\left[\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y}\right] \\
&= \left[\frac{\left(\mathbf{H}^T\mathbf{H}\right)^{-1}}{C} + \mathbf{I}\right]^{-1}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\left(\mathbf{H}^T\mathbf{H}\right)\left[\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y}\right] \\
&= \left[\frac{\left(\mathbf{H}^T\mathbf{H}\right)^{-1}}{C} + \mathbf{I}\right]^{-1}\left[\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y}\right] \\
&= \left[\frac{\left(\mathbf{H}^T\mathbf{H}\right)^{-1}}{C} + \mathbf{I}\right]^{-1}\tilde{\boldsymbol{\beta}}^{\Psi}
\end{aligned} \tag{15}$$

and can be represented as

$$\tilde{\boldsymbol{\beta}} = \frac{\sum_{i=1}^{L}\mathbf{H}_i^T\left(\mathbf{H}_i\beta_i^{\Psi} + \boldsymbol{\xi}_i\right)}{\frac{\mathbf{I}}{C} + \sum_{i=1}^{L}\mathbf{H}_i^T\mathbf{H}_i} \tag{16}$$

According to (14) and (16), when dealing with the specific tasks, the hidden layer output matrix of ELM may change largely due to the random assigned hidden layer parameters and the effects of noise, which sequentially leads to the corresponding change of the output weights. As we know, the unstable fluctuations of the output weights increase both of the structural risk and the empirical risk of ELM [30].

For the approximation capability of ELM, Huang *et al.* [4, 31] have proved that ELM can approximate an arbitrary continuous target function with a very small error δ ($\delta > 0$). Accordingly, we have the following theorem to guarantee the robustness of ELM in handling modeling tasks with unknown noise.

Theorem 1. Given any small value $\hat{\delta}$ ($\delta > \hat{\delta} > 0$), and activation function, which is infinitely differentiable in any internal, if the effects of external disturbance from the noise can be reduced or eliminated, ELM can approximate any target function with error $\hat{\delta}$.

Proof. Let $\Delta\mathbf{H}$ and $\Delta\boldsymbol{\beta}$ denote the change of the hidden layer output matrix and the output weights, caused by the external disturbance from the noise, respectively. Thus, (2) is modified as

$$(\mathbf{H} + \Delta\mathbf{H})(\boldsymbol{\beta} + \Delta\boldsymbol{\beta}) = \mathbf{Y} \tag{17}$$

$$(\mathbf{H} + \Delta\mathbf{H})(\boldsymbol{\beta} + \Delta\boldsymbol{\beta}) = \mathbf{H}\boldsymbol{\beta} \tag{18}$$

$$\mathbf{H}\Delta\boldsymbol{\beta} + \Delta\mathbf{H}\Delta\boldsymbol{\beta} = -(\Delta\mathbf{H}\boldsymbol{\beta}) \tag{19}$$

after that, we have

$$\Delta\boldsymbol{\beta} = -(\Delta\mathbf{H}\boldsymbol{\beta})(\mathbf{H} + \Delta\mathbf{H})^{\dagger} \tag{20}$$

where $(\mathbf{H} + \Delta\mathbf{H})^{\dagger}$ denotes the Moore-Penrose generalized inverse of matrix $(\mathbf{H} + \Delta\mathbf{H})$.

According to (20), we have

$$\|\Delta\boldsymbol{\beta}\| \leq \|\Delta\mathbf{H}\boldsymbol{\beta}\| \left\|(\mathbf{H} + \Delta\mathbf{H})^{\dagger}\right\| \tag{21}$$

Accordingly, the right term of (21) is the upper bound of $\|\Delta\boldsymbol{\beta}\|$, if we can reduce or eliminate the effects of external disturbance of the noise, both $\left\|(\mathbf{H} + \Delta\mathbf{H})^{\dagger}\right\|$ and $\|\Delta\mathbf{H}\boldsymbol{\beta}\|$ will be smaller, enabling $\|\Delta\boldsymbol{\beta}\|$ to be smaller, so ELM will be more robust with better generalization performance. □

Remark 1. Bartlett [32] has proved that the generalization performance of the feedforward neural network with smaller model noise and output weights should be better, and the conclusion is also true for ELM [16, 31].

Remark 2. Actually, (8) is the explicit expression of the relationship among the created model, noise, and data, and (18) is the corresponding implicit expression. In (18), the effects of noise are denoted by $\Delta\mathbf{H}$ and $\Delta\boldsymbol{\beta}$, respectively.

Generally, there are two popular modified schemes to tackle the above issue:

1) Removing or filtering the noise components of the data. This scheme aims to improve the data quality by data cleansing techniques or low-pass filters. However, it may simultaneously remove the normal data and break the original data structure, which seems to be unreasonable.

2) Optimizing the objective function, which is the sum of the output weights squares and the sum of the error squares. This scheme aims to reduce the effects of both the structural risk and empirical risk. However, it only can work well under the assumption that noise follows Gaussian distribution, which usually cannot hold. Practically, noise is usually subject to unknown distributions, such as Laplace distribution, Beta distribution, and even mixed distributions.

In order to tackle the above issues, we should propose a more robust and efficient machine learning approach to strengthen the modeling capability with unknown noise. Considering the excellent capability of mixture of Gaussian (MoG) of approximating any continuous noise distribution, we modify the original objective function of ELM by embedding MoG, in which the characteristic of noise is described using MoG [33].

3.3. Objective Function of R-ELM

According to the above theoretical analysis, the objective function of ELM in (5) is not suitable in several scenarios. If we describe the feature of noise using MoG instead of the original one in the objective function, the proposed

approach should be more robust. The probability density function of MoG is

$$P(\xi) = \sum_{j=1}^{\tilde{K}} \pi_j N_j\left(\xi|0, \sigma_j^2\right) \tag{22}$$

where $\pi_j > 0$ denotes the weighted coefficient, and $\sum_{j=1}^{\tilde{K}} \pi_j = 1$, $N_j\left(\xi|0, \sigma_j^2\right)$ denotes the Gaussian distribution with $\left(0, \sigma_j^2\right)$, and $\tilde{K}$ is the number of independent Gaussian distributions of MoG, respectively.

Thus, the objective function of R-ELM can be mathematical represented as

$$\begin{aligned} &Min: \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \frac{1}{2}C\sum_{i=1}^{\tilde{N}} \log \sum_{j=1}^{\tilde{K}}\left(\pi_j\left(\frac{1}{\sqrt{2\pi}\sigma_j}\exp\left(-\frac{\boldsymbol{\xi}_i^2}{2\sigma_j^2}\right)\right)\right) \\ &s.t., \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{y}_i - \boldsymbol{\xi}_i, i = 1, \ldots, \tilde{N} \end{aligned} \tag{23}$$

The likelihood function of $\boldsymbol{\xi}$ can be expressed as

$$P(\boldsymbol{\xi}|\vartheta) = \prod_{i=1}^{\tilde{N}} P(\boldsymbol{\xi}_i|\vartheta) = \prod_{i=1}^{\tilde{N}} \sum_{j=1}^{\tilde{K}} \pi_j N_j\left(\boldsymbol{\xi}_i|0, \sigma_j^2\right) \tag{24}$$

where $\vartheta = \left(\pi_1, \pi_2, ..., \pi_{\tilde{K}}, \sigma_1^2, \sigma_2^2, ..., \sigma_{\tilde{K}}^2, \boldsymbol{\beta}\right)$ denotes the set of parameters need to be estimated.

The corresponding log style is

$$L(\boldsymbol{\xi}|\vartheta) = \sum_{i=1}^{\tilde{N}}\left(\log \sum_{j=1}^{\tilde{K}} \pi_j\left(\frac{1}{\sqrt{2\pi\sigma_j^2}}\exp\left(-\frac{\boldsymbol{\xi}_i^2}{2\sigma_j^2}\right)\right)\right) \tag{25}$$

As mentioned above, MoG can theoretically approximate any continuous noise distribution, thus the proposed R-ELM should be more robust than ELM, especially in handling tasks with uncertainties.

Remark 3. According to the analysis of the above sections, the objective function in (5) is suitable for the noise following Gaussian distribution, and other similar objective functions can be constructed for different single noise distributions. However, noise of the real world applications is usually complex and unknown, it is difficult to character, leading to the poor robustness of the objective functions only for the specific noise distributions. Differently, we treat the noise as the random variables without prior knowledge following MoG distribution, and utilize the second term of (23) to character the features of noise.

In the new objective function, the optimal number of Gaussian mixtures κ is an important user-specified parameter. In order to guarantee the approximation capability of MoG, we should initialize a relatively large $\tilde{K}$, but there are usually redundant components. Thus, if there is no distinct difference between two components, we can merge them without performance degradation. As depicted in Fig. 2, assuming that we use the R-ELM with 5-component MoG to represent a given target function, and there are two components with the same horizontal coordinate. According to Fig.2, we can find that the line connects the centers of the other 3 components, but just goes through the middle of the two components, this indicates that these two components are redundant. If we merge them, the new component will have an inflated variance, and the approximation capability of the new 4-component MoG should be similar to the original 5-component MoG. A common pruning rule for the redundant components is that if the variance of the arbitrary two components are similar, they can be merged as a new 'big' component, which is also applied to search the optimal number of Gaussian mixtures in this paper.

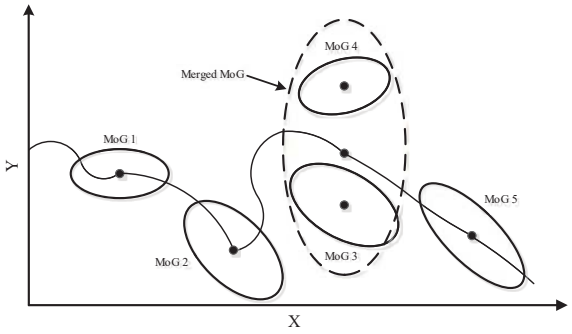


Figure 2: Redundant components in MoG

3.4. Solving Process of R-ELM

Considering that the analytical solutions of the parameters in ϑ cannot be calculated directly, we implement expectation maximization (EM) algorithm to estimate these parameters [34]. Assuming that the complete data include the observed data $\boldsymbol{\xi}_i$, and the unobservable component indicator matrix $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, ..., \boldsymbol{\omega}_{\tilde{N}})^T$, $\boldsymbol{\omega}_i = (\omega_{i1}, \omega_{i2}, ..., \omega_{i\tilde{K}})$ denotes the corresponding unobservable component indicator vector. If $\boldsymbol{\xi}_i$ comes from the jth component, then $\omega_{ij} = 1$, and others of $\boldsymbol{\omega}_i$ should be 0, thus, $\sum_{j=1}^{\tilde{K}} \omega_{ij} = 1$, and $\sum_{i=1}^{\tilde{N}} \sum_{j=1}^{\tilde{K}} \omega_{ij} = \tilde{N}$. Thus, according to (23), we have

$$L(\theta|\vartheta) = \sum_{i=1}^{\tilde{N}} \sum_{j=1}^{\tilde{K}} \omega_{ij} \left(\log \pi_j - \frac{1}{2} \log \left(2\pi\sigma_j^2 \right) - \frac{\boldsymbol{\xi}_i^2}{2\sigma_j^2} \right) \tag{26}$$

where $\theta = (\boldsymbol{\xi}, \boldsymbol{\omega})$ denotes the complete dataset.

EM algorithm involves two phases, including the E-step and M-step. In the E-step, according to $\boldsymbol{\xi}$, and the estimated $\tilde{\vartheta}_q$ after q times' iteration, we can obtain the following expression by calculating the conditional expectation of $L(\theta|\vartheta)$ with respect to $\boldsymbol{\omega}$:

$$Q\left(\vartheta|\tilde{\vartheta}_q\right) = \sum_{i=1}^{\tilde{N}} \sum_{j=1}^{\tilde{K}} \lambda_{ij} \left(\log \pi_j \right) + \sum_{i=1}^{\tilde{N}} \sum_{j=1}^{\tilde{K}} \lambda_{ij} \left(-\frac{1}{2} \log 2\pi - \frac{1}{2} \log \sigma_j^2 - \frac{\boldsymbol{\xi}_i^2}{2\sigma_j^2} \right) \tag{27}$$

where λ_{ij} stands for the posterior responsibility of the ith observation ξ_i coming from the jth component of MoG:

$$\lambda_{ij} = E\left(\omega_{ij}\right) = \frac{\pi_j^q N_j \left(\boldsymbol{\xi}_i^q | 0, \sigma_j^{2,q} \right)}{\sum_{j=1}^{\tilde{K}} \pi_j^q N_j \left(\boldsymbol{\xi}_i^q | 0, \sigma_j^{2,q} \right)} \tag{28}$$

In the M-step, Q function can be maximized iteratively, and then obtain all the parameters in ϑ. By calculating the partial derivatives of (28), we can obtain the updated expressions of π_j and σ_j^2:

$$\pi_j = \frac{\sum_{i=1}^{\tilde{N}} \lambda_{ij}}{\tilde{N}} \tag{29}$$

$$\sigma_j^2 = \sum_{i=1}^{\tilde{N}} \boldsymbol{\xi}_i^2 \left(\sum_{i=1}^{\tilde{N}} \lambda_{ij} \right)^{-1} \tag{30}$$

Actually, maximizing the above Q function equals to maximizing the following expression:

$$\begin{aligned} f_\beta &= -\sum_{i=1}^{\tilde{N}}\sum_{j=1}^{\tilde{K}}\lambda_{ij}\frac{\boldsymbol{\xi}_i^2}{2\sigma_j^2} \\ &= -\sum_{i=1}^{\tilde{N}}\left(\sum_{j=1}^{\tilde{K}}\frac{\lambda_{ij}}{2\sigma_j^2}\right)(\mathbf{y_i}-\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta})^2 \\ &= -\|\boldsymbol{\mu}\odot(\mathbf{Y}-\mathbf{H}\boldsymbol{\beta})\|^2 \end{aligned} \tag{31}$$

where $\boldsymbol{\mu}=(\mu_1,\mu_2,...,\mu_{\tilde{N}})^T$ denotes the weight vector, and $\mu_i=\sqrt{\sum_{i=1}^{\tilde{K}}\frac{\lambda_{ij}}{2\sigma_j^2}}$, $\odot$ is the Hadamard product, respectively.

The dual optimization problem of (31) can be represented as

$$\begin{aligned} &Min:\|(\boldsymbol{\mu}\odot\boldsymbol{\xi})\|^2 \\ &s.t., \mathbf{H}\boldsymbol{\beta}=\mathbf{Y}-\boldsymbol{\xi} \end{aligned} \tag{32}$$

Thus, (23) can be modified as

$$\begin{aligned} &Min:\frac{1}{2}\|\boldsymbol{\beta}\|^2+\frac{1}{2}C\|(\boldsymbol{\mu}\odot\boldsymbol{\xi})\|^2 \\ &s.t., \mathbf{H}\boldsymbol{\beta}=\mathbf{Y}-\boldsymbol{\xi} \end{aligned} \tag{33}$$

The Lagrangian function of (33) is

$$L(\boldsymbol{\beta},\boldsymbol{\xi},\boldsymbol{\alpha})=\frac{1}{2}\|\boldsymbol{\beta}\|^2+\frac{1}{2}C\|(\boldsymbol{\mu}\odot\boldsymbol{\xi})\|^2-\boldsymbol{\alpha}^T(\boldsymbol{\xi}-\boldsymbol{Y}+\mathbf{H}\boldsymbol{\beta}) \tag{34}$$

where $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\ldots,\alpha_{\tilde{N}})^T$ denotes the Lagrangian coefficient.

After that, according to the KKT theorem, by setting the gradient of the above Lagrangian function with respect to $(\boldsymbol{\beta},\boldsymbol{\xi},\boldsymbol{\alpha})$ equal to zero, we have

$$\frac{\partial L}{\partial\boldsymbol{\beta}}=0\rightarrow\boldsymbol{\beta}=\mathbf{H}^T\boldsymbol{\alpha} \tag{35a}$$

$$\frac{\partial L}{\partial\boldsymbol{\xi}}=0\rightarrow\boldsymbol{\xi}=C^{-1}diag(\boldsymbol{\mu})^{-2}\boldsymbol{\alpha} \tag{35b}$$

$$\frac{\partial L}{\partial\boldsymbol{\alpha}}=0\rightarrow\boldsymbol{\xi}-\mathbf{Y}+\mathbf{H}\boldsymbol{\beta}=0 \tag{35c}$$

Substituting (35b) in (35c), we have

$$\boldsymbol{\alpha}=\left[C^{-1}diag(\boldsymbol{\mu})^{-2}\right]^{-1}(\mathbf{Y}-\mathbf{H}\boldsymbol{\beta}) \tag{36}$$

Substituting (36) in (35a), we have

$$\boldsymbol{\beta}=\left[\mathbf{I}+\mathbf{H}^T\left[C^{-1}diag(\boldsymbol{\mu})^{-2}\right]^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^T\left[C^{-1}diag(\boldsymbol{\mu})^{-2}\right]^{-1}\mathbf{Y} \tag{37}$$

After that, we can calculate the predicted output of R-ELM:

$$f_{R-ELM}(\mathbf{x})=\mathbf{H}\boldsymbol{\beta}=\mathbf{H}\left[\mathbf{I}+\mathbf{H}^T\left[C^{-1}diag(\boldsymbol{\mu})^{-2}\right]^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^T\left[C^{-1}diag(\boldsymbol{\mu})^{-2}\right]^{-1}\mathbf{Y} \tag{38}$$

Remark 4. Due to the random feature mapping mechanism, the learning problem of ELM is reduced to that of estimating the optimal output weights. Similarly, R-ELM model also can be created based the calculated output weights using (37). According to (6), the hidden layer parameters of ELM included in the hidden layer output matrix

$\mathbf{H}$ have effects on the output weights with predetermined C and $\mathbf{Y}$. Differently, both the hidden layer parameters and $\boldsymbol{\mu}$ have effects on the output weights of R-ELM, it means that the model performance of R-ELM is partially associated with MoG.

Accordingly, R-ELM can be summarized as Algorithm 2.

Algorithm 2 R-ELM

Input: Given a training dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{\tilde{N}} \subset R^n \times R^m$, an activation function $g(\cdot)$, the number of hidden nodes L, the regularization factor C, and initialized number of Gaussian mixtures $\tilde{K}$.

1: Hidden layer parameters initialization. Randomly generating the hidden layer parameters of R-ELM $\mathbf{a}_i$ and b_i, $i = 1, \ldots, L$;

2: Update the parameters of MoG using EM algorithm. Calculate λ_{ij} using (28), update π_j and σ_j^2 using (29) and (30), sequentially update $\boldsymbol{\xi}_i$ until it reaches a predefined threshold;

3: Model building. Obtain the output weights $\boldsymbol{\beta}$ using (37), and then create the R-ELM model using (38).

4. Performance Verification on Benchmark Datasets

In this section, some selected benchmark datasets are employed to verify the effectiveness of the proposed R-ELM by comparing with a number of state-of-the-art machine learning approaches, including ELM [2], residual compensation ELM (RC-ELM) [16], PR-ELM [27], support vector machine (SVM), and back-propagation neural network (BPNN). In addition, all the experiments are conducted using Matlab 2015b running on a i5 3.2 GHz CPU with 4G RAM. In the experiments, the following root mean square error ($RMSE$) is chosen as the evaluation criterion:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{z} (y_i - \tilde{y}_i)^2}{z}} \tag{39}$$

where $\tilde{y}_i$ denotes the predicted value of y_i, and z is the number of samples used in the experiments, respectively.

4.1. Experimental Settings

In order to verify the validity of the proposed R-ELM with unknown noise, we will add the following kinds of noise into the datasets:

1) Gaussian noise: noise follows $N(0, 0.1^2)$;

2) Gaussian noise: noise follows $N(0, 1)$;

3) Laplace noise: noise follows $L(0, 0.5^2)$;

4) Mixed noise: noise includes 30% Gaussian noise following $N(0, 1)$, 30% Gaussian noise following $N(0, 0.3^2)$, 30% Gaussian noise following $N(0, 0.1^2)$, and 10% Laplace noise following $L(0, 0.1^2)$, respectively.

We firstly evaluate the performance using the following 'SinC' function:

$$y(x) = \begin{cases} \sin(x)/x, x \neq 0 \\ 1, x = 0 \end{cases} \tag{40}$$

Both of the number of the training samples and the testing samples are 5000, and all of them are created uniformly randomly distributed on the interval (-10,10), respectively. The above four types of noise are added to the training samples, while the testing samples remain noise-free.

In addition, we also select three regression datasets, including Auto-MPG, Abalone, California Housing, and a time series prediction dataset, i.e., Mackey-Glass, from the UCI Machine Learning Repository. Table 1 lists the details of the selected benchmark datasets, including attribute number, training data number, and testing data number. In the experiments, all the data are normalized to the range [0,1]. For the sake of simplicity, we choose the Sigmoid as the activation functions of ELM, RC-ELM, PR-ELM, and R-ELM, and other user specified parameters of all the machine learning approaches are determined using cross validation method.

Table 1: Details of the Selected Benchmark Datasets

Datasets	#Attributes	#Training Data	#Testing Data
Auto-MPG	7	320	72
Abalone	8	3000	1177
California Housing	8	8000	12640
Mackey Glass	4	1500	500

The Mackey-Glass data are generated by the following time-delay differential equation:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{41}$$

where $x(t)$ stands for the values of time series at time t, $\tau = 17$ denotes the delayed time. We initially set the $x(0) = 1.2$ and $x(t) = 0$ for $t < 0$. Furthermore, a sinusoid with amplitude 0.2 is added to the series to obtain the nonstationary time series, as depicted in Fig. 3. After that, we predict $y(t) = x(t+6)$ from the past values of this time series:

$$[x(t), x(t-2), x(t-12), x(t-18); x(t+6)] \tag{42}$$

We collect 2000 data pairs based on (42), in which the first 1500 input-output instances for training, while the other 500 for testing.

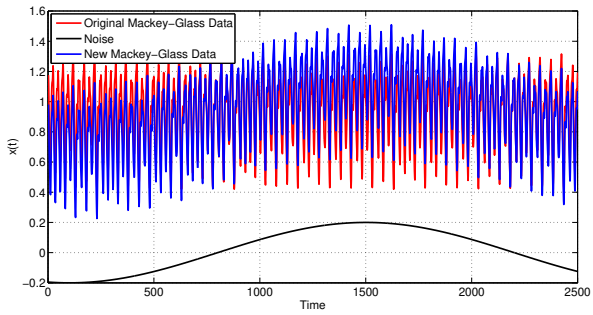
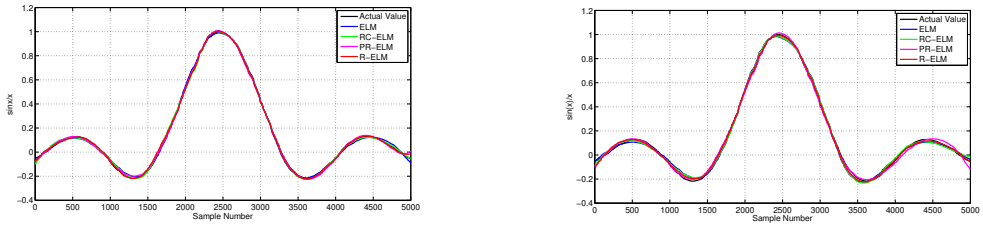


Figure 3: Mackey-Glass chaotic time series

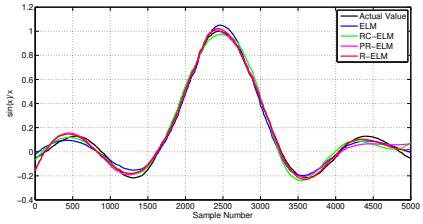
4.2. Simulation Results and Analysis

Fig. 4 illustrates the comparison results of 'SinC' with different kinds of noise. Accordingly, we can find that the performance of the four machine learning approaches are comparative and close to the actual value when the data are corrupted by Gaussian noise $N(0, 0.1^2)$. When the added noise changes as Gaussian noise $N(0, 1)$, their performance become worse than before, but all of them still can track the 'SinC' curve. However, when we add Laplace noise into the data, the accuracy of some approaches become worse. For example, ELM and RC-ELM have deviated from the actual value at the peaks. Furthermore, in Fig. 4(d), when we add mixed noise into the data, ELM, RC-ELM, and even PR-ELM seriously deviate from the actual value, but R-ELM still can fit it well in most of the situations. This is because R-ELM can approximate the mixed noise through the MoG in the modified objective function. In addition, PR-ELM is more robust than ELM and RC-ELM, because PR-ELM not only minimizes the output weights, as well as both the mean and variance of the modeling error. However, it still cannot obtain satisfactory performance with the data corrupted by unknown noise, the reason is that squared loss is still utilized in the objective function of PR-ELM.

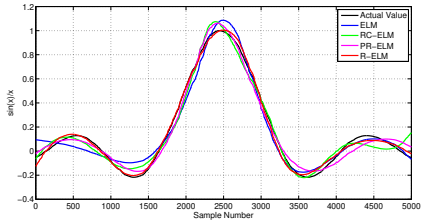


(a) Gaussian noise $N\left(0, 0.01^2\right)$

(b) Gaussian noise $N\left(0, 1\right)$



(c) Laplace noise $L(0, 0.5^2)$



(d) Mixed noise

Figure 4: Comparison results of 'SinC' with different kinds of noise

The detailed comparison results both with and without additional noise of different approaches are listed in Tables 2-5. According to Table 2, when the benchmark data are pure, the performance of all the approaches are almost comparative. In Table 3, when we add Gaussian noise into the datasets, all of the approaches are still not affected seriously. However, when we add Laplace noise into the datasets, the performance of ELM, RC-ELM, SVM, and BPNN become worse. After we add mixed noise into the datasets, even PR-ELM becomes unstable, but R-ELM still can obtain satisfactory performance.

Fig. 5 depicts the comparison results between R-ELM and PR-ELM on Mackey-Glass dataset with mixed noise. Accordingly, it is easy to observe that R-ELM outperforms PR-ELM in most of the situations, which also validates the aforementioned theoretical analysis.

Table 2: Comparison Results on Selected Benchmark Datasets without Additional Noise

Datasets	**ELM**	**RC-ELM**	**PR-ELM**	**R-ELM**	**SVM**	**BPNN**
Auto-MPG	0.1271	0.1202	0.1197	**0.1189**	0.1269	0.1355
Abalone	0.0759	0.0731	**0.0718**	0.0735	0.0782	0.0931
California Housing	0.1293	0.1160	0.1172	0.1161	**0.1158**	0.1329

Table 3: Comparison Results on Selected Benchmark Datasets with Gaussian Noise

Datasets	**ELM**	**RC-ELM**	**PR-ELM**	**R-ELM**	**SVM**	**BPNN**
Auto-MPG	0.1351	0.1232	**0.1220**	0.1223	0.1325	0.1397
Abalone	0.0812	0.0757	0.0769	**0.0751**	0.0801	0.1001
California Housing	0.1302	0.1181	0.1179	**0.1173**	0.1202	0.1319

Table 4: Comparison Results on Selected Benchmark Datasets with Laplace Noise

Datasets	ELM	RC-ELM	PR-ELM	R-ELM	SVM	BPNN
Auto-MPG	0.1467	0.1329	**0.1296**	0.1298	0.1442	0.1609
Abalone	0.1027	0.0933	0.0856	**0.0831**	0.1058	0.1103
California Housing	0.1360	0.1198	0.1181	**0.1168**	0.1299	0.1320

Table 5: Comparison Results on Selected Benchmark Datasets with Mixed Noise

Datasets	ELM	RC-ELM	PR-ELM	R-ELM	SVM	BPNN
Auto-MPG	0.1711	0.1693	0.1602	**0.1381**	0.1789	0.1920
Abalone	0.1539	0.1525	0.1439	**0.1022**	0.1421	0.1670
California Housing	0.1411	0.1203	0.1215	**0.1189**	0.1376	0.1409

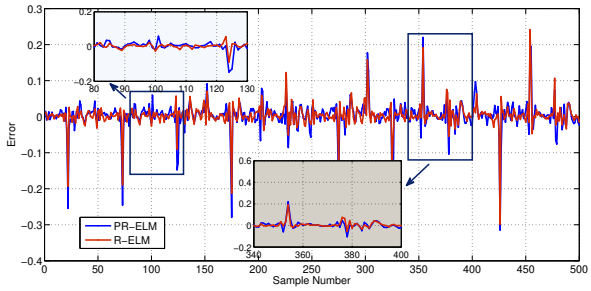


Figure 5: Comparison errors between R-ELM and RC-ELM of Mackey-Glass data

5. Performance Verification on Real World Applications

In the above section, we have conducted experiments on selected benchmark datasets to demonstrate the performance of the proposed R-ELM. Then, we further evaluate the validity of R-ELM using two real world applications, including gas utilization ratio (GUR) prediction and hot metal silicon content (HMSC) prediction in blast furnace ironmaking process.

Blast furnace is one of the dominant unit for producing molten iron in the manufacture of iron and steel with large uncertainties. It usually has multi-variable, strong nonlinear and highly complex characteristics, leading to the noise complex and unknown (see Fig. 6) [35]. GUR is one of the most concerned indicators reflecting the blast furnace status, and HMSC is a main indicator of product quality and thermal state inside blast furnace. It is critical to accurately predict GUR and HMSC for monitoring and assessing the operation state of blast furnace. Specifically, the experimental data are collected from a blast furnace with inner volume of 2500 m^3, and we randomly select 1500 data pairs for the following experiments (1000 data pairs for training, and the other 500 for testing). Some important variables are considered as the input features, such as blast volume (m^3/ min), blast pressure (kPa), blast temperature (°C), oxygen enrichment percentage (wt%), permeability index (m^3/ min ·kPa), top temperature (°C), and top pressure (kPa), etc.

5.1. Gas Utilization Ratio Prediction

Some statistical properties of the aforementioned variables of the experimental data are firstly calculated, including maximum, minimum, mean, and standard deviation (SD). The corresponding results are detailed in Table 6 and

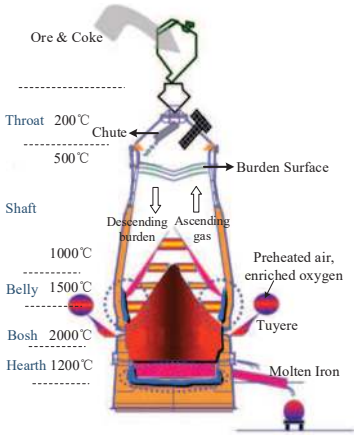


Figure 6: Inside of blast furnace

Table 7, respectively. Specifically, Table 6 lists the statistical properties of GUR in respect of the training data and testing data, and Table 7 lists the statistical properties of the selected variables. Accordingly, both of the training data and testing data demonstrate various statistical properties, indicating the characteristics of violent fluctuation of the experimental data. It means that those data can better verify the generalization performance of R-ELM. As we know, variables with large magnitude should have larger effects on the modeling than the ones with magnitude, thus all the experimental data are normalized into (0, 1) with the same magnitude to eliminate the influence of dimension among variables.

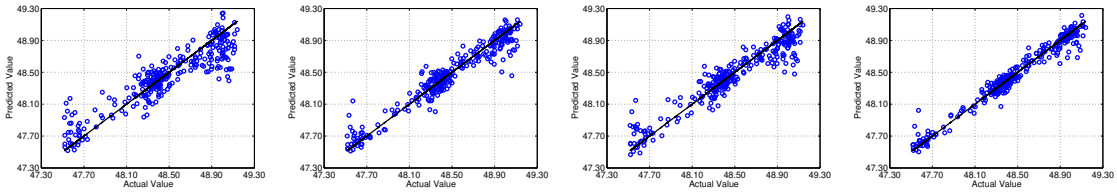
Table 6: Statistical properties of training data and testing data of GUR

Data	**Max**	**Min**	**Mean**	**SD**
Training	48.90	47.25	48.08	0.35
Testing	49.15	47.35	48.26	0.39

Table 7: Statistical properties of the variables of GUR

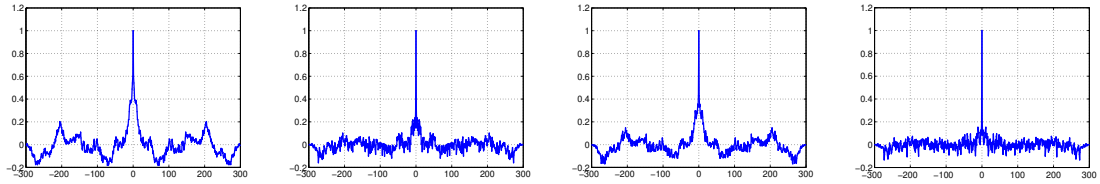
No.	**Variable**	**Max**	**Min**	**Mean**	**SD**
1	blast volume	4925.4	4866.7	4898.2	10.75
2	blast pressure	382.27	361.52	372.14	3.94
3	blast temperature	1134.1	1127.5	1130.1	1.28
4	oxygen enrichment percentage	45.96	35.22	42.43	1.21
5	permeability index	37.54	29.32	32.01	1.60
6	top temperature	308.52	202.10	276.83	23.21
7	top pressure	224.66	215.95	220.36	1.57

Fig. 7 illustrates comparison results of scatter distributions of ELM, RC-ELM, PR-ELM, and R-ELM, which can indicate the correlation between the predicted values and the actual values. Accordingly, it can be found that the scatter points of R-ELM are more concentrated in the black line (the black line $y = x$ denotes the actual value), and the predicted values of other approaches are relatively far away from the black line. In addition, Fig. 8 depicts error autocorrelation functions of those approaches, in which the error autocorrelation function of R-ELM closely



(a) ELM (b) RC-ELM (c) PR-ELM (d) R-ELM

Figure 7: Scatter distributions of GUR prediction



(a) ELM (b) RC-ELM (c) PR-ELM (d) R-ELM

Figure 8: Error autocorrelation functions of GUR prediction

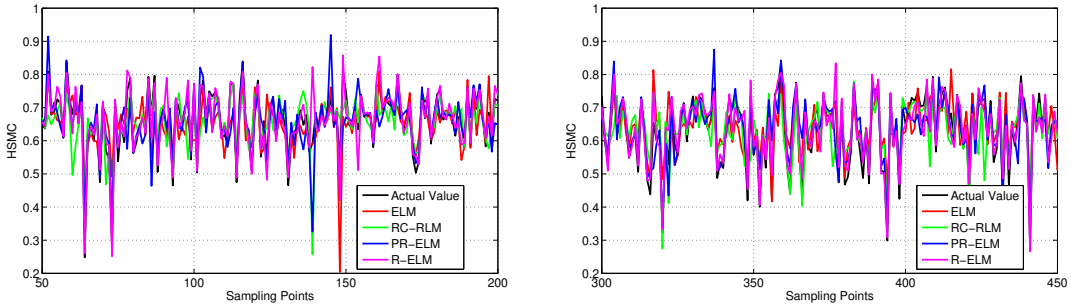
approximates to the white noise sequence. According to Fig. 7 and Fig. 8, the predicted results of those approaches are consistent with the actual GUR changing trend, but with differences in the agreement degree. We can find that the predicted result of ELM has the lowest agreement with the actual GUR values, and the predicted result of R-ELM is better than RC-ELM and PR-ELM. Actually, the difference of the predicted results between R-ELM and RC-ELM is little. The reason is that RC-RLM also can reduce the effects of noise through the hierarchical residual compensation mechanism. The detailed comparison results are summarized in Table 8. As observed from Table 8, R-ELM distinctly outperforms other approaches. Furthermore, the training RMSE and testing RMSE of R-ELM are almost two times better than ELM, SVM, and BPNN. In summarize, all the comparison results indicate the excellent modeling capability of R-ELM.

Table 8: Detailed Comparison Results of GUR Prediction

Approach	**Training RMSE**	**Testing RMSE**
ELM	0.1329	0.0927
RC-ELM	0.0605	0.0590
PR-ELM	0.0988	0.0706
R-ELM	**0.0500**	**0.0410**
SVM	0.1528	0.0855
BPNN	0.2106	0.2523

5.2. Hot Metal Silicon Content Prediction

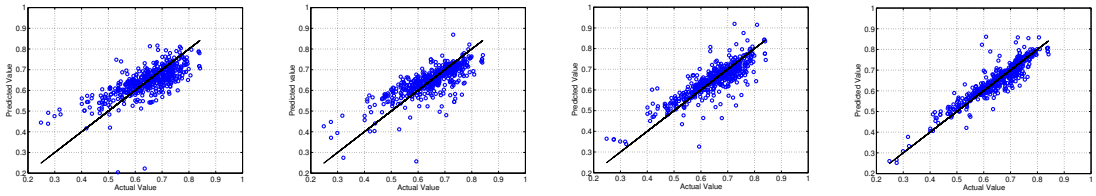
Similarly, considering the influence on the convergence speed and model complexity, all the data are firstly normalized into [0, 1] in HMSC prediction. It should note that HSMC is not normalized, because it has been controlled within (0.3, 0.7). The comparison predicted results of the selected samples of ELM, RC-ELM, PR-ELM, and R-ELM are illustrated in Fig. 9, in which the black line, red line, green line, blue line, and pink line denote the actual values, predicted results of ELM, RC-ELM, PR-ELM, and R-ELM, respectively. According to Fig. 9, R-ELM can track the actual HSMC values well, and better than other approaches. In addition, we can find that the predicted result of ELM is far from the actual HSMC values in several situations, this is because ELM has weaker tolerance capability of



(a) Predicted results from 50 to 200

(b) Predicted results from 300 to 450

Figure 9: Comparison results of HMSC prediction



(a) ELM

(b) RC-ELM

(c) PR-ELM

(d) R-ELM

Figure 10: Scatter distributions of HMSC prediction

complex and unknown noise. Differently, RC-ELM can utilize the residuals to improve the performance, PR-ELM has better robustness due to their modified objective functions considering the modeling errors caused by noise. The scatter diagrams of all the four approaches are depicted in Fig. 10, in which the regression points of R-ELM are closer to the black line. In addition, we also plot the probability density function (PDF) curves of all the four approaches in Fig. 11 for further demonstrating the dynamic tracking capability of them. According to Fig. 11, the PDF curve of R-ELM is higher and narrower, indicating that the corresponding errors are more concentrated near the mean 0 and the variance is also smaller. Thus, R-ELM can provide more accurate predicted result of HMSC.

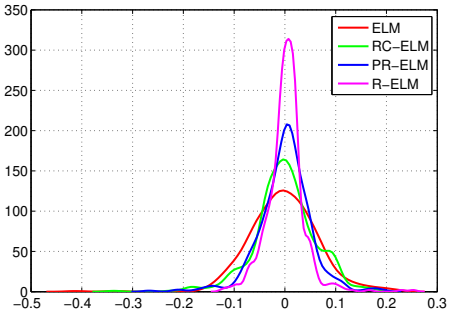


Figure 11: Probability density function curves of different approaches

6. Discussions

6.1. Influence Factors of Modeling Capability of ELM

In essence, three factors mainly confine the modeling capability of ELM, including:

1) Sensitive objective function. The objective function of ELM is sensitive to the non-Gaussian noise, which widely exists in the real world applications. In order to tackle this issue, robust variants are proposed, such as R-ELM and PR-ELM, in which the original objective function of ELM is modified to approximate the complex and unknown noise distribution.

2) Limited representation capability of the single hidden layer structure. Huang et al. [4, 31] have proven that ELM can theoretically reach the global optimum under the given network structure and user specified parameters. However, the finite training data may not guarantee the consistent good performance of ELM in the uncertain and heterogeneous environments due to its single hidden layer structure. Multilayer ELMs (ML-ELM) were designed for strengthening the representation capability of ELM. Different form conventional deep learning, ML-ELM integrates the representation learning and decision making into a whole training process, in which multiple hidden layers stacked using ELM autoencoder (ELM-AE) are for extracting high-level features, and a final layer of ELM or an ELM predictor/classifier at the last component for decision making. Accordingly, the proposed R-ELM also can be utilized as the decision maker for enhancing the performance of ML-ELM in handling tasks with complex and unknown noise. It should note that, R-ELM is mainly for regression problems, but classification problems may also involve the modeling problems with non-Gaussian noise, such as fault diagnosis [36, 37], thus how to construct robust ELM classifier by embedding MoG should be considered.

3) Random feature mapping mechanism. One of the advantages of ELM is the randomly generated hidden layer parameters, which makes ELM have very fast learning speed, even thousands of times faster than BPNN, SVM, and some conventional deep learning approaches. Simultaneously, it also leads to the instability of ELM [38, 39, 40, 41]. ELM is sensitive to the randomization range of the hidden layer parameters, whose change may cause serious performance degradation. However, there is still no clear and theoretical criterion to guide the selection of the randomization range for specific tasks. Usually, the randomization range is set as a fixed one, but different data have their own distributions, so it cannot guarantee ELM to obtain the satisfactory performance for different tasks with some specific randomization ranges.

6.2. Others

According to (1), ELM is actually a linear combination of several activation functions. Random feature mapping mechanism transforms the original nonlinear data-driven modeling into a learning problem of estimating the optimal output weights using least-square manner. Differently, according to (23) and (37), due to the embedded MoG, the analytical solutions of R-ELM cannot be calculated directly through least-square manner. Thus, we need perform EM algorithm to estimate the parameters of MoG component. In addition, the robust technique utilized in R-ELM also can be applied to the neural networks with similar objective function form of ELM, such as L_2 optimization based objective functions.

Robust modeling and robust control actually have the similar goal, which is to optimize the corresponding results with external uncertainties. Specifically, R-ELM is to create robust data-driven model for accurate feature mapping from the input to the output with complex and unknown noise, and the hierarchical nonstationary filter for networked Markov switching repeated scalar nonlinear systems proposed in [42] is to guarantee that it can track the setting values under external disturbance.

7. Conclusions

Most of the existing ELMs can theoretically obtain the optimal solutions under the assumption that the noise follows Gaussian distribution. However, in practice, noise of the real world applications is usually subject to unknown distributions, i.e., Gaussian, non-Gaussian, or even mixed distributions, which easily leads to the suboptimal solutions of these ELMs. In this paper, R-ELM is proposed to strengthen the modeling capability of classic ELM with unknown noise. Specifically, a modified objective function is constructed, in which the characteristic of noise is described utilizing MoG. Accordingly, it should be more robust even in the presence of unknown noise without performance

attenuation, due to the excellent capability of MoG for approximating any continuous noise distribution. In addition, EM algorithm is performed for solving the modified objective function of R-ELM. Comprehensive experiments on selected benchmark datasets and real world applications indicate that the proposed R-ELM outperforms state-of-the-art machine learning approaches. Similar to classic ELM, R-ELM is a batch learning algorithm, we consider to extend R-ELM for online modeling problems in our future work, enabling it more practical. In addition, R-ELM focuses on regression problems, we also consider to modify its objective function to enhance the modeling capability of classification problems with unknown noise.

Acknowledgment

This work is supported in part by China Postdoctoral Science Foundation under Grants 2019TQ0002 and 2019M660328, National Natural Science Foundation of China under Grant 61673055 and National Key Research and Development Program of China under Grant 2017YFB1401203.

References

[1] G. Huang, G.B. Huang, S.J. Song, K.Y. You, Trends in extreme learning machines: A review, Neural Netw. 61 (2015) 32-48.
[2] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: Theory and applications, Neurocomptuing 70 (2006) 489-501.
[3] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, E. Cambria, Bayesian network based extreme learning machine for subjectivity detection, J. Franklin Inst. 355 (2018) 1780-1797.
[4] G.B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17(4) (2006) 879-892.
[5] N.Y. Liang, G.B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online squential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17(6) (2006) 1411-1423.
[6] Y. Li, S. Zhang, Y. Yin, W. Xiao, J. Zhang, A novel online sequential extreme learning machine for gas utilization ratio prediction in blast furnaces, Sensors 17 (8) (2017) 1847-1870.
[7] Y. Lan, G.B. Huang, Constructive hidden nodes selection of extreme learning machine for regression, Neurocomptuing 73 (2010) 3191-3199.
[8] Y. Miche, M. Heeswijk, P. Bas, O. Simula, A. Lendasse, TROP-ELM: A double-regularized ELM using LARS and Tikhonov regularization, Neurocomptuing 74 (2011) 2413-2421.
[9] Y. Lan, Y.C. Soh, G.B. Huang, Ensemble of online sequential extreme learning machine, Neurocomptuing 72 (2009) 3391-3395.
[10] B. Mirza, Z.P. Lin, N. Liu, Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift, Neurocomptuing 149 (2015) 316-329.
[11] W. Xiao, J. Zhang, Y. Li, S. Zhang, W. Yang, Class-specific cost regulation extreme learning machine for imblanced classification, Neurocomputing 261 (2017) 70-82.
[12] Y. Li, S. Zhang, Y. Yin, W. Xiao, J. Zhang, Parallel one-class extreme learning machine for imbalance learning based on Bayesian approach, J. Amb. Intel. Hum. Comp. (2018) https://doi.org/10.1007/s12652-018-0994-x.
[13] X. Luo, Y. Xu, W. Wang, M. Yuan, X. Ban, Y. Zhu, W. Zhao, Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy, J. Franklin Inst. 355 (2018) 1945-1966.
[14] C.M. Wong, C.M. Vong, P.K. Wong, J. Cao, Kernel-based multilayer extreme learning machines for representation learning, IEEE Trans. Neural Netw. Lear. Syst. 29(3) (2018) 757-762.
[15] J. Zhang, W. Xiao, Y. Li, S. Zhang, Z. Zhang, Multilayer probability extreme learning machine for device-free localization, Neurocomputing 396 (2020) 383-393.
[16] J. Zhang, W. Xiao, Y. Li, S. Zhang, Residual compensation extreme learning machine for regression, Neurocomputing 311 (2018) 126-136.
[17] J. Zhang, Y. Lu, B. Zhang, W. Xiao, Device-free localization using empirical wavelet transform-based extreme learning machine, In: Proceedings of the 30th Chinese Control and Decision Conference, IEEE, 2018, pp. 2585-2590.
[18] J. Cao, T. Chen, J. Fan, Landmark recognition with compact BoW histogram and ensemble ELM, Multimed. Tools Appl. 75 (5) (2016) 2839-2857.
[19] J. Cao, Y. Zhao, X. Lai, M.E.H. Ong, C. Yin, Z.X. Koh, N. Liu, Landmark recognition with sparse representation classification and extreme learning machine, J. Franklin Inst. 352 (2015) 4528-4545.
[20] Y. Li, S. Zhang, Y. Yin, J. Zhang, W. Xiao, A soft sensing scheme of gas utilization prediction for blast furnace via improved extreme learning machine, Neural Process. Lett. 50 (2019) 1191-1213.
[21] Y. Li, S. Zhang, J. Zhang, Y. Yin, W. Xiao, Z. Zhang, Data-driven multi-objective optimization for burden surface in blast furnace with feedback compensation, IEEE Trans. Industr. Inform. 16 (4) (2020) 2233-2244.
[22] J. Zhang, W. Xiao, S. Zhang, S. Huang, Device-free localization via an extreme learning machine with parameterized geometrical feature extraction, Sensors 17 (4) (2017) 879-890.
[23] J. Zhang, Y. Li, W. Xiao, Data and knowledge twin driven integration for large-scale device-free localization, IEEE Internet of Things Journal (2020) doi: 10.1109/JIOT.2020.3005939.
[24] Z. Man, K. Lee, D. Wang, Z. Cao, C. Miao, A new robust training algorithm for a class of single-hidden layer feedforward neural networks, Neurocomptuing 74 (2011) 2491-2501.
[25] Y.L. He, Z.Q. Geng, Y. Xu, Q.X. Zhu, A hierarchical structure of extreme learning machine (HELM) for high-dimensional datasets with noise, Neurocomptuing 128 (2014) 407-414.

[26] X. Lu, H. Zou, H. Zhou, L. Xie, G.B. Huang, Robust extreme learning machine with its application to indoor positioning, IEEE Trans. Cybern. 46 (1) (2016) 194-205.
[27] X. Lu, L. Ming, W. Liu, H.X. Li, Probabilistic regularized extreme learning machine for robust modeling of noise data, IEEE Trans. Cybern. 48 (8) (2018) 2368-2377.
[28] Q. Hu, S. Zhang, Z. Xie, J. Mi, J. Wan, Noise model based v-support vector regression with its application to short-term wind speed forecasting, Neural Netw. 57 (2014) 1-11.
[29] H. Wang, Y. Wang, Q. Hu, Self-adaptive robust nonliear regression for unkown noise via mixture of Gaussians, Neurocomputing 235 (2017) 272-286.
[30] M. Anthony, P.L. Barlett, Neural network learning: Theoretical foundations, Cambridge University Press, Cambridge, 1999.
[31] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (2) (2012) 513-529.
[32] P.L. Bartlett, The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network, IEEE Trans. Inf. Theory 44 (2) (1998) 525-536..
[33] C.M. Bishop, Pattern recognition and machine learning, Springer, 2006.
[34] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood form incomplete data via the EM algorithm, J. Roy. Stat. Soc. Ser. B 39 (1) (1977) 1-38..
[35] Y. Li, H. Li, J. Zhang, S. Zhang, Y. Yin, Burden surface decision using MODE with TOPSIS in blast furnace ironmkaing, IEEE Access, 8 (2020) 35712-35725.
[36] Y. Wu, B. Jiang, N. Lu, A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices, IEEE Trans. Syst. Man Cybern. Syst. 49 (10) (2019) 2108-2118.
[37] Y. Wu, B. Jiang, Y. Wang, Incipient winding faults detection and diagnosis for squirrel-cage induction motors equipped on CRH trains, ISA Trans. 99 (2020) 488-495.
[38] G.B. Huang, What are extreme learning machines? Filling the gap between Frank Rosenblatts dream and John von Neumanns puzzle, Cogn. Comput. 7 (2015) 263-278.
[39] X.Z. Wang, R. Wang, C. Xu, Discovering the relationship between generalization and uncertainty by incorporating complexity of classification, IEEE Trans. Cybern. 48(2) (2018) 703-715.
[40] X. Liu, S. Lin, J. Fang, Z. Xu, Is extreme learning machine feasible? A theoretical assessment (part I), IEEE Trans. Neural Netw. Lear. Syst. 26(1) (2015) 7-20.
[41] S. Lin, X. Liu, J. Fang, Z. Xu, Is extreme learning machine feasible? A theoretical assessment (part II), IEEE Trans. Neural Netw. Lear. Syst. 26(1) (2015) 21-34.
[42] J. Cheng, J.H. Park, X. Zhao, H.R. Karimi, J. Cao, Quantized nonstationary filtering of network-based Markov switching RSNSs: A multiple hierarchical structure strategy, IEEE Trans. Automat. Contr. (2019) DOI: 10.1109/TAC.2019.2958824.

Declaration of Competing Interest

The Authors declare that we have no conflict of interest.