IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 15, 2020, accepted July 2, 2020, date of publication July 6, 2020, date of current version July 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007522

Extreme Learning Machine Under Minimum Information Divergence Criterion

CHENGTIAN SONG[1], LIZHI PAN[1], QIANG LIU[2], ZHIHONG JIANG[1], (Member, IEEE), AND JIANGUANG JIA[3]

[1]School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China
[2]Ordnance Science Institute of China, Beijing 100089, China
[3]Institute of Systems Engineering, AMS, PLA, Beijing 100091, China

Corresponding authors: Chengtian Song (songct@bit.edu.cn) and Zhihong Jiang (jiangzhihong@bit.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61973038.

ABSTRACT In recent years, extreme learning machine (ELM) and its improved algorithms have been successfully applied to various classification and regression tasks. In these algorithms, MSE criterion is commonly used to control training error. However, MSE criterion is not suitable to deal with outliers, which can exist in general regression or classification tasks. In this paper, a novel extreme learning machine under minimum information divergence criterion (ELM-MinID) is proposed to deal with the training set with noises. In minimum information divergence criterion, the Gaussian kernel function and Euclidean information divergence are utilized to substitute the mean square error (MSE) criterion to enhance the anti-noise ability of ELM. Experimental results on two synthetic datasets and eleven benchmark datasets show that this method is superior to traditional ELMs.

INDEX TERMS Extreme learning machine, minimum information divergence criterion, kernel method, gradient algorithm.

I. INTRODUCTION

Extreme learning machine (ELM) is a single hidden layer feedforward neural network (SLFN) with universal approximation capability [1], [2]. In ELM, the weights linking the input layers to the hidden layers and the hidden bias terms can be randomly initialize. Then, the corresponding weights linking the hidden layers to the output layers can be directly determined by the least square method based on the Moore-Penrose generalized inverse [3]. Different from full parameter determination algorithms such as back propagation (BP) algorithm, the hidden nodes' parameter random initialization process with an analytical weight solution can reduce computational complexity [1]. Therefore, the important advantage of ELM is the fast training speed. ELM has been widely used in many actual engineering applications, such as stock market forecasting [4], [5], image processing [6], [7], face recognition [8], and nonlinear model identification [9].

In recent years, some new improved versions of ELM have been proposed. In general, the performance of ELM is improved from two aspects. One is to optimize the network structure (like evolutionary ELM (E-ELM) [10], ELM-kernel (KELM) [11]), and the other is to improve the error statistics method (like regularized ELM (RELM) [12], outlier robust ELM (OR-ELM) [13]). In KELM, I/λ parameter is added in the hidden layer matrix to address the randomness problem of learning machine. RELM was proposed by Deng *et al.* [12], which achieves optimal trade-off between empirical risk $\|\varepsilon\|^2$ and structural risk $\|\beta\|^2$ by introducing regularization parameters, and makes the model obtain the best generalization performance. In OR-ELM [13], the ℓ_1 norm of the prediction error is used as its objective function, which can obtain better results when there are outliers in the regression task.

However, in essence, these ELMs use the mean square error (MSE) criterion to measure the error. MSE only limits the second-order statistics and shows a poor optimization ability for nonlinear and non-Gaussian (e.g. finite range or heavy-tail distributions) situations. MSE mainly focuses on the scatter aspects of the error distribution and cannot draw all the probabilistic information of the error, such as the shape (kurtosis, tails, peaks, etc.) of probability density function. To address this issue, Chen *et al.* [14]–[16] proposed a novel minimum information divergence (MinID) criterion, in which the Kullback-Leibler divergence between the actual error and

The associate editor coordinating the review of this manuscript and approving it for publication was Guangdeng Zong.

the desired error is selected as the objective function for adaptation algorithm. This criterion has been successfully used in adaptive filtering.

In order to overcome the defects of above ELMs and improve the anti-noise ability of ELM, a novel ELM-MinID algorithm is developed in this paper. In this algorithm, the MinID criterion based on Euclidean information divergence is applied to extreme learning machine (ELM). The main contributions of this paper are as follows:

1) we proposed a new method of error control: minimum information divergence criterion based on Euclidean information divergence.
2) we proposed a new ELM-MinID algorithm. Compared to the traditional ELMs, this algorithm utilizes the MinID criterion to substitute the MSE criterion, which makes ELM-MinID more resistant to noise.
3) we simulated the function fitting with synthetic data sets and the regression with benchmark data sets to verify our method.

The structure of this paper is as follows. In part A of section II, we provide a brief review of ELM. After that, the MinID criterion based on Euclidean information divergence is given in part B of section II. In section III, ELM under minimum information divergence criterion is developed. Subsequently, the performance of the algorithm is tested on synthetic data sets and benchmark data sets in section IV. Finally, conclusion is given in section V.

II. BACKGROUND

A. EXTREME LEARNING MACHINE (ELM)

For ELM, the input weights (connecting the input layer and the hidden layer) and hidden bias terms are randomly initialized, and the output weights (connecting the hidden layer and the output layer) are obtained by using Moore-Penrose generalized inverse.

We are training a single hidden-layer feedforward neural network with $\tilde{N}$ hidden neurons and activation functions $g(x)$ to learn N arbitrary distinct sample sequences $\{\mathbf{x}_k, \mathbf{t}_k\}_{k=1}^{N}$, where $\mathbf{x}_k = [x_{k1}, x_{k2}, \ldots, x_{kn}]^T \in \mathbf{R}^n$ is the kth input vector and $\mathbf{t}_k = [t_{k1}, t_{k2}, \ldots, t_{km}]^T \in \mathbf{R}^m$ is the associated desired value. In ELM, the activation function $g(x)$ is mathematically modeled as

$$\mathbf{y}_k = \sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_k + b_i), \quad k = 1, 2, \ldots, N \tag{1}$$

where $\mathbf{y}_k$ is the output weight vector of the SLFN for the kth input weight vector $\mathbf{x}_k$, $\mathbf{w}_i$ is the weight vector linking the ith hidden unit to all the input units, b_i is the hidden bias for the ith hidden unit, and β_i denotes the output weight vector linking the ith hidden unit to all the output units. In this way, the nonlinear system can be transformed to a linear system:

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

where

$$\boldsymbol{H} = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \tag{3}$$

$$\boldsymbol{\beta} = \left[\beta_1, \beta_2, \ldots, \beta_{\tilde{N}}\right]^T \tag{4}$$

$$\mathbf{Y} = [y_1, y_2, \ldots y_N]^T \tag{5}$$

and

$$\mathbf{T} = [t_1, t_2, \ldots, t_N]^T. \tag{6}$$

$\boldsymbol{\beta}$ is the vector of the weights linking the hidden layer to output layers, $\mathbf{H}$ denotes the output weight matrix of the hidden layer, $\mathbf{Y}$ is the output vector of the output layer, and $\mathbf{T}$ is the matrix of desired output.

The output weight vector $\boldsymbol{\beta}$ can be determined by minimizing the mean square error (MSE) (7)

$$\begin{aligned} e_{MSE} &= \frac{1}{N}\sum_{j=1}^{N} e_j^2 = \frac{1}{N} \parallel \mathbf{Y} - \mathbf{T} \parallel \\ &= \frac{1}{N} \parallel \boldsymbol{H}\boldsymbol{\beta} - \mathbf{T} \parallel \\ &= E(e_j^2) \end{aligned} \tag{7}$$

where E denotes the expectation operator and $e_j = \mathbf{t}_j - \sum_{p=1}^{\tilde{N}} g(\mathbf{w}_p \cdot \mathbf{x}_j + b_p)\beta_p$ is the estimation error. Usually, the solution of (7) can be determined by

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

where $\mathbf{H}^{\dagger}$ denotes the Moore-Penrose generalized inverse. When $\mathbf{H}^T\mathbf{H}$ is nonsingular, the orthogonal projection method can be used to calculate $\mathbf{H}^{\dagger}$ [1]:

$$\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T \tag{9}$$

However, there are still some shortages in the above ELM, such as the solution of the MSE function (7) is sensitive to non-Gaussian noises. The reason is that the MSE criterion captures only the second-order statistics of the residual and may perform poorly in nonlinear and non-Gaussian cases. In order to improve the robust performance in realistic situations, an alternative optimality criterion beyond the second-order statistics has been adopted in this study.

B. INFORMATION DIVERGENCE

The information divergence is a kind of distance measurement method between two distributions. Based on the Euclidean distance, a symmetric information divergence is given, called Euclidean information divergence. For two probability density functions $p(x)$ and $q(x)$, the Euclidean information divergence is given by

$$D(p \parallel q) = \int [p(x) - q(x)]^2 dx \tag{10}$$

which is always non-negative and equal to zero only if $p(x) = q(x)$. Obviously, the Euclidean information divergence is symmetric, we have $D(p \parallel q) = D(q \parallel p)$. In this work, the symmetric divergence (10) is used to measure the distance between two distributions.

In practice, the probability density functions $p(x)$ and $q(x)$ of samples are unknown. In the present paper, we adopt the kernel method [17] to estimate them. By kernel approach, the estimated PDF could be differentiable. This is the premise of the gradient calculation. The one-dimensional probability density estimator with kernel $K(.)$ is given by

$$\hat{p}(x) = \frac{1}{|S_p|\,\sigma} \sum_{x_k \in S_p} K(\frac{x - x_k}{\sigma}) \tag{11}$$

where σ is the kernel width, S_p denotes the sample sequence drawn independently from the probability density function $p(x)$, an $|S_p|$ is the total number of samples in S_p. Usually $K(.)$ will be a radially symmetric unimodal probability density function. The kernel function $K(x)$ satisfies $\int_R K(x)dx = 1$. In this work, we choose the standard Gaussian kernel function

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2) \tag{12}$$

The minimum of information divergence function (10) is called the minimum information divergence (MinID) criterion. Since divergence is insensitive to noises, it is better than the MSE especially when there is impulse noise in the samples [14].

III. ELM UNDER MINIMUM INFORMATION DIVERGENCE CRITERION

According to ELM learning theory, multiple types of feature maps can be used in ELM, so that ELM can approximate any continuous objective function. (refer to [2] for details). That is, given any continuous target function $y(x)$, there is a series of β_i to make the error equal to zero.

$$\begin{aligned} e &= \lim_{\tilde{N} \to +\infty} \left\| y_{(\cdot)} - y(x) \right\| \\ &= \lim_{\tilde{N} \to +\infty} \left\| \sum_{i=1}^{\tilde{N}} \beta_i g(\cdot) - y(x) \right\| \\ &= 0 \end{aligned} \tag{13}$$

Equation (13) is the cost function of ELM training. The purpose of ELM training is to make error between the training output and the desired output close to zero. The traditional ELM training utilizes the MSE criterion, like (7). However, the MSE criterion is sensitive to the non-Gaussian noises. In this section, the MinID criterion based on Euclidean information divergence is used as a cost function for ELM training.

Based on the MinID criterion, the ELM-MinID is proposed to minimize the information divergence between the actual error e and the desired error $e^{(d)}$ by adjusting the parameter $\boldsymbol{\beta}$. In other word, the output weight matrix $\boldsymbol{\beta}$ will be adjusted to make the PDF of error e_k close to the desired density function $p_{e^{(d)}}$. By setting the desired density function $p_{e^{(d)}}$ to a Dirac delta function at zero, the actual error e of ELM also converges around zero.

We can get a new objective function of ELM-MinID which minimizes the divergence between the actual error e and the desired error $e^{(d)}$, as follows:

$$D(\beta) = \arg\min_{\beta} D(p_e || p_{e^{(d)}}) \tag{14}$$

we use kernel method (11) and Gaussian kernel function (12) to estimate the PDF of actual error e, that is

$$p_e(e) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(e - e_i)^2}{2\sigma^2}) \tag{15}$$

where $e_i (i = 1, 2, \ldots, N)$ is the error sequence of ELM and σ is the kernel width. From (2), one can get the error of the kth output:

$$\begin{aligned} \mathbf{e} &= \mathbf{Y} - \mathbf{T} \\ &= [y_1 - t_1, y_2 - t_2, \ldots, y_N - t_N]^T \\ &= [e_1, e_2, \ldots, e_N]^T \end{aligned} \tag{16}$$

in which the error sample $e_i (i = 1, 2, \ldots, N)$ will be expressed as

$$\begin{aligned} e_i &= \sum_{q=1}^{\tilde{N}} g(\mathbf{w}_q \cdot \mathbf{x}_i + b_q)\beta_q - t_i \\ &= \mathbf{h}_i \boldsymbol{\beta} - t_i \end{aligned} \tag{17}$$

where $\mathbf{h}_i = g(\mathbf{W} \circ \mathbf{X}_i + \mathbf{b})$ is the row vector of $\mathbf{H}$ (the output matrix of the hidden layer). Here, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{\tilde{N}}]$, $\mathbf{X}_i = [\mathbf{x}_i, \mathbf{x}_i, \ldots, \mathbf{x}_i]$, $\mathbf{b} = [b_1, b_2, \ldots, b_{\tilde{N}}]$.

Theoretically, we try to make the error values as concentrated around zero as possible. For the desired error $e^{(d)}$, we can choose the δ function as the probability density function $p_{e^{(d)}}$, i.e.,

$$p_{e^{(d)}}(e) = \delta(e) \tag{18}$$

However, in practice, the above situation is difficult to operate. In real application, the estimated information divergence is used as an alternative cost function, in which the desired error distribution is also estimated by kernel method. We have the desired density function

$$\begin{aligned} p_{e^{(d)}}(e) &= \delta(e) * \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{e^2}{2\sigma^2}) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{e^2}{2\sigma^2}) \end{aligned} \tag{19}$$

The information divergence between e and $e^{(d)}$ can be written as

$$D(p_e \| p_{e^{(d)}}) = \int [p_e(e) - p_{e^{(d)}}(e)]^2 de. \tag{20}$$

TABLE 1. Parameters of algorithms in function fitting.

	ELM	RELM		ELM-RCC			ELM-MinID		
	$\tilde{N}$	$\tilde{N}$	λ	$\tilde{N}$	λ	σ	$\tilde{N}$	σ	η
$S\alpha S(1.5, 0.5)$	10	370	10^{-1}	30	10^{-1}	1	30	1	0.01
$S\alpha S(1.3, 1)$	10	70	10^{-1}	60	10^{-2}	0.1	50	0.4	0.01
Laplace	10	250	10^{2}	50	10^{-8}	1	30	1	0.05

A detailed mathematical deduction of (20) is given in Appendixes. At last, we have the function of information divergence (21). Substituting (17) into (21), we get function (22), as shown at the bottom of the page.

$$D(p_e \,\|p_{e^{(d)}}) = \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}[\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(e_j-e_i)^2}{2\sigma^2})] - 2\frac{1}{\sqrt{2\pi}\sigma N}\sum_{j=1}^{N}\exp(-\frac{e_j^2}{2\sigma^2})+\frac{1}{2\sqrt{\pi}\sigma} \quad (21)$$

One can update the parameter $\boldsymbol{\beta}$ by the following gradient algorithm:

$$\boldsymbol{\beta}(k+1) = \boldsymbol{\beta}(k) - \eta\frac{\partial D(p_e, p_{e^{(d)}})}{\partial \boldsymbol{\beta}} \quad (24)$$

where $\eta > 0$ is the step-size and $\boldsymbol{\beta}(k)$ denotes the parameter vector at iteration k.

Based on the above model optimization strategy, a robust learning algorithm for SLFNs under MinID can be obtained, which is referred to as the ELM-MinID and is described in Algorithm 1.

Algorithm 1 ELM-MinID

Input: training samples $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$, kernel widths σ, initialize the number of hidden units $\tilde{N}$, iteration step-size η, maximum iteration number K and termination tolerance ξ, the vector $\boldsymbol{\beta}(0) = 0$.

Output: weight vector $\boldsymbol{\beta}$.

1) Randomly initialize the weight vectors $\{\mathbf{w}_j\}_{j=1}^{\tilde{N}}$ together with their corresponding bias terms $\{b_j\}_{j=1}^{\tilde{N}}$.
2) Calculate the hidden layer output matrix $\mathbf{H}$.
3) Update the weight vectors $\boldsymbol{\beta}$.
   For $k = 1, 2 \ldots K$ **do**
   Compute the actual errors based on $\boldsymbol{\beta}(k-1)$: $e_i = \mathbf{h}_i\beta(k-1)_i - t_i, i = 1, 2, \ldots, N$
   Calculate the gradient of the information divergence: $\nabla D(p_e \,\|p_{e^{(d)}})$
   Update the bias term vector and the weight: $\boldsymbol{\beta}(k) = \boldsymbol{\beta}(k-1) - \eta\frac{\partial D(p_e, p_{e^{(d)}})}{\partial \boldsymbol{\beta}}$
   Until $\nabla D(p_e \,\|p_{e^{(d)}}) < \xi$
   EndFor

IV. EXPERIMENTAL RESULTS

In this part, we present experimental results to illustrate the performance of ELM-MinID proposed in the previous section. Parameters of all algorithms are chosen by grid-search method and cross validation method. In each independent trial, the training datasets and testing datasets are fixed. Average RMSE of 50 trials of simulations for each algorithm are obtained and then finally the performance obtained is

$$D(p_e \,\|p_{e^{(d)}}) = \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}[\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(\sum_{q=1}^{\tilde{N}} g(\mathbf{w}_q\cdot\mathbf{x}_j+b_q)\beta_q - t_j - \sum_{q=1}^{\tilde{N}} g(\mathbf{w}_q\cdot\mathbf{x}_i+b_q)\beta_q + t_i)^2}{2\sigma^2})] - 2\frac{1}{\sqrt{2\pi}\sigma N}\sum_{j=1}^{N}\exp(-\frac{(\sum_{q=1}^{\tilde{N}} g(\mathbf{w}_q\cdot\mathbf{x}_j+b_q)\beta_q - t_j)^2}{2\sigma^2}) + \frac{1}{2\sqrt{\pi}\sigma} \quad (22)$$

$$\begin{aligned}\nabla D(p_e \,\|p_{e^{(d)}}) &\triangleq \frac{\partial D(p_e \,\|p_{e^{(d)}})}{\partial \boldsymbol{\beta}} \\ &= \frac{1}{\sqrt{2\pi}\sigma N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\exp(-\frac{(e_j-e_i)^2}{2\sigma^2})\frac{(e_j-e_i)}{\sigma^2}(\frac{\partial e_i}{\partial \boldsymbol{\beta}} - \frac{\partial e_j}{\partial \boldsymbol{\beta}}) \\ &\quad - \frac{\sqrt{2}}{\sqrt{\pi}\sigma N}\sum_{j=1}^{N}[\exp(-\frac{e_j^2}{2\sigma^2}) - \frac{e_j}{\sigma^2}(\frac{\partial e_j}{\partial \boldsymbol{\beta}})] \\ &= \frac{1}{\sqrt{2\pi}\sigma N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\exp(-\frac{(e_j-e_i)^2}{2\sigma^2})\frac{(e_j-e_i)}{\sigma^2}(\mathbf{h}_i - \mathbf{h}_j) \\ &\quad - \frac{\sqrt{2}}{\sqrt{\pi}\sigma N}\sum_{j=1}^{N}[\exp(-\frac{e_j^2}{2\sigma^2}) - \frac{e_j}{\sigma^2}\mathbf{h}_j \end{aligned} \quad (23)$$

TABLE 2. Average testing RMSEs (*Sinc* date set).

	ELM	RELM	ELM-RCC	ELM-MinID
$S\alpha S(1.5, 0.5)$	0.0939±0.0064	0.0705±0.0001	0.0573±0.0008	**0.0436±0.0076**
$S\alpha S(1.3, 1)$	0.1745±0.0102	0.1270±0.0486	0.0605±0.0014	**0.0548±0.0021**
Laplace	0.0513±0.0022	0.0344±0.0001	0.0394±0.0003	**0.0327±0.0012**

TABLE 3. Paired t-test between the best performance and runner up (*Sinc* date set).

Datasets	Best algorithm	Runner-up algorithm	Paired t-test
$S\alpha S(1.5, 0.5)$	ELM-MinID	ELM-RCC	$t = -5.610, p = 0.000$
$S\alpha S(1.3, 1)$	ELM-MinID	ELM-RCC	$t = -10.307, p = 0.000$
Laplace	ELM-MinID	RELM	$t = -2.364, p = 0.042$

TABLE 4. Specification of the datasets.

Datasets	Features	Observations	
		Training	Testing
Concrete	9	515	515
Airfoil	5	751	751
Servo	5	83	83
Yacht	6	154	154
Slump	10	52	51
Housing	14	253	253
Wine-red	12	799	799
CCPP [1]	5	1000	1000
Fish Toxicity	7	253	253
Superconductivty	81	10631	10630
MITV[2]	9	20271	20271

[1] Combined Cycle Power Plant.
[2] Metro Interstate Traffic Volume.

TABLE 5. Parameters of algorithms in regression.

	KELM		RELM		ELM-RCC			ELM-MinID		
	γ	λ	$\tilde{N}$	λ	$\tilde{N}$	λ	σ	$\tilde{N}$	σ	η
Airfoil	2^{10}	10^{-3}	250	10^{-2}	210	10^{-1}	0.04	30	0.9	0.01
Concrete	2^{-1}	10^{2}	270	10^{5}	70	10^{2}	0.1	30	1	0.01
Servo	2^{3}	10^{5}	250	10^{-1}	50	10^{-5}	0.1	30	0.7	0.01
Yacht	2^{5}	10^{2}	190	10^{5}	100	10^{-4}	0.1	30	0.9	0.01
Slump	2^{0}	10^{-1}	50	10^{2}	160	10^{-1}	0.5	30	1	0.02
Housing	2^{-2}	10^{2}	320	10	200	10^{1}	0.02	30	0.9	0.01
Wine-red	2^{3}	10^{2}	110	10^{-2}	30	10^{4}	0.4	30	0.94	0.02
CCPP	2^{2}	10^{-1}	30	10^{-3}	220	10^{2}	0.7	30	1	0.02
Fish Toxicity	2^{0}	10^{5}	170	10^{0}	60	10^{3}	0.62	30	1	0.03
Superconductivty	2^{3}	10^{3}	190	10^{0}	60	10^{-3}	0.1	30	1	0.01
MITV	2^{6}	10^{-4}	30	10^{-1}	120	10^{-2}	0.1	30	0.4	0.05

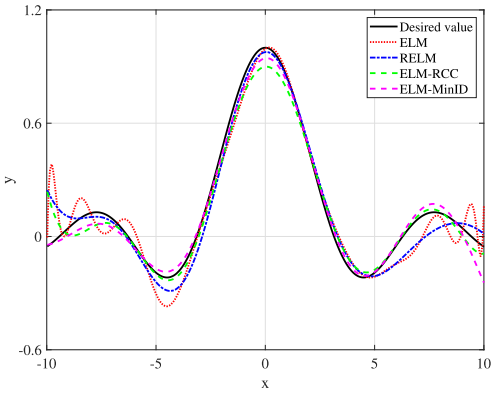
reported. All the experiments are carried out in the MATLAB R2018a environment running inInter(R) Xeon(R) E-2124G processor with the speed of 3.40GHz.

A. FUNCTION FITTING WITH SYNTHETIC DATASETS

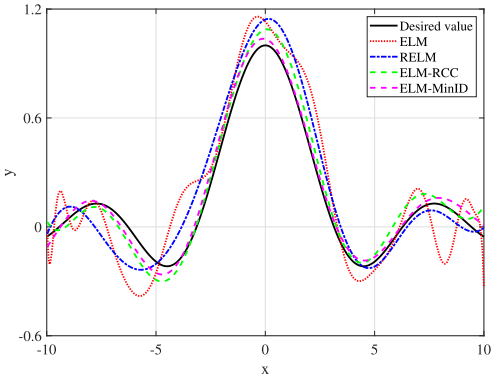
In this subsection, two synthetic datasets are utilized to validate the proposed algorithm. The description of them is as follows.

Sinc: The synthetic data set is produced by $y_i = sinc(x_i) + n$, where n denotes a noise and the *sinc* function is given as

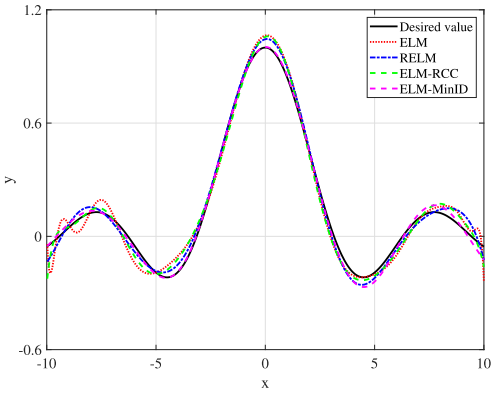
$$sinc(x) = \begin{cases} \sin(x)/x & x \neq 0 \\ 1 & x = 0. \end{cases} \tag{25}$$



(a)



(b)



(c)

FIGURE 1. Function fitting results of four algorithms upon *Sinc* with three noise: (a) α-stable noise ($\alpha' = 1.5$, $\tau = 0.5$); (b) α-stable noise ($\alpha' = 1.3$, $\tau = 1$); (c) Laplace noise (0, 0.5).

we generate 1000 data points with x_i drawn randomly from $[-10, 10]$.

Func: This artificial data set is generated by $(y_i, y_j) = func(x_i, x_j) + n$, where n is also a noise and the *func* function is given as

$$func(x_1, x_2) = x_1 \cdot e^{(-(x_1^2 + x_2^2))}. \tag{26}$$

1000 data points are constructed by randomly chosen from the evenly spaced 50×50 on $[-2, 2]$.

TABLE 6. RMSE and computing time of four algorithms (uncontaminated).

Datasets		KELM		RELM		ELM-RCC		ELM-MinID	
		Training	Testing	Training	Testing	Training	Testing	Training	Testing
Airfoil	RMSE	0.0519±0.0000	**0.0628±0.0000**	0.0792±0.0012	0.0893±0.0002	0.0969±0.0040	0.1138±0.0058	0.1281±0.0014	0.1293±0.0017
	TIME(sec)	0.0313	0.0291	0.0000	0.0000	0.1406	0.0000	2.6522	0.0033
Concrete	RMSE	0.0290±0.0000	**0.0360±0.0000**	0.0885±0.0001	0.0892±0.0000	0.0976±0.0173	0.0972±0.0191	0.0884±0.0009	0.0887±0.0018
	TIME(sec)	0.0108	0.0081	0.5156	0.0000	0.0781	0.0000	3.9110	0.0029
Servo	RMSE	0.0468±0.0000	0.0794±0.0000	0.0623±0.0023	0.0750±0.0020	0.0484±0.0004	**0.0747±0.0097**	0.0875±0.0008	0.0925±0.0000
	TIME(sec)	0.0005	0.0002	0.0000	0.0000	0.0625	0.0000	0.4291	0.0058
Yacht	RMSE	0.0739±0.0000	0.0870±0.0000	0.0816±0.0006	**0.0852±0.0009**	0.1028±0.0000	0.1177±0.0009	0.1001±0.0001	0.1143±0.0000
	TIME(sec)	0.0008	0.0004	0.0781	0.0000	0.0469	0.0000	1.6094	0.0023
Slump	RMSE	0.1425±0.0000	0.1642±0.0000	0.0004±0.0000	0.0518±0.0062	0.0001±0.0000	0.0648±0.0166	0.0559±0.0211	**0.0507±0.0193**
	TIME(sec)	0.0003	0.0001	0.0938	0.0000	0.0000	0.0000	0.6406	0.0039
Housing	RMSE	0.0490±0.0000	**0.0740±0.0000**	0.4406±0.0014	0.4203±0.0014	0.2845±0.1101	0.2710±0.1048	0.1106±0.0291	0.1193±0.0291
	TIME(sec)	0.0018	0.0016	0.0781	0.0000	0.0781	0.0000	3.2501	0.0076
Wine-red	RMSE	0.1092±0.0000	0.1147±0.0000	0.0944±0.0007	0.2249±0.0333	0.1371±0.0129	0.1628±0.0217	0.1268±0.0006	**0.0937±0.0011**
	TIME(sec)	0.0449	0.0192	0.0625	0.0000	0.2656	0.0000	3.9783	0.0040
CCPP	RMSE	0.0690±0.0000	0.0707±0.0000	0.4465±0.005	0.4465±0.0056	0.0658±0.0039	**0.0628± 0.0041**	0.0916±0.0472	0.0903±0.0473
	TIME(sec)	0.0448	0.0210	0.0781	0.0000	0.2813	0.0000	4.1025	0.0036
Fish Toxicity	RMSE	0.0358±0.0000	0.0999±0.0000	0.0731±0.0002	0.2072±0.0332	0.3255±0.0435	0.3313±0.0440	0.1018±0.0006	**0.0976±0.0009**
	TIME(sec)	0.0201	0.0105	0.0625	0.0000	0.1406	0.0000	15.3895	0.0098
Superconductivty	RMSE	0.1361±0.0000	0.1231±0.0000	0.0653±0.0069	0.0659±0.0064	0.0344±0.0015	0.0368±0.0016	0.0524±0.0019	**0.0301±0.0085**
	TIME(sec)	9.3195	2.8001	0.3906	0.2031	31.0313	0.0938	27.2688	0.0205
MITV	RMSE	0.2350±0.0000	**0.1307±0.0000**	4724±0.0077	0.4734±0.0077	0.3919±0.0.0506	0.3914±0.0511	0.2660±0.0054	0.2652±0.0088
	TIME(sec)	224.6461	113.4246	0.6875	0.3125	139.6563	0.0156	133.9968	0.0118

TABLE 7. RMSE and computing time of four algorithms (contamination rate 20%).

Datasets		KELM		RELM		ELM-RCC		ELM-MinID	
		Training	Testing	Training	Testing	Training	Testing	Training	Testing
Airfoil	RMSE	0.2761±0.0053	0.2341±0.0032	0.3752±0.0005	0.2521±0.0020	0.4857±0.0329	0.1814±0.0270	0.4497±0.0009	**0.1326±0.0026**
	TIME(sec)	0.0313	0.0231	0.0625	0.0000	0.1875	0.0000	3.7042	0.0025
Concrete	RMSE	0.1550±0.0040	0.1324±0.0019	0.2450±0.0003	0.1429±0.0003	0.2859±0.0012	0.1135±0.0019	0.2531±0.0005	**0.1052±0.0018**
	TIME(sec)	0.0101	0.0075	0.0469	0.0000	0.1250	0.0000	3.9400	0.0033
Servo	RMSE	0.2615±0.0147	0.2237±0.0108	0.2099±0.0004	0.2671±0.0154	0.4068±0.0006	0.2773±0.0097	0.2549±0.0011	**0.1475±0.0091**
	TIME(sec)	0.0005	0.0002	0.0938	0.0000	0.0781	0.0000	0.3912	00026
Yacht	RMSE	0.1694±0.0110	0.1284±0.0537	0.1782±0.0015	0.1388±0.0011	0.5870±0.0022	0.4742±0.0009	0.1953±0.0001	**0.1236±0.0000**
	TIME(sec)	0.0021	0.0.0008	0.0625	0.0000	0.0781	0.0000	1.7188	0.0017
Slump	RMSE	0.2987±0.0217	0.2524±0.0026	0.0018±0.0002	0.8130±0.0515	0.5934±0.4938	0.4228±0.6057	0.4347±0.0062	**0.0851±0.0237**
	TIME(sec)	0.0003	0.0.0001	0.0469	0.0000	0.0010	0.0000	0.6563	0.0047
Housing	RMSE	0.4252±0.0073	0.2576±0.0029	0.7308±0.0020	0.4155±0.0022	0.5671±0.0922	0.2778±0.1219	0.4641±0.0023	**0.1146±0.0053**
	TIME(sec)	0.0015	0.0007	0.0313	0.0000	0.0625	0.0000	3.1875	0.067
Wine-red	RMSE	0.3277±0.0039	0.2586±0.0032	0.3518±0.0020	0.6062±0.1139	0.8444±0.0025	0.5482±0.0002	0.4636±0.0006	**0.1363±0.0016**
	TIME(sec)	0.0489	0.0252	0.0938	0.0313	0.1875	0.0000	4.0017	0.0027
CCPP	RMSE	0.3149±0.0016	0.2578±0.0014	0.3928±0.0016	0.2260±0.0050	0.5544±0.2063	0.2716±0.3995	0.4425±0.0008	**0.0718±0.0030**
	TIME(sec)	0.0300	0.0206	0.1250	0.0000	0.3125	0.0000	4.2291	0.0022
Fish Toxicity	RMSE	0.3259±0.0037	0.2637±0.0037	0.3620±0.0006	0.8923±0.2308	0.6723±0.0371	0.3566±0.0444	0.4601±0.0005	**0.0978±0.0009**
	TIME(sec)	0.0178	0.0105	0.0000	0.0000	0.1250	0.0000	15.4412	0.0025
Superconductivty	RMSE	0.6423±0.0125	0.5094±0.0753	0.4858±0.0017	0.2353±0.0023	0.2483±0.0003	0.1948±0.0004	0.3122±0.0004	**0.1701±0.0013**
	TIME(sec)	8.9124	2.7918	0.4375	0.2031	31.0469	0.1875	27.9440	0.0781
MITV	RMSE	0.7685±0.0072	0.4828±0.0073	0.4801±0.0012	0.3310±0.0002	0.4133±0.0186	0.4052±0.0051	0.3748±0.0010	**0.3030±0.0008**
	TIME(sec)	186.0213	130.5328	0.8438	0.5000	131.4219	0.0158	120.8801	0.0111

For *Sinc* date set, we consider three long-tailed distributions of n: 1) symmetric $\alpha-stable(S\alpha S)$ distribution [18] with characteristic function $\phi(t) = \exp(-\tau|t|^{\alpha'})$, with shape parameter $\alpha' = 1.5$ and scale parameter $\tau = 0.5$; 2) $S\alpha S$ distribution with shape parameter $\alpha' = 1.3$ and scale parameter $\tau = 1$; 3) Laplace distribution with zero mean and variance 0.5. Similar Laplace noise is also added to the *Func* data set. In our simulations, 500 noisy data are used for training and another 500 clean data are used for testing. The activation function in this paper is the sigmoid function $f(x) = 1/(1 + e^{-x})$.

We contrast the performance of the proposed ELM-MinID with three existing ELMs including ELM, RELM and ELM-RCC [19]. In order to make a fair comparison, these algorithms are compared at their best fitting accuracy based on optimal parameter combination. Therefore, we need to predetermine these parameters: the number of hidden nodes $\tilde{N}$, the regularization parameters λ, the kernel width σ, and the step size η. In ELM optimization, the parameters are usually chosen by grid-search method and cross validation method, such as k-fold, as done by Inaba *et al.* [20], Kai and Luo [13], Huang *et al.* [21], Da Silva *et al.* [22] and others. Similarly, in this part, we obtain the best parameter combination by the grid search on each parameter and the five-fold cross-validation on every training set. We calculate the validation accuracy by using different parameter combinations of the hidden nodes number $\tilde{N} \in \{10, 20, \ldots, 400\}$, the regularization parameters $\lambda \in \{10^{-10}, 10^{-9}, \ldots, 10^{5}\}$, the kernel width $\sigma \in \{0.02, 0.04, \ldots, 1\}$, and the step size $\eta \in \{0.01, 0.02, \ldots, 0.1\}$. The maximum number of hidden nodes for ELMs is set to 400 because there are only 400 training data available (since we use a 5-fold cross validation on 500 training data) [23]. Additionally, in ELM-MinID, the termination tolerance ξ is 0.001 and the maximum iteration

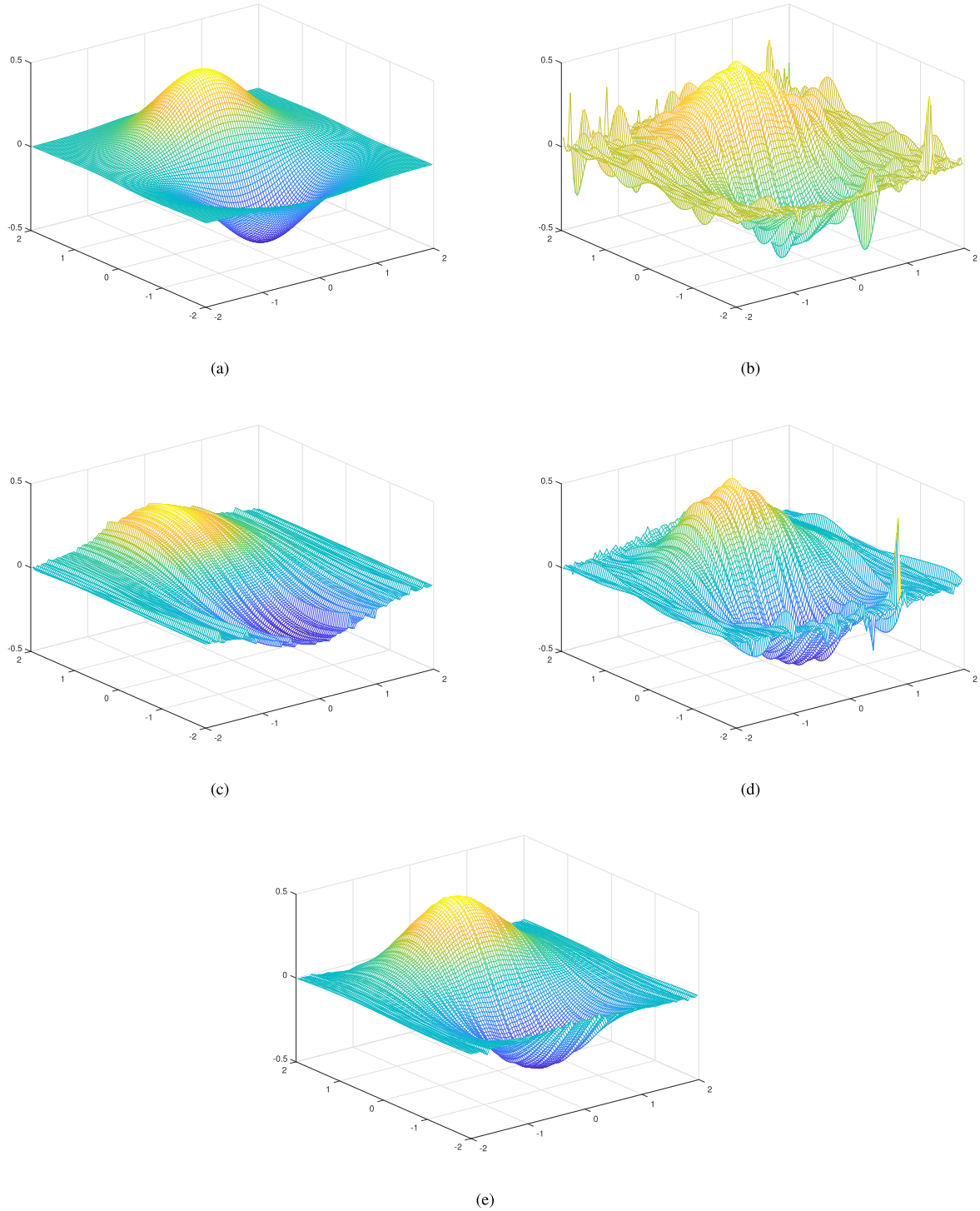


FIGURE 2. Function fitting results of four algorithms upon *func* with Laplace noise (0,0.5). (a) The original function. (b) The result of ELM. (c) The result of RELM. (d) The result of ELM-RCC. (e) The result of ELM-MinID.

number K is 300. The best parameters for each algorithm are chosen according to the validation accuracy and summarized in Table 1.

The experiments were run 50 times, using the parameters in Table 1. Fig.1 demonstrates the fitting results of the four algorithms upon *Sinc* with three different noises. Further, the average (and standard deviation) values of testing RMSEs are shown in Table 2, where the best result for each noise distribution are highlighted in bold. Table 3 is a statistical significance report between the best performance and runner-up using the paired T-test. From Table 3, $P < 0.05$, that is, there is a significant difference in the testing RMSEs between the two algorithms. This shows that ELM-MinID has a better fitting ability. Fig.2 is the fitting results of four algorithms

TABLE 8. RMSE and computing time of four algorithms (contamination rate 40%).

Datasets		KELM		RELM		ELM-RCC		ELM-MinID	
		Training	Testing	Training	Testing	Training	Testing	Training	Testing
Airfoil	RMSE	0.4087±0.0063	0.1791±0.0047	0.4669±0.0006	0.4131±0.0015	0.6747±0.0501	0.1953±0.0301	0.5987±0.0015	**0.1507±0.0007**
	TIME(sec)	0.0196	0.0150	0.0000	0.0000	0.1875	0.0000	3.0529	0.0024
Concrete	RMSE	0.2306±0.0014	0.1407±0.0031	0.2894±0.0001	0.2387±0.0001	0.3960±0.0028	0.1478±0.1039	0.2942±0.0009	**0.0848±0.0035**
	TIME(sec)	0.0130	0.0124	0.1094	0.0000	0.1250	0.0000	3.8506	0.0021
Servo	RMSE	0.3475±0.0147	0.2108±0.0131	0.2548±0.0004	0.3047±0.0120	0.5971±0.0004	0.6259±0.0097	0.2938±0.0005	**0.1940±0.0025**
	TIME(sec)	0.0005	0.0003	0.0469	0.0000	0.0000	0.0000	0.4062	0.0019
Yacht	RMSE	0.2268±0.0154	0.1916±0.0744	0.2137±0.0004	0.2042±0.0009	0.3774±0.0072	0.2265±0.0009	0.2249±0.0001	**0.1894±0.0004**
	TIME(sec)	0.0013	0.0020	0.0000	0.0000	0.0625	0.0000	1.7344	0.0000
Slump	RMSE	0.4947±0.0235	0.4052±0.0024	0.0015±0.0002	1.0764±0.0898	0.7368±0.2093	0.5576±0.4737	0.4482±0.0102	**0.1229±0.0252**
	TIME(sec)	0.0009	0.0003	0.0000	0.0000	0.0000	0.0000	0.6250	0.068
Housing	RMSE	0.5112±0.0091	0.4344±0.0042	0.9404±0.0024	0.4102±0.0026	0.7622±0.1276	0.3933±0.2286	0.6069±0.0036	**0.1309±0.0064**
	TIME(sec)	0.0024	0.0016	0.0000	0.0000	0.0625	0.0000	3.5313	0.0072
Wine-red	RMSE	0.4682±0.0048	0.1717±0.0038	0.4065±0.0022	0.8513±0.1352	1.0612±0.0021	0.5487±0.0002	0.6122±0.0008	**0.1477±0.0020**
	TIME(sec)	0.0493	0.0270	0.0781	0.0000	0.2031	0.0000	4.0245	0.0019
CCPP	RMSE	0.4744±0.0468	0.3895±0.0025	0.9159±0.0086	0.4150±0.0087	0.6851±0.0730	0.3370±0.0375	0.6076±0.0009	**0.0849±0.0029**
	TIME(sec)	0.0471	0.0265	0.0781	0.0000	0.3906	0.0000	4.3864	0.0044
Fish Toxicity	RMSE	0.4789±0.0052	0.1660±0.0028	0.4248±0.0009	0.9645±0.1469	0.0440±0.0218	0.3979±0.0243	0.6129±0.0015	**0.1099±0.0020**
	TIME(sec)	0.0333	0.0125	0.0625	0.0000	0.0625	0.0000	15.5788	0.0025
Superconductivty	RMSE	0.3241±0.0079	0.2058±0.0096	0.6540±0.0042	0.2431±0.0037	0.3817±0.0006	0.3232±0.0014	0.3732±0.0.0004	**0.1776±0.0090**
	TIME(sec)	8.7594	2.9319	0.5313	0.2188	31.5156	0.0938	27.5354	0.0098
MITV	RMSE	0.9810±0.0067	0.4896±0.0065	0.5571±0.0011	0.4781±0.0002	0.9027±0.0158	0.6535±0.0348	0.4830±0.0.0011	**0.2811±0.0003**
	TIME(sec)	237.8752	133.0665	1.0156	0.4531	134.8438	0.1875	119.6301	0.0090

TABLE 9. Paired t-test between the best performance and runner up (contamination rate 20%).

Datasets	Best algorithm	Runner-up algorithm	Paired t-test
Airfoil	ELM-MinID	ELM-RCC	$t = -17.247, p = 0.000$
Concrete	ELM-MinID	ELM-RCC	$t = -48.329, p = 0.000$
Servo	ELM-MinID	KELM	$t = -42.780, p = 0.000$
Yacht	ELM-MinID	KELM	$t = -5.058, p = 0.000$
Slump	ELM-MinID	KELM	$t = -50.192, p = 0.000$
Housing	ELM-MinID	KELM	$t = -163.782, p = 0.000$
Wine-red	ELM-MinID	KELM	$t = -27.154, p = 0.000$
CCPP	ELM-MinID	RELM	$t = -188.487, p = 0.000$
Fish Toxicity	ELM-MinID	KELM	$t = -83.486, p = 0.000$
Superconductivty	ELM-MinID	ELM-RCC	$t = -39.241, p = 0.043$
MITV	ELM-MinID	RELM	$t = -97.865, p = 0.000$

TABLE 10. Paired t-test between the best performance and runner up (contamination rate 40%).

Datasets	Best algorithm	Runner-up algorithm	Paired t-test
Airfoil	ELM-MinID	KELM	$t = -24.349, p = 0.000$
Concrete	ELM-MinID	KELM	$t = -90.631, p = 0.000$
Servo	ELM-MinID	KELM	$t = -1.044, p = 0.032$
Yacht	ELM-MinID	KELM	$t = -7.75, p = 0.017$
Slump	ELM-MinID	KELM	$t = -8.049, p = 0.000$
Housing	ELM-MinID	ELM-RCC	$t = -8.261, p = 0.000$
Wine-red	ELM-MinID	KELM	$t = -32.130, p = 0.000$
CCPP	ELM-MinID	ELM-RCC	$t = -4.362, p = 0.000$
Fish Toxicity	ELM-MinID	KELM	$t = -71.297, p = 0.000$
Superconductivty	ELM-MinID	KELM	$t = -33.978, p = 0.000$
MITV	ELM-MinID	RELM	$t = -69.256, p = 0.000$

upon *func* with Laplace noise (0,0.5). Clearly, the ELM-MinID is more robust than other algorithms under the same noises.

B. REGRESSION WITH BENCHMARK DATASETS

In the second experiment, eleven benchmark datasets from UCI machine learning repository [24] are utilized to confirm the better regression performance of the ELM-MinID compared with the KELM, RELM and ELM-RCC. The descriptions of the data sets are presented in Table 4. In order to illustrate the robustness of these algorithms, training samples with different contamination rates are generated. This is made by assigning the random values from [0, 1] to the target values of some training samples (all target values are normalized into [0, 1]).

The parameters of these algorithms are selected through grid search and five-fold cross-validation with the same parameter interval as those in section 4.1. In addition, in the KELM algorithm, the grid-search range of kernel parameter γ is $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$. The optimal parameters are summarized in Table 5, except that the iteration number K is preset to 300 and the termination tolerance ξ is fixed to 0.001.

The 50-run training and testing RMSEs are shown in Tables 6, 7, and 8, which are for uncontaminated data sets and contamination rates of 20% and 40%, respectively. The best simulation results were highlighted in bold. We can notice that when there is no contamination in the training data, all training methods can obtain similar results. When considering that 20%, 40% of each training sample is contaminated with outliers, KELM, RELM and ELM-RCC show worse regression performance than ELM-MinID. This is to be expected, because they use the ℓ_2 norm, which are not suitable to deal with the data sets with outliers. Unlike ℓ_2 norm, MinID criterion can capture the more characteristics of the error and reduce errors from many ways. Table 9 and 10 are statistical significance report between the best performance and runner-up for contamination rates of 20% and 40%, respectively. In those reports, $P < 0.05$, that is, there is a significant difference in the testing RMSEs between the two algorithms. According to the analysis above, we can draw conclusion that the proposed ELM-MinID has good robustness performance in benchmark datasets with outlier.

V. CONCLUSION

In this paper, we proposed a robust learning algorithm for single-hidden layer feedforward neural networks (SLFNs) called ELM under minimum information divergence criterion (ELM-MinID), which provides a new error control method

for ELM. The simulation results on function fitting with synthetic data and regression with benchmark data sets showed the superior noise tolerant capability and stable regression performance of the proposed method.

APPENDIXES

The information divergence between e and $e^{(d)}$ can be written as

$$\begin{aligned} D(p_e \,\|p_{e^{(d)}}) &= \int [p_e(e) - p_{e^{(d)}}(e)]^2 de \\ &= \int [p_e(e)]^2 de - 2\int p_e(e) p_{e^{(d)}}(e) de \\ &\quad + \int [p_{e^{(d)}}(e)]^2 de \\ &= A - 2B + C \end{aligned}$$

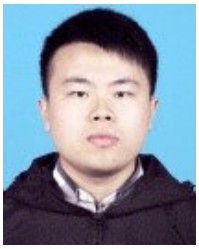
among them

$$\begin{aligned} A &= E_p[p_e(e)] \\ &= \frac{1}{N}\sum_{j=1}^{N} \hat{p}(e_j) \\ &= \frac{1}{N}\sum_{j=1}^{N} [\frac{1}{N}\sum_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(e_j - e_i)^2}{2\sigma^2})] \\ &= \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N} [\frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(e_j - e_i)^2}{2\sigma^2})] \\ B &= E_p[p_{e^{(d)}}(e)] \\ &= \frac{1}{N}\sum_{j=1}^{N} \hat{p}_{e^{(d)}}(e_j) \\ &= \frac{1}{N}\sum_{j=1}^{N} [\frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{{e_j}^2}{2\sigma^2})] \\ &= \frac{1}{\sqrt{2\pi}\sigma N}\sum_{j=1}^{N} \exp(-\frac{{e_j}^2}{2\sigma^2}) \\ C &= \int [p_{e^{(d)}}(e)]^2 de \\ &= \frac{1}{2\pi\sigma^2}\int \exp(-\frac{e^2}{\sigma^2}) de \\ &= \frac{1}{2\pi\sigma^2}\int \frac{\sqrt{2\pi}(\sigma/\sqrt{2})}{\sqrt{2\pi}(\sigma/\sqrt{2})} \exp(\frac{-e^2}{2\cdot(\sigma/\sqrt{2})^2}) de \\ &= \frac{1}{2\sqrt{\pi}\sigma}\int \frac{1}{\sqrt{2\pi}(\sigma/\sqrt{2})} \exp(\frac{-e^2}{2\cdot(\sigma/\sqrt{2})^2}) de \\ &= \frac{1}{2\sqrt{\pi}\sigma}\int \frac{1}{\sqrt{2\pi}} \exp(\frac{-\frac{e^2}{(\sigma/\sqrt{2})^2}}{2}) d(\frac{e}{\sigma/\sqrt{2}}) \\ &\xrightarrow[t=\frac{e}{\sigma/\sqrt{2}}]{e\in(-\infty,+\infty)} \frac{1}{2\sqrt{\pi}\sigma}\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp(\frac{-t^2}{2}) d(t) \\ &= \frac{1}{2\sqrt{\pi}\sigma} \end{aligned}$$

REFERENCES

[1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.
[2] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.
[3] A. J. Mayne, "Generalized inverse of matrices and its applications," *Technometrics*, vol. 15, no. 1, p. 197, 1972.
[4] A. Sorjamaa, Y. Miche, R. Weiss, and A. Lendasse, "Long-term prediction of time series using NNE-based projection and OP-ELM," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 2674–2680.
[5] J. Zhao, Z. Wang, and D. S. Park, "Online sequential extreme learning machine with forgetting mechanism," *Neurocomputing*, vol. 87, pp. 79–89, Jun. 2012.
[6] W. Jun, W. Shitong, and F.-L. Chung, "Positive and negative fuzzy rule system, extreme learning machine and image classification," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 4, pp. 261–271, 2011.
[7] B. P. Chacko, V. R. Vimal Krishnan, G. Raju, and P. Babu Anto, "Handwritten character recognition using wavelet energy and extreme learning machine," *Int. J. Mach. Learn. Cybern.*, vol. 3, no. 2, pp. 149–161, Jun. 2012.
[8] W. Zong and G.-B. Huang, "Face recognition based on extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2541–2551, Sep. 2011.
[9] J. Deng, K. Li, and G. W. Irwin, "Fast automatic two-stage nonlinear model identification based on the extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2422–2429, Sep. 2011.
[10] Q.-Y. Zhu, A. K. Qin, P. N. Suganthan, and G.-B. Huang, "Evolutionary extreme learning machine," *Pattern Recognit.*, vol. 38, no. 10, pp. 1759–1763, Oct. 2005.
[11] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
[12] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *Proc. IEEE Symp. Comput. Intell. Data Mining*, Mar. 2009, pp. 389–395.
[13] K. Zhang and M. Luo, "Outlier-robust extreme learning machine for regression problems," *Neurocomputing*, vol. 151, pp. 1519–1527, Mar. 2015.
[14] B. Chen, Y. Zhu, J. Hu, and Z. Sun, "Adaptive filtering under minimum information divergence criterion," *Int. J. Control, Autom. Syst.*, vol. 7, no. 2, pp. 157–164, Apr. 2009.
[15] J. Hu, B. Chen, F. Sun, and Z. Sun, "Adaptive filtering for desired error distribution under minimum information divergence criterion," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1215–1219.
[16] B. Chen, Y. Zhu, J. Hu, and J. C. Principe, *System Parameter Identification: Information Criteria and Algorithms*. London, U.K.: Newnes, 2013.
[17] B. D. Spurr and B. W. Silverman, "Density estimation for statistics and data analysis," *Biometrics*, vol. 44, no. 3, p. 914, 1988.
[18] W. Kramer, "Signal processing with alpha-stable distributions and applications: C.L. Nikias and Min Shoa (1995): Wiley, ISBN 0-471-10647-x, $ 50.00, pp. 168," *Comput. Statist. Data Anal.*, vol. 22, no. 3, p. 334, 1996.
[19] H.-J. Xing and X.-M. Wang, "Training extreme learning machine via regularized correntropy criterion," *Neural Comput. Appl.*, vol. 23, nos. 7–8, pp. 1977–1986, Dec. 2013.
[20] F. K. Inaba, E. O. T. Salles, S. Perron, and G. Caporossi, "DGR-ELM–distributed generalized regularized ELM for classification," *Neurocomputing*, vol. 275, pp. 1522–1530, Jan. 2018.
[21] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2004, pp. 985–990.
[22] B. L. S. D. Silva, F. K. Inaba, E. O. T. Salles, and P. M. Ciarelli, "Outlier robust extreme machine learning for multi-target regression," *Expert Syst. Appl.*, vol. 140, 2020, Art. no. 112877. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417419305871, doi: 10.1016/j.eswa.2019.112877.
[23] R. Zhang, G.-B. Huang, N. Sundararajan, and P. Saratchandran, "Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 4, no. 3, pp. 485–495, Jul. 2007.
[24] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

CHENGTIAN SONG received the B.S., M.S., Ph.D. degrees from the Beijing Institute of Technology, in 1999, 2002, and 2010, respectively, where he is currently an Associate Professor. His main research interests include intelligent target detection and recognition, and automatic test systems.



LIZHI PAN is currently pursuing the master's degree with the School of Mechatronical Engineering, Beijing Institute of Technology. His current research interests include extreme learning machine and signal processing.

QIANG LIU is currently an Engineer with the Ordnance Science Institute of China. His main research interests include intelligent target detection and recognition, and automatic test systems.

ZHIHONG JIANG (Member, IEEE) received the B.S. degree in material processing/manufacturing engineering from the Jilin University of Technology, Changchun, China, in 1998, the M.S. degree in material processing/manufacturing engineering from Jilin University, Changchun, in 2001, and the Ph.D. degree in electrical engineering and automation from Tsinghua University, Beijing, China, in 2005. He is currently a Professor with the School of Mechatronic Engineering, Beijing Institute of Technology, Beijing. His research interests include space intelligent robotics, industry robot systems, artificial intelligence and robot vision, and human–machine interaction.

JIANGUANG JIA was born in Inner Mongolia, China. He is currently an Associate Researcher of the Institute of Systems Engineering, Academy of Military Sciences, PLA. His main research interests include experimental identification technology and communication technology.