Knowledge-Based Systems 280 (2023) 111033

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Mixture extreme learning machine algorithm for robust regression

Shangrui Zhao [a], Xuan-Ang Chen [a], Jinran Wu [b,*], You-Gan Wang [b]

[a] School of Science, Wuhan University of Technology, Wuhan 430070, PR China
[b] Institute for Learning Sciences & Teacher Education, Australian Catholic University, Brisbane 4001, Australia

ARTICLE INFO

Keywords:
Extreme learning machine
The expectation maximization algorithm
The iteratively reweighted least squares algorithm
Mixture distribution
Prediction

ABSTRACT

The extreme learning machine (ELM) is a well-known approach for training single hidden layer feedforward neural networks (SLFNs) in machine learning. However, ELM is most effective when used for regression on datasets with simple Gaussian distributed error because it often employs a squared loss in its objective function. In contrast, real-world data is often collected from unpredictable and diverse contexts, which may contain complex noise that cannot be characterized by a single distribution. To address this challenge, we propose a robust mixture ELM algorithm, called Mixture-ELM, that enhances modeling capability and resilience to both Gaussian and non-Gaussian noise. The Mixture-ELM algorithm uses an adjusted objective function that blends Gaussian and Laplacian distributions to approximate any continuous distribution and match the noise. The Gaussian mixture accurately models the residual distribution, while the inclusion of the Laplacian distribution addresses the limitations of the Gaussian distribution in identifying outliers. We derive a solution to the novel objective function using the expectation maximization (EM) and iteratively reweighted least squares (IRLS) algorithms. We evaluate the effectiveness of the algorithm through numerical simulation and experiments on benchmark datasets, thereby demonstrating its superiority over other state-of-the-art machine learning methods in terms of robustness and generalization.

1. Introduction

The advancement and refinement of regression techniques have long been a persistent area of investigation. One such technique is the extreme learning machine (ELM), which is commonly incorporated in regression models [1]. Owing to its rapid learning speed, exceptional generalization performance, and other benefits, ELM models have been extensively studied in both theoretical and practical domains [2,3]. Nevertheless, the efficacy of ELM predictions is heavily contingent upon the purity of the training data [4]. Hence, researchers are continuously exploring the applicability of ELM in uncertain scenarios by employing various strategies, such as identifying and eliminating outliers, modifying objective functions, etc.

As the noise in real-world data is complex and always unknown, it cannot be well characterized by any one distribution. Moreover, in the mixture of Gaussian distributions (MoG), the noise data is assumed to be normal, and the Gaussian distribution is sensitive to outliers. Then, we construct the mixture of Gaussian and Laplace distributions as MoG, this may be more reasonable for data prediction with outliers. As is well known, there is an absolute term in the probability density function of the Laplace distribution and a square term in the Gaussian distribution. Thus, the cost function of ELM will be a l_1-norm loss function with a l_2-norm term in it. Experiments in Zhang et al. [5] reveal that when the data contain numerous outliers, methods with the l_1-norm loss function perform better than those with the l_2-norm. This implies that the l_1-norm loss will increase the generalizability and robustness of the ELM models. If the mixture of Gaussian and Laplace distributions is used, then the obtained objective function should have the properties of both the l_1-norm and l_2-norm loss functions, thereby making the model more robust.

In light of these concepts, a robust ELM, called the Mixture-ELM, is proposed in this study to enhance the modeling capacity and resilience of ELM when handling data with complex noise. A new objective function is created to map the features between input and output by using a mixture of Gaussian and Laplace distributions. This is different from existing ELMs, which minimize the output weights and modeling errors under the assumption that noise follows a Gaussian distribution. To estimate the parameters in the proposed Mixture-ELM, the expectation maximization (EM) approach is also used. The following characteristics can be used to summarize the significant contributions of this study:

(1) In this paper, the Mixture-ELM algorithm is proposed, and a new loss function is constructed, including a mixture of Gaussian and

* Corresponding author.
E-mail addresses: zhaosr@whut.edu.cn (S. Zhao), 312687@whut.edu.cn (X.-A. Chen), ryan.wu@acu.edu.au (J. Wu), you-gan.wang@acu.edu.au (Y.-G. Wang).

https://doi.org/10.1016/j.knosys.2023.111033
Received 14 April 2023; Received in revised form 20 September 2023; Accepted 22 September 2023
Available online 26 September 2023

Laplace distributions. The component of Gaussian distributions is used to deal with the normal data and the component of Laplace distributions is used to deal with the abnormal data.

(2) In Mixture-ELM, the EM algorithm is applied to identify the samples adaptively, and different loss functions are applied to the samples in different clusters to train the model.

(3) Numerical simulation and real-world public datasets are used to test the proposed Mixture-ELM approach. The EM is used to accurately estimate the parameters of the noise distributions accurately.

The remainder of this paper is organized in the following manner: Section 2 introduces the related literature. Section 3 outlines the traditional ELM and EM algorithms and presents the proposed Mixture-ELM. This section includes the modified objective function of Mixture-ELM and the corresponding solving process. Section 4 presents the numerical simulation for data with different settings. The section also presents experimental results and further analysis on selected benchmark datasets. Section 5 concludes the paper.

2. Related literature

Over the last decade, ELM has been extensively researched in theory and applications [6] as a generalized single hidden layer feedforward networks (SLFNs). Unlike most typical gradient-based SLFNs training approaches, the hidden layer parameters of ELM are randomly assigned without iterative adjustment [7]. Consequently, ELM has a fast learning rate and is simple to apply. In theory, Huang et al. [8] established the universal approximation of ELM. Moreover, ELM has been used in residual learning [9,10], online learning [11,12], structural optimization [13], ensemble learning [14,15], imbalance learning [16,17], etc. Because of the ease of implementation and great generalization [18], ELM is extensively utilized in real-world applications, such as load forecasting [19], image categorization [20], business management [21], automatic language identification [22], and COVID-19 detection.

In most of the current extreme learning machines, the data used for modeling is assumed to be pure and without noise and outliers, or solely with Gaussian error. However, sampling problems, measurement flaws, and modeling errors may cause the noise to follow an unknown distribution, thereby making data uncertainty unavoidable in real applications. In practical applications, the noises are more complex and may contain Gaussian distribution, Laplace distribution, or mixed distributions. Additionally, a data-driven predictor's effectiveness suffers greatly from cluttered or very noisy data [23]. Consequently, ELMs that do not account for the effects of uncertainty may be insufficient.

Therefore, people always search for ways to improve the modeling capability of ELM in uncertain circumstances. Based on the structural risk minimization principle and the weighted least square method, a new regularized ELM algorithm, weighted regularized ELM (WRELM) [24], is proposed. Without increasing the training time, the generalization performance of the algorithm is significantly improved in most cases. For example, FIR-ELM [25], was designed to reduce input disturbance by deleting certain undesirable signal components by FIR filtering. He et al. [26] created a hierarchical ELM for dealing with high-dimensional noisy data, in which several subnet groups were proposed for concurrently reducing data dimension and filtering noise. However, these improved ELMs based on outlier detection may mistake pure data for outliers, thereby causing the original data structure to be broken and information to be lost. Because of the presence of this flaw, people increase the model's resilience in another manner—by modifying the objective function. For example, a robust loss function is introduced to ELM instead of the l_1-norm loss, and a least-trimmed square estimator is examined for robust regression instead of the least square technique [27]. Furthermore, a robust ELM algorithm, known as outlier-robust ELM (ORELM) [28]—based on the sparsity characteristic of the outlier is proposed. The l_1-norm loss function is introduced to enhance the robustness and the fast and accurate augmenting Lagrange multiplier method is used to ensure the effectiveness and efficiency of the algorithm. It not only maintains the advantages of ELM but also has significant and stable accuracy when dealing with abnormal data.

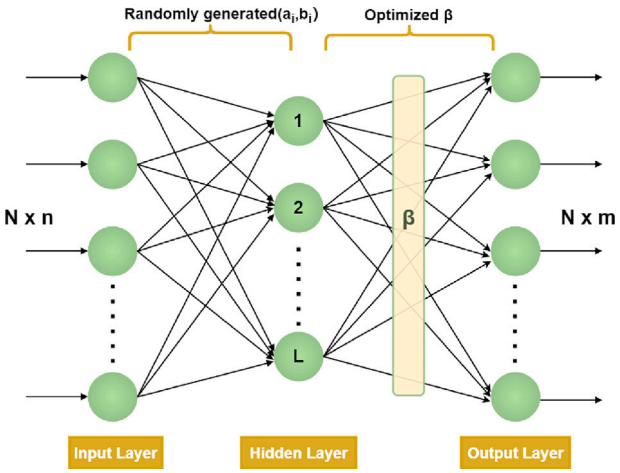


Fig. 1. ELM model.

Further, second-order cone programming, which is extensively used in robust convex optimization problems, was specifically integrated into ELM [29]. Probabilistic regularized ELM (PRELM) [30], considering the modeling error distribution, constructed a new objective function to minimize the mean and variance of the modeling error. It has good modeling performance for non-Gaussian noise or outlier problems.

Although ELMs with different goal functions can produce good results, this cannot be ensured when the noise follows non-Gaussian distributions. R-ELM was proposed by Zhang et al. [5] to improve modeling performance and robustness with Gaussian and non-Gaussian noise. A modified objective function was built in R-ELM to fit the noise using a mixture of Gaussian distributions (MoG) to approximate any continuous distribution. The tests on the selected benchmark datasets and real-world applications revealed that the proposed R-ELM outperforms state-of-the-art machine learning techniques in terms of resilience and generalization. Some algorithms are listed in Table 1.

However, in real-world applications, samples frequently contain varying degrees of outliers, whereas noises are constantly susceptible to uncertain statistical distributions. Furthermore, because the Gaussian distribution is particularly sensitive to outliers, employing MoG to match noise cannot yield a decent prediction impact for datasets with outliers. The present techniques are unable to extract the noise, especially when the data contains a large number of outliers of varying degrees.

3. Proposed method

In this section, we provide a brief introduction to the classical ELM and EM algorithms.

3.1. ELM theory

This section begins with a brief overview of the extreme learning machine (ELM) theory. An ELM is a machine learning algorithm used to train the single-layer feedforward neural network (SLFN). It was proposed to train SLFNs with only one hidden layer and the structure is the same as other neural network algorithms, as in Fig. 1. All parameters are tuned only once with this method. Iterative training is not required for the algorithm.

Table 1
Relative work.

Algorithm	Abbreviation	Year	Limitations	Source
Weighted regularized ELM	WRELM	2009	Modeling performance is heavily dependent on the accuracy of weight estimation, especially in complex real-world cases	[24]
ELM with FIR Filtering	FIR-ELM	2011	The number of iterations to reach convergence is too many, and the calculation is too large and might be trapped by local minima.	[25]
Hierarchical ELM	HELM	2013	The improved ELM based on outlier detection may mistake pure data for outliers, causing the original data structure to be broken and information lost.	[26]
Outlier-robust ELM	ORELM	2014	It creates a heavy computational cost, which often causes these models to be less effective.	[28]
ELM with second-order cone programming	SR-RELM	2016	The computational burden of the novel ELM is relatively heavy	[29]
Probabilistic regularized ELM	PRELM	2018	Its objective function still retains the square loss function (l_2-norm). However, it has been pointed out that the l_2-norm is prone to be badly affected by outliers. Typically, the ELM model with l_2-norm loss function tends to be unstable in the presence of outliers.	[30]
ELM with a mixture of Gaussian	R-ELM	2020	A mixture of Gaussian is used to fit the noise, but in the real environment, the noise follows a variety of distributions, which may lead to poor noise reduction effect. And its objective function is still a linear combination of l_2-norm loss function, which is prone to be badly disturbed by outliers.	[5]
ELM with robust loss function	PELM	2022	The hyper-parameters and penalty parameter are obtained by cross-validation method, which can only obtain a result attached to the optimal value and require much computing resources	[27]

The training sample set $x = (x_1, \ldots, x_N)$ serves as the input of the neural network in the diagram, from left to right, and there is a hidden layer in the center with n nodes. $H(x)$ is the hidden layer output, and there is a complete link between the input and the hidden layer. The output $H(x)$ can be rewritten as follows:

$$H(x) = [h_1(x), \ldots, h_n(x)]. \tag{1}$$

The output is obtained by the input multiplied by the corresponding weight and appropriate bias through a nonlinear function at each node. Moreover $h_i(x)$ is the output of the ith hidden layer node which can be treated as a nonlinear function. Different nonlinear functions can be employed for various hidden layer neurons. In real applications, $h_i(x)$ used to be represented as follows:

$$h_i(x) = g(w_i, b_i, x) = g(w_i x + b_i), w_i \in \mathrm{R}^n, b_i \in \mathrm{R}, i = 1, \ldots, n, \tag{2}$$

where w_i and b_i are the weight and bias vectors on the ith node of the hidden layer, respectively. Moreover, $g(\cdot)$ is the activation function, which fulfills the universal approximation ability theorem of the ELM. It is a nonlinear continuous function and often employed functions including the Sigmoid function and the Gaussian function, etc. According to the above diagram and formula, after passing through the hidden layer and entering the output layer, the output of ELM is:

$$f_n(x) = \sum_{i}^{n} \beta_i h_i(x) = H(x)\beta, \tag{3}$$

where $\beta = [\beta_1, \ldots, \beta_n]^T$ is the weight between n hidden layer nodes and m output layer nodes. The preceding equation can be represented as the following compact matrix:

$$\mathbf{H}\beta = \mathbf{Y}, \tag{4}$$

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

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(x_1) \\ \vdots \\ \mathbf{h}(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \cdots & h_n(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_n(x_N) \end{bmatrix}, \tag{5}$$

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

$$\mathbf{Y} = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix} = \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & \ddots & \vdots \\ y_{N1} & \cdots & y_{Nm} \end{bmatrix}. \tag{6}$$

ELM basically divides the training process for SLFN into two stages: (1) generating random feature mappings and (2) identifying linear parameters. The hidden layer parameters are initially set at random. Thereafter, ELM transfers the input data to a new feature space by using some nonlinear mapping as an activation function. A nonlinear piecewise continuous function can be used as the nonlinear mapping function in ELM and the weights and biases at the hidden layer nodes of ELM are generated at random. The hidden layer node parameters w and b are not determined by training but are randomly produced based on a continuous probability distribution.

After the first stage, w and b are randomly generated and identified; thus, the output of the hidden layer can be calculated by Eq. (2). In the second stage of ELM learning, we only need to find the weight β of the output layer.

The objective function of ELM is to minimize both the norm of the output weight β and the modeling error and it can be written as the following expression:

$$\begin{aligned} &\text{Min}: \tfrac{1}{2}\|\beta\|^2 + \tfrac{1}{2} C \textstyle\sum_{i=1}^{n} \xi_i^2 \\ &\text{s.t. } \mathbf{h}(x_i)\beta = y_i - \xi_i, i = 1, \ldots, n, \end{aligned} \tag{7}$$

where ξ_i is the model noise, and C is a regularization coefficient for improving generalization performance. For this classic optimal problem, we can quickly obtain the following solution:

$$\tilde{\beta} = \begin{cases} \mathbf{H}^T(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{Y}, n < N \\ (\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Y}, n > N \end{cases}, \tag{8}$$

where $\tilde{\beta}$ is the estimated value of β, and $\mathbf{I}$ is the identity matrix.

3.2. EM algorithm

The expectation step (E-Step) and maximization step (M-Step) are the two basic concepts of the EM algorithm. The E-Step primarily estimates the parameters by observing the data and the existing model and then uses the estimated parameters to calculate the expected value of the likelihood function. The M-Step is to identify the corresponding parameters when the likelihood function is maximized. The likelihood function will eventually converge because the procedure ensures the likelihood function will grow with each iteration. The EM algorithm

can effectively obtain the information of latent variables only according to Cappé and Moulines [31]. Therefore, the EM algorithm is frequently employed in application procedures pertaining to data mining and machine learning [32].

For a given dataset, we can assume that the samples are independent. The likelihood function is:

$$\begin{aligned} L(\theta) &= \sum_{i=1}^{n} \log P(x_i|\theta) \\ &= \sum_{i=1}^{n} \log \sum_{z} P(x_i, z|\theta), \end{aligned} \tag{9}$$

where n is the number of samples taken, θ is the parameter of samples' distribution, $P(x_i|\theta)$ is the probability of sampling x_i, and z is a latent variable, denoting the potential category to which each sample might belong. Let $Q_i(z)$ represent the distribution of the latent variable z corresponding to sample x_i; thus $Q_i(z)$ satisfies the condition $\sum_z Q_i(z) = 1, Q_i(z) \geq 0$. Then, the likelihood function is modified as follows:

$$\begin{aligned} L(\theta) &= \sum_{i}^{n} \log \sum_{z} P(x_i, z|\theta) \\ &= \sum_{i}^{n} \log \sum_{z} Q_i(z) \frac{P(x_i, z; \theta)}{Q_i(z)} \\ &\geq \sum_{i}^{n} \sum_{z} Q_i(z) \log \frac{P(x_i, z|\theta)}{Q_i(z)}, \end{aligned} \tag{10}$$

where the inequality in Eq. (10) is given by Jensen's inequality. Jensen's inequality gives that X is a random variable; then, the equal sign holds if and only if $X = E[X]$—that is X is a constant. Thus, according to Jensen's inequality, when the equal sign holds, we have:

$$\frac{P(x_i, z \mid \theta)}{Q_i(z)} = c. \tag{11}$$

Multiply both sides of the equation with $Q_i(z)$ and add up with respect to latent variables z, then:

$$\sum_{z} P(x_i, z \mid \theta) = c \sum_{z} Q_i(z). \tag{12}$$

Since $\sum_z Q_i(z) = 1$, we have:

$$\sum_{z} P(x_i, z \mid \theta) = c, \tag{13}$$

and

$$Q_i(z) = \frac{P(x_i, z \mid \theta)}{c} = \frac{P(x_i, z \mid \theta)}{\sum_z P(x_i, z \mid \theta)} = \frac{P(x_i, z \mid \theta)}{P(x_i \mid \theta)} = P(z \mid x_i, \theta). \tag{14}$$

From Eq. (10), it is obvious that when Eq. (14) holds—that is $Q_i(z) = P(z \mid x_i, \theta)$, a lower bound of the logarithmic likelihood containing latent data is given and the logarithmic likelihood equals the lower boundary. Maximizing this lower boundary also implies maximizing the logarithmic likelihood—that is, to solve the following problem:

$$\begin{aligned} L(\hat{\theta}) = \max_{\theta} \sum_{i=1}^{n} \sum_{z} Q_i(z) \log \frac{P(x_i, z \mid \theta)}{Q_i(z)} = \max_{\theta} \sum_{i=1}^{n} \sum_{z} Q_i(z) \\ \times (\log P(x_i, z \mid \theta) - log Q_i(z)). \end{aligned} \tag{15}$$

The EM algorithm to solve Eq. (15) involves first selecting the distribution $Q_i(z; \theta_0)$ for a given θ_0; this is called the E-Step. The next Step involves obtaining $\hat{\theta}$ by maximizing $L(\theta)$ with given $Q_i(z|\theta_0)$; this is called M-Step. In M-step, problem Eq. (15) is equivalent to:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \sum_{z} Q_i(z; \theta_0) \log P(x_i, z \mid \theta). \tag{16}$$

Thus, the EM algorithm flow can be summed up as follows:

Step 1: Initialize the value of the model parameter θ_0.

Step 2: **E-Step** in $k+1$th iteration: calculate the $Q-$function, $Q_i(z; \theta^{(k)})$, by:

$$Q_i(z; \theta^{(k)}) = P(z \mid x_i, \theta^{(k)}). \tag{17}$$

Step 3: **M-Step**: calculate $\theta^{(k+1)}$ by:

$$\theta^{(k+1)} = \arg\max_{\theta} \sum_{i=1}^{n} \sum_{z} Q_i(z; \theta^{(k)}) \log P(x_i, z \mid \theta). \tag{18}$$

Step 4: Repeat **E-Step** and **M-Step** until $\theta^{(k)}$ converges.

3.3. The mixture of Gaussian and Laplace distributions

As known, a prediction model can be described as:

$$y = f(x) + \xi,$$

where $f(x)$ is the developed model and ξ is the noise. In accordance with the ELM theory, the prediction model should be modified as:

$$\mathbf{H}\boldsymbol{\beta} + \xi = \mathbf{Y}. \tag{19}$$

For a given training set, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N} \subset R^n \times R^m$, assume that there are N corresponding noises. Among the N noises, a few follow the Gaussian distributions and a few follow the Laplace distributions.

When the noise $\xi_i \sim \mathcal{N}(0, \sigma^2)$, the probability density function is:

$$P(\xi_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\xi_i^2}{2\sigma^2}). \tag{20}$$

Similarly, when $\xi_j \sim \mathcal{L}(0, \lambda)$, the probability density function is:

$$P(\xi_j) = \frac{1}{2\lambda} \exp(-\frac{|\xi_j|}{\lambda}). \tag{21}$$

Suppose that there are Q_1 Gaussian distributions and Q_2 Laplacian distributions that comprise the data $\{\xi_i\}$, then the probability distribution is:

$$P(\xi) = \sum_{j=1}^{Q_1} \pi_j^{\mathcal{N}} \mathcal{N}(\xi \mid 0, \sigma_j^2) + \sum_{k=1}^{Q_2} \pi_k^{\mathcal{L}} \mathcal{L}(\xi \mid 0, \lambda_k), \tag{22}$$

where $\pi_j^{\mathcal{N}}$ and $\pi_k^{\mathcal{L}}$ are the weighting coefficients, and $\sum_{j=1}^{Q_1} \pi_j^{\mathcal{N}} + \sum_{k=1}^{Q_2} \pi_k^{\mathcal{L}} = 1$. $\mathcal{N}(\xi \mid 0, \sigma_j^2)$ denotes the Gaussian distribution with expectation zero and variance σ_j^2, and $\mathcal{L}(\xi \mid 0, \lambda_k)$ denotes the Laplace distribution with mean zero and parameter λ_k.

The likelihood function of ξ can be expressed as

$$P(\xi \mid \theta) = \prod_{i=1}^{N} P(\xi_i \mid \theta) = \prod_{i=1}^{N} (\sum_{j=1}^{Q_1} \pi_j^{\mathcal{N}} \mathcal{N}(\xi_i \mid 0, \sigma_j^2) + \sum_{k=1}^{Q_2} \pi_k^{\mathcal{L}} \mathcal{L}(\xi_i \mid 0, \lambda_k)), \tag{23}$$

where $\theta = (\pi_1^{\mathcal{N}}, \ldots, \pi_{Q_1}^{\mathcal{N}}, \pi_1^{\mathcal{L}}, \ldots, \pi_{Q_2}^{\mathcal{L}}, \sigma_1^2, \ldots, \sigma_{Q_1}^2, \lambda_1, \ldots, \lambda_{Q_2}, \boldsymbol{\beta})$ denotes the set of parameters that need to be estimated. The corresponding log style is:

$$\begin{aligned} \log P(\xi \mid \theta) = \sum_{i=1}^{N} \log \sum_{j=1}^{Q_1} (\pi_j^{\mathcal{N}} (\frac{1}{\sqrt{2\pi}\sigma_j} \exp(-\frac{\xi_i^2}{2\sigma_j^2}))) \\ + \sum_{i=1}^{N} \log \sum_{k=1}^{Q_2} (\pi_k^{\mathcal{L}} (\frac{1}{2\pi\lambda_k} \exp(-\frac{|\xi_i|}{2\lambda_k}))). \end{aligned} \tag{24}$$

The expectation maximization (EM) approach is used to estimate the parameter θ since the analytical solution cannot be derived directly.

Let us consider each ξ_i and the corresponding indicator $\boldsymbol{\omega_i} = (\omega_{i1}^{\mathcal{N}}, \ldots, \omega_{iQ_1}^{\mathcal{N}}, \omega_{i1}^{\mathcal{L}}, \ldots, \omega_{iQ_2}^{\mathcal{L}})$. If ξ_i follows the jth Gaussian (or Laplace) distribution, set $\omega_{ij}^{\mathcal{N}} = 1$ ($\omega_{ij}^{\mathcal{L}} = 1$) and the rest of $\boldsymbol{\omega_i}$ to 0. We can formulate it as:

$$\omega_{ij}^{\mathcal{N}} = \begin{cases} 1, \xi_i \sim \mathcal{N}(0, \sigma_j^2) \\ 0, else \end{cases}$$

and

$$\omega_{ik}^{\mathcal{L}} = \begin{cases} 1, \xi_i \sim \mathcal{L}(0, \lambda_k) \\ 0, else \end{cases}.$$

Thus, $\sum_{j=1}^{Q_1}\omega_{ij}^{\mathcal{N}}+\sum_{k=1}^{Q_2}\omega_{ik}^{\mathcal{L}}=1$, and $\sum_{i=1}^{N}(\sum_{j=1}^{Q_1}\omega_{ij}^{\mathcal{N}}+\sum_{k=1}^{Q_2}\omega_{ik}^{\mathcal{L}})=N$.

In the E-step of the EM algorithm, based on ξ and the estimated $\theta^{(t)}$ after the tth iteration, we obtain $Q-$function:

$$\begin{aligned}Q(\theta;\theta^{(t)})=&\sum_{i=1}^{N}(\sum_{j=1}^{Q_1}\gamma_{ij}^{\mathcal{N}}(\log\pi_j^{\mathcal{N}}-\frac{1}{2}\log(2\pi\sigma_j^2)-\frac{\xi_i^2}{2\sigma_j^2})\\&+\sum_{k=1}^{Q_2}\gamma_{ij}^{\mathcal{L}}(\log\pi_j^{\mathcal{L}}-\log(2\lambda_k)-\frac{|\xi_i|}{\lambda_k})),\end{aligned}\tag{25}$$

where $\gamma_{ij}^{\mathcal{N}}$ is the posterior responsibility of the ith observation ξ_i following the jth Gaussian distribution and $\gamma_{ik}^{\mathcal{L}}$ is that of ξ_i following the kth Laplace distribution. Thus, it is obvious that:

$$\begin{aligned}\gamma_{ij}^{\mathcal{N}}=E(\omega_{ij}^{\mathcal{N}})&=\frac{\pi_j^{\mathcal{N},(t)}\mathcal{N}(\xi_i\mid 0,(\sigma_j^{(t)})^2)}{\sum_{j=1}^{Q_1}\pi_j^{\mathcal{N},(t)}\mathcal{N}(\xi_i\mid 0,(\sigma_j^{(t)})^2)+\sum_{k=1}^{Q_2}\pi_k^{\mathcal{L},(t)}\mathcal{L}(\xi_i\mid 0,\gamma_k^{(t)})},\\ \gamma_{ij}^{\mathcal{L}}=E(\omega_{ij}^{\mathcal{L}})&=\frac{\pi_k^{\mathcal{L},(t)}\mathcal{L}(\xi_i\mid 0,\gamma_k^{(t)})}{\sum_{j=1}^{Q_1}\pi_j^{\mathcal{N},(t)}\mathcal{N}(\xi_i\mid 0,(\sigma_j^{(t)})^2)+\sum_{k=1}^{Q_2}\pi_k^{\mathcal{L},(t)}\mathcal{L}(\xi_i\mid 0,\gamma_k^{(t)})}.\end{aligned}\tag{26}$$

In the M-Step, the parameter $\theta^{(t+1)}$ is:

$$\theta^{(t+1)}=\arg\max_{\theta}Q(\theta;\theta^{(t)}),\tag{27}$$

which can be obtained by calculating the partial derivatives of Eq. (25). It should be emphasized that the sum of coefficients equals 1—that is, $\sum_{j=1}^{Q_1}\pi_j^{\mathcal{N}}+\sum_{k=1}^{Q_2}\pi_k^{\mathcal{L}}=1$. The solution of $\pi_j^{\mathcal{N}}$, $\pi_k^{\mathcal{L}}$, σ_j^2, λ_k is:

$$\begin{aligned}\pi_j^{\mathcal{N},(t+1)}&=\frac{\sum_{i=1}^{N}\gamma_{ij}^{\mathcal{N}}}{N},\\ \pi_k^{\mathcal{L},(t+1)}&=\frac{\sum_{i=1}^{N}\gamma_{ik}^{\mathcal{L}}}{N},\\ (\sigma_j^{(t+1)})^2&=\frac{\sum_i^N\gamma_{ij}^{\mathcal{N}}\xi_i^2}{\sum_i^N\gamma_{ij}^{\mathcal{N}}},\\ \lambda_k^{(t+1)}&=\frac{\sum_i^N\gamma_{ik}^{\mathcal{L}}|\xi_i|}{\sum_i^N\gamma_{ik}^{\mathcal{L}}}.\end{aligned}\tag{28}$$

3.4. The proposed Mixture-ELM

Recall the ELM theory, and take the noise of Eq. (19) to the $Q-$function. Then, it can be implied that:

$$\max_{\theta}Q(\theta;\theta^{t})=\max_{\theta}f_{\beta},$$

where

$$\begin{aligned}f_\beta=&-\sum_{i=1}^{N}\sum_{j=1}^{Q_1}\gamma_{ij}^{\mathcal{N}}\frac{\xi_i^2}{2\sigma_j^2}-\sum_{i=1}^{N}\sum_{k=1}^{Q_2}\gamma_{ik}^{\mathcal{L}}\frac{|\xi_i|}{2\lambda_k}\\=&-\sum_{i=1}^{N}(\sum_{j=1}^{Q_1}\frac{\gamma_{ij}^{\mathcal{N}}}{2\sigma_j^2})(\boldsymbol{y_i}-\boldsymbol{h(x_i)\beta})^2-\sum_{i=1}^{N}(\sum_{k=1}^{Q_2}\frac{\gamma_{ik}^{\mathcal{L}}}{2\lambda_k})\left|\boldsymbol{y_i}-\boldsymbol{h(x_i)\beta}\right|.\end{aligned}\tag{29}$$

Then, its dual optimization problem is:

$$\begin{aligned}&\min\tfrac{1}{2}\|\boldsymbol{\beta}\|_2^2+\textstyle\sum_{i=1}^{N}(\sum_{j=1}^{Q_1}\frac{\gamma_{ij}^{\mathcal{N}}}{2\sigma_j^2})(\boldsymbol{y_i}-\boldsymbol{h(x_i)\beta})^2+\sum_{i=1}^{N}(\sum_{k=1}^{Q_2}\frac{\gamma_{ik}^{\mathcal{L}}}{2\lambda_k})\left|\boldsymbol{y_i}-\boldsymbol{h(x_i)\beta}\right|\\&\text{s.t. }\boldsymbol{H\beta}=\boldsymbol{Y}-\boldsymbol{\xi}.\end{aligned}\tag{30}$$

This dual problem cannot be analytically resolved because of the l_1-norm and l_2-norm. However, Yang et al. [27] offers a general solution to this kind of issue with the following form of objective functions:

$$C\hat{\sigma}^2\sum_{i=1}^{n}\Psi(\frac{\xi_i}{\hat{\sigma}})+\frac{1}{2}\sum_{j=1}^{l}\beta_j^2.\tag{31}$$

Similarly to the noise scale estimation using M-estimation in Wang et al. [33], the scale is computed as:

$$\hat{\sigma}=\frac{\text{median}\left|\xi_i-\text{median}(\xi)\right|}{0.6745}\tag{32}$$

where $\xi_i=y_i-h(x_i)\boldsymbol{\beta}, i=1,\dots,N$. The equation is minimized using the iteratively weighted least squares (IRLS) approach from Holland and Welsch [34]. Eq. (31) is similar to the dual problem presented by KKT's theorem as follows:

$$\begin{aligned}\Gamma_n(\beta,\xi,\alpha)=&C\hat{\sigma}^2\sum_{i=1}^{n}\Psi(\frac{\xi_i}{\hat{\sigma}})+\frac{1}{2}\sum_{j=1}^{l}\beta_j^2\\=&\sum_{i=1}^{n}\alpha_i(h(x_i)\beta-y_i+\xi_i).\end{aligned}\tag{33}$$

Set the derivatives of Eq. (33) to 0 in accordance with the IRLS algorithm:

$$\begin{cases}\frac{\partial\Gamma_n}{\partial\beta}\to\beta=\sum_{i=1}^{n}\alpha_i h(x_i)^T\\ \frac{\partial\Gamma_n}{\partial x_i}\to\alpha_i=C\hat{\sigma}^2\frac{\partial\Psi(\xi_i/\hat{\sigma})}{\partial\xi_i}=Cw(\frac{\xi_i}{\hat{\sigma}})\xi_i,\\ \frac{\partial\Gamma_n}{\partial\alpha_i}\to h(x_i)\beta-y_i+\xi_i=0\end{cases}$$

where the weight function for the matching loss function is $w(\xi)=(1/\xi)(\partial\Psi(\xi)/\partial\xi)$. Then, the output weights β can be solved as follows:

$$\boldsymbol{\beta}=\begin{cases}(\frac{I}{C}+H^TW_nH)^{-1}H^TW_nY, & n\geqslant l\\ H^T(\frac{I}{C}+W_nHH^T)^{-1}W_nY, & n<l\end{cases},\tag{34}$$

where sample weight $W_n=\text{diag}(w(\frac{\xi_1}{\hat{\sigma}}),\dots,\omega(\frac{\xi_n}{\hat{\sigma}}))$ and I is the identity matrix.

By this approach, we can rewrite our objective function Eq. (30) as follows:

$$\min C\hat{\sigma}^2\sum_{i=1}^{N}(\sum_{j=1}^{Q_1}\frac{\gamma_{ij}^{\mathcal{N}}}{2\sigma_j^2}\Psi_2(\frac{\xi_i}{\hat{\sigma}})+\frac{1}{2|\hat{\sigma}|}\sum_{k=1}^{Q_2}\frac{\gamma_{ik}^{\mathcal{L}}}{2\lambda_k}\Psi_1(\frac{\xi_i}{\hat{\sigma}}))+\frac{1}{2}\sum_{i=1}^{N}\beta_i^2.\tag{35}$$

Now, the $\Psi(\frac{\xi_i}{\hat{\sigma}})$ in Eq. (31) can be represented as a linear combination of $\Psi_1(\frac{\xi_i}{\hat{\sigma}})$ and $\Psi_2(\frac{\xi_i}{\hat{\sigma}})$:

$$\Psi(\frac{\xi_i}{\hat{\sigma}})=\sum_{j=1}^{Q_1}\frac{\gamma_{ij}^{\mathcal{N}}}{2\sigma_j^2}\Psi_2(\frac{\xi_i}{\hat{\sigma}})+\frac{1}{2|\hat{\sigma}|}\sum_{k=1}^{Q_2}\frac{\gamma_{ik}^{\mathcal{L}}}{2\lambda_k}\Psi_1(\frac{\xi_i}{\hat{\sigma}}),\tag{36}$$

where the loss function $\Psi_2(\frac{\xi_i}{\hat{\sigma}})$ represents the l_2-norm loss, $\Psi_2(\frac{\xi_i}{\hat{\sigma}})=\frac{1}{2}(\frac{\xi_i}{\hat{\sigma}})^2$. The gradient of Ψ_2 is $\psi_2=\frac{\xi_i}{\hat{\sigma}}$, and the weight function is $w(\frac{\xi_i}{\hat{\sigma}})=\frac{\psi_2(\frac{\xi_i}{\hat{\sigma}})}{\frac{\xi_i}{\hat{\sigma}}}=1$. Similarly, the loss function $\Psi_1(\frac{\xi_i}{\hat{\sigma}})$ represents the l_1-norm loss, $\Psi_1(\frac{\xi_i}{\hat{\sigma}})=|\frac{\xi_i}{\hat{\sigma}}|$. The gradient is $\psi_1=sign(\frac{\xi_i}{\hat{\sigma}})$, and the weight function is $w(\frac{\xi_i}{\hat{\sigma}})=1/\max(|\frac{\xi_i}{\hat{\sigma}}|,\epsilon)$, where $\epsilon=10^{-6}$ is a small value. Now, the sample weight $W_n=\text{diag}(w(\frac{\xi_1}{\hat{\sigma}}),\dots,\omega(\frac{\xi_n}{\hat{\sigma}}))$ in Eq. (34) can be written as follows:

$$W_n=\begin{pmatrix}\mu_{11}+\mu_{21}\frac{1}{\max(|\frac{\xi_1}{\hat{\sigma}}|,\epsilon)} & 0 & \dots & 0\\ 0 & \mu_{12}+\mu_{22}\frac{1}{\max(|\frac{\xi_2}{\hat{\sigma}}|,\epsilon)} & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & \mu_{1N}+\mu_{2N}\frac{1}{\max(|\frac{\xi_N}{\hat{\sigma}}|,\epsilon)}\end{pmatrix},$$

where:

$$\mu_{1i}=\sum_{j=1}^{Q_1}\frac{\gamma_{ij}^{\mathcal{N}}}{2\sigma_j^2},\tag{37}$$

and

$$\mu_{2i}=\frac{1}{2|\hat{\sigma}|}\sum_{k=1}^{Q_2}\frac{\gamma_{ik}^{\mathcal{L}}}{2\lambda_k}.\tag{38}$$

Finally, the output weights β can be solved through Eq. (34).

The algorithm flow of finding the weight β corresponding to the maximum value of the objective function is given below:

Step 1: Initialize the original ELM without regularization term to obtain a rough initial residual ξ.
Step 2: Calculate the scale $\hat{\sigma}$ by Eq. (32).
Step 3: Update β by Eq. (34).
Step 4: Update $\xi = Y - H^T\beta$, and $W_n(i,i) = \omega(\xi_i/\hat{\sigma})$.
Step 5: Repeat Step 2 to Step 4 until β converges.

Through this flow, we can update β and the residuals ξ iteratively, and obtain the solution to the problem of Eq. (30).

Remark 1. There are many parameters in the Mixture-ELM framework, such as the number of hidden node layers L, the initial parameters required in the EM algorithm like $\sigma^{(0)}$, $\lambda^{(0)}$, $\pi^{\mathcal{N},(0)}$, $\pi^{\mathcal{L},(0)}$ and the regularization coefficient C. The following contents detail the setting of these parameters. In ELM, using too few nodes in the hidden layer will result in underfitting. Conversely, using too many neurons can also lead to overfitting. Since the EM algorithm has local convergence, if the initial parameters are not selected appropriately, the algorithm will fall into the local optimum. Therefore, the EM algorithm is extremely sensitive to initial parameters, and appropriate initial parameter selection is a necessary prerequisite for good regression results. On the other side, the value of C controls the deviation between the structural and empirical risk terms. A large C may result in minimal errors between predictions and actual observations, but can promote overfitting and produce models of great complexity. A small C efficiently reduces the difficulty of training the model and prevents overfitting. Consequently, selecting an acceptable C to make a trade-off between prediction inaccuracy and model complexity is critical. Lastly, the cross-validation method is used to establish the parameters of all machine learning algorithms.

4. Experimental setup and discussion of results

4.1. Numerical simulation

In this section, we will verify the effectiveness of the proposed Mixture-ELM by comparing it with other robust ELM algorithms, such as WRELM [24], PRELM [30], ORELM [28], and R-ELM [5].

A classical nonlinear function approximation problem "SinC" is chosen as the regression function. In the numerical simulation, we will perform regression on data with different types and degrees of noise, respectively, and use different evaluation criteria to measure the degree of regression. Note that all models select the sigmoid function $g(x) = 1/(1+\exp(-x))$ as the activation function. Parameters in WRELM, PRELM, and ORELM are selected as the recommendations in references of Martínez-Martínez et al. [24], Zhang and Luo [28] and Lou et al. [30], and the parameters of R-ELM and Mixture-ELM are selected by cross-validation. All experiments were run on the AMD3600 processor and compiled using PyCharm in the Python 3.10 environment.

4.1.1. Evaluation criteria

Several evaluation criteria were used to evaluate these ELMs with these robust ELM algorithms. Eqs. (39)–(41) present the formulas for the adopted evaluation criteria.

(1) Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mu_i - \hat{y}_i)^2}. \tag{39}$$

The root mean square error is the square root of the ratio between the projected value's squared departure from the real value and the number of observations n. It calculates the difference between the expected and true values and is sensitive to data that contains outliers.

(2) Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\mu_i - \hat{y}_i\right|. \tag{40}$$

Another frequent assessment criterion in regression issues is the MAE. It is frequently used to calculate the difference between forecasts and actual observations. Compared with MSE, MAE is less sensitive to outliers. Because MAE calculates the absolute value of the error, the penalty is fixed for any difference of any size.

(3) Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100\%}{\text{n}}\sum_{\text{i}=1}^{\text{n}}\left|\frac{\hat{\text{y}}_\text{i} - \text{y}_\text{i}}{\text{y}_\text{i}}\right|. \tag{41}$$

MAPE is often used to calculate the difference between expected and actual observed values. Because MAPE is more unbiased and equitable in comparison to the raw data, it is frequently employed as an assessment metric in algorithmic competitions.

In specific experiments, the smaller the RMSE or MAE value is, the better the forecasting model is. In most cases, RMSE and MAE have similar trends, but they measure the degree of regression in different meanings.

4.1.2. Experimental settings

We evaluate the performance using the following "SinC" function:

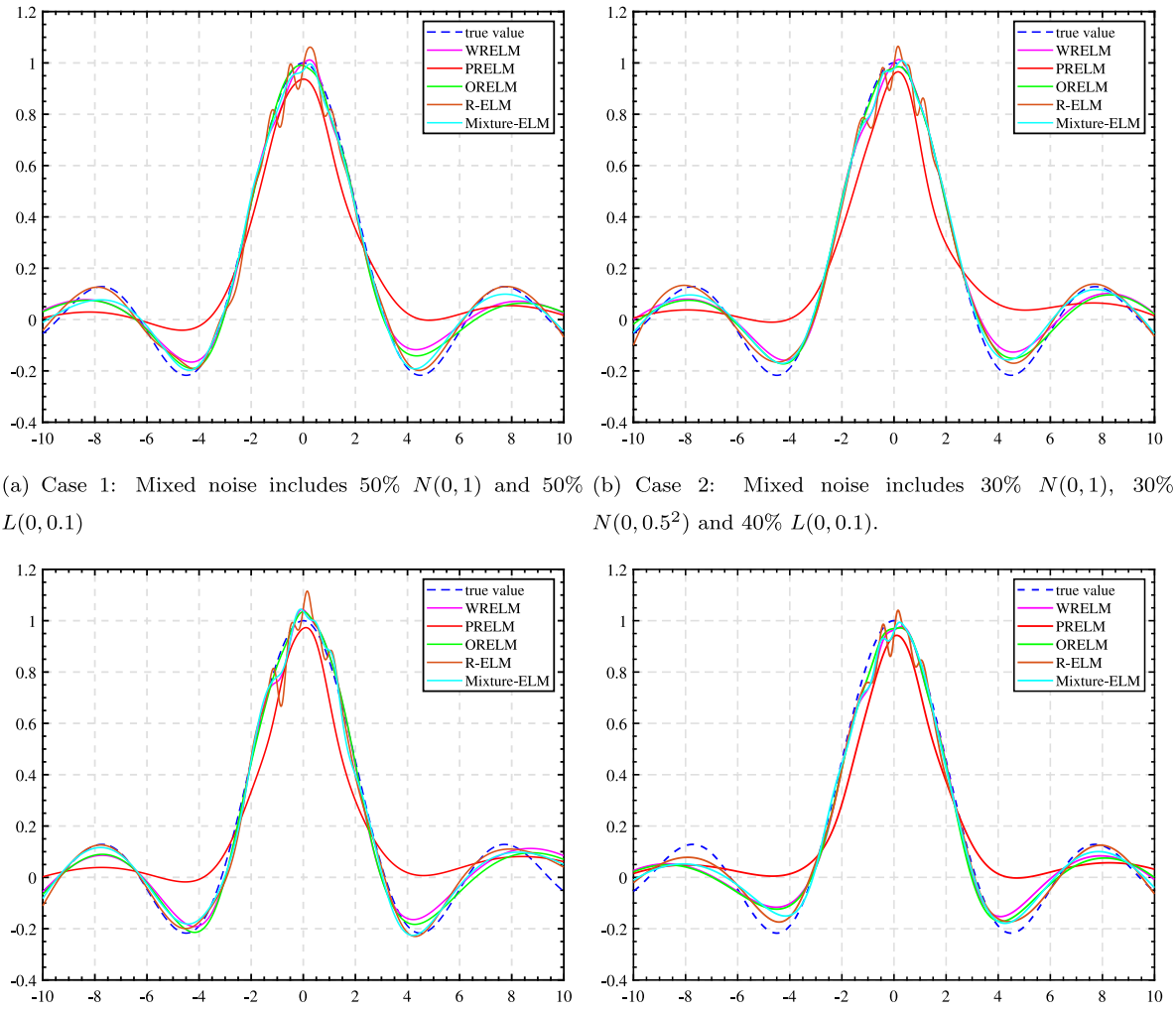
$$y(x) = \begin{cases} \sin(x)/x, & x \neq 0 \\ 1, & x = 0 \end{cases}. \tag{42}$$

Following the experimental settings in Martínez-Martínez et al. [24], Lou et al. [30], Zhang and Luo [28] and Zhang et al. [5], the number of training samples and test samples is 5000, and they are uniformly and randomly distributed in the interval $(-10, 10)$ respectively. The following four types of noise are added to the training sample for training the model, while the test sample is kept noiseless to check the experimental effect. In addition, since the hidden layer parameters are randomly generated in the first stage of the ELM algorithm, each experiment is repeated 50 times to prove the average performance of the model.

- Case 1. Mixed noise: noise includes 50% noise following Gaussian distribution $N(0,1)$, and 50% noise following Laplace distribution $L(0,0.1)$.
- Case 2. Mixed noise: noise includes 30% noise following Gaussian distribution $N(0,1)$, 30% noise following Gaussian distribution $N(0,0.5^2)$, and 40% noise following Laplace distribution $L(0,0.1)$.
- Case 3. Mixed noise: noise includes 40% following Gaussian distribution $N(0,1)$, 30% following Laplace distribution $L(0,0.5)$, and 30% following Laplace distribution $L(0,0.1)$.
- Case 4. Mixed noise: noise includes 30% following Gaussian distribution $N(0,1)$, 30% following Laplace distribution $N(0,0.5^2)$, 30% following Laplace distribution $L(0,0.1)$, and 10% following Laplace distribution $L(0,0.5)$.

4.1.3. Simulation results and analysis

Fig. 2 show the performance of five different algorithms on training data contaminated with different types of noise. The average fitting effect of different algorithms under the three valuation criteria is PRELM > WRELM > R-ELM > ORELM > Mixture-ELM. In a few cases, the results of ORELM and the Mixture-ELM are rather close—such as in the test set of Case 4, the MAE of the former is 3.66, and the latter is 3.61. Moreover, in the test set of Case 2, the RMSE of the former is 5.31 and that of the latter is 5.23. However, the stability of the Mixture-ELM results is much stronger than that of ORELM, and the corresponding standard deviations of the results are all smaller than that of ORELM. Based on the stability of the result – that is, the standard deviation



(a) Case 1: Mixed noise includes 50% $N(0,1)$ and 50% $L(0,0.1)$

(b) Case 2: Mixed noise includes 30% $N(0,1)$, 30% $N(0,0.5^2)$ and 40% $L(0,0.1)$.

(c) Case 3: Mixed noise includes 40% $N(0,1)$, 30% $L(0,0.5)$ and 30% $L(0,0.1)$.

(d) Case 4: Mixed noise includes 30% $N(0,1)$, 30% $N(0,0.5^2)$, 30% $L(0,0.1)$, and 10% $L(0,0.5)$.

Fig. 2. Comparison results of 'SinC' with different kinds of noise.

Table 2
Comparison of testing MAE by WRELM, PRELM, ORELM, R-ELM, and Mixture-ELM.

Algorithms	MAE ($\times 10^{-2}$) ± standard deviation ($\times 10^{-2}$)							
	Case 1		Case 2		Case 3		Case 4	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
WRELM	5.23 ± 0.91	5.26 ± 0.94	5.25 ± 0.74	5.30 ± 0.78	5.13 ± 1.26	5.20 ± 1.31	4.80 ± 0.87	4.86 ± 0.89
PRELM	31.78 ± 0.32	12.13 ± 0.70	12.33 ± 0.77	12.50 ± 0.74	12.19 ± 0.93	12.35 ± 0.89	12.23 ± 0.78	12.40 ± 0.76
ORELM	3.66 ± 1.14	3.71 ± 1.19	4.23 ± 0.84	4.29 ± 0.89	4.54 ± 1.43	4.51 ± 1.49	3.61 ± 1.03	3.66 ± 1.06
R-ELM	4.47 ± 0.43	4.47 ± 0.41	4.24 ± 0.48	4.23 ± 0.47	4.53 ± 0.60	4.50 ± 0.59	3.99 ± 0.37	3.97 ± 0.36
Mixture-ELM	3.51 ± 0.91	3.52 ± 0.94	4.05 ± 0.71	4.07 ± 0.73	3.92 ± 0.89	3.95 ± 0.92	3.60 ± 0.80	3.61 ± 0.80

of the result – ORELM > WRELM > Mixture-ELM > R-ELM > PRELM under MAE and RMSE, OREML > R-ELM > Mixture-ELM > WRELM > PRELM under MAPE. Despite the average performance of the Mixture-ELM in terms of stability, its average effect is substantially lower than that of algorithms with greater stability. To summarize, the Mixture-ELM algorithm achieves the best performance among these algorithms. Detailed comparisons of different methods with different additional noise are provided in Tables 2–4.

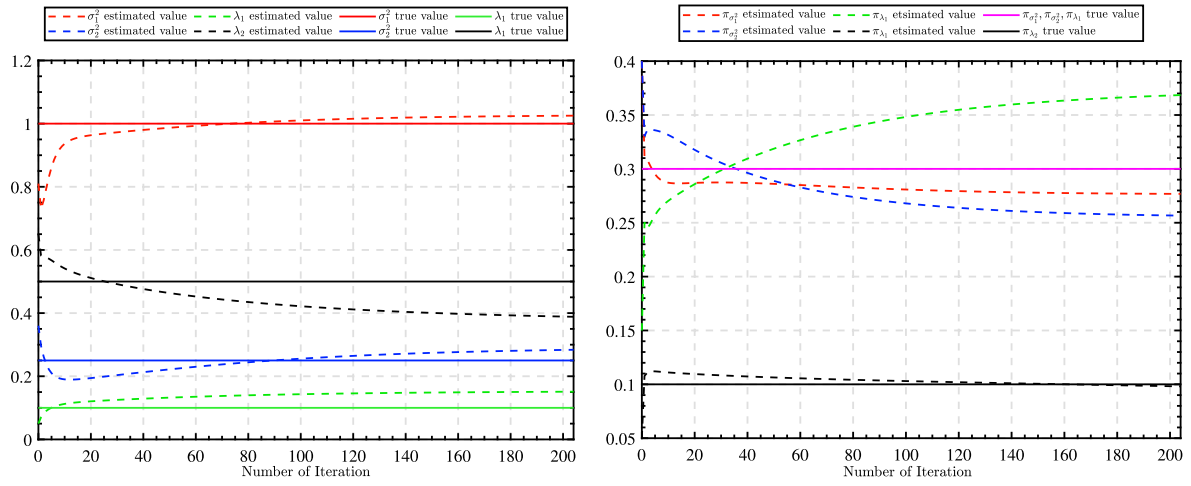
The above results reveal that although the noise distribution is more complex in all cases, our model can still extract the noise to achieve good results. The reason for this is that each noise distribution parameter can be precisely identified by the Mixture-ELM algorithm that we proposed. For example, in Case 4, the preset noise parameters are $\sigma_1 = 1, \sigma_2 = 0.5^2, \lambda_1 = 0.1, \lambda_2 = 0.5, \pi_{\sigma_1} = 0.3, \pi_{\sigma_2} = 0.3, \pi_{\lambda_1} = 0.3$, and $\pi_{\lambda_2} = 0.1$. The noise distribution parameters obtained by the Mixture-ELM algorithm in the experiment are $\sigma_1 = 1.016, \sigma_2 = 0.284, \lambda_1 =$

Table 3
Comparison of testing RMSE by WRELM, PRELM, ORELM, R-ELM, and Mixture-ELM.

Algorithms	RMSE ($\times 10^{-2}$) ± standard deviation ($\times 10^{-2}$)							
	Case 1		Case 2		Case 3		Case 4	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
WRELM	6.53 ± 1.03	6.58 ± 1.09	6.77 ± 0.83	6.82 ± 0.89	6.73 ± 1.39	6.73 ± 1.44	6.21 ± 1.04	6.29 ± 1.07
PRELM	43.95 ± 0.47	14.70 ± 1.03	14.85 ± 1.95	15.02 ± 1.01	14.71 ± 1.36	14.89 ± 1.32	14.77 ± 1.08	14.95 ± 1.05
ORELM	4.71 ± 1.46	4.75 ± 1.51	5.25 ± 1.04	5.31 ± 1.10	6.14 ± 1.77	6.08 ± 1.82	4.71 ± 1.33	4.77 ± 1.36
R-ELM	6.15 ± 0.45	6.16 ± 0.45	6.22 ± 0.41	6.26 ± 0.40	6.96 ± 0.53	6.92 ± 0.54	5.36 ± 0.37	5.36 ± 0.36
Mixture-ELM	4.25 ± 1.09	4.26 ± 1.13	5.21 ± 0.78	5.23 ± 0.81	5.10 ± 0.90	5.12 ± 0.91	4.56 ± 0.91	4.59 ± 0.91

Table 4
Comparison of testing MAPE by WRELM, PRELM, ORELM, R-ELM, and Mixture-ELM.

Algorithms	MAPE ± standard deviation							
	Case 1		Case 2		Case 3		Case 4	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
WRELM	1.49 ± 0.32	1.29 ± 0.29	1.34 ± 0.25	1.19 ± 0.20	1.54 ± 0.30	1.42 ± 0.25	1.53 ± 0.30	1.34 ± 0.26
PRELM	6.14 ± 0.18	2.01 ± 0.26	2.07 ± 0.19	2.05 ± 0.20	2.12 ± 0.31	2.07 ± 0.29	2.13 ± 0.23	2.09 ± 0.23
ORELM	1.23 ± 0.50	1.10 ± 0.43	1.14 ± 0.30	1.09 ± 0.27	1.18 ± 0.40	1.16 ± 0.34	1.19 ± 0.38	1.09 ± 0.33
R-ELM	1.17 ± 0.39	0.96 ± 0.32	0.99 ± 0.28	0.96 ± 0.27	1.36 ± 0.39	1.20 ± 0.35	1.18 ± 0.31	1.08 ± 0.26
Mixture-ELM	0.91 ± 0.34	0.85 ± 0.29	0.98 ± 0.23	0.87 ± 0.20	0.99 ± 0.34	0.91 ± 0.32	0.98 ± 0.31	0.90 ± 0.26



(a) Distributed parameter iteration diagram (b) Distributed proportion iteration diagram

Fig. 3. Iteration diagram of each parameter.

$0.125, \lambda_2 = 0.493, \pi_{\sigma_1} = 0.288, \pi_{\sigma_2} = 0.264, \pi_{\lambda_1} = 0.352$, and $\pi_{\lambda_2} = 0.096$, which is rather close to the actual value. This is sufficient proof of the validity of our suggested approach. Fig. 3 presents the matching parameter iteration diagram.

4.2. Case study

We conducted tests on selected benchmark datasets in the preceding section to illustrate the performance of the proposed Mixture-ELM algorithm. In this section, the validity of Mixture-ELM is then tested further using a real-world application—geothermal heat flux (GHF) prediction.

Rising sea levels will directly lead to the inundation of lowlands in coastal areas and the loss of land resources as a result of global warming, thereby exacerbating the extent of marine disasters; thus, it is critical to predict future sea level rise in advance and use it as the basis for the best protective measures. Glaciers, ice caps, and ice sheets from Greenland and Antarctica melt and flow into the oceans, thereby causing sea levels to increase. Therefore, being able to appropriately estimate ice sheet mass loss enhances the accuracy of projecting future sea level rise.

GHF is expected to be the least limited by observations among the various input parameters necessary for ice sheet models. However, the lack of access to bedrock makes it impossible to measure this heat directly in the ice sheet. GHF impacts ice temperature and viscosity, which can alter ice sheet shape and speed. Moreover, the genesis of ice flows in Greenland and Antarctica is frequently linked to regionally enhanced heat fluxes in the underlying bedrock. Furthermore, the heat created by the Earth's core (GHF) may be sufficient to melt the underlying layers of the ice sheet, reduce friction between ice and bedrock, and accelerate ice emissions to the ocean. In conclusion, because GHF is a critical boundary condition for accurately predicting ice sheet mass loss, it is critical to accurately quantify the geographical distribution of GHF in the underlying bedrock.

We use the Mixture-ELM algorithm proposed in this paper to obtain the statistical relationship between geological features and GHF by

Table 5
Statistical properties of the feature variable for predicting GHF.

No.	Variable type	Variable name	Max	Min	Mean	SD
1	Continuous	Global surface topography	5297.490	−1262.612	578.635	692.341
2		Depth to Moho	−14.983	−94.077	−36.914	8.217
3		Lithosphere asthenosphere boundary	309 971.000	13 886.500	143 214.645	76 716.825
4		Age	3.958	0.0713	1.640	0.684
5		Bougeur gravity anomaly	302.627	−454.792	60.820	77.307
6		Crustal thickness	76.710	11.660	38.797	6.571
7		Upper mantle density anomaly	0.072	−0.113	−0.00075	0.0257
8		Magnetic anomaly	3728.720	−9999.000	−302.952	1726.449
9		Thickness of upper crust	25.003	1.700	13.328	3.387
10		Thickness of Middle crust	25.000	2.300	13.163	3.136
11		Lat	79.500	−53.500	35.671	26.657
12		Lon	178.667	−165.333	4.340	85.643
13		Heat production provinces	2.920	0.400	1.470	0.443
14	Classification	Upper Mantle velocity structure	12	1	3.507	1.875
15		Rock type	3	1	2.664	0.698
16		thermo_tecto_age	6	1	2.260	1.649
17	Proximity	Distance to trench	5904.010	3.810	1709.269	1063.930
18		Distance to transform ridge	5476.730	3.110	2002.000	1144.039
19		Distance to young rft	3525.550	−59.10	746.851	683.103
20		Distance to volcano (5 nearest)	2914.140	23.420	1063.211	694.271
21		Distance to ridge	3446.920	6.960	1597.268	798.481
22		Distance to hot spot	86.617	0.707	25.649	20.732

Table 6
Statistical properties of training data and testing data of GHF.

Data	Max	Min	Mean	SD
Training	60.825	18.209	17.463	198.951
Testing	60.704	18.217	16.951	156.497

assuming that GHF is a complex function of geological and tectonic features. We utilized a collection of globally available geological characteristics and information from the continental crust, as in Table 5. These geological characteristics are divided into three categories:

1. Continuous data such as gravity anomalies and crust thickness.
2. Categorical data such as rock types and upper mantle velocity structure categories.
3. Proximity variables describe the distance from each point to thermally active geological features such as hot spots, ridges, and volcanoes.

First, the maximum, minimum, mean, and standard deviation (SD) of the aforementioned variables of the experimental data are computed, and shown in Table 6. Consequently, both the training data and the testing data display various statistical features, thereby revealing the characteristics of violent fluctuation of the experimental data. This implies that the data can more effectively confirm the generalization capability of the Mixture-ELM algorithm.

4.2.1. Evaluation criteria

In Section 4.1.1, Evaluation criteria, the definitions of RMSE Eq. (39), MAE Eq. (40) and MAPE Eq. (41) are given; meanings of the three evaluation criteria are expounded; and the three evaluation criteria are selected to measure the model fitting effect. Similarly, we also choose MAE, RMSE, and MAPE as evaluation criteria.

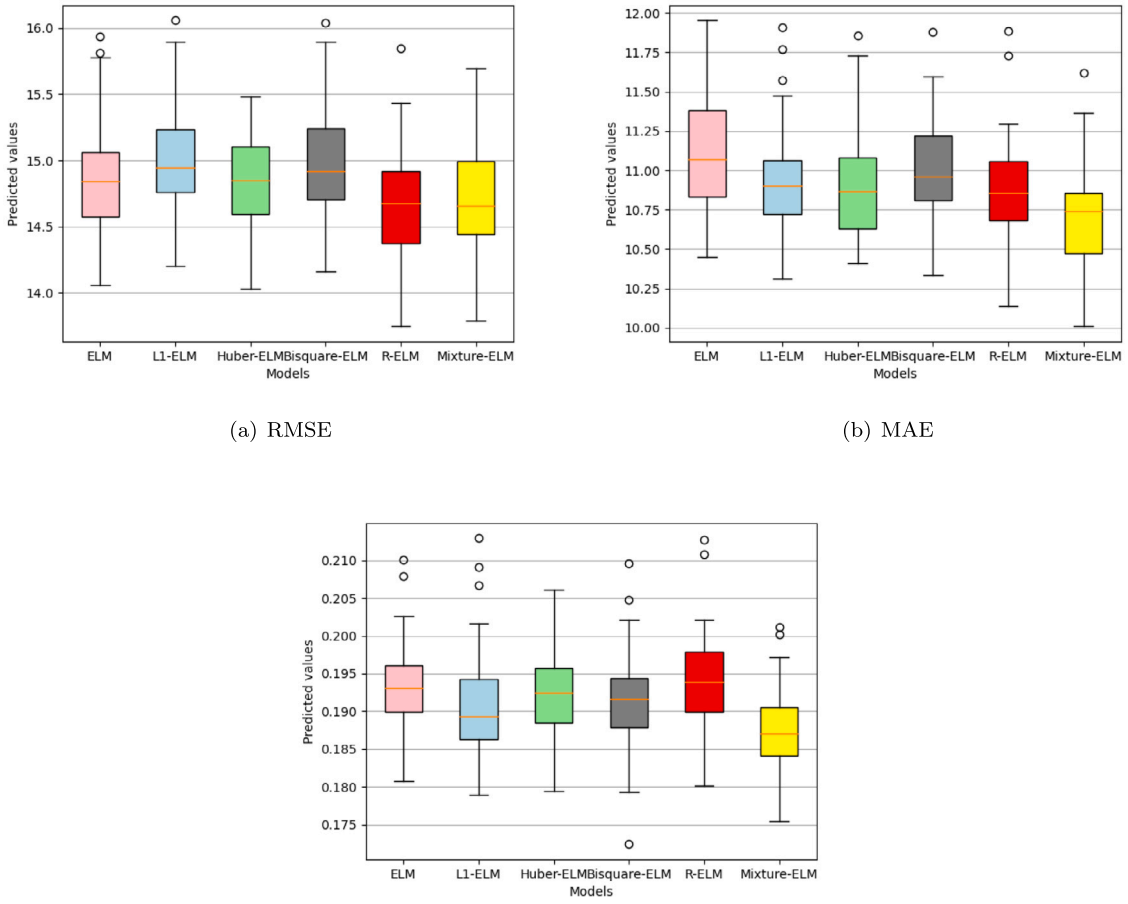
4.2.2. Experimental configuration and parameter selection

In this paper, apart from the proposed Mixture-ELM, a few other ELM models are also used in predicting GHF—that is, ELM, ELM with l_1-norm loss function (l_1-ELM), ELM with Huber-ELM loss function (Huber-ELM), ELM with Bisquare loss function (Bisquare-ELM), and R-ELM. Moreover, these models are used to reveal the performance of the proposed models in comparison with the proposed models in this work.

In specific experiments, in order to eliminate the influence of dimension among variables, all experimental data are standardized into (0, 1) with the same magnitude because it is known that variables with a big magnitude should have larger effects on modeling than variables with magnitude. The dataset was randomly divided into two groups: the training set (80% of each dataset) and the test set (the remaining data from each set). For the purpose of simplicity and fairness, we employ the "Sigmoid" as the activation function of each model. Then, since the first step of M theory is randomly generating the hidden layer parameters, to avoid randomness, each experiment is repeated 50 times to reveal the average performance of the model. Furthermore, all experiments are performed with PyCharm on an AMD 3600X CPU with a 16 GB RAM, and the experiment code is compiled with Python 3.10.

The initial parameters required to enter the EM algorithm are ξ, σ, λ, π_σ, and π_λ due to the Mixture-ELM in the EM algorithm. The EM method is sensitive to the initial settings since it theoretically may provide local optimum solutions. However, it is discovered in the experiment that almost any parameter can eventually cause the EM algorithm to find the global optimal solution as long as the order of magnitude initial parameters is adjusted to an adequate value. Consequently, the choice of these parameters is not the main emphasis of this paper.

The number of hidden layer nodes L of ELM greatly influences prediction accuracy. If the value of L is too small, the network cannot learn well and need to increase the number of training, and training accuracy is also affected. If L is too large, the training time increases and the network is easy to overfit. For the penalty item in the objective function Eq. (30), there is a regularization coefficient C. The value of C controls the bias between the structure risk and empirical risk term. A large value of C may cause a small error between predictions and real observations but would cause an overfitting problem and generate a high-complexity model. However, a small value of C can effectively reduce the complexity of the trained model and prevent the overfitting problem. Therefore, it is important to select appropriate values of L and C that can make a trade-off between forecasting error and model complexity. Similarly, we use the time-series cross-validation method



(a) RMSE

(b) MAE

(c) MAPE

Fig. 4. Results box plots of six models with different error indexes.

to determine the appropriate values of L and C and the search set is $[10, 15, 20, 25, 30]$ and $[10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}]$, respectively. The time-series cross-validation method is realized by the "TimeSeriesSplit" package in Python Sklearn.

4.2.3. Experimental results

The anticipated result of the Mixture-ELM model is superior to other models under MAE and MAPE evaluation criteria, whereas the predicted result of ELM has the lowest agreement with the actual GHF values. After 50 replications, the mean MAE and MAPE for Mixture-ELM were 10.709 and 0.187. Additionally, the Mixture-ELM MAPE values for the 50 trials had the lowest maximum and minimum values, 0.201 and 0.175, respectively. Under the RMSE evaluation criteria, the prediction accuracy of Mixture-ELM is second only to that of R-ELM, where the former is 14.689 and the latter is 14.668.

It is proven that the suggested Mixture-ELM models perform consistently better than others. We produce a box figure for the 50 trial results of each model in order to assess more precisely if there is a significant difference among the results of these models. The top and lower margins and quartile of boxes for the Mixture-ELM are smaller than others in the six sets of tests, as illustrated in Fig. 4. Therefore, a preliminary finding may be produced that mixed-performance is steadier than the others. However, it is evident from Table 7 that under different indicators, the prediction accuracy of the best model and the second-best model appears to have little difference. For example, under the MAPE evaluation criteria, the average MAPE of R-ELM attained the best value of 0.187; however, in addition, both l_1-ELM and Bisquare-ELM attained a value of 0.191.

Table 7
Detailed comparison of GHF predictions under the MAPE indicator on the test set.

Approach	RMSE	MAE	MAPE
ELM	14.862	11.111	0.193
l_1-ELM	15.001	10.935	0.191
Huber-ELM	14.806	10.903	0192
Biquare-ELM	14.990	11.008	0.191
R-ELM	**14.668**	10.886	0.194
Mixture-ELM	14.689	**10.709**	**0.187**

To more rigorously confirm the effects of the proposed Mixture-ELM, we also take the t-test to test the difference between different models under different evaluation criteria. Under the RMSE evaluation criterion, although the average value of Mixture-ELM is 14.689, second only to 14.668 (R-ELM), there is no significant difference between the results obtained by the two models after 50 experiments through the t-test. In addition, the results of Mixture-ELM are significantly better than those of the other four models. Therefore, under the RMSE evaluation criterion, the basic performance of the proposed Mixture-ELM is consistent with that of R-ELM, which is significantly better than the other four comparison models. Under the MAE evaluation criterion, through an independent sample t-test, the proposed Mixture-ELM is significantly better than the other five comparison models. Under the MAPE evaluation criterion, although the average MAPE of the proposed Mixture-ELM is 0.187, the second-best Bisquare-ELM and l_1-ELM are both 0.191, with a gap of only 0.004. However, through the independent sample t-test in Table 8, the experimental results of Mixture-ELM

Table 8
Statistical test results with investigated benchmark models under MAPE indexes.

Algorithms	Mean	Standard deviation	t	*p*-value	Average difference from Mixture-ELM	Cohen's d value
ELM	0.193	0.006	4.924	0.000***	0.006	0.990
l_1-ELM	0.191	0.008	2.641	0.010***	0.004	0.531
Huber-ELM	0.192	0.006	4.435	0.000***	0.005	0.869
Bisquare-ELM	0.191	0.006	3.105	0.002***	0.004	0.624
R-ELM	0.194	0.007	5.474	0.000***	0.007	1.100

[1] The mean and standard deviation of the Mixture-ELM results were 0.187 and 0.006.
[2] The number of tests for all models is 50, that is, the sample size is 50.
[3] The experimental parameters of the results in this table have hidden nodes of 30 and regularization coefficients of 100.
[4] ***, ** and * represent the significance level of 1%, 5% and 10% respectively.
[5] Cohen's d of 0.20, 0.50, and 0.80 correspond to small, medium, and large critical points respectively.

are significantly better than those of the other five models, although the mean value difference is small. For example, the mean values of R-ELM and Mixture-ELM in MAPE were 0.194 and 0.187, respectively. The *p*-value of the F test result is 0.00***; thus the statistical result is significant, thereby indicating that R-ELM, and Mixture-ELM showed significant differences in MAPE. In conclusion, Mixture-ELM is significantly better than other models under each evaluation criterion, and there is no significant difference between Mixture-ELM and R-ELM under the RMSE index.

5. Conclusions

In practical applications, existing ELMs can yield suboptimal solutions due to the presence of noise that often follows unknown distributions – including Gaussian, non-Gaussian, and mixed distributions – and may contain outliers. These conditions contradict the theoretical assumptions made during the ELM derivation process. To address this issue, this paper proposes a new ELM method called Mixture-ELM that enhances the modeling performance of classical ELM under unknown noise. To achieve this, we propose a superior objective function that employs mixed Gaussian and mixed Laplace distributions to describe the noise. This approach benefits from the good approximation properties of the mixture of Gaussian distribution for any continuous noise distribution. Additionally, the inclusion of a mixture Laplace distribution enhances the model's stability for anomalies and significantly reduces the sensitivity of the mixed Gaussian distribution to outliers, thereby resulting in a stronger model with no loss in performance. The modified objective function of the Mixture-ELM is resolved using the EM algorithm. In the numerical simulation, compared with other robust ELM algorithms, our proposed Mixture-ELM performs best. Moreover, in our application of forecasting GHF, we utilized the proposed model, and the results demonstrate its effectiveness in handling outlier distributions and avoiding overfitting for inaccurate forecasting. The three error indicators of RMSE, MAPE, and MAE reveal that the prediction outcomes of the Mixture-ELM model are superior to those of the comparison model, with the MAPE index performing particularly well, attaining a value of 0.187.

One limitation of this study is that the model's hyperparameters include those from the ELM model and the EM method, and cross-validation is utilized to identify the optimal hyperparameters for the experimental dataset, thereby enabling the model to make the best predictions. However, this method is not universal and consumes a significant amount of computational power, thereby necessitating the discovery of multiple ideal parameters for various data types. Therefore, in the future, adaptive methods will be sought to identify suitable hyperparameter techniques for use in this study. Moreover, in this paper, the dependent variable has only one dimension in both the numerical simulation and the actual prediction. However, the ELM theory suggests that the model is suitable for predicting high-dimensional data, thereby making it possible to extend future research to multiple dependent variable prediction tasks. Additionally, this study employs ridge regression as the regularization method. In future research, alternative regularization terms – such as lasso regression – could be explored.

CRediT authorship contribution statement

Shangrui Zhao: Writing – review & editing, Supervision, Funding acquisition. **Xuan-Ang Chen:** Writing – original draft, Visualization, Software, Formal analysis. **Jinran Wu:** Writing – review & editing, Writing – original draft, Supervision, Formal analysis. **You-Gan Wang:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request

Acknowledgments

The work is supported by the Australian Research Council project (grant number DP160104292), and "Chunhui Program" Collaborative Scientific Research Project (202202004).

References

[1] Yoan Miche, Mark Van Heeswijk, Patrick Bas, Olli Simula, Amaury Lendasse, Trop-elm: a double-regularized elm using lars and tikhonov regularization, Neurocomputing 74 (16) (2011) 2413–2421.
[2] Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[3] Cinzia Viroli, Geoffrey J. McLachlan, Deep gaussian mixture models, Stat. Comput. 29 (2019) 43–51.
[4] Ho Chun Leung, Chi Sing Leung, Eric Wing Ming Wong, Fault and noise tolerance in the incremental extreme learning machine, IEEE Access 7 (2019) 155171–155183.
[5] Jie Zhang, Yanjiao Li, Wendong Xiao, Zhiqiang Zhang, Robust extreme learning machine for modeling with unknown noise, J. Franklin Inst. B 357 (14) (2020) 9885–9908.
[6] Gao Huang, Guang-Bin Huang, Shiji Song, Keyou You, Trends in extreme learning machines: A review, Neural Netw. 61 (2015) 32–48.
[7] Iti Chaturvedi, Edoardo Ragusa, Paolo Gastaldo, Rodolfo Zunino, Erik Cambria, Bayesian network based extreme learning machine for subjectivity detection, J. Franklin Inst. B 355 (4) (2018) 1780–1797.
[8] Guang-Bin Huang, Lei Chen, Chee Kheong Siew, et al., Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
[9] Jie Zhang, Wendong Xiao, Yanjiao Li, Sen Zhang, Residual compensation extreme learning machine for regression, Neurocomputing 311 (2018) 126–136.
[10] Jie Zhang, Yifang Lu, Baoqiang Zhang, Wendong Xiao, Device-free localization using empirical wavelet transform-based extreme learning machine, in: 2018 Chinese Control and Decision Conference (CCDC), IEEE, 2018, pp. 2585–2590.
[11] Nan-Ying Liang, Guang-Bin Huang, Paramasivan Saratchandran, Narasimhan Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (6) (2006) 1411–1423.
[12] Yanjiao Li, Sen Zhang, Yixin Yin, Wendong Xiao, Jie Zhang, A novel online sequential extreme learning machine for gas utilization ratio prediction in blast furnaces, Sensors 17 (8) (2017) 1847.

[13] Yuan Lan, Yeng Chai Soh, Guang-Bin Huang, Constructive hidden nodes selection of extreme learning machine for regression, Neurocomputing 73 (16–18) (2010) 3191–3199.
[14] Bilal Mirza, Zhiping Lin, Nan Liu, Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift, Neurocomputing 149 (2015) 316–329.
[15] Yuan Lan, Yeng Chai Soh, Guang-Bin Huang, Ensemble of online sequential extreme learning machine, Neurocomputing 72 (13–15) (2009) 3391–3395.
[16] Wendong Xiao, Jie Zhang, Yanjiao Li, Sen Zhang, Weidong Yang, Class-specific cost regulation extreme learning machine for imbalanced classification, Neurocomputing 261 (2017) 70–82.
[17] Yanjiao Li, Sen Zhang, Yixin Yin, Wendong Xiao, Jie Zhang, Parallel one-class extreme learning machine for imbalance learning based on bayesian approach, J. Ambient Intell. Humaniz. Comput. (2018) 1–18.
[18] Musatafa Abbas Abbood Albadra, Sabrina Tiuna, Extreme learning machine: a review, Int. J. Appl. Eng. Res. 12 (14) (2017) 4610–4623.
[19] Jinran Wu, You-Gan Wang, Yu-Chu Tian, Kevin Burrage, Taoyun Cao, Support vector regression with asymmetric loss for optimal electric load forecasting, Energy 223 (2021) 119969.
[20] Annapareddy V.N. Reddy, Ch. Krishna, Pradeep Kumar Mallick, et al., An image classification framework exploring the capabilities of extreme learning machines and artificial bee colony, Neural Comput. Appl. 32 (8) (2020) 3079–3099.
[21] Zhan-Li Sun, Tsan-Ming Choi, Kin-Fan Au, Yong Yu, Sales forecasting using extreme learning machine with applications in fashion retailing, Decis. Support Syst. 46 (1) (2008) 411–419.
[22] Musatafa Abbas Abbood Albadr, Sabrina Tiun, Masri Ayob, Fahad Taha Al-Dhief, Taj-Aldeen Naser Abdali, Aymen Fadhil Abbas, Extreme learning machine for automatic language identification utilizing emotion speech data, in: 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), IEEE, 2021, pp. 1–6.
[23] Hui Xiong, Gaurav Pandey, Michael Steinbach, Vipin Kumar, Enhancing data analysis with noise removal, IEEE Trans. Knowl. Data Eng. 18 (3) (2006) 304–319.
[24] José M. Martínez-Martínez, Pablo Escandell-Montero, Emilio Soria-Olivas, José D. Martín-Guerrero, Rafael Magdalena-Benedito, Juan Gómez-Sanchis, Regularized extreme learning machine for regression problems, Neurocomputing 74 (17) (2011) 3716–3721.
[25] Zhihong Man, Kevin Lee, Dianhui Wang, Zhenwei Cao, Chunyan Miao, A new robust training algorithm for a class of single-hidden layer feedforward neural networks, Neurocomputing 74 (16) (2011) 2491–2501.
[26] Yan-Lin He, Zhi-Qiang Geng, Yuan Xu, Qun-Xiong Zhu, A hierarchical structure of extreme learning machine (helm) for high-dimensional datasets with noise, Neurocomputing 128 (2014) 407–414.
[27] Yang Yang, Hu Zhou, Yuchao Gao, Jinran Wu, You-Gan Wang, Liya Fu, Robust penalized extreme learning machine regression with applications in wind speed forecasting, Neural Comput. Appl. 34 (1) (2022) 391–407.
[28] Kai Zhang, Minxia Luo, Outlier-robust extreme learning machine for regression problems, Neurocomputing 151 (2015) 1519–1527.
[29] Xiaoxuan Lu, Han Zou, Hongming Zhou, Lihua Xie, Guang-Bin Huang, Robust extreme learning machine with its application to indoor positioning, IEEE Trans. Cybern. 46 (1) (2015) 194–205.
[30] Jungang Lou, Yunliang Jiang, Qing Shen, Ruiqin Wang, Zechao Li, Probabilistic regularized extreme learning for robust modeling of traffic flow forecasting, IEEE Trans. Neural Netw. Learn. Syst. 34 (4) (2023) 1732–1741.
[31] Olivier Cappé, Eric Moulines, On-line expectation–maximization algorithm for latent data models, J. R. Stat. Soc. Ser. B Stat. Methodol. 71 (3) (2009) 593–613.
[32] Nan Laird, Nicholas Lange, Daniel Stram, Maximum likelihood computations with repeated measures: application of the em algorithm, J. Amer. Statist. Assoc. 82 (397) (1987) 97–105.
[33] You-Gan Wang, Xu Lin, Min Zhu, Zhidong Bai, Robust estimation using the huber function with a data-dependent tuning constant, J. Comput. Graph. Statist. 16 (2) (2007) 468–481.
[34] Paul W. Holland, Roy E. Welsch, Robust regression using iteratively reweighted least-squares, Comm. Statist. Theory Methods 6 (9) (1977) 813–827.