Robust Extreme Learning Machine Based on p-order Laplace Kernel-Induced Loss Function

Liutao Luo[1], Kuaini Wang[2]*, Qiang Lin[3]
School of Computer Science, Xi'an Shiyou University, Xi'an, 710065, China[1]
College of Science, Xi'an Shiyou University, Xi'an 710065, China[2]
School of Business, Jiangnan University, Wuxi 214122, China[3]

***Abstract*—Since the datasets of the practical problems are usually affected by various noises and outliers, the traditional extreme learning machine (ELM) shows low prediction accuracy and significant fluctuation of prediction results when learning such datasets. In order to overcome this shortcoming, the l_2 loss function is replaced by the correntropy loss function induced by the p-order Laplace kernel in the traditional ELM. Correntropy is a local similarity measure, which can reduce the impact of outliers in learning. In addition, introducing the p-order into the correntropy loss function is rewarding to bring down the sensitivity of the model to noises and outliers, and selecting the appropriate p can enhance the robustness of the model. An iterative reweighted algorithm is selected to obtain the optimal hidden layer output weight. The outliers are given smaller weights in each iteration, significantly enhancing the robustness of the model. To verify the regression prediction of the proposed model, it is compared with other methods on artificial datasets and eighteen benchmark datasets. Experimental results demonstrate that the proposed method outperforms other methods in the majority of cases.**

***Keywords*—p-order Laplace kernel-induced loss; extreme learning machine; robustness; iterative reweighted**

I. Introduction

Extreme Learning Machine (ELM), as a generalized single hidden layer feedforward neural network, was proposed by Huang et al. [1]. Its random selection of hidden node biases and input weights, along with the use of the ordinary least square method for determining the output weight, enables a simple, fast, and straightforward implementation. It has been widely used in load forecasting [2], [3], [4], fault detection [5], [6], image processing [7], image recognition [8] and other fields.

Although ELM performs well in terms of efficiency, it is susceptible to noise and outliers due to the use of the l_2 loss function, which can amplify their interference. Therefore, in recent years, many researchers have devoted themselves to the robustness of ELM. In regularized ELM [9], the regularization term of the objective function significantly improved the learning performance of ELM by minimizing the structural risk. Deng et al. [10] put forward a weighted least square regularized ELM (Weighted ELM, WELM) to enhance robustness by iterative weighted method. The above two methods employed l_2 loss function, which was optimal only when the error of the training datasets followed the normal distribution. However, many practical applications cannot guarantee the error followed a normal distribution, which lead to a fact that ELM is highly susceptible to noise and outliers. Subsequently, the researchers proposed several loss function such as Huber [11], l_1 [12] and Pinball [13] and their corresponding ELM models. However, these loss functions were still less robust because they had a linear relationship with the training error and increased linearly with the training error. Incorporating both regularization term (l_1, l_2) and various loss functions (l_1, Huber, bisquare and Welsch), Chen et al. [14] put forward an unified robust regularized ELM, which improved the robustness of ELM.

As the research progressed, the researchers found that machine learning algorithms based on non-convex loss functions had strong robustness to datasets disturbed by noise and outliers [15], [16], [17], [18]. The loss functions in classical machine learning methods, including hinge loss, ε-insensitive loss, and l_2-loss, were replaced by non-convex loss functions to construct the corresponding robust learning algorithms. Correntropy [19] is a nonlinear local similarity measure built on a Gaussian kernel function, which can weaken the role of noise and outliers in the learning process. The correntropy loss function has better robustness to noise and outliers than the convex loss function [20]. On this basis, Xing et al. [21] developed an ELM model based on the maximum correntropy criterion to improve robustness. C-loss function [22] and non-convex smooth loss [23] derived from correntropy and their corresponding models were proved to be robust to noise and outliers. Chen et al. [24] presented a maximum correntropy criterion with variable center (MCC-VC), which is also essentially a loss function derived from the correntropy. The use of Gaussian kernels in correntropy learning is common, owing to their smoothness and strict positive definiteness. Nevertheless, Gaussian kernels may not always be the optimal choice. On the one hand, this is because that the choice should be based on specific problem and experimental results to determine the optimal kernel function and parameters. On the other hand, the exponential part of the Gaussian kernel function is in the form of l_2, which would overemphasize the role of noise and outliers, so this could potentially lead to a greater sensitivity to noise and outliers. Yang [25] introduced a new method based on the Laplace kernel (LK-loss) and demonstrated that the LK-loss serves as a reliable approximation of the zero norm. Dong et al. [26] presented a robust semi-supervised support vector machines with Laplace kernel-induced correntropy loss functtion utilizing LaplaceSVM to solve the problem of insufficient supervisory information and noise effects in practical applications. Chen et al. [27] pointed out that taking the p-order function of the error as a loss function was effective to decrease the sensitivity of the model to the noise and outliers,

*Corresponding authors. email: wangkuaini1219@sina.com

and appropriate p was conducive to improve the robustness of the model. Chen et al. [28] put forward a robust ELM based on p-order Welsch loss function, and the experiments revealed the superiority of method over the Welsch loss.

Inspired by the above studies, this paper offers the p-order loss function into the correntropy loss function induced by the Laplace kernel (p-LKI loss function) and applies it to ELM. The main contributions of this paper are as follows:

(1) This paper introduces the Laplace kernel function into the correntropy and incorporates the p-order of the loss function into it, and proposes an ELM model based on p-LKI loss function. The robustness of the model can be significantly improved by choosing a suitable p.

(2) We have proved that the p-LKI loss function is positive-definite, bounded and non-convex, and can converge to 1 with increasing error. Additionally, as the parameter p increases, the p-LKI loss function serves as a favorable approximation of the zero norm.

(3) The iterative reweighted algorithm efficiently addresses the optimization problem and converges to the optimal solution within a few iterations. We investigate that the larger the error of the sample, the smaller weight assigned to it, thus the smaller the impact on the model.

The paper is organized as follows: Section II briefly introduces ELM. In Section III, we present an ELM based on p-order Laplace Kernel-Induced loss function and the iterative reweighted algorithm is used to address the problem. The experiments are conducted in different levels of outliers in artificial dataset and benchmark datasets in Section IV. The experimental results of the proposed method are discussed and compared with other methods in Section V. And the conclution and prospect are summarized in Section VI.

II. Brief Review of ELM

Given training samples $S = \{(x_i, y_i)\}_{i=1}^N$, $x_i \in R^d$, $y_i \in R$, the mathematical representation of the output function of a single hidden layer ELM with L hidden nodes and activation functions $h_i(x)$ is as follows:

$$f(x) = \sum_{i=1}^{L} h_i(x)\beta_i = h(x)\beta \tag{1}$$

where, $\beta = [\beta_1, \beta_2, \cdots, \beta_L]^{\mathrm{T}}$ is the output weight vector, $h(x) = [h_1(x), h_2(x), \cdots, h_L(x)]$ is the hidden layer output of variable x. Let $Y = [y_1, y_2, \cdots, y_N]^{\mathrm{T}}$, hidden layer output matrix $H = [h(x_1)^{\mathrm{T}}, h(x_2)^{\mathrm{T}}, \cdots, h(x_N)^{\mathrm{T}}]^{\mathrm{T}}$, the ELM model can be expressed as the following optimization problem [1].

$$\min_{\beta} \frac{1}{2}\|\beta\|^2 + \frac{C}{2}\|Y - H\beta\|^2 \tag{2}$$

where, C is a regularization parameter. The best solution in Eq. (2) is provided by Huang et al. [1] as,

$$\beta = \begin{cases} (H^{\mathrm{T}}H + I/C)^{-1}H^{\mathrm{T}}Y, & N \geq L \\ H^{\mathrm{T}}(HH^{\mathrm{T}} + I/C)^{-1}Y, & N < L \end{cases} \tag{3}$$

where, I denotes the identity matrix.

III. Robust ELM based on p-order Laplace Kernel-induced Loss Function

The l_2 loss function in ELM gives the same weight to each training samples, which makes the outliers have a larger impact on the sum of squared errors than the rest of the samples, resulting in model that is quite sensitive to outliers. Inspired by correntropy [19] and p-order loss functions [27], this paper proposes to use the p-LKI loss function to improve the robustness of ELM.

A. P-order Laplace Kernel-induced Loss Function

In order to improve the robustness of the model, the maximum correntropy criterion (MCC) [21] is introduced. Correntropy [19] describes the measure of similarity between two samples, the principle is as follows:

$$V_\sigma(A, B) = E(k_\sigma(A, B)) \tag{4}$$

where k_σ is the kernel function, $\sigma > 0$ is the kernel bandwidth, and E is the mathematical expectation. In most cases, the joint probability distribution between variables A and B is unknown, and the mean can be used to estimate the mathematical expectation. For variables $A = (a_1, a_2, \cdots, a_N)$, $B = (b_1, b_2, \cdots, b_N)$, and $q = (q_1, q_2, \cdots, q_N)$, $q_i = a_i - b_i$. The correntropy estimation is as follows:

$$V_\sigma = \frac{1}{N}\sum_{i=1}^{N} k_\sigma(a_i, b_i) \tag{5}$$

where $k_\sigma(q_i) = \exp(-\frac{|q_i|}{\sigma})$ is Laplace kernel function.

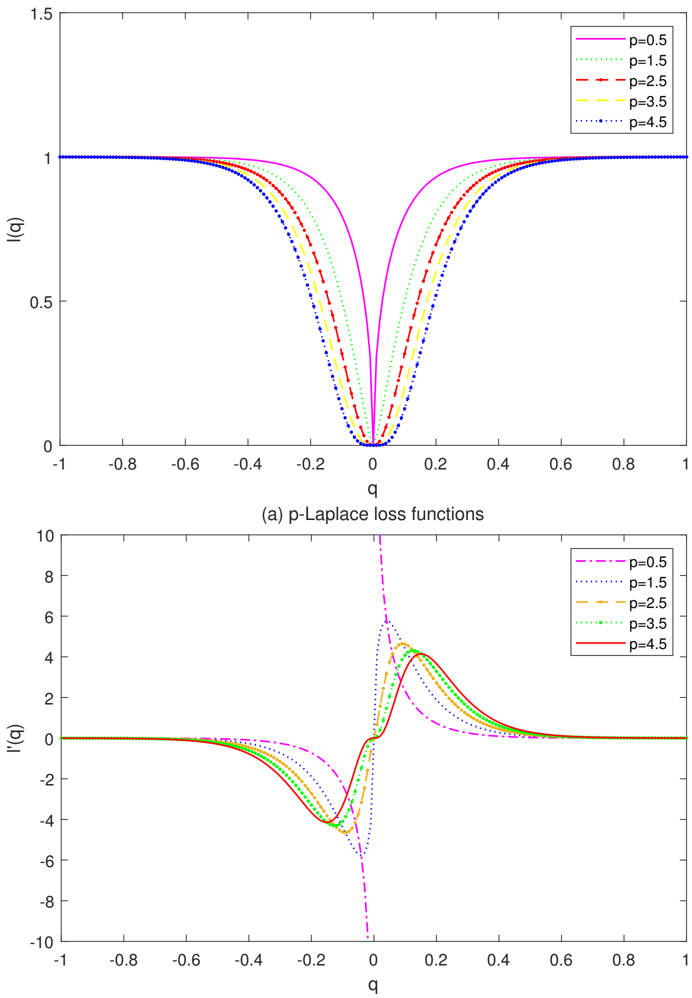
MCC [20] can be expressed as,

$$\max \frac{1}{N}\sum_{i=1}^{N} k_\sigma(q_i) = \max \frac{1}{N}\sum_{i=1}^{N} \exp(-\frac{|q_i|}{\sigma}) \tag{6}$$

To facilitate the calculation, Eq (6) is equivalent to,

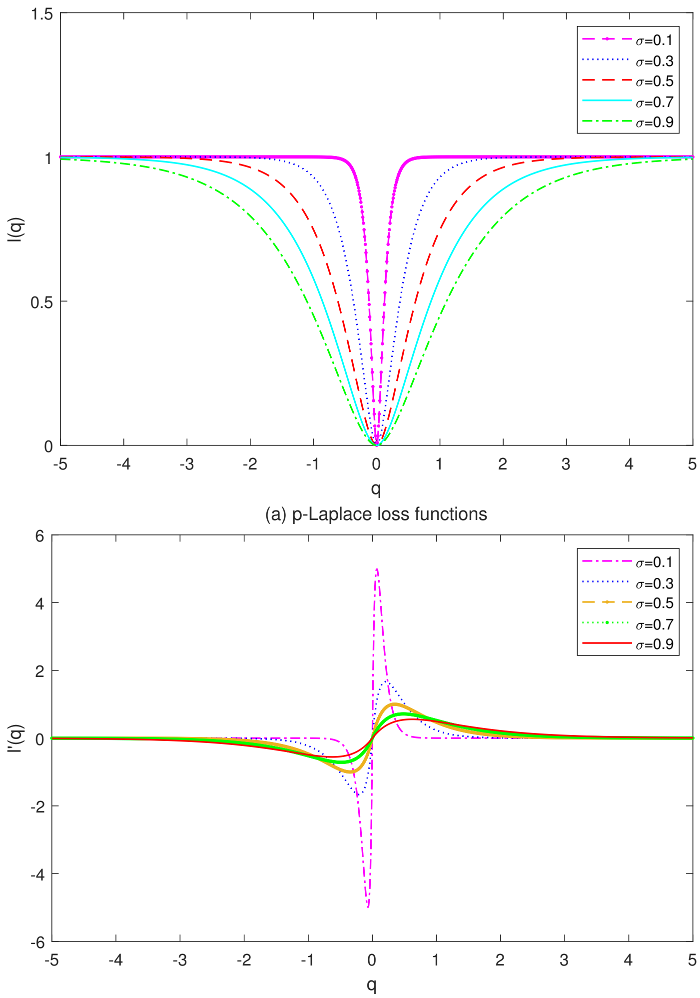
$$\min 1 - \exp(-\frac{|q|}{\sigma}) \tag{7}$$

Reference [27] pointed out that the loss function employing second-order statistical measures is susceptible to outliers, and it is not always a good choice for learning with samples that is non-Gaussian in nature. To address non-Gaussian data and noise, various non-second-order (or non-quadratic) loss functions have been proposed, such as the Huber minimum-maximum loss [15], Lorentz error loss [16], risk-sensitive loss [17], and mean p-power error (MPE) loss [27]. The MPE represents the p-th absolute moment of the error and effectively manages non-Gaussian datasets with an appropriate choice of the parameter p. Generally speaking, MPE demonstrates robustness to significant outliers for $0 < p < 2$ [27]. Inspired by the above studies, this paper proposes the following p-LKI loss function.



(b) the gradient functions of p-Laplace loss

Fig. 1. Comparison of p-LKI loss functions and their gradient functions under different p.



(b) the gradient functions of p-Laplace loss

Fig. 2. Comparison of p-LKI loss functions and their gradient functions under different σ.

$$l(q) = (1 - \exp(-\frac{|q|}{\sigma}))^p \tag{8}$$

The gradient function of p-LKI loss function is as follows:

$$\frac{\partial l(q)}{\partial q} = \frac{pq}{\sigma}\exp(-\frac{|q|}{\sigma})(1-\exp(-\frac{|q|}{\sigma}))^{p-1}\frac{1}{\max\{|q|, 10^{-6}\}} \tag{9}$$

We can observe from Fig. 1(a), $l(q)$ becomes larger as $|q|$ increases and will eventually approach 1 for any value of p when $|q|$ reaches a certain threshold. Even if the $|q|$ increases again, $l(q)$ will only approach 1 again with little change, thus reducing the influence of significant errors brought by outliers on model training. Furthermore, as depicted in Fig. 1(b), as the value of p decreases, the extreme point of $l'(q)$ will move forward with the decrease of the value of p which means that the part of $l(q)$ that is most sensitive to error changes will move forward relatively. Therefore, when p is too large, the sensitivity of $l(q)$ to outliers will increase. However, when $p = 0.5$, $l'(q)$ is discontinuous at zero which means $l(q)$ is not differentiable at zero.

Fig. 2 shows p-LKI loss function $l(q)$ and its gradient function $l'(q)$ under different values of σ. It can be seen that with the increase of the values of σ, the corresponding $|q|$ will increase correspondingly when $l(q)$ approaches 1. With the decrease of the values of σ, the extreme point of $l'(q)$ is approaching zero, and the smaller the values of σ, the stronger the robustness of the model to the outliers. Therefore, the sensitivity of $l(q)$ to outliers can be reduced by adjusting the values of p and σ. The optimal values of p and σ will be further determined by grid search.

The p-LKI loss function offers the strengths in these aspects:

1. The p-LKI loss function $l(q)$, shown in Fig. 2(a), is a positive, symmetric, and bounded function. It attains its maximum value only when $q = 0$. The p-LKI fulfills the following:

$$\frac{\partial l(q)}{\partial q} = \mathrm{sgn}(q)\frac{p}{\sigma}\exp(-\frac{|q|}{\sigma})(1-\exp(-\frac{|q|}{\sigma}))^{p-1}, q \neq 0 \tag{10}$$

We have

$$\lim_{q\to\infty}\frac{p}{\sigma}\exp(-\frac{q}{\sigma})(1-\exp(-\frac{q}{\sigma}))^{p-1} = 0 \tag{11}$$

As shown in Eq.(11) that when the error approaches infinity, the gradient function $l'(q)$ of p-LKI approaches 0, indicating that $l(q)$ does not change for the outliers. Therefore, the p-LKI loss function is resistant to outliers.

2. For $\forall q \in R^N$,

$$\lim_{\sigma\to 0^+} l(q) = \|q\|_0 \tag{12}$$

Proof: The empirical risk derived from the p-LKI loss function can be represented as:

$$R_l(f) = \sum_{i=1}^{N}(1-\exp(-\frac{|q_i|}{\sigma}))^p \tag{13}$$

By evaluating the limit as $\sigma \to 0^+$, we obtain:

$$\begin{aligned}\lim_{\sigma\to 0^+} R_l(f) &= \lim_{\sigma\to 0^+}\sum_{i=1}^{N} l(q_i) \\ &= \lim_{\sigma\to 0^+}\sum_{i=1}^{N}(1-\exp(-\frac{|q_i|}{\sigma}))^p = \|q\|_0\end{aligned} \tag{14}$$

where the zero norm $\|q\|_0$ counts the non-zero elements of q.

3. In comparison to other estimations of the zero norm, like the p-order Gaussian kernel-induced loss (p-Welsch),

$$M(q) = (1-\exp(-\frac{q^2}{2\sigma^2}))^p \tag{15}$$

Fig. 3 (a) shows the curves of the p-LKI loss function and p-Welsch loss function with $p = 0.8$ and $\sigma = 0.1$. It can be inferred that the approximation precision of the p-LKI loss function is higher than the p-Welsch loss function which means that it is closer to the zero norm. Some advantages of the zero norm are as follows:

1) Sparsity: The zero norm loss function encourages the model to produce sparse weights, i.e. only a small percentage of the weights are non-zero. This can effectively reduce the complexity of the model and prevent over-fitting [27].

2) Robustness: By making the weights sparse, the zero norm loss function can enhance the robustness of the model. Only those features that are most important to the predictions of the model are given greater weight, thus reducing over-reliance on unimportant features.

Fig. 3 shows a comparison of loss functions such as l_2 [10], l_1 [12], Welsch [22], Laplace [25], p-Welsch [28], p-LKI loss function and their gradient functions. From the figure, it is evident that in addition to the l_2-loss function and l_1-loss function, the error of each dataset in the other loss functions is controlled $[0, 1]$. The gradient function will be small after the $|q|$ exceeds a certain value and will not increase with the increase of error like the gradient function of l_2-loss and l_1-loss, thereby reducing the influence of the large error term caused by outliers on parameter estimation. Moreover, we can observe from Fig. 3 that p-LKI loss function has the closest distance from the $\|q\|_0$ ($l(q) = 1$) , so the accuracy of the zero norm approximation of the p-LKI loss function is the highest. In addition, compared with the Welsch and p-Welsch loss functions induced by the Gaussian kernel function, the Laplace and p-LKI loss function induced by the Laplace kernel have higher approximate accuracy for zero norms, where the approximate accuracy of p-LKI loss function is higher than that of Laplace loss function.

3) Robust ELM based on p-LKI loss function: By taking the p-LKI loss function in ELM, the p-LKI-ELM model is established

$$\begin{array}{ll}\min\limits_{\beta,q_i} & \frac{1}{2}\|\beta\|_2^2 + C\sum\limits_{i=1}^{N}(1-\exp(-\frac{|q_i|}{\sigma}))^p \\ s.t. & h(x_i)\beta = y_i - q_i, i = 1, 2, \cdots, N\end{array} \tag{16}$$

According to the KKT condition, Eq. (16) can be reformulated as solving the following problem:

$$\begin{aligned}L(\beta, q_i, \alpha) = \frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^{N}(1-\exp(-\frac{|q_i|}{\sigma}))^p \\ - \sum_{i=1}^{N}\alpha_i(h(x_i)\beta - y_i + q_i)\end{aligned} \tag{17}$$

where α_i is the Lagrange multiplier corresponding to each training sample.

Calculate the partial derivative of each parameter variable in Eq.(17), and let the partial derivative be zero,

$$\begin{cases}\frac{\partial L}{\partial \beta} = 0 \Rightarrow & \beta = \sum_{i=1}^{N}\alpha_i h(x_i)^{\mathrm{T}} = H^{\mathrm{T}}\alpha \\ \frac{\partial L}{\partial q_i} = 0 \Rightarrow & \alpha_i = Cq_i w(q_i) \\ \frac{\partial L}{\partial \alpha_i} = 0 \Rightarrow & h(x_i)\beta - y_i + q_i = 0\end{cases} \tag{18}$$

where

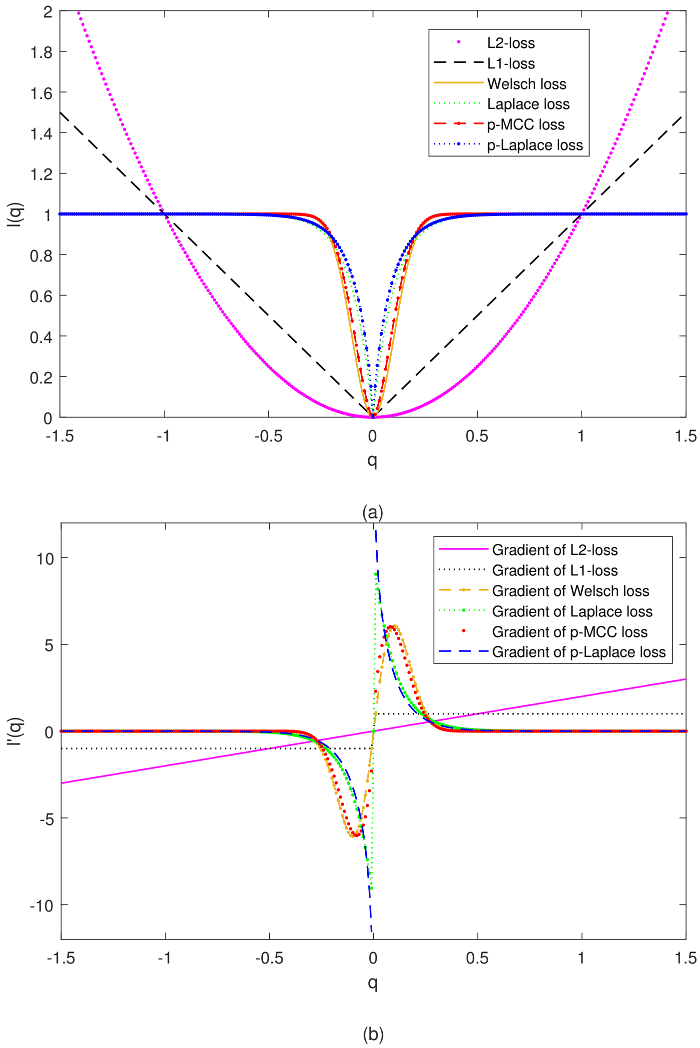


Fig. 3. Comparison of (a) Loss functions; (b) Their gradient functions.

$$w(q_i) = \frac{\partial l(q)}{q_i \partial q_i} = \frac{p}{\sigma}\exp(-\frac{|q_i|}{\sigma})(1-\exp(-\frac{|q_i|}{\sigma}))^{p-1}$$
$$= \frac{1}{\max\{|q_i|, 10^{-6}\}}$$

In this paper, we employ an iterative reweighted algorithm to obtain the optimal hidden layer output weight β. The weight of N samples can be expressed as

$$W(q) = diag(w(q_1), w(q_2), \cdots, w(q_N)) \tag{19}$$

Through Eq.(18), the output weight of the hidden layer is

$$\beta = \begin{cases} H^{\mathrm{T}}(\frac{I}{C} + W(q)HH^{\mathrm{T}})^{-1}W(q)Y, & N < L \\ (\frac{I}{C} + H^{\mathrm{T}}W(q)H)^{-1}H^{\mathrm{T}}W(q)Y, & N \geq L \end{cases} \tag{20}$$

The curve of sample weights with different parameters σ is shown below.

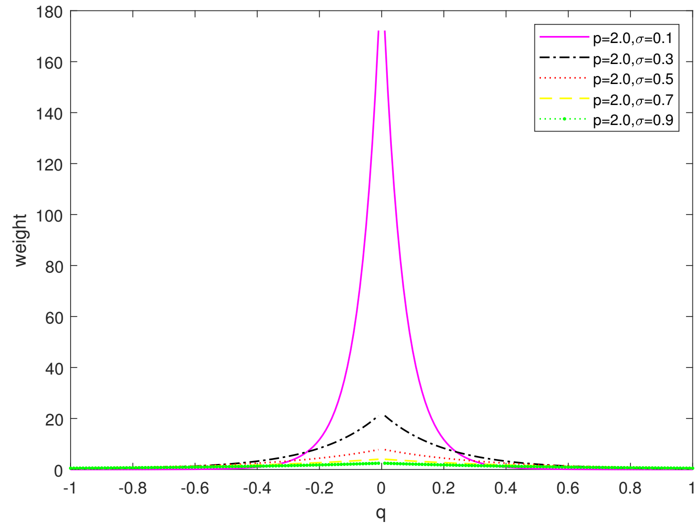


Fig. 4. Trend of sample weights with σ at $p = 2$.

Fig. 4 shows that the larger the error $|q|$ of the sample is, the smaller the weight of the sample is, then the smaller the influence on the model. Therefore, the proposed method can effectively reduce the influence of outliers and enhance the robustness of the model.

Algorithm 1 p-LKI-ELM

Input: Training dataset S, number of hidden nodes L, regularization parameter C, kernel bandwidth σ, maximum of iterations t_{max}, the hidden layer output matrix H.

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

1: Initialize $W(q)^{(0)} = I, t = 1$;

2: Calculate the optimal output weight $\beta^{(t)}$ by

$$\boldsymbol{\beta}^{(t)} = \begin{cases} H^{\mathrm{T}}(\frac{I}{C} + W(q)^{(t-1)}HH^{\mathrm{T}})^{-1}W(q)^{(t-1)}Y, & N < L \\ (\frac{I}{C} + H^{\mathrm{T}}W(q)^{(t-1)}H)^{-1}H^{\mathrm{T}}W(q)^{(t-1)}Y, & N \geq L. \end{cases} \tag{21}$$

3: Obtain $q_i^{(t)} = y_i - h(x_i)\beta^{(t)}$, and assign diagonal matrix $W(q)^{(t)}$ by (19).

4: Update $\beta^{(t+1)}$ from (21);

5: if $t > t_{max}$ or $\left\|\beta^{(t+1)} - \beta^{(t)}\right\| \leq 10^{-3}$ stop , else go to step 6.

6: Derive output value $h(x_i)\beta^{(t+1)}$. Set $t = t + 1$, and go to step 3.

IV. Experiments

We compare the proposed method with ELM [1], WELM [11], IRWELM [12], Welsch-ELM [22], Laplace-ELM [25], p-Welsch-ELM [28] on the artificial datasets and benchmark datasets. The root mean square error (RMSE) is chosen as the evaluation metric:

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}}(\mathrm{y_i} - \mathrm{t_i})^2} \tag{22}$$

where y_i and t_i represent the actual target of the sample and the corresponding prediction, respectively; m is the number of test samples.The experiments are tested in Matlab2021 a Win10 environment with 3.0 GHz CPU, 8 GB RAM and 64 bit host.

A. Experimental Settings

(1) The input weight matrix $W_{N\times L}$ and the hidden layer bias $b_{L\times 1}$ are randomly selected in [-1,1]. The hidden layer activation function is sigmoid function.

$$g(z) = \frac{1}{1+e^{-z}} \tag{23}$$

(2) Regularization parameter C is optimized by cross validation from the set $\{2^{-19}, 2^{-18}, \cdots, 2^{20}\}$ and the number of hidden nodes L is fixed as 1000.

(3) Number of algorithm iterations $t_{max} = 20$.

(4) Parameters σ and order p are also optimized by grid search, where $\sigma : \{0.1, 0.2, \cdots, 1\}$; $p : \{0.6, 0.7, \cdots, 5\}$.

B. Experimental on Artificial Datasets

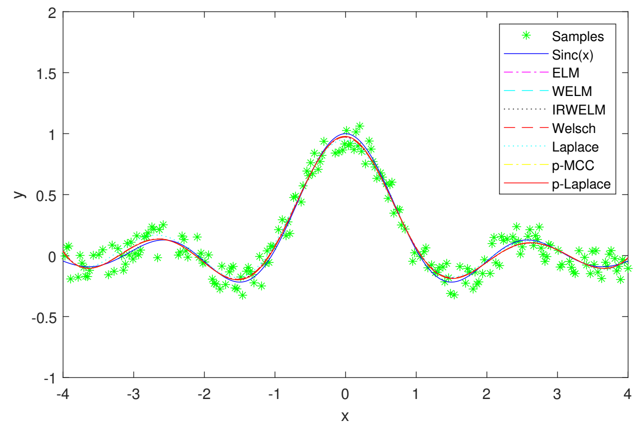
1) Experimental preparation: The artificial dataset is generated by function $y = \sin c(x)$, where,

$$\sin c(x) = \frac{\sin x}{x}, x \in [-4, 4]. \tag{24}$$

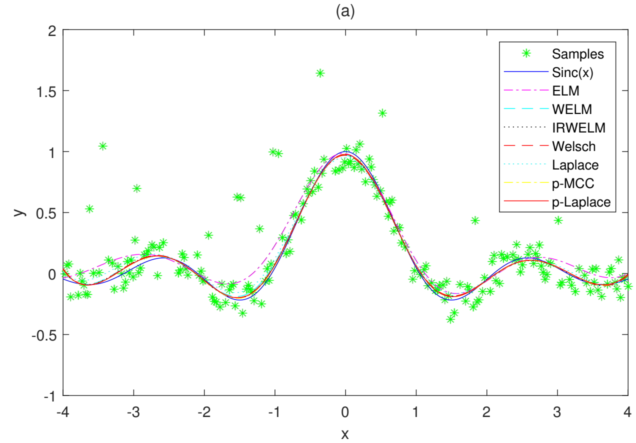
The preprocessing of artificial datasets is divided into three steps. First, 300 samples are generated from Eq.(24) and randomly divided into 200 training samples and 100 test samples. Secondly, the target of the training sample is disturbed by the uniform distribution of noise $[-0.15, 0.15]$. Finally, random values of different proportions in $[y_{\min}, y_{\max}]$ are added as outliers to the targets of some training samples generated in the second step. Outliers include 0%, 10%, 20%, 30%, and 40%. The samples used for testing are from Eq.(24) without any added outliers. To ensure fairness, 10 independent experiments are conducted for each outliers distribution.

2) Experimental results and analysis: To further confirm the robustness of the proposed algorithm, the different levels of outliers are compared. Fig. 5 illustrates the regression prediction results of these seven algorithms with different outliers levels. When the outliers level is 0%, all seven methods roughly coincide with the original position. When the outliers levels are 10% and 20%, only ELM deviates slightly from the original position and begins to shift toward the outliers, while the other six methods remain unchanged. When the outliers levels are 30% and 40%, ELM, WELM, IRWELM, Laplace-ELM and Welsch-ELM deviate from the original position and turn toward the outliers, and only p-LKI-ELM and p-Welsch-ELM are relatively close to the original position and do not have a tendency to turn towards the outliers. It can be seen that as the outliers level increases, all five methods except p-LKI-ELM and p-Welsch-ELM deviate from the original position and shift towards outliers, and the trend turn toward the outliers of p-LKI-ELM is smaller compared to p-Welsch-ELM. Therefore, it indicates that p-LKI-ELM has better stability.

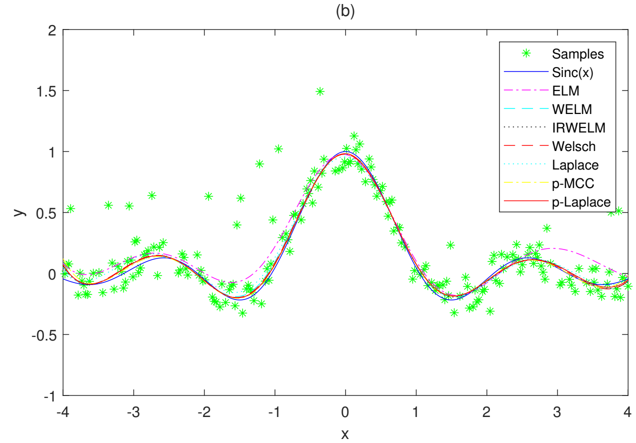
Fig. 6 reflects the variation of the RMSE of the seven methods for different outliers levels on the artificial dataset. When there are no outliers, the RMSE of p-LKI-ELM is



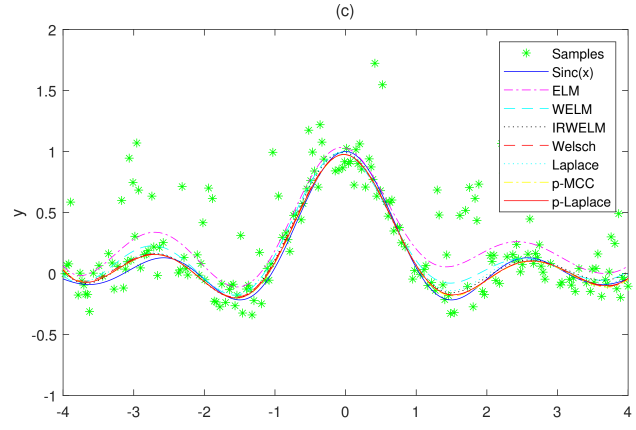
(a)



(b)



(c)



(d)

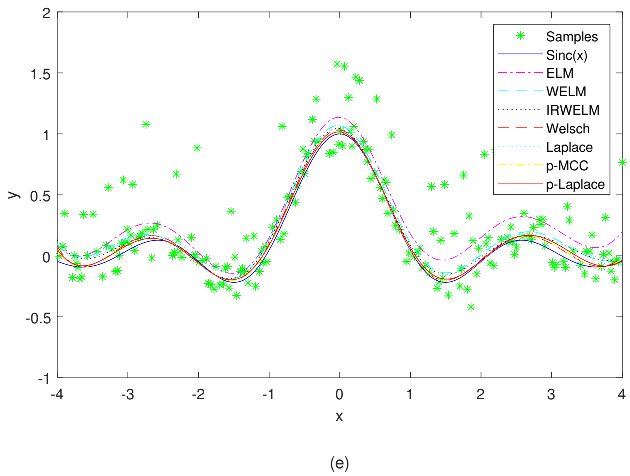


Fig. 5. Experiment results on artificial datasets with diffrent outliers levels: a(0%), b(10%), c(20%), d(30%), e(40%).

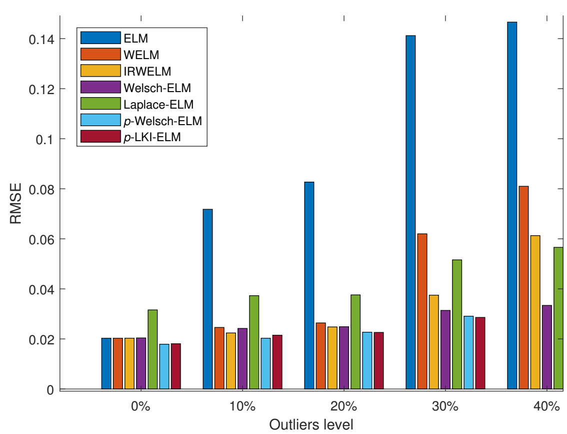


Fig. 6. RMSE of seven algorithms with different outliers levels on the artificial dataset.

slightly higher than that of p-Welsch-ELM, which ranks second among the seven methods. When the outliers level is 10%, the RMSE of ELM increases more, while the increases of p-LKI-ELM and p-Welsch-ELM are smaller compared to the other four methods and p-LKI-ELM is still ranked second among the seven methods. When the outliers level is 20%, p-LKI-ELM has the smallest RMSE among the seven methods and ranks first among the seven methods. It can be seen that the increase in RMSE of p-LKI-ELM becomes smaller and smaller as the outliers level increases. When the outliers levels are 30% and 40%, the RMSE of p-LKI-ELM is still the smallest among the seven methods and ranks first among the seven methods, and it can be concluded that the robustness of p-LKI-ELM is the best. From the aspect of the increase of RMSE, the increase of RMSE of ELM during the increase of outliers levels from 0% to 40% is the largest, while the increase of RMSE of p-LKI-ELM is the smallest, which indicates that the stability of p-LKI-ELM is the best among the seven methods.

C. Experiments on Benchmark Datasets

1) Datasets description: To further test the performance of p-LKI-ELM, the seven methods are experimented on eighteen datasets and the results are analyzed. The information on the selected dataset is shown in Table I. A portion of the datasets is randomly chosen as the training samples, while the rest is used as the test samples. To test the robustness of the model with outliers, we set 10%, 20%, 30% and 40% outliers levels, respectively.

TABLE I. BENCHMARK REGRESSION DATASETS

Dataset	Feature	Training Samples	Test Samples
Yacht	6	200	108
Servo	4	120	47
Pyrim	27	40	34
Heart	12	200	99
Fish	6	500	408
Diabetes	2	20	23
Daily	12	40	20
Concrete	8	600	430
Autompg	7	200	192
Aquatic	8	300	246
Bodyfat	14	160	92
Pollution	15	60	40
Housing	13	300	206
MG	6	700	685
Abalone	7	2000	2177
Air	6	740	313
Wine	12	1000	599
ALE	5	80	27

2) Experimental results and analysis: The RMSE values and standard deviations of the seven algorithms across various outliers levels on the nine and nine benchmark datasets are given in Tables II and III, respectively. When the outliers level is 0%, p-LKI-ELM has the lowest RMSE on four datasets in Table II and ranks first together with p-Welsch-ELM, and it has the lowest RMSE values on three datasets in Table III, ranking second among the seven methods. Overall, p-LKI-ELM ranks second among these seven methods on fifteen datasets. When the outliers level is 10%, p-LKI-ELM achieves the smallest RMSE values on seven datasets in Table II and ranks first; seven of the datasets in Table III reaches the smallest RMSE values and ranks first. In total, p-LKI-ELM ranks first among these seven methods on eighteen datasets. This shows that the rank of p-LKI-ELM increases with the addition of outliers, and p-LKI-ELM is least affected by outliers compared to the other methods. When the outliers level is 20%, p-LKI-ELM obtains the smallest RMSE value on eight datasets in Table II, the number of datasets that achieve the minimum RMSE value increases by one, and eight of the datasets in Table III win the smallest RMSE value and ranks first. With outliers levels of 30% and 40%, p-LKI-ELM achieves the smallest RMSE values on eight and seven datasets in Table II, and eight and nine datasets in Table III, respectively, and is ranked first on eighteen datasets. It can be seen that as the level of outliers increases, the number of minimum RMSE values achieved by p-LKI-ELM is increasing, which indicates that p-LKI-ELM has the best robustness compared to the other six methods. In terms of the increase of RMSE values, from outliers level of 0% to 40%, p-LKI-ELM has the lowest increase of RMSE values on all eighteen datasets among the seven methods. From the point of view of the loss function, the p-LKI loss function adopted by the p-LKI-ELM is a bounded loss function that can limit the error to a certain range and will not increase.

TABLE II. Comparisons of Seven Algorithms on Nine Benchmark Datasets

Dataset	algorithm	0%	10%	20%	30%	40%
	ELM	2.2141 ± 0.2392	6.8297 ± 0.5524	8.2801 ± 0.8043	12.5289 ± 1.2580	13.8917 ± 1.1349
	WELM	2.3363 ± 0.2924	3.3202 ± 0.8706	5.8744 ± 0.6280	9.6205 ± 1.5650	12.9183 ± 1.0998
	IRWELM	3.9779 ± 1.9951	2.8292 ± 0.3470	4.1704 ± 1.3173	6.7085 ± 1.0380	11.9374 ± 1.6171
Yacht	Welsch-ELM	1.0205 ± 0.1877	2.2840 ± 0.3704	3.3771 ± 1.2074	5.4644 ± 0.3629	6.3030 ± 1.4922
	Laplace-ELM	0.9999 ± 0.2344	2.0123 ± 0.5708	3.5665 ± 0.7810	6.4055 ± 1.7824	7.4523 ± 1.2537
	p-Welsch-ELM	0.9049 ± 0.2073	2.0516 ± 1.5746	3.2096 ± 0.7216	5.4282 ± 0.9047	6.3030 ± 1.4922
	p-LKI-ELM	**0.9006 ± 0.2069**	**1.8326 ± 0.5807**	**2.6563 ± 0.5513**	**5.3287 ± 0.3406**	**6.1246 ± 1.5251**
	ELM	11.6573 ± 5.4691	51.8964 ± 10.1203	76.6575 ± 13.6884	77.7380 ± 15.7061	78.8797 ± 13.6214
	WELM	13.8204 ± 7.0593	35.5050 ± 8.3419	49.1082 ± 14.6976	75.8675 ± 14.8694	80.6479 ± 10.6494
	IRWELM	14.7650 ± 7.9147	27.2927 ± 17.5360	34.9433 ± 12.7568	75.8575 ± 14.8827	80.6479 ± 10.6494
Daily	Welsch-ELM	11.6572 ± 5.4708	20.2666 ± 10.7994	26.9522 ± 15.8516	30.0938 ± 7.7563	34.4336 ± 9.9401
	Laplace-ELM	11.6621 ± 5.4612	17.6412 ± 9.8118	21.1682 ± 11.2175	30.6572 ± 15.2610	44.5214 ± 26.7513
	p-Welsch-ELM	**11.6526 ± 5.4708**	19.1776 ± 10.0054	25.5457 ± 17.0197	26.8373 ± 10.8504	30.9964 ± 8.9880
	p-LKI-ELM	11.6527 ± 5.4763	**16.9423 ± 8.4695**	**19.5609 ± 10.7949**	**25.6105 ± 11.4699**	**30.0912 ± 9.7413**
	ELM	2.8782 ± 0.1462	4.1298 ± 0.2136	5.9220 ± 0.3457	7.9565 ± 0.2425	7.9213 ± 0.2472
	WELM	2.8711 ± 0.1987	2.9047 ± 0.1526	3.3047 ± 0.3495	7.6725 ± 0.8660	7.9822 ± 0.2633
	IRWELM	2.9500 ± 0.1882	**2.8651 ± 0.0991**	**2.8698 ± 0.1202**	7.0622 ± 0.5588	7.9822 ± 0.2633
Autompg	Welsch-ELM	2.8680 ± 0.1748	2.8761 ± 0.1119	2.8890 ± 0.1309	2.9591 ± 0.2347	3.0852 ± 0.1235
	Laplace-ELM	2.9393 ± 0.2624	2.9237 ± 0.1281	2.9375 ± 0.1443	2.9645 ± 0.1629	3.1562 ± 0.2442
	p-Welsch-ELM	2.8664 ± 0.1796	2.8696 ± 0.0961	2.8780 ± 0.1275	2.9503 ± 0.2406	3.0112 ± 0.1286
	p-LKI-ELM	**2.8658 ± 0.1870**	2.8653 ± 0.1020	2.8739 ± 0.1251	**2.9390 ± 0.2123**	**2.9928 ± 0.1513**
	ELM	**0.3850 ± 0.0211**	0.3871 ± 0.0236	0.3957 ± 0.0222	0.4088 ± 0.0286	0.4307 ± 0.0258
	WELM	0.3855 ± 0.0206	0.3862 ± 0.0218	0.3932 ± 0.0225	0.4061 ± 0.0262	0.4290 ± 0.0261
	IRWELM	0.3862 ± 0.0188	0.3865 ± 0.0215	0.3929 ± 0.0220	0.4049 ± 0.0252	0.4288 ± 0.0270
Heart	Welsch-ELM	0.3856 ± 0.0207	0.3847 ± 0.0206	0.3879 ± 0.0188	0.3940 ± 0.0221	0.4105 ± 0.0261
	Laplace-ELM	0.4255 ± 0.0309	0.4032 ± 0.0306	0.3972 ± 0.0197	0.3988 ± 0.0242	0.4183 ± 0.0340
	p-Welsch-ELM	0.3852 ± 0.0201	**0.3845 ± 0.0215**	0.3876 ± 0.0207	0.3937 ± 0.0235	0.4062 ± 0.0244
	p-LKI-ELM	0.3854 ± 0.0202	**0.3845 ± 0.0219**	**0.3874 ± 0.0200**	**0.3930 ± 0.0235**	**0.4056 ± 0.0256**
	ELM	0.0043 ± 0.0019	0.0210 ± 0.0018	0.0214 ± 0.0022	0.0337 ± 0.0062	0.0542 ± 0.0081
	WELM	0.0031 ± 0.0015	0.0071 ± 0.0017	0.0093 ± 0.0032	0.0337 ± 0.0062	0.0542 ± 0.0081
	IRWELM	0.0030 ± 0.0017	0.0038 ± 0.0021	0.0043 ± 0.0015	0.0337 ± 0.0062	0.0542 ± 0.0081
Bodyfat	Welsch-ELM	0.0033 ± 0.0017	0.0040 ± 0.0018	0.0040 ± 0.0016	0.0046 ± 0.0016	0.0065 ± 0.0020
	Laplace-ELM	0.0029 ± 0.0018	0.0032 ± 0.0017	0.0034 ± 0.0017	0.0038 ± 0.0024	0.0042 ± 0.0022
	p-Welsch-ELM	0.0029 ± 0.0018	0.0037 ± 0.0020	0.0038 ± 0.0019	0.0038 ± 0.0021	0.0046 ± 0.0016
	p-LKI-ELM	**0.0028 ± 0.0017**	**0.0029 ± 0.0016**	**0.0032 ± 0.0015**	**0.0035 ± 0.0017**	**0.0038 ± 0.0016**
	ELM	0.1111 ± 0.0199	0.1362 ± 0.0204	0.1425 ± 0.0188	0.1553 ± 0.0145	0.1569 ± 0.0274
	WELM	0.1061 ± 0.0295	0.1102 ± 0.0326	0.1227 ± 0.0303	0.1411 ± 0.0264	0.1551 ± 0.0237
	IRWELM	0.1065 ± 0.0293	0.1071 ± 0.0332	0.1143 ± 0.0389	0.1384 ± 0.0185	0.1553 ± 0.0241
Pyrim	Welsch-ELM	0.1035 ± 0.0285	0.1054 ± 0.0315	0.1115 ± 0.0344	0.1112 ± 0.0348	0.1198 ± 0.0342
	Laplace-ELM	0.1044 ± 0.0245	0.1063 ± 0.0298	0.1127 ± 0.0318	0.1139 ± 0.0324	0.1283 ± 0.0471
	p-Welsch-ELM	**0.1021 ± 0.0275**	0.1053 ± 0.0308	0.1096 ± 0.0341	0.1111 ± 0.0346	0.1150 ± 0.0342
	p-LKI-ELM	0.1024 ± 0.0280	**0.1043 ± 0.0314**	**0.1091 ± 0.0334**	**0.1104 ± 0.0361**	**0.1144 ± 0.0363**
	ELM	0.5838 ± 0.0937	0.6523 ± 0.1043	0.6907 ± 0.1219	0.6646 ± 0.1239	0.6812 ± 0.1315
	WELM	0.5821 ± 0.0906	0.5889 ± 0.1088	0.6341 ± 0.1039	0.7033 ± 0.1333	0.6974 ± 0.1316
	IRWELM	0.5820 ± 0.0905	**0.5745 ± 0.0953**	0.5945 ± 0.0970	0.7262 ± 0.1364	0.6974 ± 0.1316
Diabetes	Welsch-ELM	0.5809 ± 0.0921	0.5752 ± 0.0927	0.5774 ± 0.0964	0.5870 ± 0.0988	0.5881 ± 0.1032
	Laplace-ELM	0.6022 ± 0.1005	0.5995 ± 0.0774	0.6303 ± 0.1037	0.6373 ± 0.1172	0.6740 ± 0.1041
	p-Welsch-ELM	**0.5748 ± 0.0804**	0.5746 ± 0.0886	**0.5742 ± 0.0942**	0.5852 ± 0.0966	0.5857 ± 0.0765
	p-LKI-ELM	0.5793 ± 0.0941	0.5746 ± 0.0971	**0.5742 ± 0.0933**	**0.5844 ± 0.0993**	**0.5826 ± 0.0831**
	ELM	0.6000 ± 0.0944	1.0198 ± 0.1596	1.1133 ± 0.1797	1.3316 ± 0.1572	1.4877 ± 0.1583
	WELM	0.5595 ± 0.1574	0.7789 ± 0.1833	0.8731 ± 0.1778	1.0145 ± 0.1605	1.4920 ± 0.1660
	IRWELM	0.5920 ± 0.1602	0.7537 ± 0.2060	0.7348 ± 0.2559	0.8599 ± 0.2012	1.4920 ± 0.1660
Servo	Welsch-ELM	0.5547 ± 0.2082	0.6610 ± 0.1921	0.7112 ± 0.2313	0.7123 ± 0.2084	0.7944 ± 0.1993
	Laplace-ELM	0.5720 ± 0.1905	0.6683 ± 0.1786	0.7086 ± 0.1834	0.7062 ± 0.2063	0.9396 ± 0.1930
	p-Welsch-ELM	**0.5514 ± 0.2116**	0.6464 ± 0.1882	0.6863 ± 0.2364	0.6959 ± 0.2038	0.7686 ± 0.2307
	p-LKI-ELM	0.5523 ± 0.2092	**0.6446 ± 0.1886**	**0.6838 ± 0.1901**	**0.6890 ± 0.2274**	**0.7619 ± 0.2352**
	ELM	35.4759 ± 6.8079	59.5479 ± 7.5005	63.7456 ± 8.5865	58.4852 ± 8.5803	66.3481 ± 10.7665
	WELM	36.8257 ± 4.8361	40.6244 ± 6.5749	48.3622 ± 8.1037	58.4852 ± 8.5803	66.3481 ± 10.7665
	IRWELM	36.8908 ± 5.6836	37.5958 ± 4.9581	39.4799 ± 5.9682	58.4852 ± 8.5803	66.3481 ± 10.7665
Pollution	Welsch-ELM	35.4153 ± 6.5356	36.2262 ± 5.9163	36.5977 ± 5.8775	38.1440 ± 5.8735	37.6558 ± 5.9691
	Laplace-ELM	35.8060 ± 6.3745	**35.8537 ± 5.6511**	36.7654 ± 10.1868	37.9770 ± 7.3431	38.3577 ± 9.0325
	p-Welsch-ELM	35.0482 ± 4.5809	35.8739 ± 6.2515	36.4647 ± 6.3492	37.7237 ± 5.9320	**37.0502 ± 6.6151**
	p-LKI-ELM	**34.2035 ± 6.1478**	**35.8537 ± 5.6511**	**36.2593 ± 9.4523**	**37.6604 ± 6.6825**	37.3199 ± 6.7331

TABLE III. COMPARISONS OF SEVEN ALGORITHMS ON NINE BENCHMARK DATASETS

Dataset	algorithm	0%	10%	20%	30%	40%
	ELM	0.9587 ± 0.1258	1.0734 ± 0.2356	1.3763 ± 0.1348	1.4139 ± 0.2314	1.4093 ± 0.2659
	WELM	0.9653 ± 0.2691	0.9737 ± 0.1645	0.9986 ± 0.3145	1.3462 ± 0.5896	1.4150 ± 0.3145
	IRWELM	0.9678 ± 0.2154	0.9757 ± 0.3245	0.9798 ± 0.1246	1.2366 ± 0.1235	1.4216 ± 0.2369
Fish	Welsch-ELM	0.9578 ± 0.3145	0.9704 ± 0.3214	0.9685 ± 0.4563	0.9757 ± 0.3145	0.9862 ± 0.3156
	Laplace-ELM	0.9625 ± 0.2145	0.9652 ± 0.2312	0.9657 ± 0.3112	0.9898 ± 0.3145	1.0323 ± 0.2136
	p-Welsch-ELM	$\mathbf{0.9563 \pm 0.2365}$	0.9687 ± 0.1345	0.9654 ± 0.3145	0.9736 ± 0.3302	0.9857 ± 0.2230
	p-LKI-ELM	0.9567 ± 0.2563	$\mathbf{0.9638 \pm 0.3145}$	$\mathbf{0.9623 \pm 0.1146}$	$\mathbf{0.9734 \pm 0.2698}$	$\mathbf{0.9850 \pm 0.2423}$
	ELM	1.1874 ± 0.0688	1.3137 ± 0.0589	1.6137 ± 0.0914	1.6683 ± 0.0636	1.7233 ± 0.0962
	WELM	1.1942 ± 0.0658	1.2053 ± 0.0455	1.2843 ± 0.0648	1.5869 ± 0.0813	1.6735 ± 0.0591
	IRWELM	1.2025 ± 0.0621	1.2046 ± 0.0540	1.2489 ± 0.0607	1.4864 ± 0.0901	1.6735 ± 0.0591
Aquatic	Welsch-ELM	$\mathbf{1.1871 \pm 0.0731}$	1.1972 ± 0.0497	1.2060 ± 0.0566	1.2160 ± 0.0518	1.2347 ± 0.0436
	Laplace-ELM	1.2011 ± 0.0590	1.1966 ± 0.0539	1.2141 ± 0.0544	1.2432 ± 0.0522	1.3141 ± 0.1120
	p-Welsch-ELM	$\mathbf{1.1871 \pm 0.0731}$	1.1958 ± 0.0514	$\mathbf{1.2051 \pm 0.0510}$	1.2154 ± 0.0544	1.2277 ± 0.0596
	p-LKI-ELM	$\mathbf{1.1871 \pm 0.0729}$	$\mathbf{1.1954 \pm 0.0520}$	$\mathbf{1.2051 \pm 0.0509}$	$\mathbf{1.2146 \pm 0.0531}$	$\mathbf{1.2250 \pm 0.0446}$
	ELM	3.2563 ± 0.2501	5.3252 ± 0.4442	7.3845 ± 0.3468	8.9692 ± 0.9645	9.1478 ± 0.4115
	WELM	3.3422 ± 0.4405	3.7507 ± 0.6523	4.7048 ± 0.5636	8.3753 ± 0.6235	9.0570 ± 0.4625
	IRWELM	3.4454 ± 0.4456	3.6482 ± 0.6741	4.1057 ± 0.5963	7.3460 ± 0.6623	9.0570 ± 0.5624
Housing	Welsch-ELM	3.2489 ± 0.3112	3.5236 ± 0.3326	3.8161 ± 0.3417	4.1932 ± 0.3918	4.7868 ± 0.3721
	Laplace-ELM	3.4086 ± 0.3056	3.6160 ± 0.3102	3.8380 ± 0.3623	4.4099 ± 0.3515	5.0003 ± 0.3120
	p-Welsch-ELM	$\mathbf{3.2404 \pm 0.3215}$	$\mathbf{3.5088 \pm 0.3625}$	3.7979 ± 0.3775	4.1684 ± 0.3023	4.1691 ± 0.3402
	p-LKI-ELM	3.2416 ± 0.3003	$\mathbf{3.5088 \pm 0.3625}$	$\mathbf{3.7709 \pm 0.3569}$	$\mathbf{4.1486 \pm 0.4021}$	$\mathbf{4.1506 \pm 0.3654}$
	ELM	6.1381 ± 0.2968	9.7657 ± 0.5098	11.7639 ± 0.5329	16.4077 ± 0.6302	16.8087 ± 0.2882
	WELM	6.2327 ± 0.3840	7.7370 ± 0.9532	8.4606 ± 0.2148	12.2520 ± 0.8875	16.9696 ± 0.3332
	IRWELM	6.4228 ± 0.3245	7.5245 ± 0.3625	7.9860 ± 0.4632	10.3756 ± 0.2564	16.9696 ± 0.4463
Concrete	Welsch-ELM	6.1330 ± 0.3456	6.6900 ± 0.5120	7.6898 ± 0.4326	8.2634 ± 0.3694	8.8897 ± 0.5213
	Laplace-ELM	6.6114 ± 0.3412	7.5818 ± 0.4362	8.1849 ± 0.4423	8.9050 ± 0.2631	10.0165 ± 0.2543
	p-Welsch-ELM	6.1241 ± 0.5023	6.6893 ± 0.4312	7.5801 ± 0.4412	$\mathbf{7.9668 \pm 0.4063}$	8.1160 ± 0.3316
	p-LKI-ELM	$\mathbf{6.1177 \pm 0.3321}$	$\mathbf{6.6817 \pm 0.4120}$	$\mathbf{7.4485 \pm 0.3216}$	7.9980 ± 0.4521	$\mathbf{8.1131 \pm 0.5623}$
	ELM	0.2267 ± 0.0031	0.2277 ± 0.0039	0.2291 ± 0.0043	0.2267 ± 0.0033	0.2362 ± 0.0054
	WELM	0.2267 ± 0.0031	0.2268 ± 0.0030	0.2267 ± 0.0031	0.2312 ± 0.0047	0.2358 ± 0.0053
	IRWELM	0.2267 ± 0.0031	0.2267 ± 0.0032	0.2268 ± 0.0031	0.2335 ± 0.0068	0.2358 ± 0.0053
MG	Welsch-ELM	0.2267 ± 0.0031	0.2267 ± 0.0031	0.2266 ± 0.0030	0.2266 ± 0.0032	$\mathbf{0.2265} \pm 0.0032$
	Laplace-ELM	0.2266 ± 0.0032	0.2273 ± 0.0028	0.2293 ± 0.0044	0.2268 ± 0.0034	0.2341 ± 0.0032
	p-Welsch-ELM	$\mathbf{0.2264 \pm 0.0031}$	$\mathbf{0.2265 \pm 0.0033}$	$\mathbf{0.2264 \pm 0.0031}$	$\mathbf{0.2265 \pm 0.0032}$	$\mathbf{0.2265 \pm 0.0032}$
	p-LKI-ELM	$\mathbf{0.2266 \pm 0.0032}$	$\mathbf{0.2265 \pm 0.0032}$	$\mathbf{0.2264 \pm 0.0031}$	$\mathbf{0.2265 \pm 0.0032}$	$\mathbf{0.2265 \pm 0.0032}$
	ELM	2.1698 ± 0.0371	2.6821 ± 0.0588	3.1988 ± 0.0538	3.2092 ± 0.0408	3.2703 ± 0.0709
	WELM	2.1869 ± 0.0359	2.1707 ± 0.0335	2.2428 ± 0.0449	3.2110 ± 0.0423	3.1981 ± 0.0340
	IRWELM	2.2253 ± 0.0456	2.1831 ± 0.0412	2.1948 ± 0.0563	2.8742 ± 0.0321	3.2033 ± 0.0364
Abalone	Welsch-ELM	2.1689 ± 0.0456	2.1769 ± 0.0356	2.1812 ± 0.0349	2.1978 ± 0.0502	2.1988 ± 0.0356
	Laplace-ELM	2.1752 ± 0.0563	2.1802 ± 0.0421	2.1901 ± 0.0314	2.1893 ± 0.0513	2.1864 ± 0.0419
	p-Welsch-ELM	$\mathbf{2.1688 \pm 0.0318}$	$\mathbf{2.1691 \pm 0.0526}$	2.1724 ± 0.0536	2.1758 ± 0.0543	2.1781 ± 0.0697
	p-LKI-ELM	2.1692 ± 0.0412	2.1696 ± 0.0316	$\mathbf{2.1710 \pm 0.0412}$	$\mathbf{2.1757 \pm 0.0346}$	$\mathbf{2.1777 \pm 0.0327}$
	ELM	2.6473 ± 0.0001	7.8888 ± 0.0102	7.9453 ± 0.0012	10.6465 ± 0.0301	15.0824 ± 0.0014
	WELM	2.6352 ± 0.0000	2.8410 ± 0.0002	3.0812 ± 0.0001	10.6465 ± 0.0023	15.0824 ± 0.0003
	IRWELM	2.6212 ± 0.0001	2.7396 ± 0.0016	2.7198 ± 0.0001	10.6465 ± 0.0012	15.0824 ± 0.0004
Air	Welsch-ELM	2.6210 ± 0.0203	2.7394 ± 0.0005	2.6902 ± 0.0101	2.6768 ± 0.0000	2.9801 ± 0.0301
	Laplace-ELM	2.6913 ± 0.0012	2.6771 ± 0.0013	2.7111 ± 0.0502	2.6769 ± 0.0107	2.9761 ± 0.0005
	p-Welsch-ELM	$\mathbf{2.6208 \pm 0.0000}$	2.6523 ± 0.0001	2.6874 ± 0.0031	2.6615 ± 0.0004	2.9798 ± 0.0015
	p-LKI-ELM	2.6213 ± 0.0013	$\mathbf{2.6516 \pm 0.0000}$	$\mathbf{2.6742 \pm 0.0001}$	$\mathbf{2.6569 \pm 0.0015}$	$\mathbf{2.9653 \pm 0.0205}$
	ELM	0.6482 ± 0.0226	0.8226 ± 0.0202	0.8243 ± 0.0205	0.8263 ± 0.0175	0.8686 ± 0.0711
	WELM	0.6506 ± 0.0230	0.6565 ± 0.0230	0.6852 ± 0.0223	0.8349 ± 0.0170	0.8686 ± 0.0711
	IRWELM	0.6515 ± 0.0234	0.6525 ± 0.0211	0.6549 ± 0.0217	0.8349 ± 0.0170	0.8686 ± 0.0711
Wine	Welsch-ELM	0.6423 ± 0.0031	0.6429 ± 0.0031	0.6512 ± 0.0030	0.7698 ± 0.0032	0.8027 ± 0.0032
	Laplace-ELM	0.6504 ± 0.0055	0.6513 ± 0.0029	0.6516 ± 0.0034	0.7742 ± 0.0034	0.8014 ± 0.0035
	p-Welsch-ELM	$\mathbf{0.6412 \pm 0.0031}$	$\mathbf{0.6416 \pm 0.0013}$	$\mathbf{0.6489 \pm 0.0038}$	0.7685 ± 0.0032	0.7746 ± 0.0006
	p-LKI-ELM	0.6414 ± 0.0056	0.6420 ± 0.0030	0.6500 ± 0.0031	$\mathbf{0.7644 \pm 0.0032}$	$\mathbf{0.7695 \pm 0.0012}$
	ELM	0.1325 ± 0.0001	0.2511 ± 0.0008	0.3261 ± 0.0102	0.3467 ± 0.0408	0.3464 ± 0.0709
	WELM	0.1427 ± 0.0009	0.1532 ± 0.0005	0.1715 ± 0.0001	0.3474 ± 0.0003	0.3456 ± 0.0000
	IRWELM	0.1342 ± 0.0006	0.1430 ± 0.0002	0.1653 ± 0.0003	0.3464 ± 0.0001	0.3456 ± 0.0004
ALE	Welsch-ELM	0.1281 ± 0.0000	0.1371 ± 0.0001	0.1371 ± 0.0009	0.1438 ± 0.0002	0.1538 ± 0.0001
	Laplace-ELM	0.1357 ± 0.0000	0.1410 ± 0.0001	0.1549 ± 0.0004	0.1450 ± 0.0003	0.1754 ± 0.0001
	p-Welsch-ELM	$\mathbf{0.1279 \pm 0.0008}$	0.1366 ± 0.0000	0.1329 ± 0.0000	0.1339 ± 0.0003	0.1503 ± 0.0007
	p-LKI-ELM	0.1289 ± 0.0002	$\mathbf{0.1356 \pm 0.0006}$	$\mathbf{0.1323 \pm 0.0002}$	$\mathbf{0.1303 \pm 0.0006}$	$\mathbf{0.1467 \pm 0.0001}$

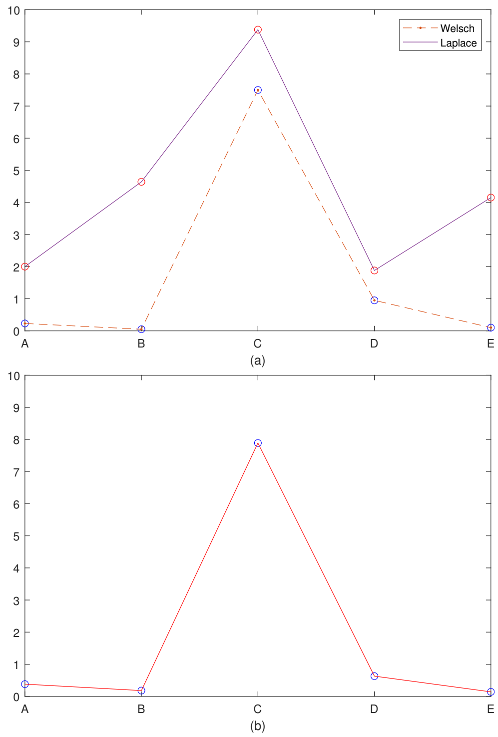


Fig. 7. (a) The reduction of RMSE for p-Welsch-ELM and p-LKI-ELM; (b) The RMSE reduction of p-LKI-ELM relative to p-Welsch-ELM.

Infinitely with the increase of outliers levels. However the l_2 loss function employed by ELM, WELM and IRWELM is an unbounded loss function that increases the error of the model as the levels of outliers increases. Compared with the loss function induced by Gaussian kernel, p-LKI is induced by Laplace kernel and the exponential part is still bounded, which makes the p-LKI-ELM the most optimal among these seven methods by choosing a suitable p.

In order to observe more intuitively the improvement effect of the loss function after the introduction of p-order and compare the effect of the two kernel functions after the introduction of p-order, we graph the experimental data.

Fig. 7 shows the reduction of p-Welsch-ELM relative to Welsch-ELM and p-LKI-ELM relative to the RMSE of Laplace-ELM on five benchmark datasets as illustrated in (a), the x-axis A, B, C, D and E represent the five datasets (Autompg, Heart, Bodyfat, Pyrim and Diabetes), respectively. The y-axis represents the reduction in RMSE (%), and Fig. 7 indicates that the RMSE reduction of p-LKI loss function relative to Laplace loss function is higher than the reduction of p-Welsch loss function relative to the RMSE of Welsch loss function. It shows that the loss function induced by the Laplace kernel is more effective for the p-order than Gaussian kernel, which indicates that the former is more suitable for the p-order.

Fig. 7(b) suggests the reduction of RMSE of the proposed model relative to the p-Welsch-ELM model on five benchmark datasets. It can be seen from (b) that the y-axis is always more significant than 0, which means that the reduction (%) of the RMSE of p-LKI-ELM relative to p-Welsch-ELM is greater than 0, which indicates that the prediction accuracy of the loss function induced by the Laplace kernel is higher than that of the Gaussian kernel.

V. Discussion

The proposed model is compared with six other models on both artificial datasets and eighteen benchmark datasets. Experimental results demonstrate that p-LKI-ELM achieves superior performance on the majority of datasets. Moreover, as the proportion of outliers increases, p-LKI-ELM is less affected compared to the other models, confirming its stronger robustness. In the future, research could be conducted from the perspective of sparsity.

VI. Conclusion

Influenced by kernel learning and correntropy learning, we propose a new loss function (p-LKI) to solve the regression problem. The proposed method is experimented on artificial datasets and benchmark datasets. In addition, the performance of the proposed method is evaluated with different outliers levels. The main work is summarized as follows:

(1) We propose a new robust loss function (p-LKI loss), which combines the advantages of the p-order loss function and the correntropy loss function. Therefore, it is insensitivity to noise and outliers. (2) The proposed method is compared against ELM, WELM, IRWELM, Welsch-ELM, Laplace-ELM, and p-Welsch-ELM on artificial datasets and eighteen benchmark datasets. The experimental results indicate that the proposed method consistently outperforms the other six models in both cases. Furthermore, the results demonstrate the superior robustness of the proposed method. (3) By comparing the reduction of p-LKI loss function induced by Laplace kernel compared with the RMSE of Laplace loss, and the reduction of RMSE induced by Gaussian kernel compared with Welsch loss, the results show that the reduction of p-LKI loss function relative to Laplace loss function is higher than the latter, which indicates that the loss function induced by Laplace kernel at order p is better than the loss function induced by Gaussian kernel at order p. By comparing the reduction in RMSE of p-LKI loss function relative to p-Welsch loss, the results demonstrate that the robustness of the p-LKI loss function is higher than that of the p-Welsch loss.

In addition, on the one hand, the number of hidden layer nodes and the activation function used in this paper are fixed, we can set a different number of hidden layer nodes and other activation function to observe the effect on the performance of the model later; on the other hand, this paper uses an iterative reweighting algorithm to solve the model, and a new algorithm can be designed to improve the training speed of the model in the future.

Acknowledgments

The work was supported by the National Science Foundation of China under Grant nos.61907033, and the Postdoctoral Science Foundation of China under Grant no.2018M642129, and the Postgraduate Innovation and Practice Ability Development Fund of Xi'an shiyou University under Grant no.YCS23213166.

References

[1] Huang G, Song S. Trends in extreme learning machines: A review[J]. Neural Networks, 2015, 61: 32-48.

[2] Baksalary O, Trenkler G. On a generalized core inverse[J]. Applied Mathematics and Computation, 2014, 236: 450-457.

[3] Lu F, Liu Y, Qi Y, et al. Short-term load forecasting based on optimized learning machine using improved genetic algorithm[J]. Journal of North China Electric Power University, 2018, 45(06): 1-7.

[4] Chen X, Liu Y, Zhang J. Short-term power load forecasting based on intelligent concentrator [J].Journal of Power System and Automation, 2020,32 (06) : 140-145.

[5] Qi Y, Fan J, Liu L,et al. Fault diagnosis of wind turbine bearings based on morphological fractal and extreme learning machine. Journal of Solar Energy, 41 (6),2020, 102-112.

[6] Song J, Shi R, et al. KELM based diagnostics for air vehicle faults[J]. Journal of Tsinghua University (Science and Technology), 2020, 60(10): 795-803.

[7] Song Y, He B, Zhao Y, et al. Segmentation of sidescan sonar imagery using markov random fields and extreme learning machine[J]. IEEE Journal of Oceanic Engineering, 2018, 44(2): 502-513.

[8] Cvetković S, Stojanović M B, Nikolić S V. Hierarchical ELM ensembles for visual descriptor fusion[J]. Information Fusion, 2018, 41: 16-24.

[9] Manoharan J. Study of variants of extreme learning machine (ELM) brands and its performance measure on classification algorithm[J]. Journal of Soft Computing Paradigm (JSCP), 2021, 3(02): 83-95.

[10] Deng W, Zheng Q, Chen L. Regularized extreme learning machine[C]//2009 IEEE symposium on computational intelligence and data mining. IEEE, 2009: 389-395.

[11] Yang Y, Zhou H, Gao Y. Robust penalized extreme learning machine regression with applications in wind speed forecasting[J]. Neural Computing and Applications, 2022, 34(1): 391-407.

[12] Zhang K, Luo M. Outlier-robust extreme learning machine for regression problems[J]. Neurocomputing, 2015, 151: 1519-1527.

[13] Ren Z, Yang L. Robust extreme learning machines with different loss functions[J]. Neural Processing Letters, 2019, 49: 1543-1565.

[14] Chen K, Lv Q, Lu Y, et al. Robust regularized extreme learning machine for regression using iteratively reweighted least squares[J]. Neurocomputing, 2017, 230: 345-358.

[15] Faccini D, Maggioni F, Potra F. Robust and distributionally robust optimization models for linear support vector machine[J]. Computers Operations Research, 2022, 147: 105930.

[16] Shen X, Niu L, Qi Z, et al. Support vector machine classifier with truncated pinball loss[J]. Pattern Recognition, 2017, 68: 199-210.

[17] Ye Y, Gao J, Shao Y. Robust support vector regression with generic quadratic nonconvex insensitive loss[J]. Applied Mathematical Modelling, 2020, 82: 235-251.

[18] Jiang W, Nie F, Huang H. Robust dictionary learning with capped l1-norm[C]//Twenty-fourth international joint conference on artificial intelligence. 2015, 232: 341-228.

[19] Liu W, Pokharel P P, Principe J C. Correntropy: Properties and applications in non-Gaussian signal processing[J]. IEEE Transactions on signal processing, 2007, 55(11): 5286-5298.

[20] Zhang X, Liu C, Suen C. Towards robust pattern recognition: A review[J]. Proceedings of the IEEE, 2020, 108(6): 894-922.

[21] Xing H, Wang X. Training extreme learning machine via regularized correntropy criterion[J]. Neural Computing and Applications, 2013, 23: 1977-1986.

[22] Radhakrishnan A, Belkin M, Uhler C. Wide and deep neural networks achieve consistency for classification[J]. Proceedings of the National Academy of Sciences, 2023, 120(14): 22087-79120.

[23] Feng Y, Yang Y, Huang X, et al. Robust support vector machines for classification with nonconvex and smooth losses[J]. Neural computation, 2016, 28(6): 1217-1247.

[24] Chen B, Wang X, Li Y, et al. Maximum correntropy criterion with variable center[J]. IEEE Signal Processing Letters, 2019, 26(8): 1212-1216.

[25] Yang L, Ren Z, Wang Y, et al. A robust regression framework with laplace kernel-induced loss[J]. Neural computation, 2017, 29(11): 3014-3039.

[26] Dong H, Yang L, Wang X. Robust semi-supervised support vector machines with Laplace kernel-induced correntropy loss functions[J]. Applied Intelligence, 2021, 51: 819-833.

[27] Chen B, Xing L, Wang X, et al. Robust learning with kernel mean p-power error loss[J]. IEEE Transactions on cybernetics, 2017, 48(7): 2101-2113.

[28] Chen B, Xing L, Wu Z, et al. Smoothed least mean p-power error criterion for adaptive filtering[J]. Digital Signal Processing, 2015, 40: 154-163.