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

Liutao Luo, Kuaini Wang, Qiang Lin
School of Computer Science, Xi’an Shiyou University, Xi’an, 710065, China
College of Science, Xi’an Shiyou University, Xi’an 710065, China
School of Business, Jiangnan University, Wuxi 214122, China

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, $\varepsilon$-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 function 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

www.ijacsa.thesai.org 1281 | Page
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 conclusion 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:

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

where, $\beta = [\beta_1, \beta_2, \cdots, \beta_L]^T$ 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 $Y = [y_1, y_2, \cdots, y_N]^T$, hidden layer output matrix $H = [h(x_1)^T, h(x_2)^T, \cdots, h(x_N)^T]^T$, the ELM model can be expressed as the following optimization problem [1].

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

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

$$\beta = \begin{cases} (H^TH + I/C)^{-1}H^TY, & N \geq L \\ H^T(H^TH + I/C)^{-1}Y, & N < L \end{cases}$$ (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:

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

where $k_\sigma$ 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:

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

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

MCC [20] can be expressed as,

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

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

$$\min 1 - \exp(-\frac{|q|}{\sigma})$$ (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.
Fig. 1. Comparison of $p$-LKI loss functions and their gradient functions under different $p$.

Fig. 2. Comparison of $p$-LKI loss functions and their gradient functions under different $\sigma$.

$l(q) = (1 - \exp(-|q|/\sigma))^p$  \hspace{1cm} (8)

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

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

p = 0.5, $l'(q)$ is discontinuous at zero which means $l(q)$ is not differentiable at zero.

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

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:
where the zero norm $\|q\|_0$ function can be represented as:

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

(10)

We have

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

(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 $\forall q \in \mathbb{R}^N$,

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

(12)

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

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

(13)

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

$$\lim_{\sigma \to 0^+} R_l(f) = \lim_{\sigma \to 0^+} \sum_{i=1}^{N} l(q_i)$$

$$= \lim_{\sigma \to 0^+} \sum_{i=1}^{N} (1 - \exp(-\frac{|q_i|}{\sigma}))^p = \|q\|_0$$

(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),

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

(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

$$\min_{\beta, q_i} \frac{1}{2} \|\beta\|_2^2 + C \sum_{i=1}^{N} (1 - \exp(-\frac{|q_i|}{\sigma}))^p$$

s.t. $h(x_i)\beta = y_i - q_i, i = 1, 2, \cdots, N$

(16)

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

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

(17)

where $\alpha_i$ 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,

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

(18)
In this paper, we employ an iterative reweighted algorithm to obtain the optimal hidden layer output weight $\beta$. The weight of $N$ samples can be expressed as

$$W(q) = \text{diag}(w(q_1), w(q_2), \cdots, w(q_N))$$  \hspace{1cm} (19)$$

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

$$\beta = \begin{cases} 
    H^T(\frac{1}{\sigma} + W(q)HH^T)^{-1}W(q)Y, & N < L \\
    (\frac{1}{\sigma} + H^TW(q)H)^{-1}HTW(q)Y, & N \geq L
\end{cases}$$  \hspace{1cm} (20)$$

The curve of sample weights with different parameters $\sigma$ is shown below.

![Graph showing the comparison of loss functions and their gradient functions.]
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.

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

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

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

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

B. Experimental on Artificial Datasets

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

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

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 $[y_{\text{min}}, y_{\text{max}}]$ 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
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.

![Fig. 5. Experiment results on artificial datasets with different outliers levels: a(0\%), b(10\%), c(20\%), d(30\%), e(40\%).](image)

![Fig. 6. RMSE of seven algorithms with different outliers levels on the artificial dataset.](image)

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 \( p \)-LKI-ELM is a bounded loss function that can limit the error to a certain range and will not increase.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>algorithm</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ELM</td>
<td>2.2141 ± 0.2392</td>
<td>6.8297 ± 0.5524</td>
<td>8.2801 ± 0.8043</td>
<td>12.5289 ± 1.2580</td>
<td>13.8917 ± 1.1349</td>
</tr>
<tr>
<td></td>
<td>WELM</td>
<td>2.3363 ± 0.2924</td>
<td>3.3202 ± 0.8706</td>
<td>5.8744 ± 0.6280</td>
<td>9.6205 ± 1.5650</td>
<td>12.9183 ± 1.0998</td>
</tr>
<tr>
<td></td>
<td>IRWELM</td>
<td>3.9779 ± 1.9951</td>
<td>2.8292 ± 0.3470</td>
<td>4.1704 ± 1.3173</td>
<td>6.7085 ± 1.0380</td>
<td>11.9374 ± 1.6171</td>
</tr>
<tr>
<td></td>
<td>Welsch-ELM</td>
<td>1.6205 ± 0.1877</td>
<td>1.2209 ± 0.3704</td>
<td>3.3771 ± 1.2074</td>
<td>5.4644 ± 0.3629</td>
<td>6.3030 ± 1.4922</td>
</tr>
<tr>
<td></td>
<td>Laplace-ELM</td>
<td>0.9999 ± 0.2344</td>
<td>2.0123 ± 0.5708</td>
<td>3.5665 ± 0.7810</td>
<td>4.6055 ± 1.7824</td>
<td>7.4523 ± 1.2537</td>
</tr>
<tr>
<td></td>
<td>p-Welsch-ELM</td>
<td>0.9049 ± 0.2073</td>
<td>2.0516 ± 1.5746</td>
<td>3.2096 ± 0.7216</td>
<td>5.4282 ± 0.9047</td>
<td>6.3030 ± 1.4922</td>
</tr>
<tr>
<td></td>
<td>p-LKI-ELM</td>
<td>0.9006 ± 0.2069</td>
<td>1.8326 ± 0.5807</td>
<td>2.6563 ± 0.5513</td>
<td>5.3287 ± 0.3406</td>
<td>6.1246 ± 1.5251</td>
</tr>
<tr>
<td></td>
<td>ELM</td>
<td>11.6573 ± 5.4691</td>
<td>51.8964 ± 10.1203</td>
<td>76.9567 ± 13.6884</td>
<td>77.7380 ± 15.7061</td>
<td>78.8797 ± 13.6214</td>
</tr>
<tr>
<td></td>
<td>Welsch-ELM</td>
<td>1.6205 ± 0.2073</td>
<td>2.0516 ± 1.5746</td>
<td>3.2096 ± 0.7216</td>
<td>5.4282 ± 0.9047</td>
<td>6.3030 ± 1.4922</td>
</tr>
<tr>
<td></td>
<td>Laplace-ELM</td>
<td>0.9999 ± 0.2344</td>
<td>2.0123 ± 0.5708</td>
<td>3.5665 ± 0.7810</td>
<td>4.6055 ± 1.7824</td>
<td>7.4523 ± 1.2537</td>
</tr>
<tr>
<td></td>
<td>p-Welsch-ELM</td>
<td>0.9049 ± 0.2073</td>
<td>2.0516 ± 1.5746</td>
<td>3.2096 ± 0.7216</td>
<td>5.4282 ± 0.9047</td>
<td>6.3030 ± 1.4922</td>
</tr>
<tr>
<td></td>
<td>p-LKI-ELM</td>
<td>0.9006 ± 0.2069</td>
<td>1.8326 ± 0.5807</td>
<td>2.6563 ± 0.5513</td>
<td>5.3287 ± 0.3406</td>
<td>6.1246 ± 1.5251</td>
</tr>
</tbody>
</table>

**TABLE II. COMPARISONS OF SEVEN ALGORITHMS ON NINE BENCHMARK DATASETS**
<table>
<thead>
<tr>
<th>Dataset</th>
<th>algorithm</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fish</td>
<td>Laplace-ELM</td>
<td>0.9574 ± 0.3145</td>
<td>0.9685 ± 0.3456</td>
<td>0.9757 ± 0.3145</td>
<td>0.9862 ± 0.3156</td>
<td>0.9921 ± 0.3156</td>
</tr>
<tr>
<td></td>
<td>Welsch-ELM</td>
<td>0.9574 ± 0.3145</td>
<td>0.9685 ± 0.3456</td>
<td>0.9757 ± 0.3145</td>
<td>0.9862 ± 0.3156</td>
<td>0.9921 ± 0.3156</td>
</tr>
<tr>
<td></td>
<td>WELM</td>
<td>0.9574 ± 0.3145</td>
<td>0.9685 ± 0.3456</td>
<td>0.9757 ± 0.3145</td>
<td>0.9862 ± 0.3156</td>
<td>0.9921 ± 0.3156</td>
</tr>
<tr>
<td></td>
<td>IRWELM</td>
<td>0.9574 ± 0.3145</td>
<td>0.9685 ± 0.3456</td>
<td>0.9757 ± 0.3145</td>
<td>0.9862 ± 0.3156</td>
<td>0.9921 ± 0.3156</td>
</tr>
<tr>
<td>Aquatic</td>
<td>Welsch-ELM</td>
<td>1.1871 ± 0.0731</td>
<td>1.1972 ± 0.0497</td>
<td>1.2060 ± 0.0566</td>
<td>1.2160 ± 0.0518</td>
<td>1.2347 ± 0.0436</td>
</tr>
<tr>
<td></td>
<td>LKI-ELM</td>
<td>1.1871 ± 0.0731</td>
<td>1.1972 ± 0.0497</td>
<td>1.2060 ± 0.0566</td>
<td>1.2160 ± 0.0518</td>
<td>1.2347 ± 0.0436</td>
</tr>
<tr>
<td></td>
<td>WELM</td>
<td>1.1871 ± 0.0731</td>
<td>1.1972 ± 0.0497</td>
<td>1.2060 ± 0.0566</td>
<td>1.2160 ± 0.0518</td>
<td>1.2347 ± 0.0436</td>
</tr>
<tr>
<td></td>
<td>IRWELM</td>
<td>1.1871 ± 0.0731</td>
<td>1.1972 ± 0.0497</td>
<td>1.2060 ± 0.0566</td>
<td>1.2160 ± 0.0518</td>
<td>1.2347 ± 0.0436</td>
</tr>
<tr>
<td>Concrete</td>
<td>Welsch-ELM</td>
<td>3.2489 ± 0.3112</td>
<td>3.5206 ± 0.3216</td>
<td>3.8161 ± 0.3417</td>
<td>4.1932 ± 0.3918</td>
<td>4.7686 ± 0.3721</td>
</tr>
<tr>
<td></td>
<td>LKI-ELM</td>
<td>3.2489 ± 0.3112</td>
<td>3.5206 ± 0.3216</td>
<td>3.8161 ± 0.3417</td>
<td>4.1932 ± 0.3918</td>
<td>4.7686 ± 0.3721</td>
</tr>
<tr>
<td></td>
<td>WELM</td>
<td>3.2489 ± 0.3112</td>
<td>3.5206 ± 0.3216</td>
<td>3.8161 ± 0.3417</td>
<td>4.1932 ± 0.3918</td>
<td>4.7686 ± 0.3721</td>
</tr>
<tr>
<td></td>
<td>IRWELM</td>
<td>3.2489 ± 0.3112</td>
<td>3.5206 ± 0.3216</td>
<td>3.8161 ± 0.3417</td>
<td>4.1932 ± 0.3918</td>
<td>4.7686 ± 0.3721</td>
</tr>
<tr>
<td>Abalone</td>
<td>Welsch-ELM</td>
<td>2.1752 ± 0.0563</td>
<td>2.1802 ± 0.0462</td>
<td>2.1901 ± 0.0314</td>
<td>2.1893 ± 0.0315</td>
<td>2.1864 ± 0.0419</td>
</tr>
<tr>
<td></td>
<td>LKI-ELM</td>
<td>2.1752 ± 0.0563</td>
<td>2.1802 ± 0.0462</td>
<td>2.1901 ± 0.0314</td>
<td>2.1893 ± 0.0315</td>
<td>2.1864 ± 0.0419</td>
</tr>
<tr>
<td></td>
<td>WELM</td>
<td>2.1752 ± 0.0563</td>
<td>2.1802 ± 0.0462</td>
<td>2.1901 ± 0.0314</td>
<td>2.1893 ± 0.0315</td>
<td>2.1864 ± 0.0419</td>
</tr>
<tr>
<td></td>
<td>IRWELM</td>
<td>2.1752 ± 0.0563</td>
<td>2.1802 ± 0.0462</td>
<td>2.1901 ± 0.0314</td>
<td>2.1893 ± 0.0315</td>
<td>2.1864 ± 0.0419</td>
</tr>
<tr>
<td>Air</td>
<td>Welsch-ELM</td>
<td>2.6210 ± 0.0203</td>
<td>2.7394 ± 0.0005</td>
<td>2.6769 ± 0.0017</td>
<td>2.8369 ± 0.0031</td>
<td>3.0761 ± 0.0052</td>
</tr>
<tr>
<td></td>
<td>LKI-ELM</td>
<td>2.6210 ± 0.0203</td>
<td>2.7394 ± 0.0005</td>
<td>2.6769 ± 0.0017</td>
<td>2.8369 ± 0.0031</td>
<td>3.0761 ± 0.0052</td>
</tr>
<tr>
<td></td>
<td>WELM</td>
<td>2.6210 ± 0.0203</td>
<td>2.7394 ± 0.0005</td>
<td>2.6769 ± 0.0017</td>
<td>2.8369 ± 0.0031</td>
<td>3.0761 ± 0.0052</td>
</tr>
<tr>
<td></td>
<td>IRWELM</td>
<td>2.6210 ± 0.0203</td>
<td>2.7394 ± 0.0005</td>
<td>2.6769 ± 0.0017</td>
<td>2.8369 ± 0.0031</td>
<td>3.0761 ± 0.0052</td>
</tr>
<tr>
<td>Wine</td>
<td>Welsch-ELM</td>
<td>0.6423 ± 0.0031</td>
<td>0.6429 ± 0.0031</td>
<td>0.6512 ± 0.0030</td>
<td>0.7698 ± 0.0032</td>
<td>0.8207 ± 0.0032</td>
</tr>
<tr>
<td></td>
<td>LKI-ELM</td>
<td>0.6423 ± 0.0031</td>
<td>0.6429 ± 0.0031</td>
<td>0.6512 ± 0.0030</td>
<td>0.7698 ± 0.0032</td>
<td>0.8207 ± 0.0032</td>
</tr>
<tr>
<td></td>
<td>WELM</td>
<td>0.6423 ± 0.0031</td>
<td>0.6429 ± 0.0031</td>
<td>0.6512 ± 0.0030</td>
<td>0.7698 ± 0.0032</td>
<td>0.8207 ± 0.0032</td>
</tr>
<tr>
<td></td>
<td>IRWELM</td>
<td>0.6423 ± 0.0031</td>
<td>0.6429 ± 0.0031</td>
<td>0.6512 ± 0.0030</td>
<td>0.7698 ± 0.0032</td>
<td>0.8207 ± 0.0032</td>
</tr>
<tr>
<td>ALE</td>
<td>Welsch-ELM</td>
<td>0.1325 ± 0.0001</td>
<td>0.2511 ± 0.0008</td>
<td>0.3261 ± 0.0102</td>
<td>0.3467 ± 0.0439</td>
<td>0.4364 ± 0.0709</td>
</tr>
<tr>
<td></td>
<td>LKI-ELM</td>
<td>0.1325 ± 0.0001</td>
<td>0.2511 ± 0.0008</td>
<td>0.3261 ± 0.0102</td>
<td>0.3467 ± 0.0439</td>
<td>0.4364 ± 0.0709</td>
</tr>
<tr>
<td></td>
<td>WELM</td>
<td>0.1325 ± 0.0001</td>
<td>0.2511 ± 0.0008</td>
<td>0.3261 ± 0.0102</td>
<td>0.3467 ± 0.0439</td>
<td>0.4364 ± 0.0709</td>
</tr>
<tr>
<td></td>
<td>IRWELM</td>
<td>0.1325 ± 0.0001</td>
<td>0.2511 ± 0.0008</td>
<td>0.3261 ± 0.0102</td>
<td>0.3467 ± 0.0439</td>
<td>0.4364 ± 0.0709</td>
</tr>
</tbody>
</table>

**TABLE III: COMPARISONS OF SEVEN ALGORITHMS ON NINE BENCHMARK DATASETS**
Influenced by kernel learning and correntropy learning, we propose a new loss function ($\text{p-LKI}$) 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 ($\text{p-LKI}$ 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 $\text{p-LKI}$-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 $\text{p-LKI}$ 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 $\text{p-LKI}$ 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 $\text{p-LKI}$ loss function relative to Welsch loss, the results demonstrate that the robustness of the $\text{p-LKI}$ 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 functions 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


