Robust Extreme Learning Machine with Exponential Squared Loss via DC Programming

Kuaini Wang[1], Xiaoxue Wang[2], Weicheng Zhan[3], Mingming Wang[4], Jinde Cao[5]
Southeast University, School of Mathematics, China[1,5]
Xi'an Shiyou University, College of Science, China[1]
Xi'an Shiyou University, School of Computer Science, China[2,3,4]
Yonsei University, Yonsei Frontier Lab, South Korea[5]

***Abstract*—Extreme learning machines (ELM) have recently attracted considerable attention because of its fast learning rate, simple model structure, and good generalization ability. However, classical ELM with least squares loss function is prone to overfitting and lack robustness in dealing with datasets containing noise and outliers in the real world. In this paper, inspired by the maximum correntropy criterion, an exponential squared loss function is introduced, which is nonconvex and insensitive to noise and outliers. A robust ELM with exponential squared loss (RESELM) is presented to overcome the overfitting problem. The proposed model with nonconvexity is difficult to be directly optimized. Considering the superior performance of difference of convex functions (DC) programming in solving nonconvex problems, this paper optimizes the model by expressing the objective function as a DC function and employing DC algorithm (DCA). To examine the effectiveness of the proposed algorithm in noisy environment, different levels of outliers are added to the training samples in the experiments. Experimental results on benchmark data sets with different outliers levels illustrate that the proposed RESELM achieves significant advantages in generalization performance and robustness, especially in higher outliers levels.**

***Keywords*—Extreme learning machine; exponential squared loss; DC programming; DCA; robust regression**

I. Introduction

To improve the slow learning rate of single hidden layer feedforward neural networks (SLFNs), Huang and his team proposed extreme learning machine (ELM) in 2004 [1], [2], [3]. Traditional algorithms for feedforward neural networks require iterative adjustment of all parameters in the whole network. However, the experiments in [4], [5] illustrate that input weights and hidden layer bias of SLFNs may not need to be adjusted. The hidden layer input weights and biases of ELM are determined by random generation, which reduces the number of parameters to be solved in the network by a large part. ELM can be regarded as a simple linear system with only the output weights to be solved. ELM is widely used in various real-world problems relying on its fast learning rate, simple model structure, and good generalization ability [6], [7], [8].

However, samples in real-world problems are different from the clean and uncontaminated samples used in the laboratory, which is potentially polluted in the process of both generation and acquisition [9]. Training ELM with samples containing outliers can exacerbate the discrepancy between the true and predicted values, leading to longer learning time and poorer model prediction accuracy [10], [11]. The loss function plays a crucial role in ELM training. Classical ELM employs the least squares loss function, which is easy to be solved and can improve the learning rate of the model. However, its squared effect leads to more sensitivity to outliers. When the outliers are larger, the empirical risk of the model becomes higher, which eventually affects the accuracy of the model [12].

In order to minimize the disturbance of outliers, researchers have turned to finding alternative loss functions to obtain a more robust algorithm [13], [14]. Deng et al. proposed an improved ELM based on a weighted 2-norm loss function (WELM) [15], which assigned weights to the samples depending on the residuals, improved the model's generalization. Zhang et al. developed outlier robust ELM (ORELM) by applying the 1-norm loss function to ELM [16]. The 1-norm loss function grows slower than the 2-norm loss function as the residuals increase, thus obtaining a better accuracy than ELM. Chen et al. constructed a robust ELM that can use four loss functions (1-norm, Huber, Bisquare, Welsch) [17]. The experiment's optimal accuracy was obtained by the model that used Bisquare or Welsch loss functions, both of which are nonconvex loss functions. The 1-norm and Huber loss functions are both convex loss functions and has a linear relationship with residuals, which is still not robust. When the residuals are enormous, the penalty imposed on the sample by the convex loss function can also be very large. Models usually treat outliers as normal values to reduce the large value's loss caused by outliers in empirical risk at the cost of sacrificing the model's generalization, and nonconvex loss functions can compensate for this deficiency [18].

The nonconvex loss function has a strong learning capability in terms of both generalization and robustness. The capped type of nonconvex loss functions can directly limit the maximum penalty value caused by noise and outliers and explicitly suppress the negative impact of such samples on the decision hyperplane to build models with excellent robustness [18], [19]. Different capped 2-norm loss functions are constructed in [20] and [21], respectively, which have shown stronger robustness and generalization. In recent years, with the development of information theory, Liu et al. proposed correntropy in 2007 [22], which is a measure of similarity of two sets of random variables and widely used in robust learning. Xing et al. proposed a robust ELM model based on the regularized correntropy criterion [23], which showed that the proposed model has better robustness and can effectively handle scenes with outlier interference.

This paper proposes a nonconvex exponential squared loss

function inspired by the above literature and the maximum correntropy criterion. This loss function is applied to ELM, which leads to a new robust ELM. RESELM can sufficiently suppress the negative impact of outliers on the robustness of the model and effectively improve the model's generalization. However, nonconvexity makes the model difficult to optimize. Considering the advantages of DCA [24] in solving nonconvex problems, this paper converts the objective function into DC programming [25] and then uses DCA to obtain the optimal output nodes.

The main contributions of this paper can be summarized as follows:

(1) A new loss function is constructed based on the maximum correntropy criterion, called the exponential squared loss function, which is nonconvex and can deal with training samples with noise and outliers.

(2) Robust ELM with the exponential squared loss function (RESELM) is developed. The nonconvexity of the proposed model makes it difficult to optimize directly by classical convex optimization algorithms. Therefore, it is transformed into a DC programming, and solved by DCA.

(3) The RESELM is tested in the case with 0%-20% lower outliers levels and with 25%-40% higher outliers levels, respectively. The experimental results show that RESELM improves the robustness and has excellent generalization ability, especially in the case of higher outliers levels.

The remainder of the paper is organized as follows. Section II briefly reviews the ELM. In Section III, the proposed model of this paper and the process of optimization by adopting DC programming are elaborated in detail. In Section IV, the experimental results of RESELM with different outliers levels are shown and analyzed. In the fifth part, the work of this paper is summarized.

II. Related Works

For *N* arbitrary samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in R^d$ is the input variable and $y_i \in R$ is the corresponding target in regression estimation, the output of ELM with *L* hidden nodes can be described as follows:

$$f(\mathbf{x}) = \sum_{j=1}^{L} h_j(\mathbf{x})\boldsymbol{\beta_j} = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{1}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_L]^T$ is the output weights vector that connects the hidden layer to the output node, and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), ..., h_L(\mathbf{x})]$ is the output of the hidden layer and $h_j(\mathbf{x})$ is the activation function. The formulation of regularized ELM [26] can be expressed as the following optimization:

$$\min_{\boldsymbol{\beta}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{N} e_i^2 \tag{2}$$

$$s.t. \quad \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = y_i - e_i, i = 1, ..., N, \tag{3}$$

where e_i denotes the error of training sample $\mathbf{x}_i$, and *C* is the regularization parameter. The optimal solution $\boldsymbol{\beta}$ of (2)-(3) is given by [26]

$$\boldsymbol{\beta} = \begin{cases} (\mathbf{H}^T\mathbf{H} + \dfrac{\mathbf{I}}{C})^{-1}\mathbf{H}^T\mathbf{y}, & N \geq L, \\ \mathbf{H}^T(\mathbf{H}\mathbf{H}^T + \dfrac{\mathbf{I}}{C})^{-1}\mathbf{y}, & N < L. \end{cases} \tag{4}$$

where $\mathbf{y} = [y_1, y_2, ..., y_N]^T$ and

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \mathbf{h}(\mathbf{x}_2) \\ . \\ . \\ . \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & ... & h_L(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & ... & h_L(\mathbf{x}_2) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ h_1(\mathbf{x}_N) & h_2(\mathbf{x}_N) & ... & h_L(\mathbf{x}_N) \end{bmatrix} \tag{5}$$

is the output matrix of the hidden layer.

In Regularized ELM (2)-(3), the least squares loss function may result in poor performance of ELM in dealing with the training samples containing noise and outliers. The reason for this is that least squares loss function assumes that the training samples obey a normal error distribution [12]. However, it can not guaranteed in the real world, which mistakenly considers the role of outliers with large residuals. This paper will focus on suppressing the effect of outliers by introducing a new exponential squared loss function for the ELM.

III. Research Methodology

A. Exponential Squared Loss Function

In information theory, the maximum correntropy criterion [22] is used to deal with the analysis of signals affected by various noises, which can effectively improve the robustness of signal analysis, and it is defined as follows:

$$V_\sigma(A, B) = E[k_\sigma(A - B)], \tag{6}$$

where $k_\sigma(\cdot)$ is a kernel function and $E[\cdot]$ is the mathematic expectation. In general, the joint probability distribution between variables A and B is unknown, so the average value is used to estimate the mathematical expectation. Then the maximum correntropy criterion is expressed as

$$V_\sigma(A, B) = \frac{1}{m}\sum_{i=1}^{m} k_\sigma(A_i, B_i), \tag{7}$$

where $k_\sigma(A_i, B_i) = \exp\left(-\frac{\|A_i - B_i\|^2}{\sigma^2}\right)$ is Gaussian kernel function.

In order to overcome the drawback of the least squares loss function, this paper constructs the exponential squared loss function based on the correntropy,

$$\ell_\sigma(z) = \sigma^2\left[1 - \exp\left(-\frac{z^2}{\sigma^2}\right)\right], \tag{8}$$

where σ^2 is the upper bound of the exponential squared loss function. Fig. 1 demonstrates the different curves of exponential squared loss with respect to the different of σ^2. As shown in Fig. 1, the proposed loss function is bounded.

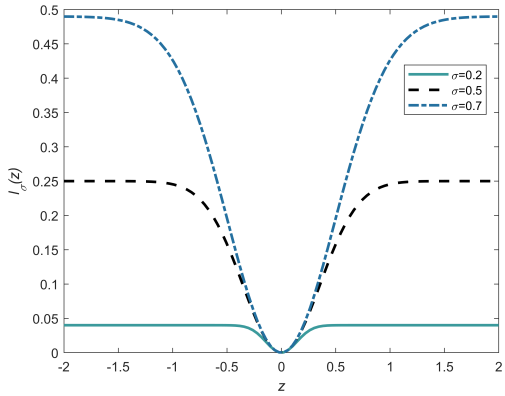


Fig. 1. Exponential squared loss function with different σ.

B. Robust ELM with Exponential Squared Loss Function

In this subsection, a robust ELM with exponential squared loss function is developed to improve the robustness of ELM, and the corresponding optimization problem can be obtained as

$$\min_{\boldsymbol{\beta}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\ell_\sigma(z_i) \tag{9}$$

where training error $z_i = y_i - \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta}$ and $\ell_\sigma(z_i)$ is the exponential squared loss function. The expression (8) can be written in the following equivalent form.

$$\ell_\sigma(z) = \ell_1(z) - \ell_2(z) \tag{10}$$

where $\ell_1(z) = z^2$, $\ell_2(z) = z^2 - \sigma^2\left[1 - \exp\left(-\frac{z^2}{\sigma^2}\right)\right]$. Substituting $\ell_1(z)$ and $\ell_2(z)$ into the optimization problem (9) can be derived as follows:

$$\min_{\boldsymbol{\beta}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\ell_1(z_i) - \frac{C}{2}\sum_{i=1}^{N}\ell_2(z_i) \tag{11}$$

Mark $L_1(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{\beta}\| + \frac{C}{2}\sum_{i=1}^{N}\ell_1(z_i)$, $L_2(\boldsymbol{\beta}) = \frac{C}{2}\sum_{i=1}^{N}\ell_2(z_i)$. According to the DCA, the optimal solution of the optimization problem (11) is obtained by solving the following iterations:

$$\boldsymbol{\beta}^{(t+1)} = \arg\min_{\boldsymbol{\beta}}\left\{L_1(\boldsymbol{\beta}) - L_2^{'}(\boldsymbol{\beta}^{(t)})\cdot\boldsymbol{\beta}\right\} \tag{12}$$

where $L_2^{'}(\boldsymbol{\beta}^{(t)})$ represents the derivative of $L_2(\boldsymbol{\beta})$ at $\boldsymbol{\beta}^{(t)}$. For a certain $\boldsymbol{\beta}$, the derivative expression is as follows:

$$L_2^{'}(\boldsymbol{\beta}) = \frac{C}{2}\sum_{i=1}^{N}\frac{\partial\ell_2(z_i)}{\partial z_i}\cdot\frac{\partial z_i}{\partial\boldsymbol{\beta}} = \frac{C}{2}\sum_{i=1}^{N}\left(\frac{\partial\ell_2(z_i)}{\partial z_i}\right)\cdot\left(-\mathbf{h}^T(\mathbf{x}_i)\right) \tag{13}$$

Denote $s_i = \frac{C}{2}\cdot\frac{\partial\ell_2(z_i)}{\partial z_i}$, and then define

$$s_i = Cz_i\left[1 - \exp\left(-\frac{z_i^2}{\sigma^2}\right)\right] \tag{14}$$

From (13) and (14),

$$\begin{aligned} -L_2^{'}\left(\boldsymbol{\beta}^{(t)}\right)\cdot\boldsymbol{\beta} &= \sum_{i=1}^{N}s_i^{(t)}\cdot\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} \\ &= \mathbf{s}^T\mathbf{H}\boldsymbol{\beta} \end{aligned} \tag{15}$$

where $\mathbf{s} = [s_1, s_2, ..., s_N]^T$, then (12) can be transformed into solving the following optimization problem

$$\boldsymbol{\beta}^{(t+1)} = \arg\min\left\{\frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\|\mathbf{y} - \mathbf{H}\boldsymbol{\beta}\|^2 + \mathbf{s}^T\mathbf{H}\boldsymbol{\beta}\right\} \tag{16}$$

Following the line [26], the optimal solution of (12) in the (t+1) iteration is obtained

$$\boldsymbol{\beta} = \begin{cases} \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\left(\mathbf{y} - \frac{\mathbf{s}}{C}\right) & N > L, \\ \mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\left(\mathbf{y} - \frac{\mathbf{s}}{C}\right) & N \le L. \end{cases} \tag{17}$$

Next is the step to solve RESELM by DC algorithm

Algorithm 1 RESELM

Input: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, set t=0 and choose an initial point $\mathbf{s}^{(0)}$, L, C, t_{max} and $\varepsilon > 0$ is a sufficient small number
Output: $\boldsymbol{\beta}$
repeat
Compute $s_i^{(t)}$ by (14) and $\mathbf{s}$.
Calculate (17) to obtain $\boldsymbol{\beta}^{(t+1)}$.
Let t=t+1.
until $\left\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^{(t+1)}\right\| \le \varepsilon$ or $t > t_{max}$

IV. Results and Discussion

To validate the efficacy of RESELM, it is compared with four related methods, regularized ELM [26], weighted ELM (WELM) [15], outlier robust ELM (ORELM) [16] and iteratively reweighted ELM (IRWELM) [27] on 18 benchmark data sets. In the first part, to simulate the samples in real world, different outliers levels are added to each of the 18 benchmark data sets, which can better reflect the robustness. of each algorithm. In the second part, the effects of the number of hidden nodes L and the parameter σ on the performance of the algorithms are studied. The root mean squares error (RMSE) [28] is used to measure the performance of the five regression algorithms. In the experiments, three parameters should be selected, L, C, σ. The maximum number of iteration t_{max} = 20, and the number of hidden nodes L=200. C is chosen from $\{2^{-19}, 2^{-18}, 2^{-17}, ..., 2^{18}, 2^{19}, 2^{20}\}$, and the width parameter of exponential squared loss function in RESELM is chosen from $\{0.05, 0.1, 0.15, 0.2, ..., 0.95, 1\}$.

A. Experimental Results on Benchmark Data Sets

This section shows the accuracy of the five algorithms on the benchmark data sets with different outliers levels. Table 1 demonstrates the algorithms' RMSE on 10 benchmark data sets with different outliers levels (0%, 5%, ...,35%, 40%). Fig. 2 adopts a line chart to intuitively exhibit the accuracy variation

TABLE I. EXPERIMENTAL RESULTS ON BENCHMARK DATA SETS WITH DIFFERENT OUTLIERS LEVELS

Data set	Outliers levels	ELM (RMSE ± Std)	WELM (RMSE ± Std)	ORELM (RMSE ± Std)	IRWELM (RMSE ± Std)	RESELM (RMSE ± Std)
Diabetes	0%	0.5832±0.0935	**0.5818±0.0908**	0.5946±0.1012	**0.5818±0.0907**	0.5819±0.0926
	5%	0.6480±0.0995	0.5757±0.0944	0.5985±0.0726	**0.5734± 0.0898**	0.5740±0.0906
	10%	0.6486±0.1028	0.5897±0.1036	0.6151±0.0709	**0.5781±0.0982**	0.5787±0.0828
	15%	0.6658±0.1131	0.6091±0.1140	0.6342±0.1031	0.5853±0.0946	**0.5745±0.0837**
	20%	0.6625±0.1232	0.6514±0.0934	0.6314±0.1057	0.5967±0.0925	**0.5795±0.0953**
	25%	0.6978±0.1244	0.7372±0.1979	0.6346±0.1081	0.7666±0.1686	**0.5801±0.0982**
	30%	0.6732±0.1243	0.7437±0.1923	0.6445±0.1097	0.7570±0.2105	**0.5823±0.0941**
	35%	0.6641±0.1230	0.6641±0.1230	0.6638±0.1022	0.6641±0.1230	**0.6036±0.0963**
	40%	0.6623±0.1184	0.6647±0.1192	0.6601±0.1051	0.6655±0.1196	**0.6224±0.1120**
Pollution	0%	36.0203±6.6156	37.0793±4.1013	36.2977±5.6007	37.2221±4.0031	**35.9914±6.4602**
	5%	56.7646±6.9775	37.7693±6.2724	37.6903±6.1268	37.2284±5.8706	**36.1362±6.7496**
	10%	57.4947±7.8601	39.5232±6.5899	39.5366±6.3062	38.1760±5.7391	**36.6225±5.9713**
	15%	59.5536±9.4949	43.3163±4.5347	43.0358±6.8460	38.3071±6.1082	**36.9604±5.3304**
	20%	59.3307±7.8055	48.0174±11.8658	45.0034±7.3071	37.9233±6.2045	**37.5655±5.9203**
	25%	64.6370±12.6879	58.7079±9.5638	49.6946±9.4306	65.1266±12.7976	**38.6563±6.0681**
	30%	61.5882±11.2617	61.5882±11.2617	56.4876±13.6835	61.5882±11.2617	**39.2598±5.7079**
	35%	58.8763±9.0385	58.8763±9.0385	57.8867±8.6369	58.8763±9.0385	**39.7484±6.1533**
	40%	59.8161±8.7125	59.8161±8.7125	57.8640±7.6647	59.8161±8.7125	**40.4083±5.1636**
Pyrim	0%	0.1114±0.0204	0.1060±0.0297	0.1084±0.0256	0.1077±0.0291	**0.1052±0.0318**
	5%	0.1305±0.0276	0.1083±0.0327	0.1098±0.0277	0.1078±0.0317	**0.1049±0.0317**
	10%	0.1315±0.0248	0.1127±0.0346	0.1134±0.0293	0.1103±0.0335	**0.1099±0.0352**
	15%	0.1402±0.0283	0.1199±0.0344	0.1182±0.0318	0.1092±0.0339	**0.1091±0.0342**
	20%	0.1451±0.0208	0.1289±0.0264	0.1207±0.0320	0.1107±0.0351	**0.1104±0.0339**
	25%	0.1469±0.0257	0.1405±0.0239	0.1284±0.0371	0.1190±0.0358	**0.1135±0.0322**
	30%	0.1456±0.0199	0.1381±0.0315	0.1313±0.0400	0.1293±0.0369	**0.1178±0.0325**
	35%	0.1549±0.0396	0.1556±0.0341	0.1389±0.0304	0.1559±0.0330	**0.1199±0.0363**
	40%	0.1583±0.0217	0.1587±0.0429	0.1396±0.0199	0.1580±0.0427	**0.1241±0.0356**
Servo	0%	0.6177±0.1013	**0.5547±0.1686**	0.6007±0.1416	0.6151±0.1666	0.5881±0.2043
	5%	0.8056±0.1097	0.7016±0.2281	0.6612±0.1732	0.7022±0.2212	**0.6455±0.1954**
	10%	0.9855±0.1307	0.7931±0.2099	0.7144±0.1969	0.7172±0.1868	**0.6827±0.1888**
	15%	1.0425±0.1378	0.8122±0.1500	0.7128±0.1816	**0.6935±0.1978**	0.6992±0.2195
	20%	1.0965±0.1633	0.9197±0.1655	0.7981±0.1871	0.7821±0.2080	**0.7521±0.1905**
	25%	1.3023±0.1662	0.9976±0.1218	0.7994±0.2028	0.7898±0.1995	**0.7341±0.2163**
	30%	1.4394±0.1417	1.1480±0.1802	0.9342±0.1977	0.9709±0.1605	**0.8323±0.2103**
	35%	1.4957±0.1387	1.4858±0.1487	1.1048±0.1575	1.4600±0.2192	**0.8663±0.2129**
	40%	1.4938±0.1681	1.4999±0.1360	1.2504±0.1558	1.4996±0.1361	**0.9211±0.1720**
Triazines	0%	0.1478±0.0169	0.1494±0.0189	0.1508±0.0203	0.1509±0.0198	**0.1471±0.0183**
	5%	0.1525±0.0180	0.1491±0.0197	0.1518±0.0200	0.1502±0.0196	**0.1487±0.0191**
	10%	0.1593±0.0153	0.1502±0.0212	0.1533±0.0205	0.1502±0.0199	**0.1496±0.0204**
	15%	0.1598±0.0170	0.1513±0.0192	0.1554±0.0206	0.1517±0.0200	**0.1507±0.0197**
	20%	0.1612±0.0138	0.1588±0.0157	0.1578±0.0160	0.1546±0.0154	**0.1527±0.0183**
	25%	0.1609±0.0140	0.1593±0.0165	0.1580±0.0164	0.1578±0.0200	**0.1574±0.0215**
	30%	0.1622±0.0177	0.1602±0.0173	0.1596±0.0172	0.1599±0.0162	**0.1577±0.0196**
	35%	0.1652±0.0122	0.1646±0.0196	0.1597±0.0157	0.1643±0.0188	**0.1583±0.0206**
	40%	0.1652±0.0122	0.1646±0.0186	**0.1595±0.0167**	0.1636±0.0176	0.1600±0.0154
MCPU	0%	54.0322±21.9741	59.0387±25.1172	**51.6258±25.1990**	54.8509±24.0867	51.7506±21.6742
	5%	79.5067±24.9900	55.1369±22.8733	58.2620±23.6020	**53.3418±23.6794**	55.1113±26.0012
	10%	114.6473±7.7572	61.2738±23.1768	58.5051±21.3603	57.3894±26.6030	**53.2878±23.2898**
	15%	116.6586±11.1286	70.2727±24.2272	63.9712±20.8597	60.6010±26.4665	**60.4078±25.1280**
	20%	145.8573±9.8285	78.6395±13.3020	72.2602±17.3181	**68.3295±26.7257**	70.3807±26.6680
	25%	158.6027±46.1825	84.5457±9.5708	73.1470±15.5764	78.3080±24.5267	**70.9848±22.4826**
	30%	159.1254±44.2106	96.0923±19.2900	72.4842±17.2320	86.3810±19.4879	**66.6127±25.0384**
	35%	159.8816±42.8364	143.0014±24.4416	79.7272±17.8036	124.9419±26.7260	**73.9987±18.2163**
	40%	158.9709±48.5852	160.5926±41.8149	88.6661±16.9791	160.4793±41.9984	**86.0494±25.0028**
Bodyfat	0%	0.0027±0.0015	0.0022±0.0017	**0.0021±0.0018**	**0.0021±0.0018**	0.0022±0.0018
	5%	0.0210±0.0018	0.0027±0.0015	**0.0022±0.0018**	**0.0022±0.0018**	0.0026±0.0016
	10%	0.0221±0.0026	0.0028±0.0015	**0.0022±0.0018**	**0.0022±0.0018**	0.0026±0.0016
	15%	0.0197±0.0021	0.0034±0.0014	**0.0022±0.0018**	**0.0022±0.0018**	0.0027±0.0015
	20%	0.0227±0.0033	0.0052±0.0018	**0.0022±0.0017**	**0.0022±0.0018**	0.0028±0.0015
	25%	0.0201±0.0011	0.0233±0.0056	**0.0023±0.0018**	0.0215±0.0040	0.0028±0.0015
	30%	0.0204±0.0025	0.0204±0.0025	**0.0024±0.0018**	0.0204±0.0025	0.0028±0.0015
	35%	0.0234±0.0048	0.0234±0.0048	0.0032±0.0018	0.0234±0.0048	**0.0030±0.0014**
	40%	0.0232±0.0039	0.0232±0.0039	0.0069±0.0024	0.0232±0.0039	**0.0034±0.0013**
AutoMPG	0%	2.8927±0.1359	**2.8697±0.1619**	2.9273±0.1445	2.9391±0.1228	2.8811±0.1753
	5%	3.4544±0.1374	2.9082±0.1223	2.9241±0.1439	2.9038±0.1366	**2.8781±0.1393**
	10%	4.1467±0.2067	2.9201±0.1447	2.9290±0.1170	2.8888±0.0989	**2.8869±0.0785**
	15%	5.2720±0.3020	3.0450±0.2378	2.9650±0.1801	**2.8866±0.1050**	2.8976±0.0992
	20%	5.9322±0.3485	3.3159±0.3372	2.9931±0.1597	2.9208±0.1188	**2.8964±0.1243**
	25%	7.6352±0.3434	5.1168±0.5234	3.1203±0.2035	4.0939±0.8859	**2.9122±0.1548**
	30%	7.9646±0.2458	7.8468±0.7013	3.3235±0.2356	7.1337±0.4989	**2.9339±0.1536**
	35%	8.0838±0.2344	8.0070±0.2422	3.8620±0.5010	8.0042±0.2448	**3.0375±0.2151**
	40%	8.2152±0.2736	8.1623±0.2287	4.9058±0.4914	8.1617±0.2289	**3.1374±0.1495**

Data set	Outliers levels	ELM (RMSE ± Std)	WELM (RMSE ± Std)	ORELM (RMSE ± Std)	IRWELM (RMSE ± Std)	RESELM (RMSE ± Std)
BH	0%	3.3436±0.2752	3.4172±0.4053	3.4892±0.4099	3.5044±0.4771	**3.3299±0.2751**
	5%	4.4329±0.5026	3.5431±0.4199	3.6025±0.4778	3.6465±0.3939	**3.4886±0.3914**
	10%	5.3008±0.4367	3.7162±0.7069	3.7211±0.6146	3.6627±0.5360	**3.5933±0.5327**
	15%	6.4081±0.3513	3.9997±0.4288	3.8031±0.4840	3.6272±0.3429	**3.6239±0.4093**
	20%	7.4551±0.3277	4.5851±0.5451	4.0599±0.4876	3.9450±0.5606	**3.8697±0.5311**
	25%	8.7010±0.2298	6.0587±0.4800	4.4545±0.6318	4.8437±0.5504	**4.1625±0.5608**
	30%	9.1915±0.5201	8.1431±0.6531	4.9708±0.6748	7.2347±0.6227	**4.3934±0.8690**
	35%	9.0939±0.4877	9.2483±0.4948	5.7650±0.4941	9.2744±0.4934	**4.7559±0.9979**
	40%	9.0767±0.4749	9.1630±0.5199	6.7831±0.6937	9.1791±0.5311	**5.2372±0.6306**
Concrete	0%	**6.6012±0.3947**	6.7518±0.4287	6.9974±0.5071	6.7847±0.4490	6.6016±0.3934
	5%	8.5968±0.4431	7.1756±1.0593	7.3170±0.6440	7.0391±0.8160	**6.7927±0.8174**
	10%	9.8699±0.5149	7.8765±0.4552	7.9495±0.5064	7.6314±0.4311	**7.3815±0.3971**
	15%	11.6955±0.5531	8.2027±0.3171	8.3026±0.4968	7.8751±0.3722	**7.7131±0.4287**
	20%	11.8756±0.5302	8.5947±0.2092	8.6132±0.4671	8.0975±0.3241	**7.9589±0.6622**
	25%	13.3383±0.5713	9.4221±0.2403	9.2141±0.5663	8.6074±0.2260	**8.3641±0.3404**
	30%	16.4737±0.6057	12.3405±0.8420	10.1421±0.7395	10.5091±0.9161	**8.4881±0.4038**
	35%	17.2334±0.3850	16.9023±0.9519	10.9747±0.6482	15.6942±1.0658	**8.9344±0.5123**
	40%	17.1472±0.3705	17.1077±0.3583	12.0187±0.5392	17.0847±0.3563	**9.0388±0.4906**

of the five algorithms on 8 benchmark data sets with 0% to 40% outliers levels.

In the experiments, each data set is randomly divided into training set and test set. Then different proportions of outliers are added to the training set, which is determined by the targets of the training set. The experiment select different proportions of outliers from $[y_{min},\ y_{max}]$ and add these outliers to the training samples randomly. The test set does not take any operation. The 10-fold cross-validation is applied on benchmark data sets, and taking the average RMSE of these ten independent experiments as the final result.

Observing the results in Table I, RESELM performs the best in the case of no outliers and achieves the optimal RMSE on four data sets. The accuracy on the other data sets is similar to the optimal RMSE. The worst performer is ELM, which achieves the optimum only on the Concrete data set. The performance of ORELM and IRWELM is close to each other. On the data set with 5% outliers level, the accuracy of ELM decreases most significantly, with a larger RMSE than the other algorithms on each data set. The accuracy of IRWELM is better than it would have been in the case of no outliers. RESELM obtains more comparable robustness, and its accuracy is optimal on most of the data sets. When the outliers level rises to 10%, the accuracy of ELM is still not very competitive, and the RMSE of WELM is better than that of the ELM. RESELM maintains its advantage in robustness, obtaining the optimal RMSE on eight data sets. The accuracy of the IRWELM is next to that of the RESELM in most cases. In the case of 15% and 20% outliers levels, the accuracy of RESELM obtains the best RMSE on most of the data sets. At the same time, ELM fails to obtain the optimal accuracy on any of data sets and has the worst RMSE of the five algorithms in most cases.

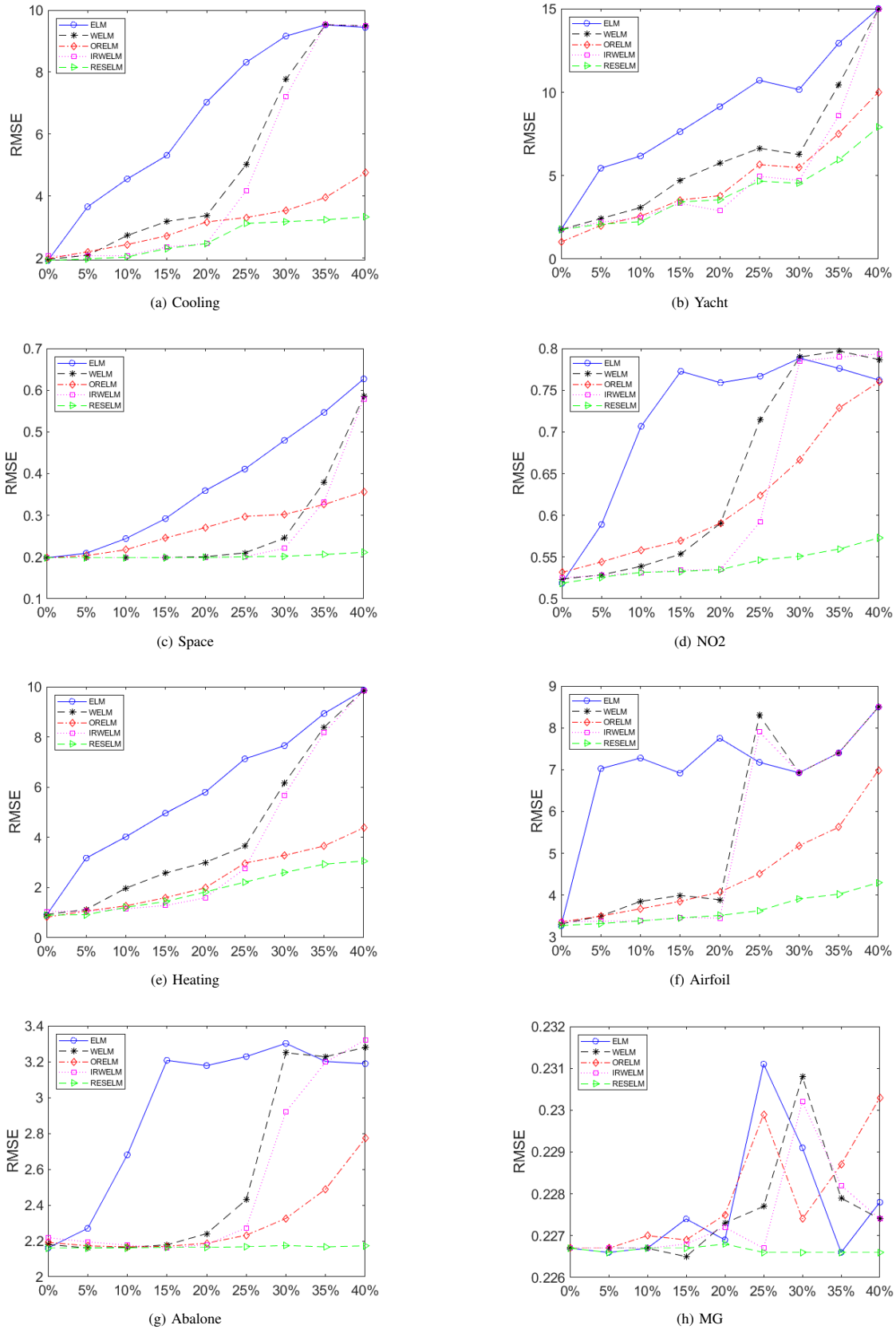
In the lower outliers levels (0%-20%), the optimal RMSEs are obtained for ELM only on Concrete. The most optimal RMSEs is RESELM with 34 times, and RESELM ranks second in accuracy for most other cases. It can be seen that ELM, WELM and ORELM are most negatively affected as the outliers increase. In comparison, RESELM is hardly affected, which shows that RESELM can effectively suppress the effect of outliers.

This paper focuses on the improvement of RESELM in terms of robustness. In the case of lower outliers levels, the robustness of RESELM improves but does not achieve the optimal RMSE on all data sets, such as on the data sets Body-fat, which is mainly determined by the loss function, where the difference between the true value and the predicted value is larger, the bigger the corresponding loss function value. Researchers have proposed various loss functions in order to reduce the effect of outliers so that the robustness of the model will be better [18], [19]. As shown in Table I, the loss function of RESELM does not have a significant advantage in the lower outliers levels, so experiments with higher outliers levels are conducted to investigate the robustness of the proposed algorithm.

From Table I, it can be seen that the robustness of ELM remains uncompetitive in the case of higher outlier levels, and its RMSE increases with the outliers level. ELM applies a least squares loss function, and when the residuals are small, the least squares loss function does not differ significantly from the exponential squared loss function. Therefore, ELM can produce more accurate results when there are no outliers. However, this loss function increases infinitely with the growth of the residuals and is growing exponentially, so it cannot effectively constrain the effect of outliers, which makes the ELM less robust.

In the case of 25%-40% outliers levels, WELM does not obtain the optimal RMSE, but it has better accuracy than ELM. A weighted 2-norm loss function is proposed in WELM, which assigns different weights according to the value of the residuals to improve the robustness. However, the algorithm of WELM relies too much on the initial accuracy of the model, and the results obtained are not satisfactory.

ORELM has a better increase in accuracy than the lower outliers levels, and has the smallest difference from the optimal RMSE in most cases. ORELM uses the 1-norm loss function, which has a larger function value than the others when the residuals are small, and therefore has poorer accuracy. However, after the residuals increase, it has an advantage over the least squares loss function and the weighted 2-norm loss function. This is the reason that why ORELM is worse than IRWELM in most cases in the lower outliers levels but has better RMSEs than IRWELM in the cases of 25% to 40% outliers levels.



(a) Cooling

(b) Yacht

(c) Space

(d) NO2

(e) Heating

(f) Airfoil

(g) Abalone

(h) MG

Fig. 2. The performance of ELM, WELM, ORELM, IRWELM and RESELM on data sets with different outliers levels.

IRWELM is the opposite of ORELM. The accuracy of IRWELM ranks second 25 times in the lower outliers levels, and its ability to suppress outliers is worse than ORELM when the outliers level increases. IRWELM and WELM apply the same loss function, but IRWELM is more robust than WELM because IRWELM's solution method is an iterative reweighting algorithm. Compared with WELM which solves the output nodes directly, the iterative approach of IRWELM makes it obtain more accurate output nodes than those of WELM.

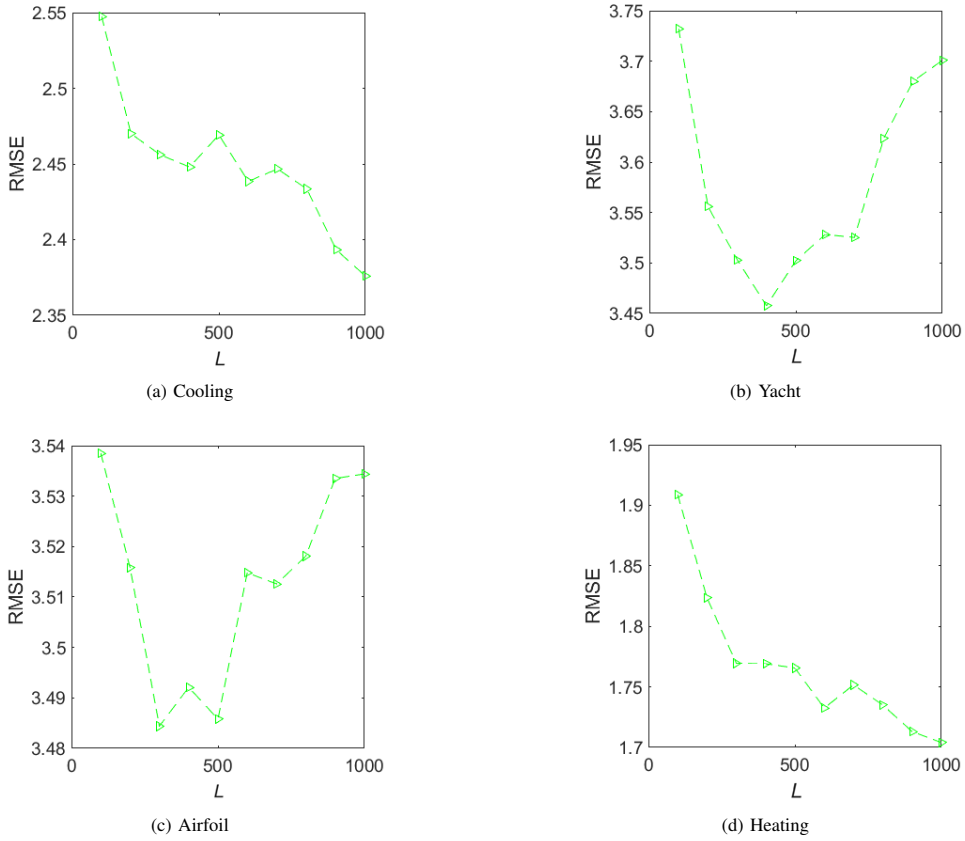
In the higher outliers levels, the RESELM has a significant advantage. The optimal RMSE is achieved 37 times due to the insensitive nature of the exponential squared loss function to higher outliers, it does not grow indefinitely, and the optimization problem is solved using an iterative approach in RESELM. Therefore, RESELM can effectively suppress outliers and has excellent generalizability.

Compared with ELM, WELM, ORELM, and IRWELM, RESELM obtains better accuracy than them in most cases. The RMSE is usually used in the study of improved models for ELM to illustrate the performance of the model in terms of accuracy, and experiments are conducted in [15], [16], [27] to verify the ability of the model to suppress outliers. The experiments in this paper show that the RMSE of RESELM is smaller than the other four models, so the proposed method effectively improves the robustness of ELM and has excellent generalization.

In Fig. 2, 8 benchmark data sets are chosen to show the variation of RMSE with the increasing outliers level for the five algorithms by line chart. It is more intuitive to observe the outliers level's effect on the algorithms' accuracy by the line chart. The line of ELM is almost always at the top of the axis and is most obvious on the Cooling, Yacht, Space, and Heating data sets, which have higher fold line from 0% outlier level to 40% outliers level than the other algorithms. WELM and IRWELM sometimes have worse RMSE than ELM in the higher outliers levels, as seen on the NO2, Airfoil, Abalone, and MG data sets. The fold line of ORELM are in the middle of ELM and RESELM on all data sets except the MG data set, which illustrates that the robustness of ORELM has improved somewhat compared to ELM but still suffer some effect in the higher outlier level compared to RESELM. The fold line of RESELM is always below all the folds except on the MG data set. Its accuracy is least affected by outliers, which means it has the best robustness.

B. Parameter Influence

Different parameters have an effect on the performance of the model. Next, the effect of hidden layer nodes L and the upper bound parameter σ of the exponential squared loss function on the performance of RESELM is examined. The experiments are conducted on four data sets without outliers, Cooling, Yacht, Airfoil, and Heating. In the experiments, the optimal parameters are selected for all parameters except the ones to be studied. The experiments reflect the effects of the parameters by RMSE.



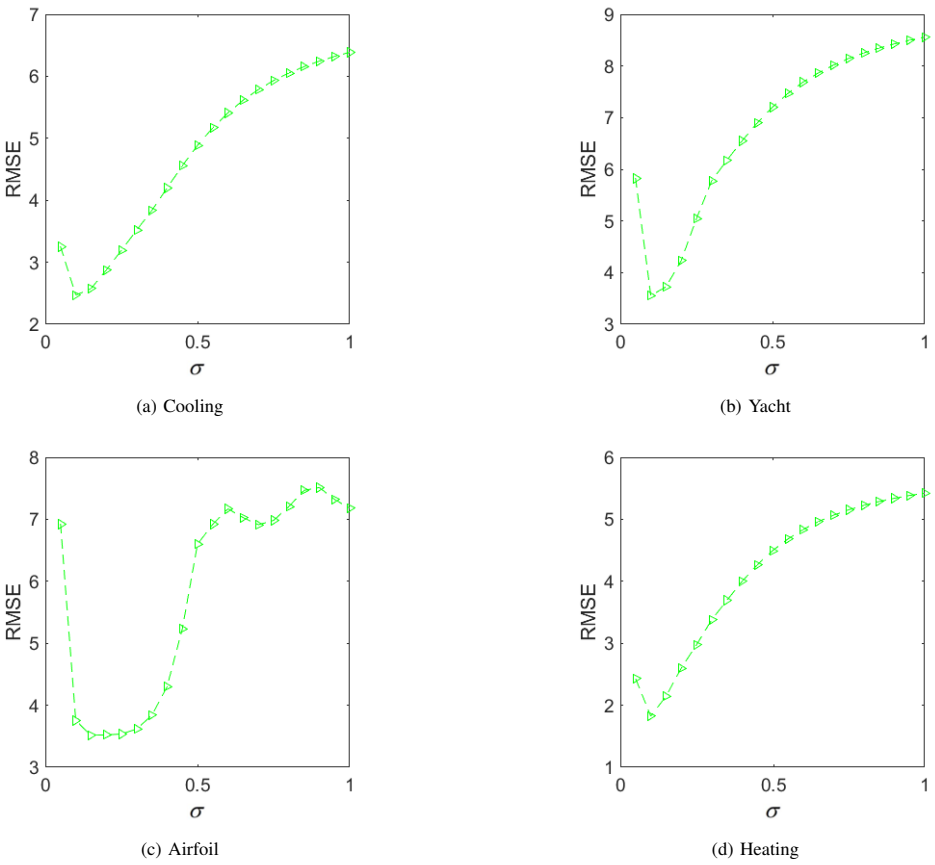
(a) Cooling

(b) Yacht

(c) Airfoil

(d) Heating

Fig. 3. The effect of L on RMSE.



(a) Cooling (b) Yacht

(c) Airfoil (d) Heating

Fig. 4. The effect of σ on RMSE.

There is no exact method to determine the number of hidden layer nodes L in ELM, and in practice previous experience or experimental methods within a certain range are usually used, but the number of L can affect the model's accuracy. When the L is too small, the model may have difficulty dealing with more complex problems. When L is too large, some nodes will not be meaningful to the model performance, which makes the model training time longer without improving the model accuracy. Therefore, the appropriate L is significant for the model's performance.

The effect of L on the model's accuracy is shown in Fig. 3, with L taking values from $\{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$. The figure visualizes the effect of L. In Fig. 3(a) and (d), the curves of RMSE show a decreasing trend. The RMSE at L=1000 is the smallest, indicating that a minimum value is achieved in the given range. In Fig. 3(b) and (c), the RMSE decreases and then increases, and the optimal L is found in [300, 500]. The curves on four data sets are not smooth, which indicates that RESELM is a little sensitive to the network size. Therefore, when conducting experiments, choosing a more appropriate number of hidden layer nodes has an essential impact on the performance of RESELM.

To investigate the effect of σ on accuracy, experiments are conducted on four data sets using different σ chosen from the range [0, 1] with an interval of 0.05. In Fig. 4, the curves on the four data sets demonstrate that the RMSE first decreases and then continues to increase as σ increases. The global optimum is found in [0,0.5]. Although slightly different on the Airfoil data set, the global optimum is still in [0, 0.5]. The optimal RMSE can be obtained when the σ is small. By observing Fig. 1, it can be observed when the σ is small, the upper bound of the loss function also becomes smaller. When the residuals are larger, the value of the proposed loss function is also smaller for smaller σ, and the robustness of the model is better.

V. Conclusion

This paper proposes a robust ELM based on the exponential squared loss function (RESELM) for training samples contaminated with noise and outliers. The loss function used in the model is obtained based on correntropy. The nonconvexity of the exponential squared loss function enables RESELM to control the effect of outliers effectively. However, the nonconvexity also makes the model difficult to optimize. The proposed model is solved by formulating it as a DC programming and then adopting DCA. Experiments were conducted to verify the performance of RESELM on benchmark data sets with different outliers levels. The experimental results demonstrated that RESELM is non-sensitive to outliers and can obtain better robustness with a significant advantage in accuracy, especially in the case of 25% to 40% outliers levels. This paper discusses offline ELM, but in real-world problems, online learning is usually required, so future work considers

extending RESELM to online sequential learning for better application to real-world problems.

ACKNOWLEDGMENT

The work was supported by the National Natural Science Foundation of China under Grant No. 61833005 and 61907033, the Postdoctoral Science Foundation of China under Grant No. 2018M642129.

REFERENCES

[1] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, vol. 2, pp. 985–990, Ieee, 2004.

[2] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[3] G. Huang, L. Chen, C. Siew, *et al.*, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[4] G. Huang, Q. Zhu, and C. Siew, "Real-time learning capability of neural networks," *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 863–878, 2006.

[5] G. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE transactions on neural networks*, vol. 14, no. 2, pp. 274–281, 2003.

[6] D. Zheng, Z. Hong, N. Wang, and P. Chen, "An improved lda-based elm classification for intrusion detection algorithm in iot application," *Sensors*, vol. 20, no. 6, pp. 1–19, 2020.

[7] J. Zeng, B. Roy, D. Kumar, A. S. Mohammed, D. J. Armaghani, J. Zhou, and E. T. Mohamad, "Proposing several hybrid pso-extreme learning machine techniques to predict tbm performance," *Engineering with Computers*, pp. 1–17, 2021.

[8] S. S. Chakravarthy and H. Rajaguru, "Automatic detection and classification of mammograms using improved extreme learning machine with deep learning," *Irbm*, vol. 43, no. 1, pp. 49–61, 2022.

[9] A. Boukerche, L. Zheng, and O. Alfandi, "Outlier detection: Methods, models, and classification," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–37, 2020.

[10] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.

[11] D. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artificial intelligence review*, vol. 33, pp. 275–306, 2010.

[12] P. Meer, C. V. Stewart, and D. E. Tyler, "Robust computer vision: An interdisciplinary challenge," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 1–7, 2000.

[13] S. Mehrkanoon, X. Huang, and J. A. Suykens, "Non-parallel support vector classifiers with different loss functions," *Neurocomputing*, vol. 143, pp. 294–301, 2014.

[14] A. Ghosh, H. Kumar, and P. S. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.

[15] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," *IEEE symposium on computational intelligence and data mining*, pp. 389–395, 2009.

[16] K. Zhang and M. Luo, "Outlier-robust extreme learning machine for regression problems," *Neurocomputing*, vol. 151, pp. 1519–1527, 2015.

[17] K. Chen, Q. Lv, Y. Lu, and Y. Dou, "Robust regularized extreme learning machine for regression using iteratively reweighted least squares," *Neurocomputing*, vol. 230, pp. 345–358, 2017.

[18] Y. Feng, Y. Yang, X. Huang, S. Mehrkanoon, and J. A. Suykens, "Robust support vector machines for classification with nonconvex and smooth losses," *Neural computation*, vol. 28, no. 6, pp. 1217–1247, 2016.

[19] X. Wang, K. Wang, Y. She, and J. Cao, "Zero-norm elm with non-convex quadratic loss function for sparse and robust regression," *Neural Processing Letters*, pp. 1–33, 2023.

[20] K. Wang, J. Cao, and H. Pei, "Robust extreme learning machine in the presence of outliers by iterative reweighted algorithm," *Applied Mathematics and Computation*, vol. 377, p. 125186, 2020.

[21] H. Pei, K. Wang, Q. Lin, and P. Zhong, "Robust semi-supervised extreme learning machine," *Knowledge-Based Systems*, vol. 159, pp. 203–220, 2018.

[22] W. Liu, P. Pokharel, and J. Principe, "Correntropy: Properties and applications in non-gaussian signal processing," *IEEE Transactions on signal processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

[23] H. Xing and X. Wang, "Training extreme learning machine via regularized correntropy criterion," *Neural Computing and Applications*, vol. 23, pp. 1977–1986, 2013.

[24] L. An and P. Tao, "The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems," *Annals of operations research*, vol. 133, no. 1, pp. 23–46, 2005.

[25] R. Horst and N. Thoai, "Dc programming: overview," *Journal of Optimization Theory and Applications*, vol. 103, no. 1, pp. 1–43, 1999.

[26] G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2011.

[27] P. Horata, S. Chiewchanwattana, and K. Sunat, "Robust extreme learning machine," *Neurocomputing*, vol. 102, pp. 31–44, 2013.

[28] T. Hodson, "Root mean square error (rmse) or mean absolute error (mae): when to use them or not," *Geoscientific Model Development Discussions*, vol. 15, no. 14, pp. 5481–5487, 2022.