Hindawi
Scientific Programming
Volume 2022, Article ID 6446080, 10 pages
https://doi.org/10.1155/2022/6446080

Research Article

Robust Extreme Learning Machine Using New Activation and Loss Functions Based on M-Estimation for Regression and Classification

Adnan Khan,[1] **Amjad Ali**,[1] **Naveed Islam**,[2] **Sadaf Manzoor**,[1] **Hassan Zeb**,[1] **Muhammad Azeem**,[3] **and Shumaila Ihtesham**[1]

[1]*Department of Statistics Islamia College University Peshawar, Peshawar, Pakistan*
[2]*Department of Computer Science Islamia College University Peshawar, Peshawar, Pakistan*
[3]*Department of Statistics, University of Malakand, Peshawar, Pakistan*

Correspondence should be addressed to Adnan Khan; adnan@icp.edu.pk

Received 18 May 2022; Accepted 18 August 2022; Published 15 October 2022

Academic Editor: Sadiq Hussain

This paper provides an analysis of the combining effect of novel activation function and loss function based on M-estimation in application to extreme learning machine (ELM), a feed-forward neural network. Due to the computational efficiency and classification/prediction accuracy of ELM and its variants, they have been widely exploited in the development of new technologies and applications. However, in real applications, the performance of classical ELMs deteriorates in the presence of outliers, thus, negatively impacting the precision and accuracy of the system. To further enhance the performance of ELM and its variants, we proposed novel activation functions based on the psi function of M and redescend the M-estimation method along with the smooth $\ell 2$-norm weight-loss functions to reduce the negative impact of the outliers. The proposed psi functions of several M and redescending M-estimation methods are more flexible to make more distinct features space. For the first time, the idea of the psi function as an activation function in the neural network is introduced in the literature to ensure accurate prediction. In addition, new robust $\ell 2$ norm-loss functions based on M and redescending M-estimation are proposed to deal with outliers efficiently in ELM. To evaluate the performance of the proposed methodology against other state-of-the-art techniques, experiments have been performed in diverse environments, which show promising improvements in application to regression and classification problems.

1. Introduction

Neural networks (NNs) are biological-inspired predictive techniques that mimic the behavior and neural processing of the biological nervous system. NN has been extensively and successfully applied to pattern recognition, time series prediction and modeling, adaptive control, classification, and other areas of artificial intelligence (AI). Advancements in AI applications depend upon robust machine learning algorithms. The shortcomings of traditional NN were detached by Zhu et al. [1] developed a single-layer feed-forward neural network called ELM due to its high speed and accuracy, and further Huang et al. [2] observed its performance over the popular back propagation neural network(BPNN) and support vector machine (SVM) in regression and classification problems. ELM has been used widely in numerous application domains, such as biomedical engineering, system identification, computer vision, control, and robotics [3]. Harikumar et al. [4] developed ELM-based classifier for epilepsy identification from EEG signals and compared the computational performance with BPNN. In reference [5], Li et al. deployed ELM for daily stream forecasts and showed better performance than random forest. ELM performed considerably faster without a significant loss in accuracy. Bhatia et al. [6] used ELM in plant disease prediction for a highly imbalanced dataset. Fabric wrinkle evaluation model with regularized ELM based on improved Harris's Hawk optimization was

discussed in [7]. In short, more applications of ELM were found in the literature such as mine reclamation based on remote sensing information and error compensation, cooperative spectrum sensing for cognitive radio networks, detection of total iron content, evaluation of shape factor impact on the discharge coefficient of side orifices, coal exploration based on a multilayer extreme learning machine and satellite images, e-mail spam filtering techniques, and emotion recognition in election day tweets and compared the performance with existing classification approaches and noted the highest accuracy [8–14]. In machine learning, ELM has introduced a better alternative to existing algorithms used in several supervised and unsupervised learnings. There is no need to iteratively tune the inputs, hidden and output layer weights, and biases like BPNN [1, 15–19]. Due to this, the ELM is capable of lower cost and high speed learning with good generalization accuracy and performance. ELM has the capacity to introduce nonlinearity if any, using different differentiable or nondifferentiable activation functions in its training/testing phase, and possession of a unique solution to a different complex problem in practice [20]. Furthermore, ELM avoids the problem of overfitting due to analytical solutions and local minima [1, 20]. ELM needs fewer hyperparameters such as activation function and hidden layer size to be optimized as compared to other techniques, as in conventional neural networks, SVM, and least-square SVM, with similar computational costs. ELM filled the gap between biological learning and conventional learning machines [21]. Instead of the great merits of classical ELM, it has several deficiencies such as contamination in data and ill-posed structure due to which sometimes analytical solution of output weights was not possible due to noninvertibility and sensitivity of hyperparameters. Deng et al. [22] and Horata et al. [23] introduced regularized and weighted regularized ELM (WRELM) to solve the problem of overfitting and noninvertibility. Following Deng et al. [22] and Horata et al. [23], Barreto et al. [24] used robust M-estimators based on cost functions to downweigh outliers and avoid their negative effects in the computational process in image classification with salt and pepper noise. Zhang and Luo [25] developed an outlier robust extreme learning machine for regression and classification purposes based on l_1–the norm and augmented language multiplier (ALM), a novel variant of ELM, and compared its performance with a weighted regularized extreme learning machine by taking real benchmark data from the UCI machine learning repository. Chen et al. [26] used some popular M estimators-based weight-loss functions to regression problems in the presence of outliers instead of the Huber loss function, as it has a linear relationship with error and has no smoothing criteria to downweigh outliers properly. Recently, more robust M-estimators are developed to properly filter and smoothly reduce the negative effects of outliers by many researchers in statistical regression analysis. The detailed information about existing and recently developed M-estimation based objectives, psi, and weight functions is given in Table 1 [27–29]. Almost all researchers including [1, 2, 6–26] used sigmoid, sine, cosine, Gaussian, tan-sigmoid ReLu, radial basis function (RBF), and their modified version as activation functions in ELM and also used their variants to introduce nonlinearity in hidden layer space of the neural network. Unlike the traditional typical gradient-based learning algorithms which only work for differentiable activation functions, it is easily detected that ELM could be used to train with many nondifferentiable activation functions. Huang et al. [2] discussed the limitations of popular activation functions that behave S type bounded shape function between 0 and 1 or −1 and 1, which observed the problem of diminishing gradient necessary to differentiate between good and bad observations at extreme edges during the training process. Due to this mismanagement, significant information may be lost. Liu et al. [30] introduced a robust activation function (RAF) to keep activation function output away from zero as much as possible and make inputs fully informative. The very same problem may happen with tan sigmoid as well as with RAF, which was introduced by Liu et al. [30] in ELM and still has the problem of robustness against outliers. SIBI et al. [31] studied the effects of different activation functions while training BPNN to extract useful information by transforming inputs into output signals. They concluded that there is no significant difference found among them to prefer it over one another. Gomes et al. [32] analyzed the performance of different activation functions in NN to accurately forecast time series data. Later, Essai and Ellah [33] performed experiments using robust M-estimators objective functions as activation functions which outperformed the activation functions used earlier in the literature. Freire and Barreto [34] used the idea of batch intrinsic plasticity (BIP) to maximize hidden layer information combined with robust estimation of the output weights. This paper proposes several redescending M-estimators $\psi - function$ as activation functions in ELM and in their variants complemented by weight-loss functions to smoothly avoid the negative impact of contamination. A $\psi - function$ is a piecewise continuous $\psi - function$ redescending towards zero $\psi(x) = 0\, x \geq C$ for, and C is often called the rejection point to real outliers. This study aims to extend the applicability of high breakdown M-estimator's psi-functions as activation functions against other competitors, complemented by robust loss functions in ELM and its variants. To evaluate the performance of the proposed methodology against other state-of-the-art techniques, experiments have been performed in diverse environments, which shows promising improvements in application to regression and classification problems. The details of the remaining paper is as in section 2. First, an overview of related work of ELM and its variants is discussed and extended to the proposed methodology. In section 3 experimental design of the simulation study is defined, in section 4 results and discussions are mentioned, and in the last section conclusions and future work is discussed. Figure 1 shows all loss functions.

1.1. Extended ELM Based on Convex and Nonconvex 2 Norm Loss Functions. The overfitting problem in ELMs using $\ell$the 2-norm loss function is intrinsically caused by the

Table 1: Commonly used continuous and differentiable activation functions.

Sigmoid	Tan-sigmoid or hyperbolic tangent	Sine symmetric	Cosine	Bent line	RAF

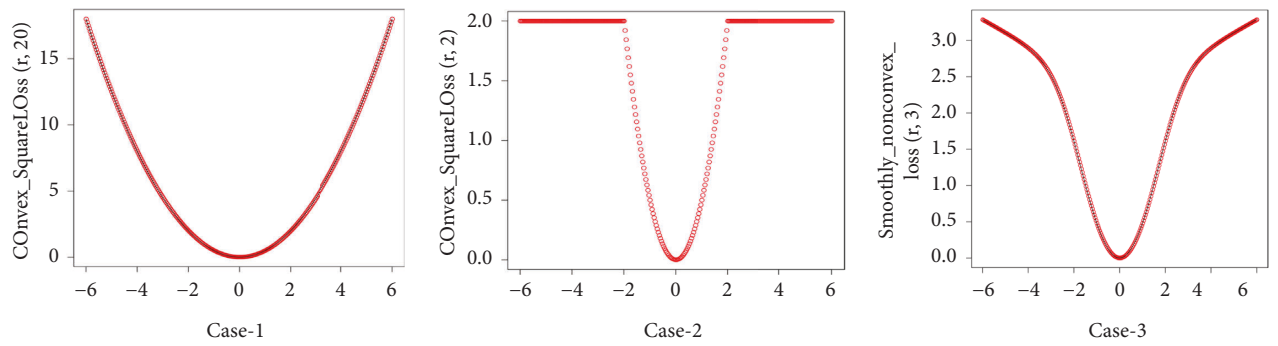

Figure 1: $\ell 2$-norm loss function, strict nonconvex $\ell 2$-norm loss function, and smooth nonconvex $\ell 2$-norm loss function.

number of outliers in data. In a real scenario far away data can transfers statistical results into a biased analysis. Wang et al. [35] have exploited the strict nonconvex loss function to mitigate the effect of wild observation if any while training the desired network. Though the outcomes produced are satisfactory in specific applications, yet, these good results negatively impact the overall performance in terms of accuracy and stability in general applications.

In our approach, a nonconvex 2-norm smooth loss function based on M and redescending M-estimation theory was incorporated in ELM, as inspired by Wang et al. [35] because strict nonconvex loss function sometimes loses valuable information. Graphically shown in Figure 1 where (a) $\ell 2$-norm loss function which uses all data while training model even outliers, (b) strict nonconvex loss function which holds data in original form and excludes observations from a specific point, and (c) smooth nonconvex $\ell 2$-norm loss function to assign weights in such way weights decreases as residuals increases.

$$l_c(z) = \min\left\{c^2, \rho(z)\right\} = \begin{cases} \rho(z), |z| \le C, \\ c^2, |z| > C. \end{cases} \tag{1}$$

Special cases case 1. *If* $\rho(z) = z^2$ *and* $C = \infty$ It reduces to a $\ell 2$-norm convex loss function that is applied to minimize the training error while keeping outliers as well if any.

case 2. If $\rho(z) = z^2$ and Cis predefined constant, then it reduces to a strict nonconvex $\ell 2$-norm loss function applied by Wang et al. [35] in ELM to reduce the negative impact of outliers during the training model.

case 3. If $\rho(z)$is function of z whose derivative is nonlinear and C defined optimizing constantis proposed loss function $\rho(z)$is based on M and re-descending M-estimation theory, where outliers are down weighed to normalize training data while training a classifier.

The conventional form of supervised learning data $(x_i, y_i) \subset R^d \times R$, the general form of the objective function of ELM.

$$\sum_{i=1}^{\widetilde{N}} \beta_i g(a_i, b_i, x_i) = y_j, j = 1, 2, \ldots N, \tag{2}$$

$$\min_{\beta} \frac{1}{2}\|\beta\|^2 + \zeta \sum_{i}^{N} l(e_i), \tag{3}$$

$$s.th(x_i)\beta = y_i - e_i, i = 1, 2, \ldots, N, \tag{4}$$

where $a_i = [a_{i1}, a_{i2}, a_{i3}, \ldots, a_{in}]^T$is the weights vector and bias term b_i randomly generated from any continues distribution, connecting the hidden nodes and input nodes; $\beta_i = [\beta_{i1}, \beta_{i2}, \beta_{i3}, \ldots, \beta_{im}]^T$is the weight vector connecting hidden nodes with output neurons

$$.H = \begin{bmatrix} h(x_1) \\ h(x_2) \\ . \\ . \\ . \\ h(x_N) \end{bmatrix} = \begin{bmatrix} g(a_1, b_1, x_1) \ldots & g(a_{\widetilde{N}}, b_{\widetilde{N}}, x_1) \\ . & . \\ . & . \\ . & . \\ g(a_1, b_1, x_N) & g(a_{\widetilde{N}}, b_{\widetilde{N}}, x_{\widetilde{N}}) \end{bmatrix}_{N\times}$$

$$\widetilde{N}, \beta = \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ . \\ . \\ . \\ \beta_{\widetilde{N}}^T \end{bmatrix}_{\widetilde{N}\times m}, Y = [y_1, y_2, \ldots y_N]^T.$$

The Lagrange function for optimization of (3) and (4) becomes as follows:

$$l(\beta, e_i, \alpha_i) = \frac{1}{2}\|\beta\|^2 + \zeta \sum_{i}^{N} \rho(e_i) + \sum_{i=1}^{N} \alpha_i (y_i - h(x_i)\beta - e_i), \tag{5}$$

where $\beta \in \widehat{R^N}$ is the output weights vector and $l(.,.)$: $R \times R \longrightarrow R$ is the loss function. The parametenetwork as a regularization agent to maintain a bias-variance trade-off. Traditional ELM uses a simple square loss function which is highly sensitive to outliers. Therefore, M-estimator and redescending M-estimator loss functions have been used to

Table 2: The $\rho(.)$, $\psi(.)$, and weight function $\mathbf{w}(.)$, of M and redescending M- estimators.

Method	Weight-loss function $\rho(e_i)$	ψ –functions $\psi = d/r(\rho(e_i))$ Proposed activation functions	Weight-function $w(e) = \psi(e)/e$	C
Lm1stf	$\log(1 + 1/2e^2)$	$e/1 + 1/2e^2$	$1/1 + 1/2e^2$	NA
ALARM	$2c^2/3[1 - 2(1 + 3e^{(\frac{-}{c})^2})/(1 + e^{(\frac{e}{c})^2})^3]$	$16ee^{-2(e/c)^2}/(1 + e^{-(e/c)^2})^4 0$	$(4e^{-(e/c)^2})^2/(1 + e^{-(e/c)^2})^4 0$	3
OLS	$2c^2/3[1 - 2(1 + 3e)/(1 + e)^3]$ $1/2e^2$	e	1	NA
Insha	$[c^2/4[\text{Arctan}(e/c)^2 + (ce)^2/c^4 + e^4]]$	$[e[1 + ((e/c))^4]^{-2}]$	$[1 + ((e/c))^4]^{-2}$	1.5
Welsch	$c^2/2[1 - \exp(-((e/c)^2)]$	$e.\exp(-((e/c)^2)$	$\exp(-((e/c)^2)$	4.654
Tukey bisquare	$c^2/6[1 - \{1 - (e/c)^{23}]c^2/6$	$e[1 - (e/c)^2]^2 0$	$e[1 - (e/c)^2]^2 0$	2.985
GMtf	$e^2/2/1 + e^2$	$e/(1 + e^2)^2$	$1/(1 + e^2)^2$	NA
Ali	$2e/3(1 - (e/c)^4)$	$2e/3(1 - (e/c)^4)$	$2/3(1 - (e/c)^4)$	2
Cauchy	$c^2/2\log(1 + (e/c^2)$	$e/1 + (e/c)^2$	$1/1 + (e/c)^2$	3
Khalil et al.(2017	$3/2e[1 - (e/c)^4]^2 \sin(2/3(1 - e/c))^2$	$3/2e[1 - (e/c)^4]^2 \sin(2/3(-e/c))^2$	$3/2[1 - (e/c)^4]^2 \sin(2/3(1 - e/c))^2$	3
Logistic	$c^2 \log[\cosh(e/c)]$	$e(e/c)^{-1} \tanh(r/c)$	$(e/c)^{-1} \tanh(e/c)$	1.205

Table 3: Proposed Activation Functions in Extreme Learning Machine and its variants.

Proposed 1	Lm1stf ψ –function
Proposed 2	Alarm ψ –function
Proposed 3	OLS ψ –function
Proposed 4	Insha ψ –function
Proposed 5	Welsch ψ –function
Proposed 6	Bisquare ψ –function
Proposed 7	Ali ψ –function
Proposed 8	Qadir ψ –function
Proposed 9	Cauchy ψ –function
Proposed 10	Khalil ψ –function
Proposed 11	Logistic ψ –function

$W1$ = Tukey Bisquare M-estimator weights-loss function, $W2$ = Welsch weights-loss function, $W3$ = Gmtf, $W4$ = ALARM weights function, $W5$ = Insha weights-loss function, and $W6$ = Cauchy weights-loss function.

enhance the robustness of ELM against outliers, which $\rho(.)$ denotes robust loss function and e_i is the standardized residuals. The psi function of $\rho(e)$ is $\psi(e) = \partial\rho(e)/\partial(e)$, and corresponding weight function is $w(e) = \psi(e)/e$. In the present work, efficient M-estimation-based loss functions are studied along with their psi function as activation function complemented by their loss function to gain maximum accuracy. The detailed information on the proposed M-estimation with their objective, psi function, and weight function is given in Table 2.

After simplification, the output weights estimate ofβ, the objective function (5) using l_2-norm smooth loss function regularization term can be written as follows:

$$\beta = \begin{cases} \left(H^T W H + \frac{I}{\zeta}\right)^{-1} H^T y, N \geq \widetilde{N} \\ \\ H^T \left(W H^T H + \frac{I}{\zeta}\right)^{-1} y, N < \widetilde{N}, \end{cases} \quad (6)$$

where $w_i = \partial\rho(e_i)/\partial e_i/e_i = \begin{cases} \partial\rho(e_i)/\partial e_i/e_i, |e_i| \leq c \\ 0, |e_i| > c. \end{cases}$ weight function for training data. Setting the diagonal matrix $W = diag\{w_1, w_2, \ldots, w_N\}$. Cases 1 and 2 are special cases of solution given in (6).

1.2. Proposed Iterative Reweighted Algorithm for Robust ELM. Input: training data $(x_i, y_i) \subset R^d \times R$,number of hidden nodes$\widetilde{N}$, maximum of iterations, $k_{\max}$, and activation function g (.) given below in Table 2 and in Table 2 only psi function as activation function. Calculate the hidden layer output matrix $H$and initiate the weight matrix $W^{(0)} = I\&k = 1$.

Step 1. Compute initial output weights by equation-(6)

$$\beta^{(k)} = \begin{cases} (H^T W^{(k)} H + I/\zeta)^{-1} H^T y, N \geq \widetilde{N} \\ H^T (W^{(k)} H^T H + I/\zeta)^{-1} y, N < \widetilde{N}. \end{cases}$$

The estimate function is given in (2).

Step 2. Obtain residual $e_i^{(k)} = y_i - \widehat{y}_i$ and standardize it using robust location and scale parameter and assign weights using existing and proposed weight function based on M-estimation given in Table 2 to update $W^{(k)} = w_i^{(k)} I$.

Step 3. Update $\beta^{(k+1)}$ computed in Step 1.

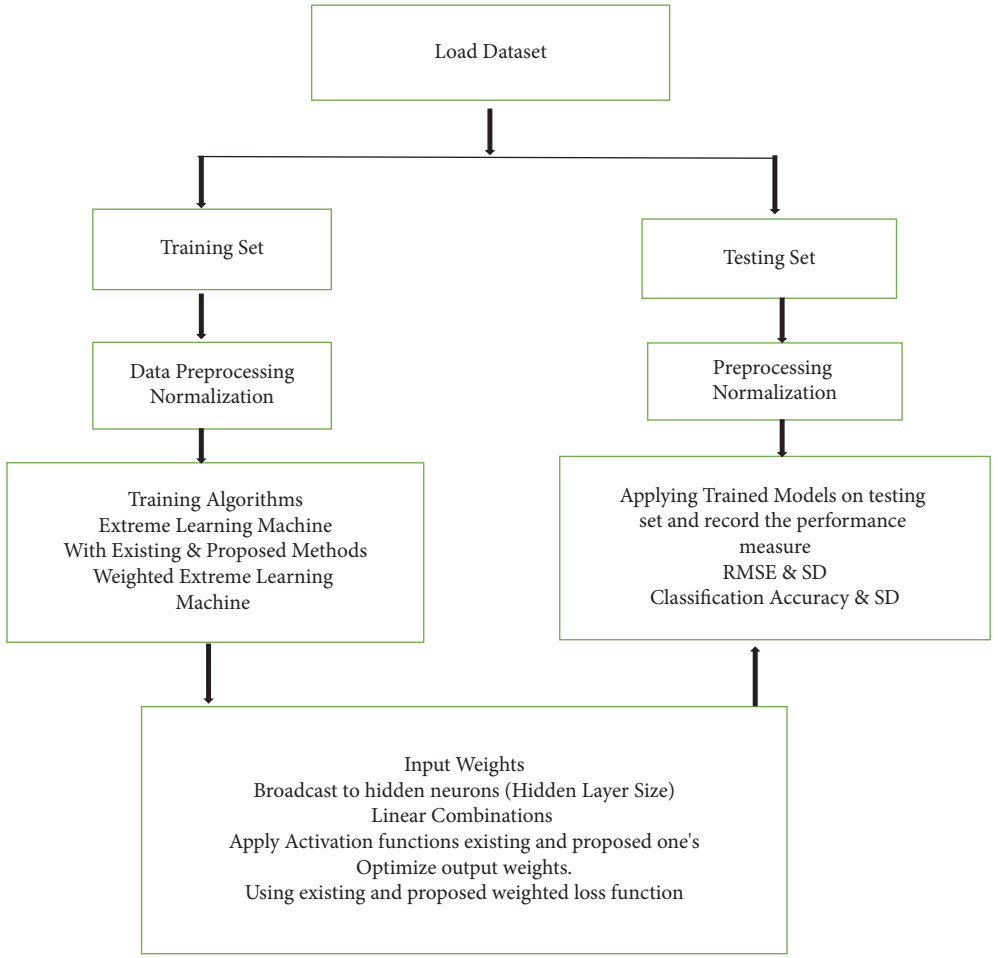

Figure 2: Block diagram of the proposed study.

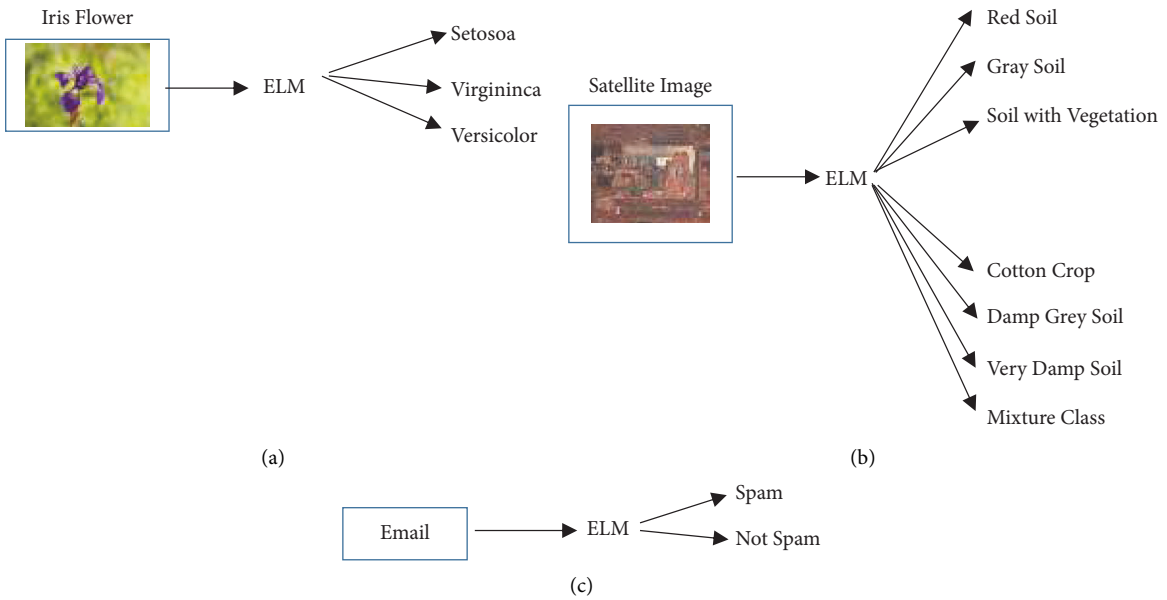

Figure 3: (a) Iris flower detection, (b) SatImage classification, and (c) emails spamming filtering.

Table 4: Testing RMSE and SD of existing and proposed weighted ELM in clean abalone data.

Af	ELM	W1ELM	W2ELM	W3ELM Proposed	W4ELM Proposed	W5ELM Proposed	W6ELM Proposed
Sigmoid	2.144143 0.1442943	2.175000 0.136926	2.348306 0.30963177	2.205588 0.16094535	2.191642 0.13586263	2.173392 0.1334078	2.189162 0.138817
Tan-sig	2.150399 0.1926798	2.123554 0.1643963	2.298845 0.24419012	2.157511 0.20161524	2.135203 0.16191567	2.126287 0.1626033	2.132039 0.164313
Sine	2.186667 0.2009761	2.129157 0.1452062	2.330158 0.15104044	2.161264 0.15971286	2.139394 0.13502665	2.133264 0.1450795	2.136425 0.137740
Cosine	2.213637 0.2063120	2.186199 0.1776403	2.384910 0.19514964	2.248835 0.20166515	2.196252 0.16658765	2.185097 0.1719300	2.195070 0.1720428
BentIdle	2.132008 0.1605604	2.129026 0.1561023	2.277284 0.09741951	2.156375 0.17502070	2.137727 0.14920465	2.132926 0.1559677	2.134048 0.1507907
RAF	2.145623 0.1463326	2.161926 0.1426574	2.374974 0.17473441	.235957 0.23472574	2.180706 0.14963018	2.159569 0.1337737	2.178774 0.1537632
Proposed 1	2.166058 0.1874100	2.131701 0.14846	2.284179 0.24488947	2.157106 0.15988542	2.137113 0.13883288	2.136875 0.1488099	2.134278 0.1413268
Proposed 2	2.151613 0.1661444	2.133518 0.1576740	2.250627 0.26301000	2.151150 0.18135924	2.138221 0.14918027	2.138806 0.1575292	2.135509 0.1522103
Proposed 3	2.380380 0.3186960	2.286334 0.2351961	2.428068 0.28907843	2.324140 0.26964348	2.283338 0.21956250	2.282506 0.2278787	2.286572 0.2275768
Proposed 4	2.192119 0.2036608	2.161120 0.1775893	2.224148 0.12617183	2.154010 0.17595357	2.159759 0.16575919	2.168203 0.1807883	2.157113 0.1676623
Proposed 5	2.145925 0.1535060	2.162594 0.1468836	2.397043 0.30858832	2.238244 0.23296731	2.183591 0.15598956	2.162214 0.1404016	2.180518 0.1585908
Proposed 6	2.139137 0.1634156	2.162571 0.1887972	2.399246 0.23987476	2.212400 0.21515337	2.179009 0.18678814	2.160730 0.1799538	2.176938 0.1914957
Proposed 7	2.321377 0.2732022	2.195224 0.1883943	2.358959 0.11484580	2.245618 0.25699135	2.210419 0.19094163	2.193319 0.1768587	2.207753 0.1976999
Proposed 8	2.167783 0.2002444	2.633435 0.1051162	2.609206 0.11752655	2.850021 0.02805903	2.748345 0.07666315	2.616541 0.1089142	2.730395 0.0814038
Proposed 9	2.155725 0.1738043	2.143697 0.1383195	2.321712 0.16859567	2.195728 0.19172220	2.158484 0.13940778	2.145574 0.1338223	2.155555 0.1426305
Proposed 10	2.130854 0.1575525	2.134321 0.1507116	2.179927 0.12246446	2.158516 0.13017982	2.145262 0.14917700	2.136971 0.1535475	2.142324 0.1481731
Proposed 11	2.138887 0.1472135	2.120890 0.1389861	2.242633 0.05301144	2.164488 0.18233037	2.129773 0.13005945	2.125429 0.1396772	2.126699 0.1324753

Step 4. If $k > k_{\max}$ or $\|\beta^{(k+1)} - \beta^{(k)}\| \le 0.001$, stop, and $\widehat{\beta} = \beta^{(k+1)}$; else go to step 5.

Step 5. Finally, the estimate function is given in equation (1)

1.3. Experimental Design and Simulation Studies. This section elaborates on the mechanism to know the performance of the proposed method against RELM and the existing weighted RELM. Several redescending M-estimators based on psi-functions are considered as activation functions to build hidden layer nonlinear space from the input space. As in ELM-related literature, common activation functions are used such as logistic sigmoid, tan-sigmoid, ReLU, softsign, Sin, Cos, leakyReLU, BentIdle, and Arc Tan in ELM and its variants including [1–25] respectively. Furthermore, we use redescending M-estimators based on nonconvex loss functions to reduce the effect of outliers in our proposed studies. The details of the nonconvex loss functions based on M-estimation are mentioned in Table 2 and in Table 3. The proposed psi functions are mentioned for convenience. However, we use the different numbers of hidden layer neurons to assess the performance of the proposed strategy. The results are shown here only for a single number, as our objective is not here to optimize hidden layer size. All experiments are carried out in an R-studio environment running on an Intel Core m3 7th Gen PC. In each experiment datasets are broken into two halves with a ratio of 70 : 30 where training and testing data sets are 70% and 30%, respectively. We have checked the performance of the proposed methods using two benchmark regression-related datasets, such as Boston Housing Price data and abalone age prediction data; however, only the results of the abalone data set with and without artificial outliers are kept to assess the performance of each method due to space limitation. Different scaling techniques were used in literature to reduce the size of data such as linear scaling minimax (0, 1) or (–1, 1) or statistical standardization We have considered minimax techniques to scale all data available on attributes and response variables to the range of (0, 1) before training the proposed and existing networks. The training dataset is contaminated in each trial with 20% outliers generated from uniform distribution but highly distant from the remaining dataset, which trains both the existing RELM and proposed RELM and check their performance on the test set. Proposed RELM and existing methodologies are repeatedly performed

Table 5: Testing RMSE and SD of existing and proposed weighted ELM with 20% outliers in abalone data.

Af	ELM	W1ELM	W2ELM	W3ELM Proposed	W4ELM Proposed	W5ELM Proposed	W6ELM Proposed
Sigmoid	4.754750	3.227287	2.414179	2.287563	2.172357	3.757036	2.234866
	0.06268815	0.1091655	0.09269088	0.12418925	0.038659	0.103437	0.06961570
Tan-sig	4.703597	3.132457	2.406314	2.276477	2.170417	3.615910	2.239491
	0.13000821	0.1208053	0.01755072	0.07965595	0.010285181	0.134645	0.010221790
Sine	4.666985	3.060303	2.361913	2.265762	2.150898	3.528365	2.210075
	0.20622199	0.2181420	0.06551598	0.10267846	0.006244196	0.2596405	0.033152012
Cosine	4.698598	3.180699	2.399364	2.308385	2.198570	3.701441	2.249796
	0.20381093	0.2577677	0.21437287	0.14727834	0.0875391569	0.2523779	0.145378848
BentIdle	4.713415	3.143851	2.390502	2.287222	2.163265	3.644029	2.228988
	0.13830989	0.1155777	0.03791609	0.11066652	0.0004587613	0.1344872	0.018264087
RAF	4.750438	3.196679	2.378543	2.267268	2.133434	3.729898	2.204034
	0.10228568	0.1157485	0.09912061	0.12104854	0.0453002613	0.1163393	0.072324507
Proposed 1	4.705598	3.089672	2.378945	2.265295	2.057172	3.565513	2.219889
	0.14179679	0.2089739	0.06826299	0.09770380	0.0025575652	0.2448967	0.032597902
Proposed 2	4.751669	3.097359	2.375285	2.276195	2.119978	4.582049	2.223024
	0.14523476	0.1859790	0.06077762	0.09643253	0.00707343	0.2195192	0.037928300
Proposed 3	4.409515	3.089344	2.462548	2.350298	2.271832	3.504199	2.318664
	0.66763602	0.4398715	0.24848033	0.20780245	0.0850197195	0.5013264	0.160961103
Proposed 4	4.754798	3.120053	2.426455	2.297376	2.205088	3.583513	2.267244
	0.23081970	0.2150026	0.05665887	0.08989809	0.0134925439	0.2651077	0.036196250
Proposed 5	4.599507	3.074987	2.396853	2.315319	2.190741	3.530176	2.247327
	0.30391626	0.2746488	0.10360959	0.15672495	0.0127734590	0.3270949	0.061538856
Proposed 6	4.595273	3.117355	2.435072	2.321735	2.218868	3.569725	2.280369
	0.29349235	0.2979396	0.13227469	0.17585465	0.0340319304	0.3348503	0.088922964
Proposed 7	5.291630	3.723273	3.230150	2.554555	2.965076	4.081953	3.003771
	1.54315324	1.2721507	1.31582437	0.51404176	1.1100654156	1.2158311	1.144834877
Proposed 8	4.733634	3.314672	2.763327	2.598141	2.501335	3.696463	2.558568
	0.17460431	0.1572617	0.10634390	0.09347673	0.0801301294	0.1703520	0.099964180
Proposed 9	4.606322	3.073412	2.388054	2.289723	2.179600	3.532195	2.237992
	0.33914493	0.2470656	0.09013738	0.14496423	0.0151039952	0.2858742	0.060854415
Proposed	4.818178	3.134719	2.378025	2.268276	2.147976	3.638479	2.212687
10	0.23783341	0.1284443	0.06425884	0.08654497	0.0408592032	0.1393702	0.059960084
Proposed	4.716515	3.106628	2.378284	2.274507	2.122296	3.590920	2.217427
11	0.12247669	0.1597634	0.02981345	0.06553342	0.0231154939	0.1990009	0.002427482

Table 6: Experimental results on real–world dataset abalone with 0%–20% Outlier's levels.

Wang et al. [35] used sigmoid activation function in existing ELM and its variants					
ELM		WELM	ORELM	IRRELM	IRRELM
Outliers	(RMSE, SD)	(RMSE, SD)	(RMSE, SD)	(RMSE, SD)	(RMSE, SD)
0%	2.1382, 0.0692	2.1532, 0.0737	2.1909, 0.0576	2.2106, 0.0646	2.1350, 0.0625
5%	2.3182, 0.0533	2.1557, 0.0609	2.1712, 0.0589	2.1928, 0.0632	2.1455, 0.0552
10%	2.6679, 0.0533	2.1603, 0.0597	2.1667, 0.082	2.1724, 0.0579	2.1499, 0.0621
15%	3.1917, 0.0763	2.1701, 0.0909	2.1713, 0.0820	2.1492, 0.0603	2.1508, 0.0643
20%	3.2161, 0.0976	2.2297, 0.0636	2.2168, 0.0741	2.1694, 0.0755	2.1633, 0.0716

50 times, and computed training and testing root mean square (RMSE) and their standard deviation (SD) are recorded in the case of regression. The step-by-step abstract block diagram of the proposed strategy is defined in Figure 2. While knowing the performance of the proposed strategy in classification, we consider a benchmark dataset IRIS, satellite image, and e-mails spam filtering datasets considered from the UCI machine repository. The photos of classification applications are shown in Figure 3 to know the importance of the proposed work easily. We have used three random choices of weights initialization from standard normal, uniform (–1, 1), and exponential distribution. However, due to space limitations, the results of standard normal distribution are kept into consideration as there is found no significant impact on initial weights.

1.4. Performance Metrics Root. Mean Square Error (RMSE) = $\sqrt{\sum_{i=1}^{N}((y-\hat{y})^2)/N}$ *Mean RMSE* = $\sum_{j=1}^{50} RMSE/50$, $SD = \sqrt{(\sum_{j=1}^{5}{}^{0}(RMSE - meanRMSE)^2/5 \quad 0)}$,

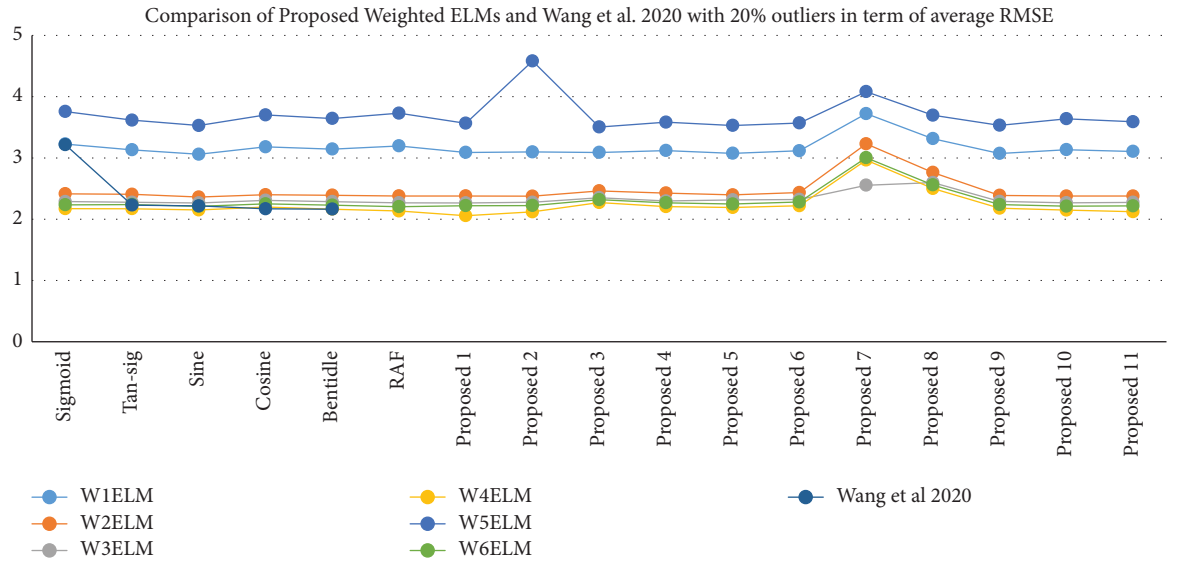

Figure 4: Comparison of proposed weighted ELMS and Wang et al. 2020 weighted ELMs.

Table 7: Percent accuracy with SD of different existing and proposed activation functions in basic ELM.

Af	ELM 50 IRIS	ELM 300 SatImage	ELM 1000 E-mail spam filtering
Sigmoid	95.67, 4.83	87.5, 0.518874500	88.00000, 1.387233
Tan-sigmoid	96.00, 2.11	87.7857, 0.6992932	88.00000, 1.732051
Sine	95.33333, 4.216370	88.5714, 0.9376145	88.66667, 1.527525
Cosine	98.00000, 2.810913	88.3571, 0.6333237	88.66667, 1.527525
Bentidle	97.00000, 3.314763	86.21429, 0.8418974	90.33333, 0.5773503
RAF	96.00000, 3.784308	88.92857, 0.8287419	76.33333, 0.5773503
Proposed-1	97.66667, 2.249829	88.07143, 0.6157279	88.33333, 1.527525
Proposed-2	95.66667, 3.531166	88.21429, 0.6112498	88.66667, 1.154701
Proposed-3	85.66667, 6.675920	75.00, 0.6793662000	88.66667, 1.527525
Proposed-4	96.00000, 4.097575	89.85714, 0.8644378	88.66667, 1.527525
Proposed-5	96.66667, 2.721655	87.78571, 0.5789342	89.00000, 1.732051
Proposed-6	97.00000, 4.830459	86.85714, 0.7449463	89.00000, 1.732051
Proposed-7	96.33333, 1.892154	60.0, 5.7912400000	70.66667, 1.154701
Proposed-8	98.33333, 2.357023	88.07143, 0.9168748	93.66667, 0.5773503
Proposed-9	95.66667, 2.744242	87.64286, 0.7449463	93.66667, 0.5773503
Proposed-10	96.66667, 3.513642	77.78571, 1.5281250	71.66667, 0.5773503
Proposed-11	95.33333, 4.499657	88.28571, 0.6112498	71.66667, 0.5773503

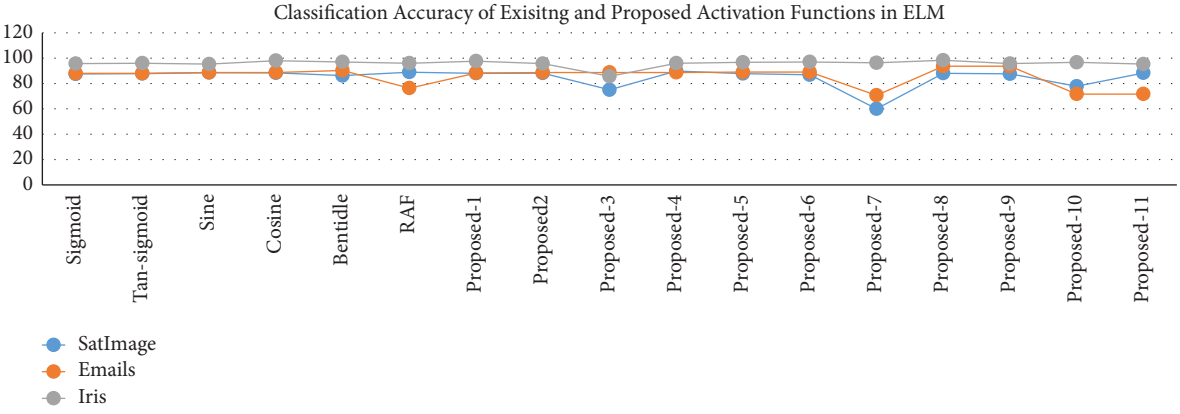

Figure 5: Classification accuracy of existing and proposed activation functions in ELM.

Accuracy = No. of correctly classified classes by classifier/total cases $*$ 100.%SD = $\sqrt{(\sum_{j=1}^{50} (Accuracy - \overline{Accuracy})^2/50)}$.

2. Results and Discussions

The results of the simulation study revealed in Tables 4–5, explain the performance of each activation function complemented with their corresponding loss function in terms of RMSE and SD in regression application.

In Table 4 and Table 5, the application of proposed methods extends to well-known data abalone age prediction, where proposed methods outperform in terms of RMSE and SD as compared to other state-of-the-art techniques in both clean and contaminated data as well. For further confirmation of the performance of the proposed methods, we compared robust ELM using the sigmoid activation function by Wang et al. [35], called iterative reweighted ELM (IRRELM) compared with ELM, WELM, ORELM, and IRWELM, whose results are mentioned in Table 6 with different contamination levels and further performance of the proposed methods are shown in Figure 4**.** If we see their results and compar them with our proposed methods, they clearly show improvement in efficiency even at all levels of outliers. To further check the performance of proposed methods, we extended their applications to classification problems. We considered three popular benchmark data sets to show the classification accuracy of the proposed methods. The following table demonstrates the efficiencies of existing methods and proposed methods.

Table 7 along with Figure 5 describes the performance of the proposed methods in terms of percentage testing accuracy with SD using well-known low- and high-dimensional data clearly depicted in Figure 4. To clarify our results given in Table 7, for each trial of simulation of Iris, Satimage, and emails spam filtering, the training, and testing datasets are randomly generated from their database. Fifty trials have been set for the ELM algorithm using different activation functions trained under a fixed hidden layer size, and at the end, average testing accuracy was measured with standard deviation. In the case of the Satimage dataset, one of our proposed activation functions reaches an average accuracy of 89.86% with a 0.8644 standard deviation. These results are compared with Huang et al. [2]. The ELM classifier with sigmoid activation function got 89.04% accuracy with a standard deviation of 1.57 using 500 nodes in the hidden layer, whereas the proposed activation function achieved a higher accuracy with only 300 hidden nodes in the hidden layer, which clearly showed that ELM under the proposed activation function makes the desired classifier have less computational complexity with higher accuracy. The remaining proposed activation functions in the same experiment showed almost similar performance to all competitors. In the case of another high-dimensional data, "emails" proposed 8 and 9 number activation functions outperform all existing and proposed activation functions with an higher average accuracy of 93.66. In Iris data, the classification accuracy of the proposed and existing activation functions are the same nearly.

3. Conclusion

This paper proposed a new robust activation function complemented by weight-loss on M and redescending M-estimation in ELM with l_2-norm regularization criteria for solving regression and classification problems. The focus of this work was to introduce the psi function of different redescending M-estimators as activation functions in ELM, complemented by existing and some new weight-loss functions. In the task of prediction, the proposed methods show improvement in terms of accuracy and precision over existing methods for predicting the age of abalone with and without adding outliers to the training set. Several combinations of activation function and loss function are studied and compared with their performance with proposed combinations. The performance of the proposed combination of activation and weight-loss functions outperformed existing methods in regression problems in the presence of contaminations. Moreover, the application of the proposed activation function in ELM is extended to know the classification accuracy in low and high-dimensional data sets. In almost all classification applications, the predictive performance of proposed activation functions in ELM outperformed. For instance, in the case of regression application using an abalone dataset, the proposed activation function along with the weighted loss function performed better than the existing combination of activation and weight-loss function in extreme learning machine in the presence of outliers. Furthermore, the application of proposed activation functions was deployed to classification problems using the famous Iris, Satimage, and emails datasets, where some of the proposed activation functions outperform their existing competitors. In the future, the role of the proposed activation function in ELM can be studied with different convolutional neural networks (CNNs), such as Google Net, Alex Net, VGG-16, and ResNet, using feature selection techniques from image data along with famous robust statistical feature selection methods. Moreover, in the future, the applications of proposed activation functions in extreme learning machines can be extended to analyze their performance in epilepsy identification from EEG signals, emotion recognition in Election Day tweets, total iron detection, and fault diagnostic of electric impact drills using thermal imaging.

Data Availability

Data and programming codes available on request. The manuscripts are available upon contacting the first author. Moreover, used datasets are available on Kaggle repository.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Q. Y. Zhu, G. B. Huang, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," *Neural Networks*, vol. 2, pp. 985–990, 2004.

[2] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[3] M. A. A. Albadra and S. Tiuna, "Extreme learning machine: a review," *International Journal of Applied Engineering Research*, vol. 12, no. 14, pp. 4610–4623, 2017.

[4] R. Harikumar, C. Ganesh Babu, and M. Gowri Shankar, "Extreme learning machine (ELM) based performance analysis and epilepsy identification from EEG signals," *IETE Journal of Research*, vol. 67, pp. 1–11, 2021.

[5] X. Li, J. Sha, and Z. L. Wang, "Comparison of daily streamflow forecasts using extreme learning machines and the random forest method," *Hydrological Sciences Journal*, vol. 64, no. 15, pp. 1857–1866, 2019.

[6] A. Bhatia, A. Chug, and A. Prakash Singh, "Application of extreme learning machine in plant disease prediction for highly imbalanced dataset," *Journal of Statistics & Management Systems*, vol. 23, no. 6, pp. 1059–1068, 2020.

[7] J. Li, W. Shi, and D. Yang, "Fabric wrinkle evaluation model with regularized extreme learning machine based on improved Harris Hawks optimization," *Journal of the Textile Institute*, vol. 113, no. 2, pp. 199–211, 2022.

[8] D. Xiao, H. Xie, Y. Fu, and F. Li, "Mine reclamation based on remote sensing information and error compensation extreme learning machine," *Spectroscopy Letters*, vol. 54, no. 2, pp. 151–164, 2021.

[9] M. K. Giri and S. Majumder, "Cooperative spectrum sensing using extreme learning machines for cognitive radio networks," *IETE Technical Review*, vol. 39, no. 3, pp. 698–712, 2021.

[10] Y. Fu, L. Jiang, Y. Zhang, L. Wan, and D. Xiao, "Detection of total iron content based on improved extreme learning machine," *Spectroscopy Letters*, vol. 55, no. 4, pp. 284–289, 2022.

[11] M. Heydari, S. Shabanlou, and B. San Ahmadi, "Evaluation of shape factor impact on discharge coefficient of side orifices using boost simulation model with extreme learning machine data-driven," *Network: Computation in Neural Systems*, vol. 32, no. 2-4, pp. 83–109, 2021.

[12] B. T. Le, D. Xiao, Y. Mao et al., "Coal exploration based on a multilayer extreme learning machine and satellite images," *IEEE Access*, vol. 6, pp. 44328–44339, 2018.

[13] N. Nisar, N. Rakesh, and M. Chhabra, "Review on email spam filtering techniques," *International Journal of Performability Engineering*, vol. 17, no. 2, p. 178, 2021.

[14] F. N. Funding, P. R. P V G D, and K. Venkata Rao, "Emotion recognition in election day tweets using optimised kernel extreme learning machine classifier," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 34, pp. 1–19, 2022.

[15] G. B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International journal of machine learning and cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[16] G. B. 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, 2012.

[17] R. Zhang, Y. Lan, G. B. Huang, and Z. B. Xu, "Universal approximation of extreme learning machine with adaptive growth of hidden nodes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 2, pp. 365–371, 2012.

[18] S. Ding, X. Xu, and R. Nie, "Extreme learning machine and its applications," *Neural Computing & Applications*, vol. 25, no. 3-4, pp. 549–556, 2014.

[19] G. B. Huang, "What are extreme learning machines? Filling the gap between," *Cognitive Computation*, vol. 7, pp. 263–278, 2015.

[20] G.-B. Huang, "Frank Rosenblatt's dream and John von Neumann's puzzle," *Cognitive Computation*, vol. 7, no. 3, pp. 263–278.

[21] G. Huang, G. B. Huang, S. Song, and K. You, "Trends in extreme learning machines: a review," *Neural Networks*, vol. 61, pp. 32–48, 2015.

[22] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Data Mining*, pp. 389–395, IEEE, Nashville, TN, USA, March 2009.

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

[24] A. Freire and G. Barreto, "A Robust and Regularized Extreme Learning Machine," *Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2014)*, vol. 2014, pp. 1–6, 2014.

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

[26] 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.

[27] I. Ullah, M. F. Qadir, and A. Ali, "Insha's redescending M-estimator for robust regression: a comparative study," *Pakistan Journal of Statistics and Operation Research*, vol. 2, no. 2, pp. 135–144, 2006.

[28] Alamgir, S. A. Khan, D. M. Khan, and U. Khalil, "A new efficient re-descending M-estimator: alamgir redescending M-estimator," *Research Journal of Recent Sciences ISSN*, vol. 2277, p. 2502, 2013.

[29] D. M. Khan, M. Ali, Z. Ahmad, S. Manzoor, and S. Hussain, "A new efficient redescending M-estimator for robust fitting of linear regression models in the presence of outliers," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–11, 2021.

[30] S. Liu, L. Feng, Y. Xiao, and H. Wang, "Robust activation function and its application: semi-supervised kernel extreme learning method," *Neurocomputing*, vol. 144, pp. 318–328, 2014.

[31] P. Sibi, S. A. Jones, and P. Siddarth, "Analysis of different activation functions using back propagation neural networks," *Journal of Theoretical and Applied Information Technology*, vol. 47, no. 3, pp. 1264–1268, 2013.

[32] G. S. D. S. Gomes, T. B. Ludermir, and L. M. Lima, "Comparison of new activation functions in a neural network for forecasting financial time series," *Neural Computing & Applications*, vol. 20, no. 3, pp. 417–439, 2011.

[33] M. H. Essai and A. R. A. Ellah, "M-estimators-based activation functions for robust neural network learning," in *Proceedings of the 2014 10th International Computer Engineering Conference (ICENCO)*, pp. 70–75, IEEE, Giza, Cairo, Egypt, 2014 December.

[34] A. Freire and G. Barreto, "A robust and regularized extreme learning machine," *Encontro Nacional De Inteligncia Artificial E Com-Putacional*, vol. 2014, pp. 1–6, 2014.

[35] 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, Article ID 125186, 2020.