Neural Processing Letters (2024) 56:5
https://doi.org/10.1007/s11063-024-11443-0

An Enhanced Extreme Learning Machine Based on Square-Root Lasso Method

Murat Genç[1]

Accepted: 30 October 2023 / Published online: 6 February 2024
© The Author(s) 2024

Abstract
Extreme learning machine (ELM) is one of the most notable machine learning algorithms with many advantages, especially its training speed. However, ELM has some drawbacks such as instability, poor generalizability and overfitting in the case of multicollinearity in the linear model. This paper introduces square-root lasso ELM (SQRTL-ELM) as a novel regularized ELM algorithm to deal with these drawbacks of ELM. A modified version of the alternating minimization algorithm is used to obtain the estimates of the proposed method. Various techniques are presented to determine the tuning parameter of SQRTL-ELM. The method is compared with the basic ELM, RIDGE-ELM, LASSO-ELM and ENET-ELM on six benchmark data sets. Performance evaluation results show that the SQRTL-ELM exhibits satisfactory performance in terms of testing root mean squared error in benchmark data sets for the sake of slightly extra computation time. The superiority level of the method depends on the tuning parameter selection technique. As a result, the proposed method can be considered a powerful alternative to avoid performance loss in regression problems .

Keywords Extreme learning machine · Ridge · Lasso · Square-root lasso · Regression

1 Introduction

Feedforward neural networks (FNN) have been comprehensively studied and widely used in many fields of science because of approximating complex nonlinear mappings and modeling of various natural and artificial phenomena. Many researchers investigate the universal approximation capabilities of standard multi-layer FNN architectures, including [24, 25, 38]. Real data applications of the neural network are based on a finite training set. [28] proved that a finite number of distinct observations can be learned with zero error by a single-hidden layer feedforward neural network (SLFN) with at most the same finite number of hidden neurons and with almost any nonlinear activation function.

✉ Murat Genç
muratgenc@tarsus.edu.tr

[1] Department of Management Information Systems, Faculty of Economics and Administrative Sciences, Tarsus University, 33400 Mersin, Turkey

Although SLFN is one of the most popular FNNs, the learning algorithms used to train the SLFN are slow since its parameters are tuned using iterative procedures and these parameters may get stuck at a local minimum [12]. To deal with these drawbacks, [33] proposed the extreme learning machine (ELM), which is a fast learning algorithm based on the SLFN. The applications of ELM are primarily regression, classification and clustering tasks [29, 34]. The main distinction of ELM from traditional learning algorithms for neural networks is that it does not need to determine the input weights iteratively. In the ELM algorithm, the learning parameters (input weights and biases) of hidden layer units are determined by random assignments and the output weights are obtained by the generalized inverse operation. In the last step of the training process of ELM, a linear or non-linear transformation is applied to the output weights. This procedure leads to notable benefits including faster learning speed, promising generalization capability and less human intervention [30, 31]. Since ELM is proposed by [33], ELM has been successfully applied to many real-life problems (see, [7, 35–37, 40, 49, 50, 55, 56, 58, 60, 61]). Although ELM has shown remarkable success on many real data sets, it may give poor results in terms of stability, generalization performance and sparsity depending on the nature of the underlying data set. Multicollinearity occurs when the columns of the output matrix of the hidden layer are moderately or highly correlated with one another. The multicollinearity leads the ELM results to be unstable and affects the generalization performance, adversely. Also, ELM may suffer from overfitting or underfitting because of interference of redundant information when the number of hidden neurons is not properly selected [42].

Regularization methods have been proposed to deal with the drawbacks of basic ELM. There are ℓ_1 and ℓ_2 norm-based regularization methods to improve the underlying model in points of multicollinearity, overfitting and sparsity. [23] proposed an ℓ_2 norm-based regularization method called ridge regression to handle the multicollinearity for linear models. Although ridge regression improves the model in the case of multicollinearity, it does not carry out automatic variable selection. [53] introduced the lasso as an alternative to ridge regression, which is an ℓ_1 norm-based regularization method. The lasso performs automatic variable selection as a result of the nature of the ℓ_1 norm. [69] proposed the elastic net, which employs a linear combination of ℓ_1 and ℓ_2 norm regularizations. The elastic net tends to group highly correlated variables and improves the lasso in case of multicollinearity. [4] proposed a modified version of the lasso, called the square-root lasso by modifying the form of the objective function of the lasso. The square-root lasso carries out the automatic variable selection in a similar way to the lasso while it is a pivotal method in that the method does not rely on the knowledge or pre-estimate of the standard deviation of the noise. The square-root lasso also does not rely on the normality assumption of the noise. Hence, the square-root lasso potentially outperforms the lasso in variable selection.

Recently, many algorithms have been proposed as variants of basic ELM to improve the performance of the method depending on the structure of the underlying data set. The majority of these works are ℓ_1 and ℓ_2 norm-based regularization methods. [48] proposed the pruned-ELM, a systematic and automated method for ELM classifier, based on constituting the appropriate classifier network by measuring the relevance of the hidden nodes through statistical methods. [17] introduced the error-minimized extreme learning machine that updates the output weights incrementally during the growth of the network to determine the number of hidden nodes automatically. [32] examined the ELM for classification in the aspect of the standard optimization method and extended the ELM to the support vector network, a specific type of generalized SFLN. [11] proposed regularized ELM considering the structural risk minimization principle and weighted least squares to tackle the drawbacks of ELM in cases of heteroskedasticity, outliers and over-fitting. [43] introduced the optimally pruned

ELM (OP-ELM) to reduce the effect of the irrelevant variables on the ELM predictions. The OP-ELM uses the leave-one-out (LOO) validation method to achieve the best number of neurons. Since the computation of the LOO error is based on PRESS statistic, the OP-ELM is sensitive to the high correlations among variables. To cope with this situation, [44] proposed a double-regularized ELM called Tikhonov regularized ELM (TROP-ELM) to improve the numerical stability with efficient pruning of neurons compared to OP-ELM. [42] proposed a regularized ELM for regression problems by using ridge, lasso and elastic net methods. The idea of the proposed method is to prune the irrelevant nodes from a large number of hidden nodes determined initially, hence automating the architectural design of ELM. [39] proposed ridge regression ELM by applying ridge regression in the computation of output weights to handle the multicollinearity in the data set. [16] proposed ℓ_1 regularized ELM which is based on using lasso-type regularization in the computation of output weights to promote the sparsity and, hence, to improve the performance of the ELM. [2] proposed the knowledge-based ELM formulation by incorporating nonlinear prior knowledge in the form of implications into the ELM. [3] considered the primal form of optimization-based ELM [32], whose solution is obtained by solving an absolute value equation problem by a functional iterative method. [41] proposed $\ell_1 - \ell_2$-ELM, a unified form of ELM that combines the grouping property of ℓ_2 norm and sparsity property of ℓ_1 norm to control the complexity of the network and deal with over-fitting. [5] proposed the implicit Lagrangian twin random vector functional-link networks through the unconstrained minimization problem for better generalization performance in ELM. [62] observed that the regularization parameter selection criteria affected the performance of the ridge regression ELM and investigated the performance of the ridge regression ELM by various regularization parameter selection criteria. [9] proposed the bootstrapped lasso as a regularization and resampling method to select the most related neurons for the model. [57] proposed $\ell_{2,1}$-norm-based regularization by the difference of convex functions program to obtain a more robust and stable model by reducing the influence of outliers. [21] proposed a robust regularized extreme learning machine by using the asymmetric Huber loss function to reduce the effect of noise and outliers in the data. [22] proposed regularized based implicit Lagrangian twin extreme learning machine in primal as a pair of unconstrained convex minimization problems where the regularization term is added to follow the structural risk minimization principle. [47] proposed a novel online sequential algorithm for ELM with $\ell_{2,1}$-norm regularization to deal with real-time sequential data. [64] proposed the Liu-lasso ELM as a regularization and variable selection method and compared the performance of the method with ridge, liu, lasso and elastic net regularized ELM methods. [66] developed a novel elastic net regularized ELM algorithm based on the over-relaxed alternating direction method of multipliers to reduce the model training time by using the results of the previous iteration to the next iteration. [26] proposed a least squares ELM with derivative characteristics in which weights and biases of the network are specified by a twice least squares method to improve the regression accuracy of the network. [65] proposed a combination of ridge and Liu regressions in a unified form in the ELM context as a remedy to the disadvantages of the basic ELM algorithm. [59] established a connection between the variance and weights of the ELM for regression problems and proposed maximum likelihood-based estimates for model parameters. [67] proposed a regularized functional extreme learning machine that uses the regularization function instead of a preset regularization parameter to select appropriate regularization parameters adaptively.

In this study, we present square-root lasso ELM (SQRTL-ELM), an enhanced ELM based on the pivotal recovery of signals in the hidden layer output matrix. The SQRTL-ELM calculates output weights by square-root lasso to achieve better stability and generalization capability compared to the other regularized ELM methods.

The rest of the paper is organized as follows: A brief review of the algorithms is given in Sect. 2. The details and computational properties of the proposed method are presented in Sect. 3. In Sect. 4, an experimental study is carried out to examine the performances of all the methods on real data sets. Finally, the study is concluded in Sect. 5.

2 A Review of Algorithms

2.1 Extreme Learning Machine (ELM)

In this section, the ELM algorithm is briefly reviewed. The ELM is a special SLFN in which the weight matrix between the input layer and output layer is chosen randomly. Hence, the estimation problem of ELM is reduced to a solution of a set of linear equations and the output layer weights can be estimated by least squares [27, 34]. As a result of this training process, ELM is a faster learning algorithm compared to the other algorithms such as backpropagation for training SLFN.

Consider N distinct training samples $(\mathbf{x}_i, \mathbf{t}_i)$ where $\mathbf{x}_i \in \mathbb{R}^N$ is the extracted feature vector and $\mathbf{t}_i \in \mathbb{R}^p$ is the target vector. The mathematical model for the SFLN with L hidden nodes and activation function $g(x)$ is

$$\sum_{i=1}^{L} \boldsymbol{\beta}_i^T g\left(\mathbf{w}_i \cdot \mathbf{x}_j + b_i\right) = \mathbf{t}_j, \;\; j = 1, 2, \ldots, N, \tag{1}$$

where $\mathbf{w}_i \in \mathbb{R}^N$ is the weight vector that connects the input neurons to the ith hidden neuron, $\boldsymbol{\beta}_i$ is the weight vector that connects the output neurons to i the hidden neuron, b_i is the bias of the ith hidden neuron, and $\mathbf{w}_i \cdot \mathbf{x}_j$ denotes the inner product of the vectors $\mathbf{w}_i$ and $\mathbf{x}_j$. The structure of basic ELM is given in Fig. 1.

The matrix form corresponding to Eq. (1) is written as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T},$$

where

$$\mathbf{H} = \begin{bmatrix} g\left(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1\right) & \cdots & g\left(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L\right) \\ \vdots & \ddots & \vdots \\ g\left(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1\right) & \cdots & g\left(\mathbf{w}_L \cdot \mathbf{x}_N + b_L\right) \end{bmatrix}_{N \times L}, \tag{2}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}^T \end{bmatrix}_{L \times p} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times p}. \tag{3}$$

The $\mathbf{H}$ matrix in Eq. (3) is called the hidden layer output matrix of the neural network. Therefore, the weight matrix $\boldsymbol{\beta}$ can be obtained analytically by the least squares approach

$$\hat{\boldsymbol{\beta}}_{\text{ELM}} = \mathbf{H}^{+}\mathbf{T}, \tag{4}$$

where $\|\cdot\|_2^2$ denotes the ℓ_2-norm of the vector and $\mathbf{H}^{+}$ is the Moore-Penrose generalized inverse of the matrix $\mathbf{H}$.

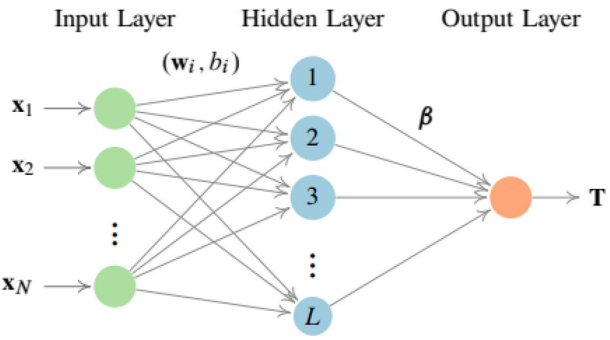


Fig. 1 The structure of ELM with one output

2.2 Regularized ELM

If the $\mathbf{H}$ matrix suffers from multicollinearity, the stability and generalization capability of the ELM are adversely affected since, in the case of multicollinearity, ELM results cannot be obtained or they will be unstable [39, 64]. Several ELM variants are proposed to deal with the multicollinearity in the hidden layer output matrix. Consider the setup for a single-output regression model

$$\mathbf{T} = \mathbf{H}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{5}$$

where $\boldsymbol{\varepsilon}$ is the noise vector which consists of elements with mean zero and constant variance σ^2. The ELM based on the ordinary least squares (OLS) estimates is formulated as the minimization of the loss function

$$\hat{\boldsymbol{\beta}}_{\text{ELM}} = \text{argmin}_{\boldsymbol{\beta}} \left[\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_2^2 \right], \tag{6}$$

which has a closed-form solution

$$\hat{\boldsymbol{\beta}}_{\text{ELM}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{T}$$

if $\mathbf{H}$ has full column rank. In the case of multicollinearity, $\hat{\boldsymbol{\beta}}_{\text{ELM}}$ may suffer from a large variance and yield unsatisfactory estimates [63].

[39] proposed the ridge regression ELM (RIDGE-ELM) as the solution to the problem

$$\hat{\boldsymbol{\beta}}_{\text{R-ELM}} = \text{argmin}_{\boldsymbol{\beta}} \left[\frac{1}{2N} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right], \tag{7}$$

where $\lambda > 0$ is the tuning parameter of RIDGE-ELM. The problem in Eq. (7) has a closed-form solution as

$$\hat{\boldsymbol{\beta}}_{\text{R-ELM}} = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}^T\mathbf{T},$$

where $\mathbf{I}$ is the $L \times L$ identity matrix. Introducing the regularization term reduces the variance of the estimates of $\boldsymbol{\beta}$ for the sake of some bias. The main benefit of the Ridge-ELM is that it ensures the stability of the model in case of multicollinearity in the $\mathbf{H}$ matrix. [44] proposed LASSO-ELM by replacing ℓ_2 regularization with ℓ_1 regularization in the criterion in Eq. (7) as

$$\hat{\boldsymbol{\beta}}_{\text{L-ELM}} = \text{argmin}_{\boldsymbol{\beta}} \left[\frac{1}{2N} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right]. \tag{8}$$

The LASSO-ELM aims to improve the performance of ELM by pruning some hidden nodes using a ℓ_1-type regularization term. However, the LASSO-ELM selects at most L variables, which is a limiting property for a variable selection method [41, 42]. To deal with the limitation of the LASSO-ELM in variable selection, [42] proposed to use a convex combination of ℓ_1 and ℓ_2 regularization, which leads to ENET-ELM as

$$\hat{\boldsymbol{\beta}}_{\text{ENET}-\text{ELM}} = \text{argmin}_{\boldsymbol{\beta}} \left[\frac{1}{2N} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_2^2 + \lambda \left(\alpha \|\boldsymbol{\beta}\|_1 + (1-\alpha) \|\boldsymbol{\beta}\|_2^2 \right) \right] \tag{9}$$

where α is a tuning parameter that controls the trade-off between the RIDGE-ELM ($\alpha = 0$) and LASSO-ELM ($\alpha = 1$). The ENET-ELM performs much like the LASSO-ELM for the small values of α, while dealing with the wild behavior arising from high correlations. However, tuning the second parameter α leads to an increase in the computational cost of the ENET-ELM.

3 The Proposed Algorithm: SQRTL-ELM

Despite the attractive properties of LASSO-ELM and ENET-ELM, their estimations depend on the standard deviation σ of the noise in the model (5). Considering this issue, [4] proposed the square-root LASSO (SQRTL) to provide pivotal recovery of sparse signals in the scope of regularized regression methods. The SQRTL is pivotal in terms of its regularization level since the regularization level is not dependent on the standard deviation of the noise. With the pivotal recovery property, The SQRTL can be used to enhance the performance of the ELM. Therefore, we propose square-root LASSO-ELM (SQRTL-ELM) by applying ℓ_1 type regularization to the square root of the loss function in (6):

$$\hat{\boldsymbol{\beta}}_{\text{SQRTL}-\text{ELM}} = \text{argmin}_{\boldsymbol{\beta}} \left[\frac{1}{2N} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_2 + \lambda \|\boldsymbol{\beta}\|_1 \right]. \tag{10}$$

In the SQRTL-ELM, the standard deviation σ becomes a constant factor that drops out of the minimization. This property of SQRTL-ELM enables the derivation of pivotal estimators for the lasso tuning parameter since there is no need to estimate σ as a part of the estimation formula of the tuning parameter. Hence, the SQRTL-ELM potentially outperforms the LASSO-ELM and ENET-ELM methods with the properly selected tuning parameter value.

3.1 Computational Properties of Sqrt-Lasso-Elm

The SQRTL-ELM method given by problem (10) does not have a closed-form solution because of the nature of the ℓ_1 norm. [4] proposed an interior-point method and a first-order method to obtain the solution to problem (10) by using the fact that the SQRTL is equivalent to a conic programming problem holding strong duality. [51] proposed the scaled lasso which has an equivalent form of SQRTL in coefficient estimates. [51] proposed the alternating minimization algorithm to obtain the scaled lasso estimates, which is based on solving a sequence of LASSO problems. The LASSO solution can be obtained by the coordinate descent algorithm, efficiently [18]. Also, the computational cost of solving the LASSO problem can be reduced by a large amount by using strong rules for screening out variables from the model [54]. Therefore, we use a modified version of the alternating minimization algorithm, which is based on altering the design matrix in SQRTL to the hidden layer output matrix to obtain the SQRTL-ELM solution in Eq. (10).

The selection of the tuning parameter λ is critical for the SQRTL-ELM method to give successful results in the prediction of the target vector. In this paper, we use three different techniques for the selection of the tuning parameter. Each technique has its theoretical background.

(λ_1) The first technique is k-fold cross-validation (CV) which is a commonly used technique to estimate the prediction error [15, 19, 45, 68]. In the k-fold cross-validation procedure, the data set is split into k subsets of approximately equal size and the model is trained and tested k times. In each replication, one of the subsets is used as testing data and the rest $k-1$ subsets serve as training data. Then, we combine these k estimates of the prediction error to obtain the cross-validation error. We select the tuning parameter that gives the model with minimum cross-validation error.

(λ_2) The second technique is to use the universal penalty level [13],

$$\lambda_2 = \sqrt{2\log(L)/N}. \tag{11}$$

[51] used the universal penalty level in the scope of the scaled LASSO estimator. [8] studied the theoretical properties of the universal penalty level.

(λ_3) The third technique is the quantile-based penalty proposed by [52]. Let $Q(u)$ be the negative u-quantile function of the standard normal distribution. The penalty level in this approach is determined by

$$\lambda_3 = \sqrt{2/N} \cdot Q(k/L) \tag{12}$$

where k is the solution of the equation $k = Q^4(k/L) + 2Q^2(k/L)$. [52] investigated the theoretical properties of the quantile-based penalty and provided oracle inequalities for prediction bounds based on this penalty level.

We give Algorithm 10 to summarize the details of the implementation of the SQRTL-ELM method. As seen from Algorithm 10, the CV technique selects the best model from a group of candidate models while the universal or quantile-based penalty technique is based on a single model corresponding to a single tuning parameter. Therefore, the computational cost of applying the CV technique is higher than the computational cost of applying other techniques.

4 Performance Evaluation of SQRTL-ELM

In this section, a performance comparison is carried out on several benchmark regression data sets to verify the effectiveness of the proposed method SQRTL-ELM. For this purpose, the data sets Auto MPG, Strikes, Delta Ailerons, Concrete and Yacht are retrieved from the UCI machine learning repository [14] and Sleep data from [1]. Each data set is split into training and testing sets by the proportions of 70% and 30%, respectively, and all the inputs and outputs are normalized for each data set into the interval $[-1, 1]$ before the trials of the experiments. The properties of the data sets are summarized in Table 1.

All the experiments are carried out on an `Intel Core i5 CPU 2.40 GHz with 8 GB RAM`. All the numerical computations of the algorithms are performed in `C++`, and the algorithms are implemented as `R` functions. Each experiment is repeated 50 times to reduce the effect of randomness. Also, the number of hidden nodes is fixed at 50 for the experiments, and the sigmoidal activation function is used for all the data sets.

The accuracy of the regularized ELM algorithms on the testing set is critically dependent on the right choice of the tuning parameters [6]. When the value of the tuning parameter is

Input : a training set: $\{(\mathbf{x}_i, \mathbf{t}_i)\}$;
an activation function: $g(x)$;
the maximum hidden node number: L;

Output: $\hat{\boldsymbol{\beta}}_{\mathrm{SL-ELM}}$

Generate learning parameters of hidden nodes $\mathbf{w}_i$ and b_i, $1 \leq i \leq L$, randomly;
Calculate the hidden layer output matrix $\mathbf{H}$ in Eq. (2);
if *tuning* == *CV* **then**
  Solve the problem in Eq. (10) for a grid of tuning parameter values as $\hat{\boldsymbol{\beta}}_{\mathrm{CV}}$;
  for $1 \leq m \leq size\left(\hat{\boldsymbol{\beta}}_{\mathrm{CV}}\right)$ **do**
    Calculate CV error corresponding to $\hat{\boldsymbol{\beta}}_{\mathrm{CV}}(m)$;
  end
  Find m^* corresponding to the minimum CV error among all possible models;
  Select the optimum weights vector as $\hat{\boldsymbol{\beta}}_{\mathrm{selected}} = \hat{\boldsymbol{\beta}}_{\mathrm{CV}}(m^*)$;

else if *tuning* == *universal* **then**
  Solve the problem in Eq. (10) for the universal penalty in Eq. (11) as $\hat{\boldsymbol{\beta}}_{\mathrm{univ}}$;
  Select the optimum weights vector as $\hat{\boldsymbol{\beta}}_{\mathrm{selected}} = \hat{\boldsymbol{\beta}}_{\mathrm{univ}}$;

else if *tuning* == *quantile* **then**
  Solve the problem in Eq. (10) for the quantile-based penalty in Eq. (12) as $\hat{\boldsymbol{\beta}}_{\mathrm{quantile}}$;
  Select the optimum weights vector as $\hat{\boldsymbol{\beta}}_{\mathrm{selected}} = \hat{\boldsymbol{\beta}}_{\mathrm{quantile}}$;
end

Algorithm 1: SQRTL-ELM.

Table 1 The properties of benchmark data sets used in the study

Data sets	Attributes	Sample size	Training set	Testing set
Auto Mpg	7	392	275	117
Strikes	7	625	438	217
Delta Ailerons	5	7129	4991	2138
Concrete	9	1030	721	309
Sleep	8	62	44	18
Yacht	7	308	216	92

very small, the algorithms tend to result in an over-fitted model. In the experiments, the tuning parameter λ is selected from the range $\left\{2^{-10}, 2^{-4}, \ldots, 2^{13}, 2^{14}\right\}$ by following [20, 46], and five-fold cross-validation is conducted to determine the best tuning parameter value for all the regularized ELM models. Also, the SQRTL-ELM models are fitted by using universal and quantile-based tuning parameters detailed in Sect. 3 to compare these techniques with the cross-validation approach. Moreover, since the ENET-ELM has two tuning parameters, α is fixed at 0.5 to fit the ENET-ELM model.

To compare the performance of the algorithms, the mean of the root mean squares error (RMSE) is computed for the training and testing sets and the mean of the optimal λ values is obtained. The standard deviations of the training and testing RMSE for each method are given to evaluate the stability performance of the methods. The number of hidden layer nodes is reported in order to compare the sparsity levels of the models. Also, the reduction rates are computed as $\mathrm{RR} = 100 \times \frac{\mathrm{RMSE(any\ algorithm)} - \mathrm{RMSE(SQRTL\text{-}ELM)}}{\mathrm{RMSE(any\ algorithm)}}$ to determine the gain in percent from using the proposed method. Moreover, the mean of the computation time of

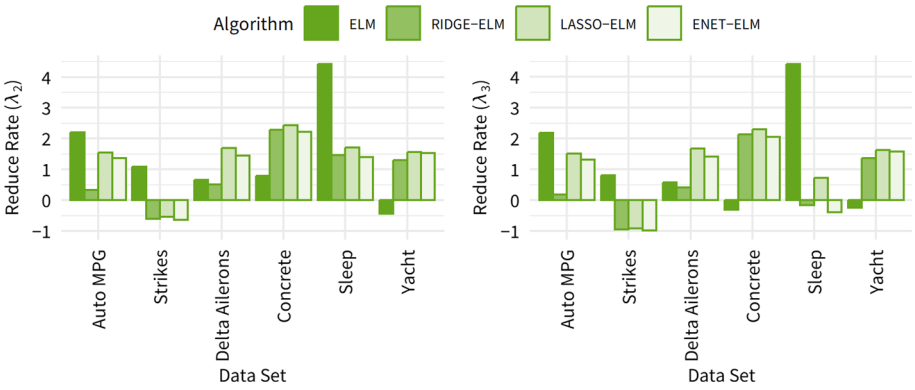


Fig. 2 Reduction rates in testing RMSE obtained by SQRTL-ELM based on universal (left) and quantile-based (right) penalties

each algorithm is given as a computational efficiency measure. All the criteria are reported in Table 2.

The results in Table 2 are summarized as follows:

- In Table 2, λ_1, λ_2 and λ_3 represent the value of the tuning parameter based on cross-validation, universal penalty technique, and quantile-based technique, respectively. Table 2 reports that the ELM yields better training RMSE values followed by SQRTL-ELM for at least one tuning parameter selection method. The SQRTL-ELM method outperforms the basic ELM and regularized ELM methods in terms of testing RMSE with at least one tuning parameter selection technique for the data sets other than the Yacht data in which the ELM has a better testing RMSE. However, SQRTL-ELM with universal and quantile-based penalties is superior to other regularized ELM algorithms for the Yacht data set. The cases where SQRTL-ELM has a better performance in terms of Testing RMSE are presented in bold in Table 2.
- Fig. 2 shows the reduction rates in log-scale gained by SQRTL-ELM based on universal and quantile-based penalties respectively, compared to other regularization methods. Considering Fig. 2 together with Table 2, the SQRTL-ELM method with universal and quantile-based penalties shows considerable reduction rates compared to the competing models.
- According to Table 2, the universal penalty outperforms the other penalty levels in Auto-MPG, Delta Ailerons, Concrete and Sleep data sets while the cross-validation-based tuning parameter is superior in the Strikes data set. While the quantile-based penalty is dominated by the other penalty levels, it outperforms at least one of the other regularized ELM methods except the Strikes data set in terms of testing RMSE.
- Considering the sparsity results in Table 2, the lasso and elastic net give more sparse models at all penalty levels of the SQRTL-ELM except Sleep data in which the cross-validation-based SQRTL-ELM outperforms the ENET-ELM in terms of sparsity and testing RMSE. When the sparsity results are considered together with the testing RMSE results, it can be stated that SQRTL-ELM makes a trade-off between model complexity and testing RMSE. In other words, SQRTL-ELM compromises from sparsity to achieve a more precise RMSE value in the testing set.
- Table 2 presents the standard deviation of the testing RMSE values to examine the stability of the SQRTL-ELM. According to Table 2, the standard deviations obtained from the

Table 2 Performance comparison of the algorithms in terms of training and testing RMSE

Data Set	Algorithm	Node	Training		Testing		RR(%) (λ_1)	RR(%) (λ_2)	RR(%) (λ_3)	λ	Timing
			RMSE	SD	RMSE	SD					
Auto	ELM	50.00	0.12226	0.00096	0.20260	0.00580	4.12	**8.05**	7.88	–	0.0065
MPG	RIDGE-ELM	50.00	0.13983	0.00104	0.18701	0.00408	−3.87	0.39	0.20	0.0011	2.5150
	LASSO-ELM	12.96	0.15087	0.00098	0.19343	0.00381	−0.42	3.70	3.51	0.0010	0.5394
	ENET-ELM	20.76	0.14698	0.00099	0.19184	0.00415	−1.25	2.90	2.71	0.0010	0.9256
	SQRTL-ELM (λ_1)	21.68	0.14740	0.00712	0.19424	0.00747	–	4.10	3.91	0.5400	3.0274
	SQRTL-ELM (λ_2)	32.92	0.13043	0.00099	**0.18628**	0.00433	−4.27	-	−0.19	0.1687	0.7956
	SQRTL-ELM (λ_3)	38.92	0.12868	0.00097	0.18664	0.00476	−4.07	0.19	–	0.1055	0.9306
Strikes	ELM	50.00	0.14282	0.00135	0.23345	0.01026	**2.89**	1.99	1.25	–	0.0043
	RIDGE-ELM	50.00	0.15473	0.00168	0.22696	0.01084	0.12	−0.81	−1.57	0.0284	0.4779
	LASSO-ELM	10.76	0.15664	0.00149	0.22716	0.01065	0.21	−0.72	−1.48	0.0019	0.2217
	ENET-ELM	17.30	0.15562	0.00145	0.22677	0.01071	0.03	−0.89	−1.66	0.0033	0.3189
	SQRTL-ELM (λ_1)	15.08	0.15415	0.00160	**0.22669**	0.01055	–	−0.93	−1.69	2.7800	2.0117
	SQRTL-ELM (λ_2)	40.18	0.14536	0.00140	0.22880	0.01054	0.92	–	−0.76	0.1337	0.4695
	SQRTL-ELM (λ_3)	43.78	0.14447	0.00136	0.23053	0.01037	1.66	0.75	–	0.0836	0.5840
Delta	ELM	50.00	0.08003	0.00083	0.16111	0.00815	−3.31	0.94	0.78	–	0.0455
airelons	RIDGE-ELM	50.00	0.08341	0.00085	0.16067	0.00857	−3.59	0.67	0.51	0.0010	15.9527
	LASSO-ELM	6.46	0.08616	0.00087	0.16708	0.00869	0.38	**4.48**	4.33	0.0010	1.4653
	ENET-ELM	10.12	0.08513	0.00085	0.16501	0.00873	−0.87	3.28	3.13	0.0010	2.6140
	SQRTL-ELM (λ_1)	11.00	0.09638	0.00504	0.16645	0.00860	–	4.12	3.96	7.6000	14.6686
	SQRTL-ELM (λ_2)	49.92	0.08132	0.00083	**0.15959**	0.00829	−4.29	–	−0.16	0.0396	11.2899
	SQRTL-ELM (λ_3)	49.98	0.08137	0.00083	0.15985	0.00837	−4.13	0.16	-	0.0248	11.4499

Table 2 continued

Data Set	Algorithm	Node	Training		Testing		RR(%) (λ_1)	RR(%) (λ_2)	RR(%) (λ_3)	λ	Timing
			RMSE	SD	RMSE	SD					
Concrete	ELM	50.00	0.19073	0.00119	0.22690	0.00261	−24.17	1.20	−0.35	–	0.0068
	RIDGE-ELM	50.00	0.22416	0.00134	0.24604	0.00234	−14.51	8.88	7.45	0.0018	3.9162
	LASSO-ELM	19.30	0.22971	0.00136	0.25008	0.00240	−12.66	10.36	8.95	0.0010	1.3068
	ENET-ELM	30.02	0.22412	0.00141	0.24430	0.00199	−15.33	8.23	6.79	0.0011	1.9808
	SQRTL-ELM (λ_1)	21.92	0.26101	0.01266	0.28175	0.01289	–	**20.43**	19.18	1.5400	3.7981
	SQRTL-ELM (λ_2)	48.46	0.19142	0.00132	**0.22419**	0.00264	−25.68	-	−1.57	0.1042	1.5494
	SQRTL-ELM (λ_3)	49.08	0.19278	0.00100	0.22771	0.00293	−23.74	1.55	–	0.0651	1.5871
Sleep	ELM	50	0.18625	0.00641	2.35350	0.19866	81.94	**82.38**	81.74	–	0.0009
	RIDGE-ELM	50	0.31667	0.00525	0.42909	0.00783	0.94	3.35	−0.18	0.0331	0.2918
	LASSO-ELM	7.66	0.32257	0.00537	0.43440	0.00867	2.15	4.53	1.05	0.0071	0.1133
	ENET-ELM	15.36	0.31397	0.00465	0.42785	0.00856	0.65	3.07	−0.47	0.0078	0.1649
	SQRTL-ELM (λ_1)	8.54	0.32046	0.00431	0.42506	0.00733	–	2.43	−1.13	0.6412	0.2724
	SQRTL-ELM (λ_2)	9.3	0.31177	0.00293	**0.41474**	0.00851	−2.49	–	−3.65	0.4217	0.0085
	SQRTL-ELM (λ_3)	10.92	0.30015	0.00339	0.42986	0.00919	1.12	3.52	–	0.2637	0.0156
Yacht	ELM	50.00	0.04687	0.00077	**0.17660**	0.00712	−27.89	−0.55	−0.27	–	0.0046
	RIDGE-ELM	50.00	0.07443	0.00067	0.18240	0.00634	−23.82	2.65	2.92	0.0010	0.9541
	LASSO-ELM	13.98	0.08493	0.00084	0.18458	0.00643	−22.36	3.80	4.07	0.0010	0.2950
	ENET-ELM	20.64	0.08150	0.00064	0.18417	0.00629	−22.63	3.59	3.85	0.0010	0.4282
	SQRTL-ELM (λ_1)	23.02	0.12705	0.02071	0.22585	0.01766	–	21.38	21.60	0.4800	1.8806
	SQRTL-ELM (λ_2)	34.18	0.06219	0.00070	0.17756	0.00671	−27.19	–	0.27	0.1903	0.4449
	SQRTL-ELM (λ_3)	40.04	0.05961	0.00085	0.17707	0.00678	−27.54	−0.27	–	0.1190	0.5044

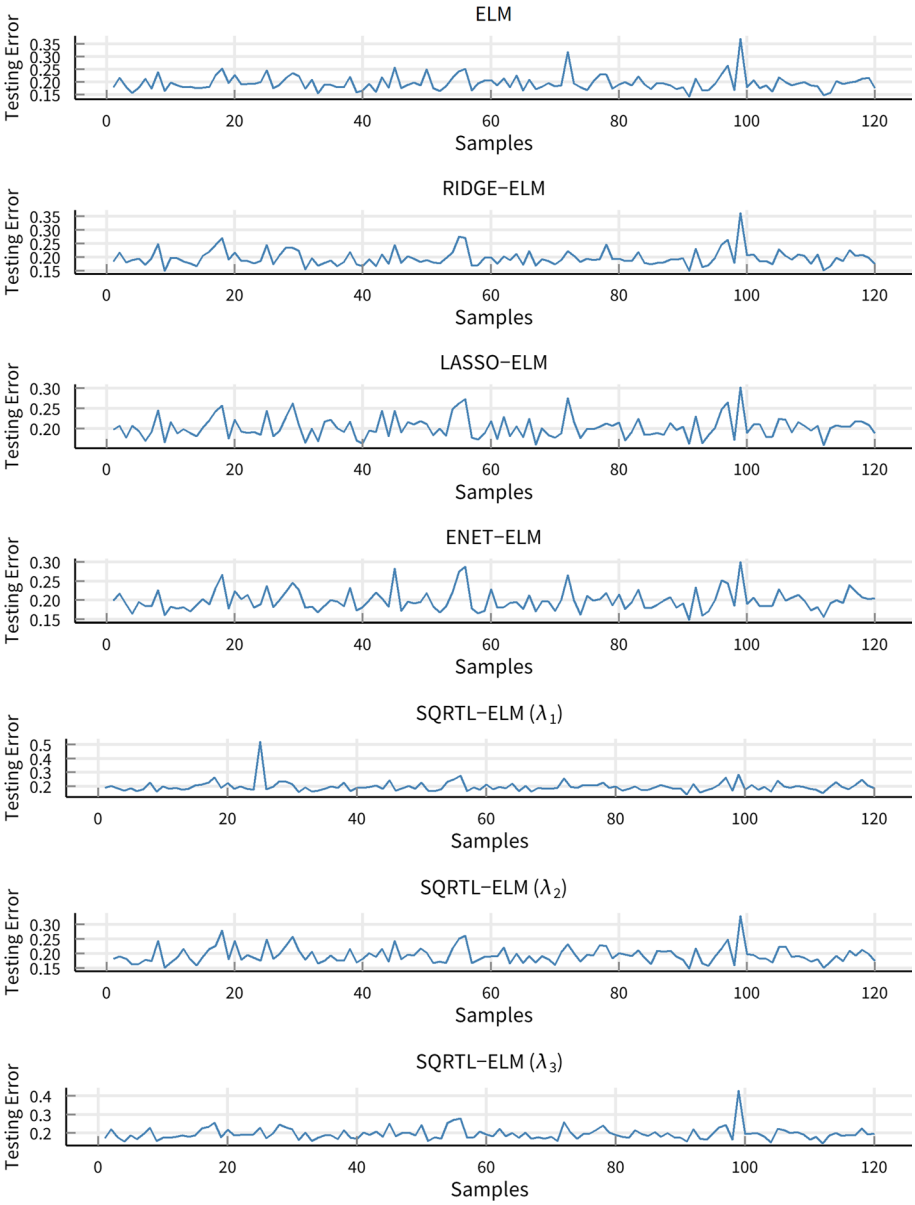


Fig. 3 Testing errors of ELM, RIDGE-ELM, LASSO-ELM, ENET-ELM and SQRTL-ELM ($\lambda_1, \lambda_2, \lambda_3$) algorithms for Auto MPG data set

SQRTL-ELM are larger than that of the other regularized ELM methods by a small margin in general. Also, the standard deviations of the testing RMSE values obtained from universal and quantile-based penalties for SQRTL-ELM are better than those of the cross-validation-based penalty level.

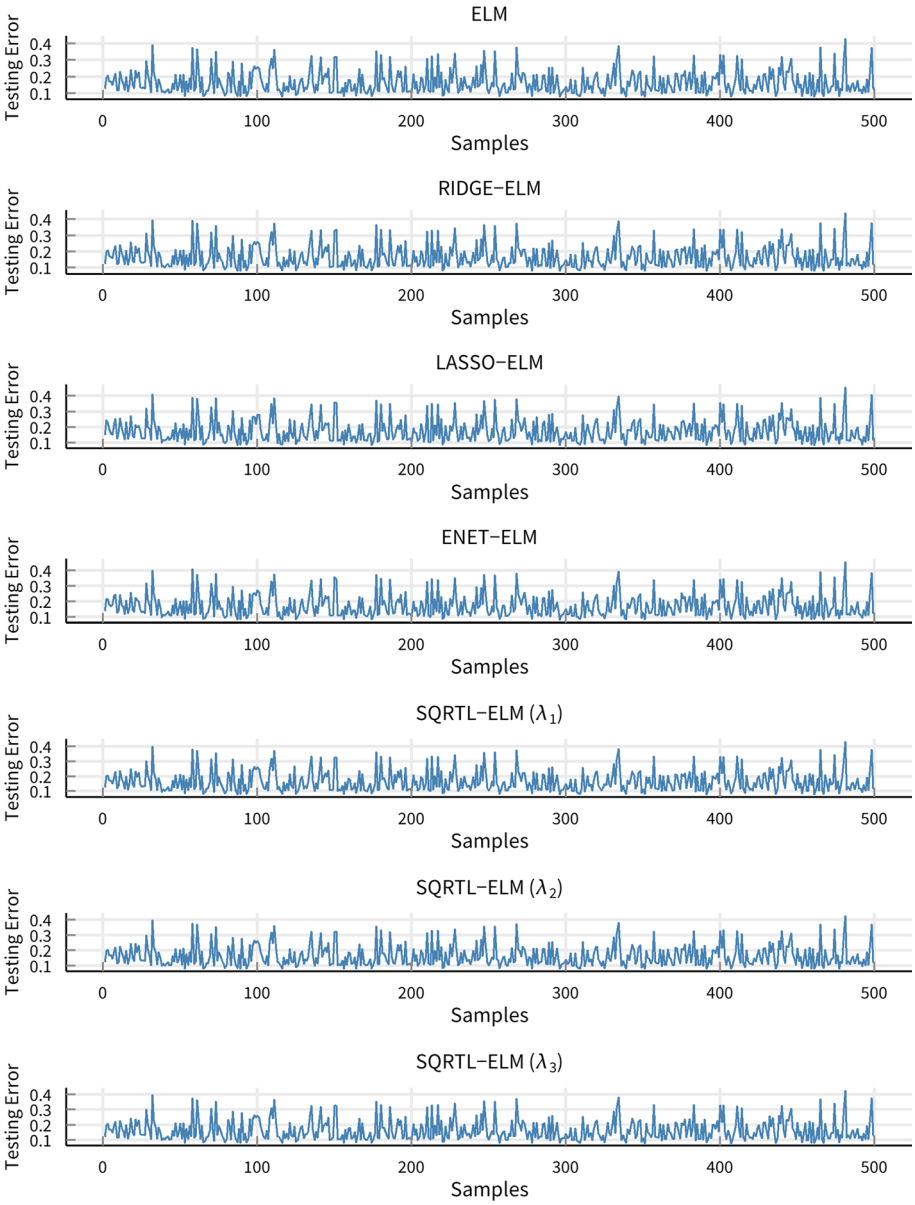


Fig. 4 Testing errors of ELM, RIDGE-ELM, LASSO-ELM, ENET-ELM and SQRTL-ELM ($\lambda_1, \lambda_2, \lambda_3$) algorithms for Delta Ailerons data set

- The training time of the basic ELM is lower than the regularized ELM algorithms for all the data sets. The SQRTL-ELM method with universal or quantile-based tuning has a lower training time than RIDGE-ELM. Among the models built based on cross-validation, LASSO-ELM and ENET-ELM methods, which give smaller node sizes than SQRTL-ELM, have lower training times than other algorithms. Also, The SQRTL-

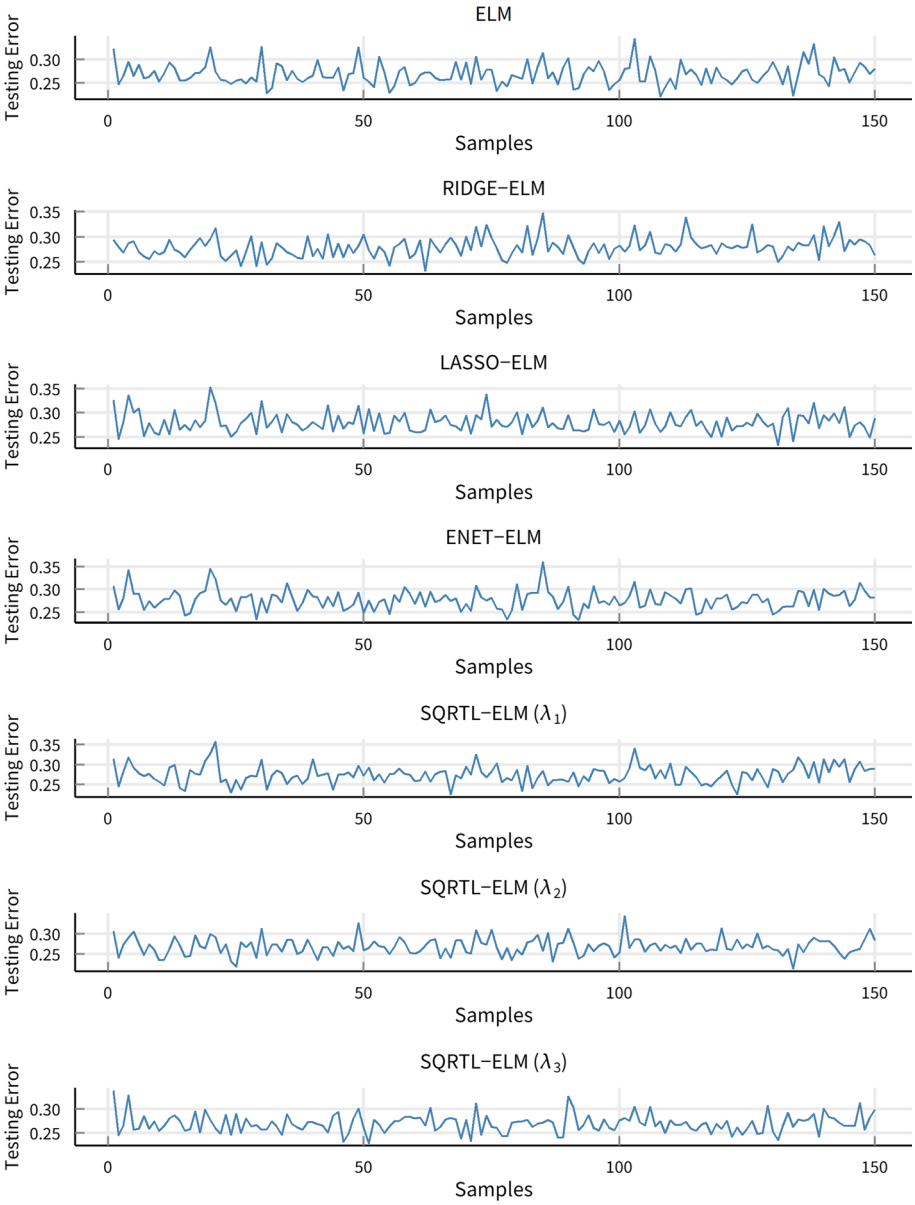


Fig. 5 Testing errors of ELM, RIDGE-ELM, LASSO-ELM, ENET-ELM and SQRTL-ELM ($\lambda_1, \lambda_2, \lambda_3$) algorithms for Concrete data set

ELM with universal penalty yields a lower training time than the cross-validation and quantile-based penalties.

Overall, the SQRTL-ELM outperforms the basic ELM and regularized ELM in terms of testing RMSE in general. However, it is slightly poorer than the other methods in terms of standard errors. Moreover, although the training time of SQRTL-ELM is higher than

that of the basic ELM, the training times of the universal and quantile-based SQRTL-ELM algorithms are comparable to those of other regularization methods.

Figures 3, 4 and 5 show the change of errors on the test sets of all the algorithms for Auto MPG, Delta Ailerons and Concrete data sets, respectively. The errors are obtained from the fitted models at the optimal parameter values. Figures 3, 4 and 5 show the stability performance of the algorithms, the smaller spread in testing RMSE will be more homogeneous. When the range of errors is analyzed in Figs. 3 and 4, it is observed that the algorithms show similar stability for the Delta Ailerons data set and better performance for at least one of the penalty levels of SQRTL-ELM for the auto MPG and Concrete data sets.

To check the performance of the algorithms statistically, the non-parametric Friedman test is conducted with the Nemenyi post hoc tests [10] based on testing RMSE of the data sets. Based on Friedman test results, the differences between RMSE scores of fifty trials are found to be statistically significant ($p < 0.05$ for the data sets). Also, Nemenyi (z) test statistics with corresponding $p-$values are reported in Table 3.[1] For the cases where SQRTL-ELM performs better, the $p-$values that indicate significant differences at the 0.05 level in testing RMSE are shown in bold.

Considering the results in Table 3, the following interpretations for SQRTL-ELM with cross-validation, universal and quantile-based penalties can be presented:

- SQRTL-ELM is statistically superior to ELM for Auto MPG, Strikes and Sleep data sets.
- SQRTL-ELM is statistically superior to RIDGE-ELM for Concrete and Yacht data sets.
- SQRTL-ELM is statistically superior to LASSO-ELM and ENET-ELM for Auto MPG, Delta Ailerons, Concrete and Yacht data sets.

As a result, SQRTL-ELM with cross-validation, universal and quantile-based penalties may overcome ELM, RIDGE-ELM, LASSO-ELM and ENET-ELM statistically, in terms of testing performance for at least one data set. Considering Tables 2 and 3 together, it can be said that the SQRTL-ELM algorithm for at least one tuning parameter selection method provides generally better results for the mean of testing RMSE than the compared regularized ELM methods RIDGE-ELM, LASSO-ELM and ENET-ELM.

5 Conclusions

In this paper, we propose a novel regularized ELM method called SQRT-ELM to improve the basic ELM and its regularized extensions. The proposed method uses the pivotal recovery property of square-root lasso to handle the drawbacks of basic ELM like instability, poor generalizability and overfitting problems. The experimental studies on well-known benchmark data sets show that the SQRTL-ELM generally improves the testing RMSE performance of basic ELM and outperforms its variants including RIDGE-ELM, LASSO-ELM and ENET-ELM depending on the data set for the sake of slightly larger standard deviations of the testing RMSE. The amount of improvement in the test RMSE and sparsity level varies according to the applied tuning parameter technique. The training time of SQRTL-ELM is higher than that of the basic ELM and competes with the training time of regularized ELM methods depending on the underlying data set. Also, SQRTL-ELM with universal and quantile-based penalty outperforms SQRTL-ELM with cross-validation in terms of the lower training time.

[1] For the sake of simplicity and keeping the study shorter, only the main results of the whole statistical process are presented.

Table 3 Post hoc results on testing RMSE for each data set

Data Set	Algorithm	Statistic	ELM	RIDGE-ELM	LASSO-ELM	ENET-ELM	SQRTL-ELM (λ_1)	SQRTL-ELM (λ_2)
Auto MPG	SQRT-ELM(λ_1)	z	5.8264	1.3093	5.6955	3.8625	–	–
		p	**0.0007**	0.9685	0.0011	0.0906	–	–
	SQRT-ELM(λ_2)	z	7.1357	0.0001	7.0048	5.1718	1.3093	–
		p	**<0.0001**	0.9999	**<0.0001**	**0.0048**	0.9685	–
	SQRT-ELM(λ_3)	z	8.0522	0.9165	7.9213	6.0883	2.2258	0.9165
		p	**<0.0001**	0.9952	**<0.0001**	**0.0003**	0.6992	0.9952
Strikes	SQRT-ELM(λ_1)	z	7.2012	0.3928	0.7201	0.3928	–	–
		p	**<0.0001**	0.9999	0.9987	0.9999	–	–
	SQRT-ELM(λ_2)	z	5.7610	1.0474	0.7201	1.8330	1.4402	–
		p	**0.0009**	0.9900	0.9987	0.8540	0.9500	–
	SQRT-ELM(λ_3)	z	2.8150	3.9934	3.6661	4.7790	4.3862	2.9459
		p	0.4206	0.0709	0.1282	0.0129	**0.0317**	0.3627
Delta ailerons	SQRT-ELM(λ_1)	z	1.8985	2.5531	2.8805	1.8330	–	–
		p	0.8319	0.5444	0.3912	0.8540	–	–
	SQRT-ELM(λ_2)	z	2.4877	1.8330	7.2667	6.2192	4.3862	–
		p	0.5759	0.8540	**<0.0001**	**0.0002**	**0.0317**	–
	SQRT-ELM(λ_3)	z	2.2258	1.5712	7.0048	5.9573	4.1243	0.2619
		p	0.6992	0.9250	**<0.0001**	**0.0005**	0.0548	0.9999

Table 3 continued

Data Set	Algorithm	Statistic	ELM	RIDGE-ELM	LASSO-ELM	ENET-ELM	SQRTL-ELM (λ_1)	SQRTL-ELM (λ_2)
Concrete	SQRT-ELM(λ_1)	z	7.7904	0.0655	2.2913	0.4583	–	–
		p	<0.0001	0.9999	0.6692	0.9999	–	–
	SQRT-ELM(λ_2)	z	0.5892	8.3141	10.6709	7.9213	8.3796	–
		p	0.9996	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**	–
	SQRT-ELM(λ_3)	z	0.1964	7.5285	9.8853	7.1357	7.5940	0.7856
		p	0.9999	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**	0.9979
Sleep	SQRT-ELM(λ_1)	z	11.8492	0.5892	1.8330	0.3273	–	–
		p	**<0.0001**	0.9996	0.8540	0.9999	–	–
	SQRT-ELM(λ_2)	z	13.6823	2.4222	3.6661	2.1604	1.8330	–
		p	**<0.0001**	0.6074	0.1282	0.7282	0.8540	–
	SQRT-ELM(λ_3)	z	10.4090	0.8510	0.3928	1.1129	1.4402	3.2733
		p	**<0.0001**	0.9968	0.9999	0.9863	0.9500	0.2368
Yacht	SQRT-ELM(λ_1)	z	3.5351	0.9165	4.1898	2.8805	–	–
		p	0.1592	0.9952	0.0480	0.3912	–	–
	SQRT-ELM(λ_2)	z	0.0655	4.5171	7.7904	6.4811	3.6006	–
		p	0.9999	**0.0237**	**<0.0001**	**0.0001**	0.1431	–
	SQRT-ELM(λ_3)	z	0.0655	4.5171	7.7904	6.4811	3.6006	**<0.0001**
		p	0.9999	**0.0237**	**<0.0001**	**0.0001**	0.1431	0.9999

Consequently, as a novel regularized ELM algorithm, SQRTL-ELM shows applicability and usefulness in regression tasks in data-driven studies.

Declarations

Conflict of interest The author declares that he has no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Allison T, Cicchetti DV (1976) Sleep in mammals: ecological and constitutional correlates. Science 194(4266):732–734
2. Balasundaram S, Gupta D (2016) Knowledge-based extreme learning machines. Neural Comput Appl 27:1629–1641
3. Balasundaram S, Gupta D (2016) On optimization based extreme learning machine in primal for regression and classification by functional iterative method. Int J Mach Learn Cybern 7:707–728
4. Belloni A, Chernozhukov V, Wang L (2011) Square-root lasso: pivotal recovery of sparse signals via conic programming. Biometrika 98(4):791–806
5. Borah P, Gupta D (2020) Unconstrained convex minimization based implicit lagrangian twin extreme learning machine for classification (ultelmc). Appl Intell 50(4):1327–1344
6. Cawley GC, Talbot NLC (2010) On over-fitting in model selection and subsequent selection bias in performance evaluation. J Mach Learn Res 11(70):2079–2107
7. Cui D, Huang G-B, Liu T (2018) Elm based smile detection using distance vector. Pattern Recogn 79:356–369
8. Dalalyan AS, Hebiri M, Lederer J (2017) On the prediction performance of the lasso. Bernoulli 23(1):552–581
9. de Campos Souza PV, Bambirra Torres LC, Lacerda Silva GR, Braga ADP, Lughofer E (2020) An advanced pruning method in the architecture of extreme learning machines using l1-regularization and bootstrapping. Electronics 9(5):811
10. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
11. Deng W, Zheng Q, Chen L (2009) Regularized extreme learning machine. In *2009 IEEE symposium on computational intelligence and data mining*, pp 389–395. IEEE
12. Ding S, Xu X, Nie R (2014) Extreme learning machine and its applications. Neural Comput Appl 25(3):549–556
13. Donoho DL, Johnstone IM (1994) Minimax risk overl p-balls forl p-error. Probab Theory Relat Fields 99(2):277–303
14. Dua D, Graff C (2019) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. Accessed on January 17, 2022
15. Efron B (2004) The estimation of prediction error: covariance penalties and cross-validation. J Am Stat Assoc 99(467):619–632
16. Fakhr M.W, Youssef E-N.S, El-Mahallawy M.S (2015) L1-regularized least squares sparse extreme learning machine for classification. In *2015 international conference on Information and Communication Technology Research (ICTRC)*, pp 222–225. IEEE
17. Feng G, Huang G-B, Lin Q, Gay R (2009) Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Trans Neural Netw 20(8):1352–1357
18. Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. J Stat Softw 33(1):1
19. Fushiki T (2011) Estimation of prediction error by using k-fold cross-validation. Stat Comput 21(2):137–146

20. Genç M, Özkale M.R (2022) Lasso regression under stochastic restrictions in linear regression: An application to genomic data. *Communications in Statistics-Theory and Methods*, pp 1–24
21. Gupta D, Hazarika BB, Berlin M (2020) Robust regularized extreme learning machine with asymmetric huber loss function. Neural Comput Appl 32(16):12971–12998
22. Gupta U, Gupta D (2021) Regularized based implicit Lagrangian twin extreme learning machine in primal for pattern classification. Int J Mach Learn Cybern 12(5):1311–1342
23. Hoerl AE, Kennard RW (1970) Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12(1):55–67
24. Hornik K (1991) Approximation capabilities of multilayer feedforward networks. Neural Netw 4(2):251–257
25. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366
26. Hou S, Wang Y, Jia S, Wang M, Wang X (2022) A derived least square extreme learning machine. Soft Comput 26(21):11115–11127
27. Huang G, Huang G-B, Song S, You K (2015) Trends in extreme learning machines: a review. Neural Netw 61:32–48
28. Huang G-B, Babri HA (1998) Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. IEEE Trans Neural Netw 9(1):224–229
29. Huang G-B, Bai Z, Kasun LLC, Vong CM (2015) Local receptive fields based extreme learning machine. IEEE Comput Intell Mag 10(2):18–29
30. Huang G-B, Chen L (2008) Enhanced random search based incremental extreme learning machine. Neurocomputing 71(16–18):3460–3468
31. Huang G-B, Chen L, Siew CK et al (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892
32. Huang G-B, Ding X, Zhou H (2010) Optimization method based extreme learning machine for classification. Neurocomputing 74(1–3):155–163
33. Huang G.-B, Zhu Q.-Y, Siew C-K (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, volume 2, pp 985–990. IEEE
34. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501
35. Khan MA, Arshad H, Khan WZ, Alhaisoni M, Tariq U, Hussein HS, Alshazly H, Osman L, Elashry A (2023) Hgrbol2: human gait recognition for biometric application using Bayesian optimization and extreme learning machine. Futur Gener Comput Syst 143:337–348
36. Lan Y, Hu Z, Soh YC, Huang G-B (2013) An extreme learning machine approach for speaker recognition. Neural Comput Appl 22(3):417–425
37. Larrea M, Porto A, Irigoyen E, Barragán AJ, Andújar JM (2021) Extreme learning machine ensemble model for time series forecasting boosted by PSO: application to an electric consumption problem. Neurocomputing 452:465–472
38. Leshno M, Lin VY, Pinkus A, Schocken S (1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw 6(6):861–867
39. Li G, Niu P (2013) An enhanced extreme learning machine based on ridge regression for regression. Neural Comput Appl 22(3):803–810
40. Lipu MSH, Hannan MA, Hussain A, Saad MH, Ayob A, Uddin MN (2019) Extreme learning machine model for state-of-charge estimation of lithium-ion battery using gravitational search algorithm. IEEE Trans Ind Appl 55(4):4225–4234
41. Luo X, Chang X, Ban X (2016) Regression and classification using extreme learning machine based on l1-norm and l2-norm. Neurocomputing 174:179–186
42. Martínez-Martínez JM, Escandell-Montero P, Soria-Olivas E, Martín-Guerrero JD, Magdalena-Benedito R, GóMez-Sanchis J (2011) Regularized extreme learning machine for regression problems. Neurocomputing 74(17):3716–3721
43. Miche Y, Sorjamaa A, Bas P, Simula O, Jutten C, Lendasse A (2009) Op-elm: optimally pruned extreme learning machine. IEEE Trans Neural Netw 21(1):158–162
44. Miche Y, Van Heeswijk M, Bas P, Simula O, Lendasse A (2011) Trop-elm: a double-regularized elm using lars and tikhonov regularization. Neurocomputing 74(16):2413–2421
45. Molinaro AM, Simon R, Pfeiffer RM (2005) Prediction error estimation: a comparison of resampling methods. Bioinformatics 21(15):3301–3307
46. Park C, Yoon YJ (2011) Bridge regression: adaptivity and group selection. J Stat Plan Inf 141(11):3506–3519

47. Preeti Bala R, Dagar A, Singh RP (2021) A novel online sequential extreme learning machine with l2 1-norm regularization for prediction problems. Appl Intell 51(3):1669–1689
48. Rong H-J, Ong Y-S, Tan A-H, Zhu Z (2008) A fast pruned-extreme learning machine for classification problem. Neurocomputing 72(1–3):359–366
49. Saputra D.C.E, Sunat K, Ratnaningsih T (2023) A new artificial intelligence approach using extreme learning machine as the potentially effective model to predict and analyze the diagnosis of anemia. In *Healthcare*, vol 11, pp 697. MDPI
50. Singh M, Chauhan S (2023) A hybrid-extreme learning machine based ensemble method for online dynamic security assessment of power systems. Electric Power Syst Res 214:108923
51. Sun T, Zhang C-H (2012) Scaled sparse linear regression. Biometrika 99(4):879–898
52. Sun T, Zhang C-H (2013) Sparse matrix inversion with scaled lasso. J Mach Learn Res 14(1):3385–3418
53. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J Royal Stat Soc Series B (Methodological) 58:267–288
54. Tibshirani R, Bien J, Friedman J, Hastie T, Simon N, Taylor J, Tibshirani RJ (2012) Strong rules for discarding predictors in lasso-type problems. J Royal Stat Soc: Series B (Statistical Methodology) 74(2):245–266
55. Tong R, Li P, Lang X, Liang J, Cao M (2021) A novel adaptive weighted kernel extreme learning machine algorithm and its application in wind turbine blade icing fault detection. Measurement 185:110009
56. Vidal C, Malysz P, Kollmeyer P, Emadi A (2020) Machine learning applied to electrified vehicle battery state of charge and state of health estimation: State-of-the-art. IEEE Access 8:52796–52814
57. Wang K, Pei H, Cao J, Zhong P (2020) Robust regularized extreme learning machine for regression with non-convex loss function via dc program. J Franklin Inst 357(11):7069–7091
58. Wang X, Sun Q, Kou X, Ma W, Zhang H, Liu R (2022) Noise immune state of charge estimation of li-ion battery via the extreme learning machine with mixture generalized maximum correntropy criterion. Energy 239:122406
59. Yang L, Tsang EC, Wang X, Zhang C (2023) Elm parameter estimation in view of maximum likelihood. Neurocomputing 557:126704
60. Yang Y, Zhou H, Wu J, Ding Z, Wang Y-G (2022) Robustified extreme learning machine regression with applications in outlier-blended wind-speed forecasting. Appl Soft Comput 122:108814
61. Yaseen ZM, Sulaiman SO, Deo RC, Chau K-W (2019) An enhanced extreme learning machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction. J Hydrol 569:387–408
62. Yildirim H, Özkale MR (2019) The performance of elm based ridge regression via the regularization parameters. Expert Syst Appl 134:225–233
63. Yıldırım H, Özkale MR (2020) An enhanced extreme learning machine based on liu regression. Neural Process Lett 52:421–442
64. Yıldırım H, Özkale MR (2021) Ll-elm: a regularized extreme learning machine based on l1-norm and liu estimator. Neural Comput Appl 33(16):10469–10484
65. Yıldırım H, Özkale MR (2023) A combination of ridge and liu regressions for extreme learning machine. Soft Comput 27(5):2493–2508
66. Zhang Y, Dai Y, Wu Q (2022) An accelerated optimization algorithm for the elastic-net extreme learning machine. Int J Mach Learn Cybern 13(12):3993–4011
67. Zhang Y, Dai Y, Wu Q (2023) A novel regularization paradigm for the extreme learning machine. Neural Process Lett 55:7009–7033
68. Zhao Y, Wang K (2014) Fast cross validation for regularized extreme learning machine. J Syst Eng Electron 25(5):895–900
69. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J Royal Stat Soc: Series B (Statistical Methodology) 67(2):301–320

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.