Ship roll motion prediction based on $\ell_1$ regularized extreme learning machine

Binglei Guan$^{1,2}$, Wei Yang$^3$, Zhibin Wang$^4$, Yinggan Tang$^4*$

1 Logistics Engineering College, Shanghai Maritime University, Shanghai, 200135, China, 2 School of Electronic & Information Engineering, Ningbo University of Technology, Ningbo, Zhejiang, 315016, China, 3 State GRID Quzhou Power Supply Company, No. 6, XinHe road, Quzhou, 253000, Zhejiang, China, 4 Institute of Electrical Engineering, Yanshan University, Qinhuangdao, Hebei, 066004, China

* ygtang@ysu.edu.cn

Abstract

In this paper, a new method is proposed for prediction of ship roll motion based on extreme learning machine (ELM). To improve the prediction accuracy and avoid over or under fitting, two techniques are adopted to select the appropriate structure of ELM. First, the inputs of the ELM are selected from the roll motion time series using Lipschitz quotient method. Second, the number of hidden layer nodes is determined via $\ell_1$ regularized technique. Finally, the $\ell_1$ regularized ELM is solved by least angle regression (LAR) algorithm. The effectiveness of the proposed method is demonstrated by ship roll motion prediction experiments based on the real measured ship roll motion time series.

Introduction

Roll motion is one of the important motion modes for ship navigating in sea, which is caused by external environmental factors such as strong wind, waves and currents. Ship’s roll motion is undesirable especially under the condition of rough sea because it is harmful for ship’s stability, affects the safety of crew and cargos and gives rise to working inefficiency of seafarers. Therefore, ship roll motion prediction is very necessary because prediction information can give operator sufficient time to avoid serious events. However, ship roll motion prediction is a difficult problem because the dynamics of ship’s roll motion is a complex nonlinear system with time varying characteristics [1]. Moreover, roll motion is also coupled with and affected by other motion modes such as heave and pitch. So, it is hard to establish a precise model to predict ship’s roll motion.

Many researchers had put attention to ship roll motion prediction and put forward many prediction methods. Most of these prediction methods are based on time series analysis. In [2], a minor component analysis (MCA) was proposed to predict ship motion. Since the ship motion is a nonlinear processes, the nonlinear time series analysis method, neural network (NN) [3, 4] for example, is suitable to establish a prediction model. Yin et al. established radial basis function (RBF) neural networks to predict ship roll motion, where the structure and parameters of RBF network were adjusted online via sequential learning algorithm [5]. Since the ship motion is complex and time varying, wavelet analysis is a time-frequency signal...
analysis method, which can capture the changes of signals. Observing this, a ship roll motion prediction method based on wavelet network was proposed in [6], the wavelet network was adjusted through a coarse to fine process. While in [7], the ship roll motion time series data is first decomposed into different subbands, and then the subbands were used as inputs to train a variable RBF network, which is finally used as predictor of ship roll motion.

Extreme learning machine (ELM) is a new algorithm to train single hidden layer feed-forward neural networks (SLFNs) proposed by Huang in [8]. ELM transforms the training of SLFNs to a standard least square problem by randomly choosing the input weights and hidden bias, and thus is more efficient than conventional training algorithms in terms of training speed and computation efficiency. In [9], a ship roll motion predictor based on ELM was proposed. In [10], an improved OS-ELM was proposed to predict ship roll motion, where the number of nodes in hidden layer is determined using Akaike information criterion (AIC). In [11], a new ship roll motion prediction method was proposed by combining grey theory and OS-ELM, where the original time series was firstly processed to obtain a new accumulated time series by using the accumulated generation operation (AGO) in grey theory, and then, the mapping relationship between the accumulated time series and its prediction was built using OS-ELM. The actual prediction is obtained by the inverse accumulated generation operation (IAGO) performed on the predicted accumulated time series through OS-ELM.

For time series prediction based on neural network, determining the input variables is an important problem because it greatly affects the prediction accuracy. Moreover, determining the number of hidden nodes for neural networks is also an important problem. If large number of nodes are selected, it is possible to occur over-fitting. On the contrary, if small number of nodes are selected, under-fitting may occurs. Actually, for neural network based time series prediction, the above two problems are structure selection of neural networks. In the literatures related to ship roll motion prediction, the above two problems are less addressed simultaneously. In [5, 11, 12], sequential learning algorithm was adopted to obtain a variable structure RBF neural network for ship roll motion prediction, in which the number of hidden nodes is determined real time. However, the input variable selection was not considered.

To overcome the above drawbacks, in this paper, a new prediction method for ship roll motion based on $\ell_1$-regularized ELM is proposed. In the proposed method, ELM is used as prediction model. The main contributions of this paper include two aspects. First, the predictors, namely the input variables and its number of prediction model, are determined from the view of function continuity, which is characterized by Lipschitz quotients. The proposed approach is different from phase space reconstruction method, where a fixed embedded dimension and time lag are assumed. Second, a $\ell_1$-regularized technique is utilized to select the node number of hidden layer of ELM. The training and structure determination of ELM are fulfilled simultaneously.

The rest of this paper is organized as follows. The principle of Lipschitz quotients is briefly reviewed in section 1. The $\ell_1$ regularized ELM is introduced in section 2. Section 1 presents the roll motion prediction process using $\ell_1$ regularized ELM. Finally, the simulated prediction results are presented in section.

### Lipschitz quotients

Originally, Lipschitz quotients is a ratio of two distances in Euclidean space. In [13], He and Asada adopted it to identify the orders of nonlinear dynamic system. More specifically, Lipschitz quotient was used as a measure if a variable is missed in a nonlinear function or is redundantly added in the function based on the continuity of the nonlinear function. In this paper, it is utilized to determine the appropriated inputs for time series prediction.
Considering a nonlinear function as

\[ y = f(x_1, x_2, \ldots, x_n), \]

where \( n \) is the number of input variables. For the sake of convenience, denote \( x = [x_1, x_2, \ldots, x_n]^T \). Here, we pay our attention to the selection of input variables in reconstruction of function \( f \) from input-output pairs \((x(i), y(i)), (i = 1, 2, \ldots, N)\). If function \( f \) is continuous, its Lipschitz quotient \( q_{ij}^{(n)} \), which is defined as

\[ q_{ij}^{(n)} = \frac{|y(i) - y(j)|}{|x(i) - x(j)|}, i \neq j, \]

is bounded. In Eq (2), \(|a - b|\) represents the Euclidean distance between two points \( a \) and \( b \). For function \( f \) in (1) with \( n \) input variables, its Lipschitz quotient \( q_{ij}^{(n)} \) can be calculated by extending Eq (2) as

\[ q_{ij}^{(n)} = \frac{|y(i) - y(j)|}{\sqrt{(x_1(i) - x_1(j))^2 + \cdots + (x_n(i) - x_n(j))^2}}. \]

where the superscript \( n \) in \( q_{ij}^{(n)} \) denotes the correct number of input variables in (1). He and Asada revealed in [13] that if an input variable, \( x_n \) for example, is missed in reconstruction of \( f \), its Lipschitz quotient \( q_{ij}^{(n-1)} \) will be extreme large or is larger than \( q_{ij}^{(n)} \), which relies on \( x_n \) is independent of other variables \( x_1, x_2, \cdots, x_{n-1} \) or not. On the other hand, if two or more redundant input variables are included in the reconstruction of \( f \), its Lipschitz quotient \( q_{ij}^{(n+1)} \) will be very close to \( q_{ij}^{(n)} \) [13]. From the above findings, Lipschitz quotient can be used to select or determine the optimal number of input variables for reconstruction of \( f \). In practice, there may be noise in input or output variables, the Lipschitz quotient maybe incorrect. To avoid the impact of measurement noise, a modified index as

\[ q^{(n)} = \left( \prod_{i=1}^{p} \sqrt{nq^{(n)}(i)} \right)^{1/p}, \]

are suggested in [13] for variable selection or order identification. In (4), \( q^{(n)}(i) \) is the \( i \)-th largest Lipschitz quotient among all \( q^{(n)} \) and \( p \) is a positive number usually selected to be \( p \in [0.01N, 0.02N] \). In practical application, a stop criterion defined as

\[ \frac{|q^{(n+1)} - q^{(n)}|}{\max(1, |q^{(n)}|)} < \varepsilon, \]

is used to terminate the algorithm, where the threshold \( \varepsilon = 0.1 \) is suggested in [13].

\section*{\( \ell_1 \)-regularized extreme learning machine (ELM)}

\textbf{Basics of ELM}

ELM was proposed by Prof. Huang in [8] to train a single hidden layer feedforward neural network (SFLN), aiming at simplifying and speeding up the training process of SFLN. Different from traditional training methods, the input weights and bias in ELM are randomly initialized, and only the output weights are calculated analytically using Moore–Penrose (M–P) generalized inverse.
The typical architecture of a SLFN is shown in Fig 1 [14], in which there are \( n \) input nodes, \( m \) output nodes and \( L \) hidden nodes. Denote \( \mathcal{D} = \{(x_i, y_i) | x_i \in \mathbb{R}^n, y_i \in \mathbb{R}^m, i = 1, 2, \ldots, N\} \) be a set of training sample. The output of SLFN is calculated as

\[
\sum_{i=1}^{L} \beta_i g(w_i \cdot x_i + b_i) = \hat{y}_i, \quad i = 1, 2, \ldots, N, \tag{6}
\]

where \( w_i = [w_{i1}, w_{i2}, \ldots, w_{in}] \) is called input weight vector, which connects the input nodes and \( \ell \)th hidden node; \( \beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T \) is hidden layer weight vector building link between the \( \ell \)th hidden neuron and the output nodes, \( b_i \in \mathbb{R} \) is the bias of the \( \ell \)th hidden neuron, \( w_i \cdot x_i \) denotes the inner product between vector \( w_i \) and \( x_i \), \( g(\cdot) \) is the activation function of hidden layer nodes. In ideal case, one expects the output of SLFN \( \hat{y}_i \) are perfectly equal to the actual output \( y_i \), i.e.,

\[
\sum_{i=1}^{L} \beta_i g(w_i \cdot x_i + b_i) = y_i, \quad i = 1, 2, \ldots, N. \tag{7}
\]

Writing Eq (7) in matrix form, one can get

\[
H\beta = Y, \tag{8}
\]

where

\[
H = \begin{bmatrix}
g(w_1 x_1 + b_1) & \cdots & g(w_L x_1 + b_L) \\
\vdots & \ddots & \vdots \\
g(w_1 x_N + b_1) & \cdots & g(w_L x_N + b_L)
\end{bmatrix}_{N \times L}. \tag{9}
\]
\[
\beta = \begin{bmatrix}
\beta_1^T \\
\vdots \\
\beta_L^T
\end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix}
\tilde{y}_1^T \\
\vdots \\
\tilde{y}_N^T
\end{bmatrix}.
\]

\(H\) is called the hidden layer output matrix. In Eq (8), the output weights \(\beta\) is unknown. Prof. Huang [8] proposed to calculate \(\beta\) using the following M-P pseudo inverse,

\[
\beta = H^\dagger Y,
\]

where \(H^\dagger\) is the Moore-Penrose (M-P) generalized pseudo-inverse of the hidden layer output matrix. The solution presented in Eq (11) means that the smallest error can be obtained.

**\(\ell_1\)-regularized ELM**

In practice, determining the number of hidden layer nodes for SLFN is an important problem. If the number of hidden layer nodes is selected too large, over-fitting occurs, and vice versa, if the number of hidden layer nodes is selected too small, under-fitting may occur. In the literatures, some pruning technologies had been proposed to select the appropriate number of hidden layer nodes [15, 16]. Although it has advances in training speed and accuracy, ELM itself can not automatically determine the appropriate number of hidden layer nodes.

It can be seen that the solution of Eq (11) is a least square solution of the following minimization problem,

\[
\hat{\beta} = \arg\min_{\beta} \| Y - H\beta \|_2^2.
\]

The least square estimation of output weight \(\beta\) has smaller variance for training set, however, it has large variance for test set. That is to say, the generalization of ELM is not so satisfactory. On the other hand, least square hasn’t the ability of variable selecting. In ELM, determining the number of hidden layer nodes can be viewed as a problem of variable selecting. Thus, we can add a \(\ell_1\) penalty term on the output weight as following,

\[
\hat{\beta} = \arg\min_{\beta} \| Y - H\beta \|_2^2 + \lambda \| \beta \|_1,
\]

where the first term \(\| Y - H\beta \|_2^2\) forces the output of ELM is as close as possible to the actual output, the second term \(\lambda \| \beta \|_1\) is a \(\ell_1\)-regularized term of output weight \(\beta\) and \(\lambda\) is the regularization parameter, which is used to perform a tradeoff between the approximation error and the sparseness of the weights. Solving the \(\ell_1\) regularized problem (13) leads to a sparse solution, i.e., the output weight vector \(\beta\) is sparse. It means that most of elements of \(\beta\) are zero or near zero. Therefore, the link between the hidden node and the output is disconnected. The corresponding hidden node can be removed, and thus, the purpose of selecting of the number of hidden nodes is achieved.

There are many methods to solve minimization problem (13), for example, coordinate descent method and gradient descent method. In deed, the problem is also a LASSO (Least absolute shrinkage and selection operator) problem, it can be solved using least angle regression (LAR) algorithm. In this paper, the LAR method is adopted. For more details about LAR, one can refer to [17].

The regularization parameter \(\lambda\) affects the approximation error and the complexity of the model. In most algorithms, the value of \(\lambda\) decrease from \(\lambda_{\text{max}}\) from \(\lambda_{\text{min}}\) in a log manner to form a sequence with \(K\) elements. Each regularization parameter \(\lambda(k)\) corresponds to a
solution path or a model, in other words. In order to select the best model, several criteria can be adopted. The commonly used criteria includes adjusted $R^2$, Akaike information criterion (AIC) and Bayesian information criterion (BIC) [18]. In this paper, BIC criteria is used to select the best model. The BIC for $k$-th model is defined as

$$BIC(k) = \|Y - H\beta(k)\|_2^2 + \log(N)M(k)\sigma$$  \hspace{1cm} (14)

where $N$ is the number of samples, $M(k)$ is the number of nonzero elements of $\beta(k)$ and $\sigma$ is the residual variance of a low-bias model defined as

$$\sigma = \frac{1}{N}\|Y - HH^T\beta(k)\|_2^2$$  \hspace{1cm} (15)

where $H^T$ is the Moore–Penrose pseudo-inverse of $H$.

**Ship roll motion prediction based on $\ell_1$ regularized ELM**

The process of using $\ell_1$ regularized ELM for ship roll motion prediction can be summarized as follows.

**Step 1** Input the ship roll motion time series $\{y(k), k = 1, 2, \cdots, n\}$;

**Step 2** Calculating Lipschitz quotient $q^{(l)}$ of time series $\{y(k)\}$. From $q^{(l)}, l = 1, 2, \cdots$, determine the input variables.

**Step 3** Constructing training set. The input training set is constructed as

$$X = \begin{bmatrix} y(1) & y(2) & \cdots & y(m) \\ y(2) & y(3) & \cdots & y(m+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(n-m) & y(n-m+1) & \cdots & y(n-1) \end{bmatrix},$$

and the output is as

$$Y = \begin{bmatrix} y(m+1) \\ y(m+2) \\ \vdots \\ y(n) \end{bmatrix}$$

where $m$ is the number of input variable determined in Step 2. In $X$, each row is a training sample.

**Step 4** Training $\ell_1$ regularized ELM. Set the hidden layer nodes of ELM to be a large number, and solve the optimization problem (13) using LAR algorithm to obtain $K$ models.

**Step 5** Select the best model using BIC criteria in (14). Let the best model be $\beta^*$.

**Step 6** Prediction using the best model. Let $\mathcal{F}(\cdot)$ represent the map relationship of the best model. One step ahead prediction is achieved by

$$\hat{y}(k+1) = \mathcal{F}(y(k), y(k-1), \cdots, y(k-m-1)),$$  \hspace{1cm} (16)
and the second step prediction is achieved as
\[ \hat{y}(k + 2) = \mathcal{F}(\hat{y}(k + 1), y(k), \ldots, y(k - m - 2)). \]  
(17)

\( p \)-step ahead prediction can be obtained in a similar way as Eqs (16) and (17).

### Simulation studies

To validate the effectiveness of the proposed ship roll motion prediction method, simulation studies are conducted. All the algorithms in this paper are implemented using MATLAB 2016b programming language and executed on Thinkpad T440, a laptop computer with Intel® Core™ i5-4200U processor, 8.0G random access memory (RAM). The roll motion data is measured from Yu Kun, a scientific research and training ship. The sea trial condition and characteristic of Yu Kun, one can refer to [11]. The measured roll angles are shown in Fig 2.

### Determining the structure of ELM

In this paper, the maximum number of hidden nodes of the ELM is set to 500, i.e., \( L = 500 \). The activation function is sigmoidal function. The training algorithm is LAR algorithm.

The Lipschitz quotient method described in section 1 is used to select the input variables of ELM. The Lipschitz quotient \( q(l) \) of measured roll angles is shown in Fig 3. In Fig 3, \( q^{(1)} = \text{NaN} \), \( q^{(2)} = \infty \), \( q^{(3)} = 36.82 \), \( q^{(4)} = 22.31 \), \( q^{(5)} = 12.72 \), \( q^{(6)} = 9.48 \), \( q^{(7)} = 9.21 \). Since, \( q^{(6)} \) is smaller than \( q^{(5)} \) and \( q^{(6)} \) is very close to \( q^{(7)} \), according to the principle of Lipschitz quotient, the input variables are six, i.e., the input of ELM is \( \{y(k - 1), y(k - 2), \ldots, y(k - 6)\} \), the output of ELM is \( y(k) \). Therefore, the ELM has initially 6 inputs and 500 hidden nodes and one output. The LAR algorithm is used to training \( \ell_1 \) regularized ELM. Fig 4 shows the solution path of LAR.

![Fig 2. Measured roll angles.](https://doi.org/10.1371/journal.pone.0206476.g002)
Fig 3. Lipschitz quotients $q^l$.  
https://doi.org/10.1371/journal.pone.0206476.g003

Fig 4. Solution path of LAR.  
https://doi.org/10.1371/journal.pone.0206476.g004
According to BIC criterion, the finally model contains 256 non-zeros elements in $\beta$, i.e., the ELM finally has 256 hidden layer nodes.

**Simulation results**

In this section, the simulation results of prediction and prediction performance of the proposed method are presented. The RMSE (root mean square error), defined as

$$
RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^{N} [y(k) - \hat{y}(k)]^2}
$$

is used to evaluate the prediction performance. In (18), $y(k)$ is the real measured roll angle at time $k$ and $\hat{y}(k)$ is the predicted roll angle. The real and the one-step predicted roll angle are shown in Fig 5. Also, the prediction error is shown. It can be seen from Fig 5 that the prediction is accurate with small prediction error. The RMSE of the prediction is $1.9819 \times 10^{-4}$.

To show its effectiveness, the proposed method is compared with autoregression (AR) method, conventional ELMs with different inputs and hidden layer nodes. Table 1 lists the comparison results of RMSEs of AR with different number of inputs, ELM with different number of inputs and different number of hidden layer nodes, and $\ell_1$-ELM. It can be seen from Table 1 that with the increase of the number of inputs, the RMSE of AR and ELM decrease, on the other hand, with the increase of the number of hidden layer nodes, the RMSE of ELM also decreases. This fact implies that selecting the appropriate inputs and hidden layer nodes has
great effect on the prediction accuracy. For our proposed method, the inputs and number of hidden layer nodes are objectively determined using suitable algorithms and can obtain more accurate prediction result than conventional ELM and AR methods. This demonstrates the advantage of the proposed method. To show the advantage of the proposed method, the prediction error of AR method with 6 inputs and ELM with 6 inputs and 250 hidden layer nodes are shown in Fig 6.

**Conclusion**

A $\ell_1$ regularized ELM based scheme is proposed for ship roll motion prediction. The proposed method combines Lipschitz quotient and $\ell_1$ regularized technique to determine appropriate structure of ELM for the purpose of obtaining high accurate prediction. Real measured roll motion data is used to validate the effectiveness of the proposed method. Simulated prediction results show that the proposed method can achieve more accurate prediction than conventional ELM and AR prediction method.

Table 1. RMSEs of different methods.

| Inputs— | AR | ELM(# hidden layer nods) | $\ell_1$-ELM |
|--------|----|----------------------------|--------------|
|        |    | 50            | 100          | 150          | 200          | 250          | 256          |
| 2      | 0.5031 | 0.4475 | 0.4053 | 0.4044 | 0.4036 | 0.4031 | -            |
| 3      | 0.5014 | 0.4487 | 0.3927 | 0.3405 | 0.254  | 0.1645 | -            |
| 4      | 0.4866 | 0.4371 | 0.3807 | 0.2966 | 0.2049 | 0.0646 | -            |
| 5      | 0.4844 | 0.4322 | 0.3855 | 0.287  | 0.1819 | 0.0501 | -            |
| 6      | 0.4791 | 0.4263 | 0.3663 | 0.3197 | 0.2111 | 0.0422 | 1.9819 $\times 10^{-4}$ |

Fig 6. Prediction error of AR, ELM and $\ell_1$-ELM.

https://doi.org/10.1371/journal.pone.0206476.g006
Author Contributions

Conceptualization: Binglei Guan.

Data curation: Wei Yang.

Formal analysis: Zhibin Wang.

Investigation: Yinggan Tang.

Writing – original draft: Binglei Guan.

Writing – review & editing: Yinggan Tang.

References

1. Fossen TI. Handbook of marine craft hydrodynamics and motion control. Chichester, UK: John Wiley; 2011.
2. Zhao X, Xu R, Kwan C. Ship-motion prediction: algorithms and simulation results. In: Proc. and Signal Processing 2004 IEEE Int. Conf. Acoustics, Speech. vol. 5; 2004. p. V–125–8 vol.5.
3. Zheng M, Li L, Peng H, Xiao J, Yang Y, Zhang Y, et al. Finite-time stability and synchronization of memristor-based fractional-order fuzzy cellular neural networks. Communications in Nonlinear Science and Numerical Simulation. 2018; 58:272–291. https://doi.org/10.1016/j.cnsns.2017.11.025
4. Chen C, Li L, Peng H, Yang Y. Adaptive synchronization of memristor-based BAM neural networks with mixed delays. Applied Mathematics & Computation. 2018; 322:100–110.
5. Yin JC, Zou ZJ, Xu F. On-line prediction of ship roll motion during maneuvering using sequential learning RBF neural networks. Ocean Engineering. 2013; 61:139–147. https://doi.org/10.1016/j.oceaneng.2013.01.005
6. Huang BG, Zou ZJ, Ding WW. Online prediction of ship roll motion based on a coarse and fine tuning fixed grid wavelet network. Ocean Engineering. 2018; 160:425–437. https://doi.org/10.1016/j.oceaneng.2018.04.065
7. Yin JC, Perakis AN, Wang N. A real-time ship roll motion prediction using wavelet transform and variable RBF network. Ocean Engineering. 2018; 160:10–19. https://doi.org/10.1016/j.oceaneng.2018.04.058
8. Huang GB, Zhu QY, Siew CK. Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE International Joint Conference on Neural Networks. vol. 2; 2004. p. 985–990.
9. Fu HX, Wang YC, Zhang HM. Ship rolling motion prediction based on extreme learning machine. In: 2015 34th Chinese Control Conference (CCC); 2015. p. 3468–3472.
10. Yu C, Yin JC, Hu JQ, Zhang AQ. Online ship rolling prediction using an improved OS-ELM. In: the 33th chinese control conference; 2014. p. 5043–5048.
11. Yin JC, Zou ZJ, Xu F, Wang NN. Online ship roll motion prediction based on grey sequential extreme learning machine. Neurocomputing. 2014; 129:168–174. https://doi.org/10.1016/j.neucom.2013.09.043
12. Yin J, Wang N, Perakis AN. A real-time sequential ship roll prediction scheme based on adaptive sliding data window. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2017; p. 1–11.
13. He X, Asada H. A new method for identifying orders of input-output models for nonlinear dynamic systems. In: American Control Conference, 1993; 1993. p. 2520–2523.
14. Tang Y, Li Z, Guan X. Identification of nonlinear system using extreme learning machine based Hammerstein model. Communications in Nonlinear Science and Numerical Simulation. 2014; 19(9):3171–3183. https://doi.org/10.1016/j.cnsns.2013.12.006
15. Wan W, Hirasa kwa, Hu J, Jin C. A new method to prune the neural network. In: IEEE-INNS-ENNS International Joint Conference on Neural Networks; 2000. p. 6449.
16. Guo H, Ren X, Li S. A New Pruning Method to Train Deep Neural Networks; 2018.
17. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Annals of Statistics. 2004; 32 (2):407–451. https://doi.org/10.1214/009053604000000067
18. Sjöstrand K, Clemmensen LH, Larsen R, Ersbøll B, Einarssson G. SpaSM: A MATLAB Toolbox for Sparse Statistical Modeling. Journal of Statistical Software. 2018; 84(10).1–24.