Pruned Broad Learning System Based on Sparse Ridge Fusion

Fei Chu

chufeizhufe@sina.com

China University of Mining and Technology

Xinyu Lu

China University of Mining and Technology

Jiaming Su

Northeastern University

Tao Liang

Northeastern University

C. L. Philip Chen

South China University of Technology

Xuesong Wang

China University of Mining and Technology

Research Article

Keywords: Broad learning system, Sparse ridge fusion, Output weight, Model interpretability

Posted Date: October 3rd, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3387037/v1

Additional Declarations: No competing interests reported.

Pruned Broad Learning System Based on Sparse Ridge Fusion

Fei Chu[a,b,*](IEEE senior member), Xinyu Lu, Jiaming Su[a], Tao Liang[c], C. L. Philip Chen[d](IEEE Fellow), Xuesong Wang[a]

a. Artificial Intelligence Research Institute, China University of Mining and Technology, Xuzhou 221116, China
b. School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China
c. College of Information Science and Engineering, Northeastern University, Shenyang 110004, China
d. School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

***Abstract*—Broad learning system is an emerging method, which has achieved outstanding performance in regression and classification problems. This paper proposes a novel algorithm called Pruned Broad Learning System (PBLS) to reduce model size and improve model interpretability. The proposed PBLS introduces the sparse ridge fusion penalty into BLS, which combines the L_1-norm regularizer and second order difference penalty together. The L_1-norm regularizer is used to sparse the model by making insignificant output weights toward zero, while second order difference penalty is used to smooth output weights by penalizing the roughness of the model. Then, the unimportant nodes are pruned from the model. The remaining useful nodes are divided in the form of block structure, and their output weights change slowly and smoothly in their respective block structure. The experiments on several commonly used regression data sets are carried out to verify the feasibility of the proposed pruned BLS. The experiment results indicate that pruned BLS can reduce model complexity without loss of prediction accuracy and improve the model interpretability.**

***Index Terms*—Broad learning system, Sparse ridge fusion, Output weight, Model interpretability**

***Impact Statement* —This paper concern with PBLS (Pruned BLS), an approach to reduce model size and improve model interpretability. The proposed PBLS introduces the sparse ridge fusion penalty into BLS, which combines the L1-norm regularizer and second order difference penalty together. The L1-norm regularizer is used to sparse the model by making insignificant output weights toward zero, while second order difference penalty is used to smooth output weights by penalizing the roughness of the model. Then, the unimportant nodes are pruned from the model. The remaining useful nodes are divided in the form of block structure, and their output weights change slowly and smoothly in their respective block structure.The experiment results indicate that pruned BLS can reduce model complexity without loss of prediction accuracy and improve the model interpretability.**

I. Introduction

BROAD learning system (BLS) is an efficient incremental neural network that does not require a deep structure[1]. It takes the advantage of random vector functional-link neural network (RVFLNN)[2]. Unlike RVFLNN, the main advantage of BLS is the input data can be mapped to a series of random feature spaces, Then the sparse autoencoder can extract the sparse features from given training data and the output weights can be determined by ridge regression. On the basis of the RVFLNN, Wang[3]proposed a stochastic configuration network(SCN), SCNs randomly assign the implied layer parameters within an adjustable interval, and innovatively introduce a supervisory mechanism to constrain them, ensuring their infinite approximation property. Wang[4] extend SCN to deep structures and proposed DeepSCN, DeepSCN has good generalization and learning performance. Wang[5] proposed a robust SCN for contaminated data modeling problem. Unlike some popular deep networks[6], which require a lot of time to learn excessive parameters, BLS can provide an alternative algorithm for fast universal approximation in various applications. The training process is quickly reconstructed in broad expansion through incremental learning algorithms[7]. If the network needs to be expanded, there is no need for a retraining process. Johan[8] pointed out the computational limitations of shallow neural networks. BLS is different from general shallow neural networks in that it can flexibly expand the network structure and has the ability to extract features efficiently. Due to the outstanding performance, BLS has been applied in many fields. Jin and Chen[9] proposed the regularized robust BLS for uncertain data modeling. Zhang [10] proposed an incremental BLS with a semantic feature extraction layer, and this new approach improves its fast modeling capability. Huang proposed a multi-label broad learning system, which aims to improve the classification performance of the model when dealing with large-scale data. Jin [12] proposed a novel robust broad learning system to handle classification problems with labeled noise based on a probabilistic framework. Jin [13] proposed a

novel discriminative group sparse broad learning system to enhance the classification ability of BLS in visual recognition. Liu[14] proposed a regularized broad learning system to address outliers and Gaussian noise in data modeling

BLS can expand the network horizontally to meet different task requirements, which raises the problem of how far the network needs to expand. On the one hand, the fewer nodes, the narrower the network, and the simpler the model, which may lead to poor performance of the model. On the other hand, if there are too many nodes, the network will be overly wide and it will produce a complex model, which may lead to the over-fitting problem. Currently, in order to obtain a suitable model, a common method is to train multiple models of different sizes and then select a model with the minimum generalization error[7]. This method is time consuming, and it is very unfavorable to apply BLS to some situations, such as edge computing[15].

The output weights of the network nodes play an important role in BLS modeling. It is necessary to make the output weights of the nodes in the final model have certain sense. The nodes in the model do not all have a high contribution to the prediction[16]. The nodes with higher output weights are more important for prediction, while those with lower output weights may have little or even no contribution[17]. The existence of these low-contribution nodes in the network makes the model redundant[18]. The interpretability of the model becomes very poor if the model is redundant[19]. Therefore, it is necessary to prune the model to avoid redundancy[20]-[21]. Sparse output weights are a practical method of pruning model. However, the L_2-norm regularizer in the standard BLS does not have the ability of output weight sparseness[22]. In addition, whether the output weight of the model is smooth or not has attracted a lot of attention[23]-[25]. Smoothness makes the adjacent output weights in the network have similar values. Therefore, the adjacent output weights change slowly. This can capture the relation between the nodes, and encourage the nodes with similar effect to be selected simultaneously in the form of blocks, and present slow and smooth changes in their respective blocks. In the existing improved BLS, lasso[26] and elastic-net[22] are used to sparse the model. Nevertheless, the improved BLS based on lasso and elastic-net cannot produce smooth output weights, and the interpretability of the model may be poor.

In order to reduce model size and generate sparse and smooth output weights to improve the model interpretability to a certain extent, a new algorithm called Pruned Broad Learning System (PBLS) is proposed. PBLS replaces L_2-norm regularizer used in BLS with sparse ridge fusion (SRF)[27], where L_1-norm regularizer is used for weight sparseness and second order difference penalty ($\sum_{j=2}^{p-1}\left(W_{j+1}-2W_j+W_{j-1}\right)^2$)[27] is used for weight smoothness. In addition, the L_2-loss function used in BLS is replaced by L_1-loss function to improve the robustness of the model. In summary, the main characteristics of the proposed PBLS are as follows

(1) PBLS begins with an initial large number of nodes to train the model. Then, the unimportant nodes get zero output weight, and prune the model to avoid redundancy by removing these nodes. Next, the remaining nodes with similar effect are divided into several blocks. Finally, a simplified and interpretable model is obtained.

(2) PBLS can smooth output weights by the penalty $\sum_{j=2}^{p-1}\left(W_{j+1}-2W_j+W_{j-1}\right)^2$. PBLS can reduce the irregular fluctuation between the output weights of three or more adjacent nodes and make the fluctuation curve more stable. This means that the adjacent output weights change slowly. Therefore, PBLS can capture the relationship between the adjacent nodes. The nodes with similar effect are divided into blocks, and the output weights in each block change slowly and smoothly.

(3) PBLS reduces the model complexity without losing the prediction accuracy. More importantly, PBLS can produce meaningful output weights, which improves the interpretability of the model to a certain extent.

The rest of the paper is organized as follows. Section 2 gives a brief review of BLS and sparse ridge fusion penalty. The details of the proposed PBLS algorithm is then described in Section 3. Section 4 presents the performance comparison of PBLS to other algorithms based on commonly used regression datasets. Finally, the conclusion is given in section 5.

II. PRELIMINARIES

A. *Broad learning system*

Broad learning system is a new type of neural network for fast universal approximation[7]. It inherits the advantages of random vector function link neural network(RVFLNN)[2] and develops an extendable flat network structure on this basis. This part will briefly review the broad learning system.

Fig.1 shows the network structure of BLS. The specific learning process of BLS is as follows.

Supposing that the training set is $X\in R^{t\times D}, X=\left[x_1,x_2,\ldots,x_t\right]^T$, where t is the number of samples and D is the dimension of samples, and there are n groups of feature nodes. Then the mapped features are obtained by transforming the training set X

$$Z_i=\phi(XW_{e_i}+\beta_{e_i}),i=1,...,n \tag{1}$$

where W_{e_i} and β_{e_i} are randomly generated. Z_i represents the ith group of mapped features. All the groups of mapped features are denoted as $Z^n\equiv[Z_1,Z_2,\ldots,Z_n]$. Similarly, the jth group of enhanced nodes is obtained by transforming all the mapped features

$$H_j = \xi(Z^n W_{h_j} + \beta_{h_j}), j = 1,\ldots,m \tag{2}$$

where W_{h_j} and β_{h_j} are randomly generated, $\xi(\cdot)$ is the sigmoid activation function. The m groups of enhanced nodes are denoted as $H^m \equiv [H_1, H_2, \ldots, H_m]$.

The output of BLS could be represented as

$$\hat{Y} = AW \tag{3}$$

where $A = [Z^n \mid H^m]$ and W is the output weight that connects the mapped features and the enhanced nodes to the output layer. Finally, W is optimized by the following formula

$$\arg\min_W \|Y - AW\|_2^2 + \lambda \|W\|_2^2 \tag{4}$$

where λ is the ridge parameter. Then, the output weight can be approximated by the ridge regression

$$W = (A^T A + \lambda I)^{-1} A^T Y \tag{5}$$

where I is the identity matrix.

B. *Sparse ridge fusion*

Sparse ridge fusion is a new regularization and variable selection method, which is used to generate sparse and smooth regression coefficients. The technique of the SRF is mixture of $L_1 - norm$ regularizer and the penalty $\sum_{j=2}^{p-1}\left(W_{j+1} - 2W_j + W_{j-1}\right)^2$

$$\|W_j\|_1 + \sum_{j=2}^{p-1}\left(W_{j+1} - 2W_j + W_{j-1}\right)^2 \tag{6}$$

equation (6) is proposed by Mahmood[27],where W are the regression coefficients, p is the dimension of the samples, $\| \ \|_1$ stands for the L_1 norm of a vector which equals the sum of absolute values of the vector's entries.

The $L_1 - norm$ regularizer is used to shrink some coefficients toward zero and make some coefficients exactly equal to zero. The penalty $\sum_{j=2}^{p-1}\left(W_{j+1} - 2W_j + W_{j-1}\right)^2$ [27]is a constraint form of second order difference. It can reduce the irregular fluctuation between the coefficients and make the fluctuation curve more stable. Therefore, it is used to penalize the roughness of three consecutive coefficients and achieve a more stable solution. In general, sparse ridge fusion can obtain more meaningful solutions, which is beneficial to improve the interpretability of the model.

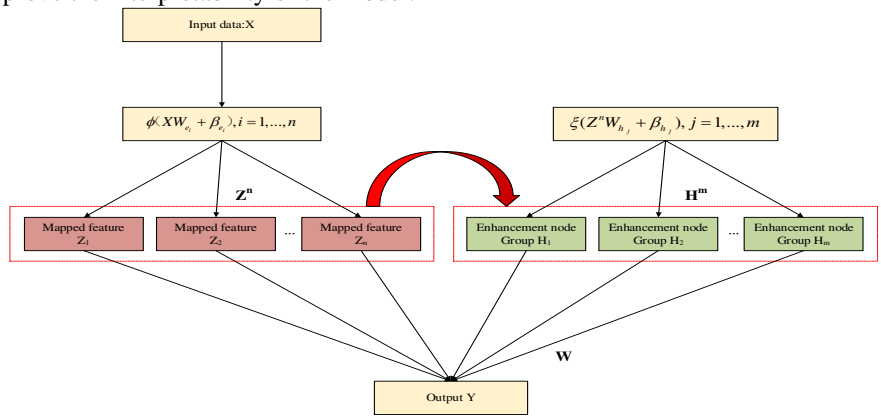

Fig.1 The network structure of BLS

III. PRUNED BROAD LEARNING SYSTEM

In this section, the details of the proposed Pruned Broad Learning System algorithm are described. Compared with the standard BLS, PBLS modifies the objective function, where the $L_2 - loss$ function is replaced by the $L_1 - loss$ function and the sparse ridge fusion is introduced. First, the initial model structure is set to be larger than actually needed. Then, PBLS prunes the model by removing unimportant nodes, and the remaining important nodes are divided into several blocks.

A. *PBLS Model*

In order to improve the robustness, PBLS replaces the $L_2 - loss$ function with the $L_1 - loss$ function. Moreover, PBLS utilizes the sparse ridge fusion penalty to produce sparse and smooth output weights. First, the corresponding feature layer and enhancement layer output are obtained by equation (1) and (2). Then, the feature layer and the enhancement layer are connected together to the output layer to obtain an expanded input matrix, which can be expressed as $A = \left[Z^n \mid H^m\right]$. Finally, the model is

denoted as $Y = AW$. Based on the above description, the objective function of PBLS can be expressed as

$$\arg\min_{W} \|Y - AW\|_1 + \lambda_1 \|W\|_1 + \lambda_2 \sum_{2}^{p-1} \left(W_{i+1} - 2W_i + W_{i-1}\right)^2 \tag{7}$$

The first term in the penalty leads to a tendency, like the lasso[24], to set the output weight of the irrelevant nodes to zero. The second term tends to give smooth solutions to the three consecutive nodes. The resulting output weight vectors are composed of zero sequences and some non-zero sequences clustered in blocks[28].

B. OPTIMIZATION OF PBLS

After obtaining the corresponding expanded matrix A , the optimization objective function (7) of PBLS cannot be solved directly. Here, an iterative optimization calculation method is used to solve (7) by the Augmented Lagrange Multiplier (ALM) method[29]-[30]. According to ALM, several variables F , G are used to replace $Y - AW$ and W . In this way, equation (7) can be simplified as

$$\begin{aligned} &\arg\min_{G,W} \|F\|_1 + \lambda_1 \|G\|_1 + \lambda_2 \|CW\|_2^2 \\ &s.t.\ \ F = Y - AW, G - W = 0 \end{aligned} \tag{8}$$

where C is a square matrix and can be defined as

$$C = \begin{bmatrix} 1 & -2 & 1 & & & & & \\ & 1 & -2 & 1 & & & & \\ & & \cdot & \cdot & \cdot & & & \\ & & & \cdot & \cdot & \cdot & & \\ & & & & & 1 & -2 & 1 \\ & & & & & & & 0 \end{bmatrix}$$

Then, according to the calculation strategy of ALM, equation (7) is rewritten as an augmented Lagrangian function

$$\begin{aligned} L_1(F,G,W,H_1,H_2) = &\|F\|_1 + \lambda_1 \|G\|_1 + \lambda_2 \|CW\|_2^2 \\ &+ H_1^T (Y - AW - F) + H_2^T (G - W) \\ &+ \frac{\mu}{2}\left(\|Y - AW - F\|_2^2 + \|G - W\|_2^2\right) \end{aligned} \tag{9}$$

where H_1 , H_2 are Lagrange multipliers and μ is the penalty parameter.

For the convenience of the following calculation, equation (8) can be further simplified as

$$\begin{aligned} L_2(F,G,W,H_1,H_2) = &\|F\|_1 + \lambda_1 \|G\|_1 + \lambda_2 \|CW\|_2^2 \\ &+ \frac{\mu}{2}\left\|Y - AW - F + \frac{H_1}{\mu}\right\|_2^2 \\ &+ \frac{\mu}{2}\left\|G - W + \frac{H_2}{\mu}\right\|_2^2 \end{aligned} \tag{10}$$

The variables F,G,W,H_1,H_2 in L_2 can be optimized via the block coordinate descent (BCD) method[31]. The details are as follows.

(1)Optimize W : Fix the variables F,G,H_1,H_2 and remove the terms which are not related to W , then W can be

$$\arg\min_{w} \frac{\mu}{2}\left(\|J - AW\|_2^2 + \|K - W\|_2^2\right) + \lambda_2 \|CW\|_2^2 \tag{11}$$

calculated by solving the following formula

where $J = Y - F + \frac{H_1}{\mu}, K = G + \frac{H_2}{\mu}$. Then, equation (10) is expanded to get the following formula

$$\begin{aligned} E(W) = &\frac{\mu}{2}\left[(J - AW)^T (J - AW) + (K - W)^T (K - W)\right] \\ &+ \lambda_2 \|CW\|_2^2 \end{aligned} \tag{12}$$

By deriving the W of the formula (11), the problem can be solved as

$$W = \left(A^T A + \frac{2\lambda_2}{\mu} C^T C + I\right)^{-1} \left(A^T J + K\right) \tag{13}$$

where I is the identity matrix.

(2)Optimize F : Fix the variables G,W,H_1,H_2 and remove the terms which are not related to F , the optimization of F can be obtained via the following formula

$$\arg\min_{F} \|F\|_1 + \frac{\mu}{2}\|M - F\|_2^2 \tag{14}$$

where $M = Y - AW + \frac{H_1}{\mu}$. Because the formula (13) involves the $L_1 - norm$, it is not easy to solve directly. Then soft thresholdin[32] is used to obtain F

$$\begin{aligned} F &= shrink\left(M, \frac{1}{\mu}\right) \\ &= \max\left\{|M| - \frac{1}{\mu}, 0\right\} \times sign(M) \end{aligned} \tag{15}$$

(3)Optimize G : Fix the variables F, W, H_1, H_2 and remove the terms which are not related to G, the optimization of G can be obtained via the following formula

$$\arg\min_{G} \lambda_1 \|G\|_1 + \frac{\mu}{2}\|N - G\|_2^2 \tag{16}$$

where $N = W - \frac{H_2}{\mu}$. Like the method of optimizing F, G can be obtained as

$$\begin{aligned} G &= shrink\left(N, \frac{\lambda_1}{\mu}\right) \\ &= \max\left\{|N| - \frac{\lambda_1}{\mu}, 0\right\} \times sign(N) \end{aligned} \tag{17}$$

The Lagrange multipliers H_1, H_2 are updated in each iteration by the following formula

$$\begin{aligned} H_1^{'} &= H_1 + \mu(Y - AW - F) \\ H_2^{'} &= H_2 + \mu(G - W) \end{aligned} \tag{18}$$

Based on the above description, the algorithm steps of PBLS are shown in Algorithm 1.

Algorithm 1 :Pruned Broad Learning System

Input: Training set $\{X, Y\}$

Initialization: F, G, W, H_1, H_2

1 Random $W_{e_i}, \beta_{e_i}, i = 1, 2, \ldots, n$;

2 Calculate (1) $Z_i = \phi\left(XW_{e_i} + \beta_{e_i}\right)$, set the groups of mapped features as $Z^n \equiv [Z_1, Z_2, \ldots, Z_n]$;

3 Random $W_{h_j}, \beta_{h_j}, j = 1, 2, \ldots, m$;

4 Calculate (2) $H_j = \xi\left(Z^n W_{h_j} + \beta_{h_j}\right)$, set the groups of enhanced nodes as $H^m \equiv [H_1, H_2, \ldots, H_m]$;

5 Set $A = \left[Z^n \mid H^m\right]$;

6 Optimize the objective function (7) as follows:

7 While the number of iterations does not reach the preset value

1 Optimize W by calculating (13);

2 Optimize F by calculating (15);

3 Optimize G by calculating (17);

4 Update H_1 and H_2 by calculating (18);

End while

Output: Output weight W

IV. EXPERIMENTS

In this section, some regression datasets are used to evaluate the performance of the proposed PBLS. The datasets are collected from the University of California at Irvine (UCI) Machine Learning Repository[33] and a real-world application dataset[35]. The real-world application involved a multi-stage centrifugal compressor of a spare power plant of a steelwork. The specific information of the datasets is shown in Table 1.

Table 1 The specific information of the data sets

Data set	Attributes	Samples	
		Train	Test
Basketball	4	64	32

Cleveland	13	202	101
Abalone	8	2784	1393
Quake	3	1452	726
Weather	9	974	487
Seoul Bike	10	5840	2920
Compressor	5	400	100

In the experiments, RVFLNN[2] ,L2-L1-RVFL[34]and the standard BLS and two improved algorithms of BLS (L1RBLS, ENRBLS) are used to compare with the proposed PBLS. The L1RBLS and ENRBLS were proposed in [9]. The two improved algorithms used lasso and elastic net respectively, in which L1RBLS combined $L_1 - loss$ function with lasso, and ENRBLS combined $L_1 - loss$ function with elastic net. L2-L1-RVFL which combines $L_1 - loss$ function with elastic net. In all BLS-type (BLS, L1RBLS, ENRBLS, PBLS) methods, the sigmoid function $f(x) = 1/(1+\exp(-x))$ is selected as the activation function[7]. This paper uses root mean square error (RMSE) to measure the prediction performance of the models. The RMSE can be computed as follows

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{19}$$

where y_i and $\hat{y}_i$ are the actual values and the prediction values respectively, N represents the number of samples.

In addition, in order to make the experimental data more in line with the actual collected data, some outliers are added to these experimental datasets. We added random values (Δy_{noise})to the output of some samples to generate outliers

$$y_{outlier} = y + \Delta y_{noise}, -y_{\min} \le \Delta y_{noise} \le y_{\max} \tag{20}$$

In the experiments, the data contamination rate is set to 20% (the number of outliers/the number of samples). Obviously, the data contamination rate can be set to any other value. However, this paper focuses on the problem of model selection rather than outlier processing. Therefore, all experiments in this paper are only carried out under the 20% data contamination rate.

In RVFLNN and L2-L1-RVFL method, The search range of hidden nodes is set to $[1,1200]$, The regularization parameters are selected by the grid search method in the candidate range $\{2^{-10},2^{-9},\ldots,2^{10}\}$ .In all BLS-type methods, there are two main parameters: the network structure parameters, including mapping groups, feature nodes of each group, enhancement nodes, and the regularization parameters. The BLS network structure parameters are determined by the grid search method in the range of $[1,30]*[1,30]*[1,300]$. The other algorithms get the optimal network structure by pruning the model. The regularization parameters are selected by the grid search method in the candidate range $\{2^{-10},2^{-9},\ldots,2^{10}\}$. All the experiments are carried out using Matlab (2017b) on a computer with Intel Core i5-8400 processor at 2.80GHz and 8GB RAM.

Table 2-8 shows the performance of the six algorithms (RVFLNN, L2-L1-RVFL, BLS, L1RBLS, ENRBLS, PBLS) on seven datasets. The specific information given in the table is as follows: the regularization parameters, the mean finial number of nodes for 100 tests, the training time, the mean RMSE of 100 tests and its standard deviation. From table 2-8, the following results can be found:

(1) Take into account the factor of the prediction accuracy, RVFLNN and BLS have similar prediction accuracy. The proposed PBLS achieve better performance than the other four algorithms. The PBLS and ENRBLS have similar prediction accuracy on Seoul Bike, Abalone and Basketball data sets. Both L2-L1-RVFL and ENRBLS are modeled by elastic net, but L2-L1-RVFL have poor ability of deal with outliers, so the prediction performance of L2-L1-RVFL is lower than ENRBLS. In other cases, the proposed PBLS outperforms the ENRBLS. Compared to training time between these datasets, the training time of BLS outperforms the other four algorithms. The output weight of PBLS are obtained using ALM and BCD. The algorithm itself involves the iterative convergence of various parameters, so the training time is longer than BLS.

(2) Compared with RVFLNN and BLS, the proposed PBLS can select fewer nodes. If we only consider the single factor of network structure, L1RBLS is clearly the best performing algorithm because it can select the fewest nodes. Overall, the model pruning rate of L2-L1-RVFL is lower than that of PBLS and higher than that of ENRBLS, However, L1RBLS performs poorly in prediction accuracy. Compared with ENRBLS, the proposed PBLS can obtain a simpler network structure by pruning the model while obtaining similar or even better prediction accuracy.

(3) Considering the two factors of prediction accuracy and the number of nodes simultaneously, the proposed PBLS performs better than the other five algorithms. The PBLS can have a better balance between prediction accuracy and network structure. In other words, PBLS can obtain a simpler network structure while maintaining a certain degree of prediction accuracy.

Table 2 Performance of different algorithms on Abalone

	Algorithm					
	RVFLNN	L2-L1-RVFL	BLS	L1RBLS	ENRBLS	PBLS
regularization parameters	5	(-1,2)	5	-4	(-3,2)	(-2,0)

node number	301	278	312	170	393.3	212
train time	0.0218	0.0413	0.0202	0.0521	0.0513	0.0266
RMSE $\pm$ STD($\times 10^{-2}$)	12.33 ± 0.19	7.88 ± 0.06	12.28 ± 0.04	7.91 ± 0.12	7.70 ± 0.02	7.71 ± 0.03

Table 3 Performance of different algorithms on Basketball

	Algorithm					
	RVFLNN	L2-L1-RVFL	BLS	L1RBLS	ENRBLS	PBLS
regularization parameters	3	(-2,2)	2	-1	(-2,5)	(-2,4)
node number	239	144.3	215	8.4	146.7	128.8
train time	0.0127	0.0276	0.0124	0.0164	0.0388	0.0156
RMSE $\pm$ STD($\times 10^{-2}$)	10.66 ± 0.67	7.96 ± 0.15	10.76 ± 0.24	8.48 ± 0.82	7.72 ± 0.04	7.71 ± 0.04

Table 4 Performance of different algorithms on Cleveland

	Algorithm					
	RVFLNN	L2-L1-RVFL	BLS	L1RBLS	ENRBLS	PBLS
regularization parameters	5	(-2,4)	5	1	(-1,5)	(0,9)
node number	268	193.1	255	40.1	180.1	114.9
train time	0.0123	0.03991	0.0125	0.03795	0.03799	0.01782
RMSE $\pm$ STD($\times 10^{-2}$)	14.02 ± 0.22	13.66 ± 0.23	13.96 ± 0.19	14.93 ± 0.74	13.40 ± 0.25	12.73 ± 0.32

Table 5 Performance of different algorithms on Quake

	Algorithm					
	RVFLNN	L2-L1-RVFL	BLS	L1RBLS	ENRBLS	PBLS
regularization parameters	2	(-1,4)	4	-5	(-5,5)	(-2,7)
node number	266	266.4	253	121.5	372.3	122.5
train time	0.01402	0.0511	0.01395	0.1033	0.0633	0.04353
RMSE $\pm$ STD($\times 10^{-2}$)	20.97 ± 0.04	19.33 ± 0.41	20.96 ± 0.01	17.95 ± 0.11	17.65 ± 0.05	17.49 ± 0.19

Table 6 Performance of different algorithms on Weather

	Algorithm					
	RVFLNN	L2-L1-RVFL	BLS	L1RBLS	ENRBLS	PBLS
regularization parameters	1	(-3,3)	2	-2	(-5,5)	(-4,5)
node number	324	289.4	314	192.4	327.6	244.5
training time	0.01755	0.1901	0.01746	0.02015	0.01852	0.01847
RMSE $\pm$ STD($\times 10^{-2}$)	2.71 ± 0.05	2.15 ± 0.07	2.12 ± 0.06	2.62 ± 0.03	2.08 ± 0.014	2.04 ± 0.062

Table 7 Performance of different algorithms on Seoul Bike

	Algorithm					
	RVFLNN	L2-L1-RVFL	BLS	L1RBLS	ENRBLS	PBLS
regularization parameters	3	(-3,4)	1	-6	(-5,4)	(-8,4)
node number	317	311.2	321	32.7	542.6	246.7
training time	0.06113	0.06814	0.06047	0.07345	0.06934	0.07196
RMSE $\pm$ STD($\times 10^{-2}$)	0.43 ± 0.021	0.31 ± 0.014	0.36 ± 0.051	0.32 ± 0.017	0.27 ± 0.014	0.26 ± 0.018

Table 8 Performance of different algorithms on Compressor

	Algorithm					
	RVFLNN	L2-L1-RVFL	BLS	L1RBLS	ENRBLS	PBLS
regularization parameters	3	(-1,1)	5	-4	(-2,1)	(-3,2)
node number	377	289.5	408	148.2	346.2	241.6
train time	0.1023	0.1104	0.0955	0.1143	0.1204	0.1106
RMSE $\pm$ STD($\times 10^{-2}$)	11.23 ± 0.07	6.41 ± 0.04	10.04 ± 0.04	4.07 ± 0.07	3.94 ± 0.07	3.91 ± 0.08

In order to show the performance of different algorithms more intuitively, fig.2 and 3 show the accuracy improvement rate and model pruning rate of RVFLNN, L2-L1-RVFL, L1RBLS, ENRBLS, PBLS compared to BLS. Fig.2 shows the improvement in RMSE. The percentage is calculated by the following formula:

$$Percentage = \frac{RMSE_{BLS} - RMSE_{(addational\ algorithm)}}{RMSE_{BLS}} * 100 \tag{21}$$

where $RMSE_{BLS}$ is the prediction accuracy of BLS and $RMSE_{(addational\ algorithm)}$ is the prediction accuracy of the other five algorithms. As can be seen from the fig.2, in addition to the Cleveland dataset, ENRBLS and PBLS have similar performance on the other six datasets, which can greatly improve the prediction accuracy. In Cleveland dataset, the performance of L1RBLS is the worst, the prediction accuracy is reduced by 6.9%. In general, PBLS performs best in all datasets, and the prediction accuracy is improved by 28.3%, 8.8% ,16.5% ,37.2% 22.13%, 27.7% and 61.2% respectively. This proves that PBLS has a greater improvement in prediction accuracy compared to other algorithms.

Fig.3 shows the model pruning rate of several improved algorithms compared to the standard BLS. The model pruning rate is

calculated the same as Equation (21), but the RMSE is replaced by the model size. It can be seen that L1RBLS has the highest model pruning rate on these seven datasets. However, as can be seen from the Fig.2, the prediction accuracy of L1RBLS is poor. In addition to Basketball and Cleveland datasets. ENRBLS selects more nodes than BLS, so it does not achieve the effect of pruning the model. The proposed PBLS can obtain a simpler network structure, and the model pruning rate exceeds 30% in all cases, the highest is 54.9%. In addition, Fig.2 shows that PBLS can also have good prediction accuracy. Therefore, the proposed PBLS can achieve a good balance between prediction accuracy and model size.

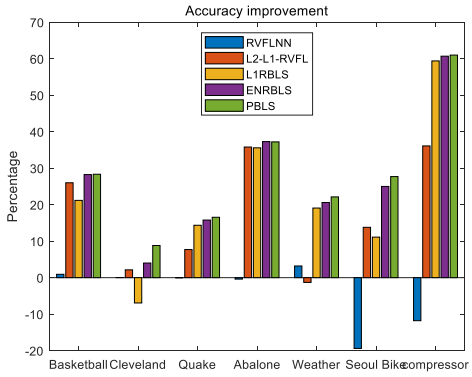

Fig.2 The accuracy improvement rate of RVFLNN, L2-L1-RVLFN, L1RBLS, ENRBLS and PBLS compared with BLS

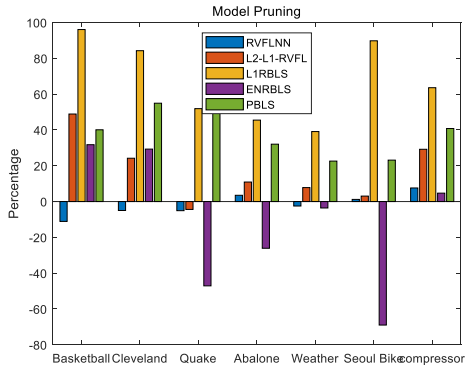

Fig.3 The model pruning rate of RVFLNN, L2-L1-RVFLN L1RBLS, ENRBLS and PBLS compared with BLS

Output weight is an important parameter of the model. Therefore, the output weights of each model on the four datasets are shown in Fig.5. In the figure, from top to bottom are RVFLNN, BLS, L1RBLS, ENRBLS, PBLS, L2-L1-RVFL, and from left to right are Abalone, Basketball, Cleveland and Quake. The following information can be obtained from the figure.

(1) For the output weights obtained by RVFLNN, BLS, L1RBLS and ENRBLS, L2-L1-RVFL, it can be clearly seen that the difference of the adjacent output weights is erratic. The output weights are disordered and present irregular fluctuations. The reason for the weight distribution may be that $L_1 - norm$ regularizer and $L_2 - norm$ regularizer used in the five algorithms only consider each output weight independently, but ignore the association between successive weights.

(2) In view of the output weights obtained by the proposed PBLS, it can be observed that the weights are clustered into some blocks. It means that the nodes with similar effect are grouped together. The output weights of each block change smoothly. Therefore, the output weights obtained by PBLS are no longer disordered. From the previous analysis, we can know that PBLS imposes constraints on the output weights of three adjacent nodes, which can capture the relationship between the nodes to a certain extent. Moreover, as can be seen from the figure, each block contains more than three output weights. Therefore, PBLS actually has a wider range to constrain the weights. Compared with the model with disordered output weights, PBLS can obviously produce a more interpretable model. After obtaining such a more meaningful solution of output weights, we may be able to further analyze the information contained in the nodes.

In this part, The experimental results show that PBLS can simplify the model by pruning the model without losing the prediction accuracy. Furthermore, PBLS can produce smooth and slowly changing output weights, which makes PBLS more competitive in model interpretability.

PBLS requires analytical solutions for W, F, G, H1, H2. Fig.4 shows RMSE of PBLS methods with respect to the number of iterations versus different datasets. It can be seen from Fig.4 that the RMSE of proposed PBLS increased at the beginning and finally reach a stable value. The BCD method will possibly have the problem of convergence reversal, so the RMSE appeared to rise during the initial iteration, and when the convergence direction is consistent, the RMSE tends to stabilize. After ten iterations, The RMSE of PBLS can reach a stable value. In this paper. the iteration values are set to 50.

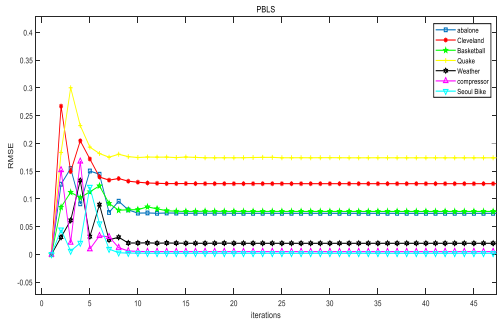

Fig.4 RMSE of PBLS methods with respect to the number of iterations versus different datasets

V. CONCLUSION

In this paper, pruned BLS is proposed for simplifying the model and improving model interpretability. The sparse ridge fusion contains both $L_1 - norm$ regularizer and the penalty $\sum_{j=2}^{p-1}\left(\beta_{j+1} - 2\beta_j + \beta_{j-1}\right)^2$. In addition, the $L_2 - loss$ function in BLS is replaced by $L_1 - loss$ function to improve the robustness of the model. An iterative optimization calculation method based on the Augmented Lagrange Multiplier is used to solve the improved objective function.

PBLS trains a large enough initial network with many nodes. Then, PBLS uses sparse ridge fusion to constrain the output weight, which makes it sparse and smooth. The unimportant nodes are removed from the network and the remaining nodes with similar effect are divided into several blocks. Finally, a simplified and interpretable model is obtained.

The experimental results show that the proposed PBLS have a better balance between model size and prediction accuracy. Most importantly, PBLS can generate meaningful output weights, which is conducive to improving the interpretability of the model.

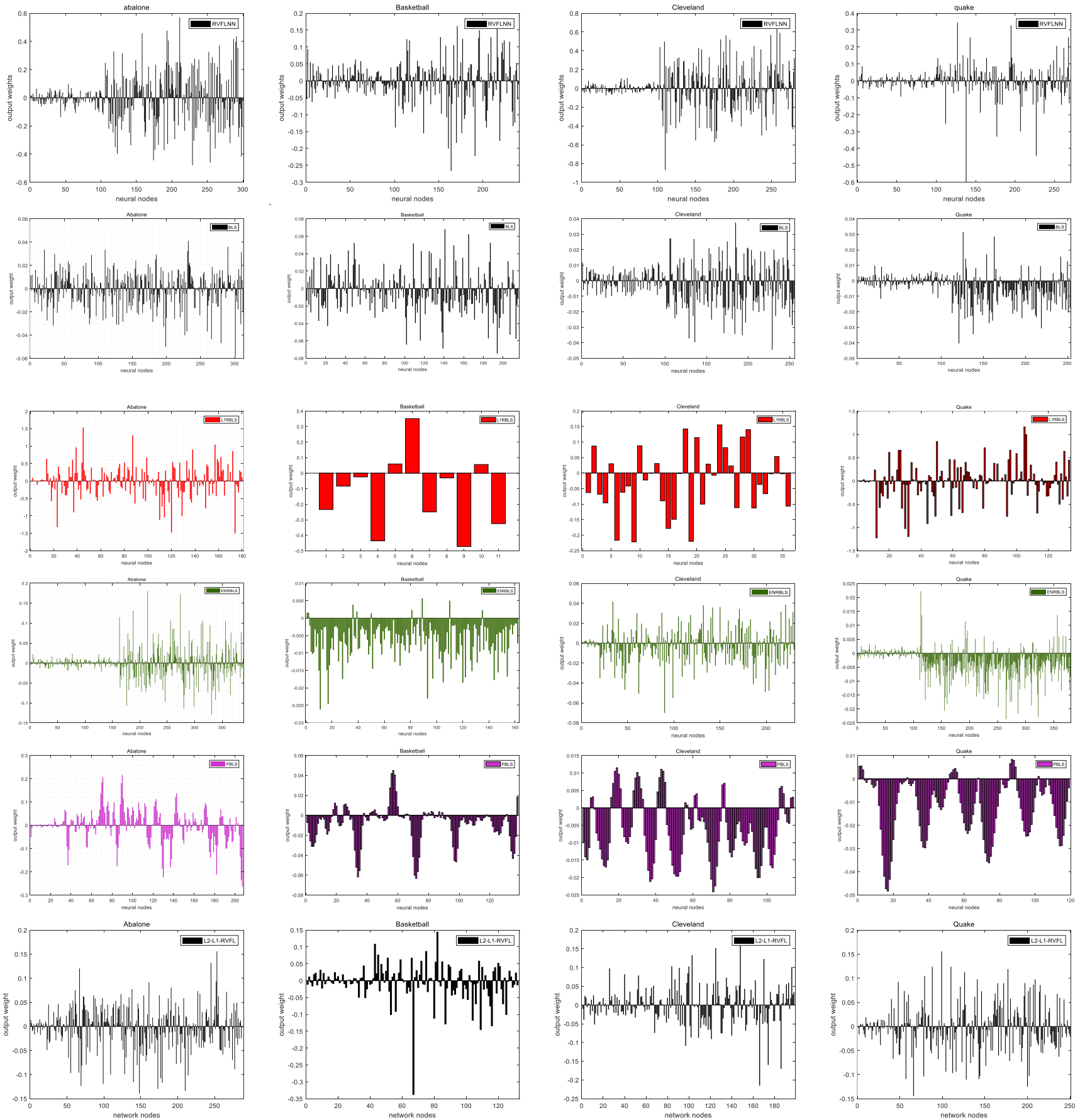

Fig.5 Output weights of the six algorithms (from top to bottom: RVFLNN, BLS, L1RBLS, ENRBLS , PBLS,L2-L1-RVFL) on the Abalone, Basketball, Cleveland and Quake datasets (from left to right)

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61973304, Grant 61873049, and Grant 62073060, in part by the Selection and Training Project of High-level Talents in the Sixteenth/Six Talent Peak of Jiangsu Province under Grant DZXX-045, in part by the Science and Technology Plan Project of Jiangsu Province under Grant BK20191339, Open Project of Autonomous Control Technology of Aircraft, Engineering Research Center of the Ministry of Education [grant numbers NJ2020004] Open Project of Autonomous Control Technology of Aircraft, Engineering Research Center of the Ministry of Education [grant numbers NJ2020004].

REFERENCES

[1] Chen C L P, Liu Z. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture[J]. IEEE transactions on neural networks and learning systems, 2017, 29(1): 10-24.
[2] Pao Y H, Takefuji Y. Functional-link net computing: theory, system architecture, and functionalities[J]. Computer, 1992, 25(5): 76-79.
[3] Wang D , Ming L . Stochastic Configuration Networks: Fundamentals and Algorithms[J]. IEEE Transactions on Cybernetics, 2017, 47(10):3466-3479.
[4] Wang D, Li M. Deep stochastic configuration networks with universal approximation property[C]//2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018: 1-8.
[5] Li M, Huang C, Wang D. Robust stochastic configuration networks with maximum correntropy criterion for uncertain data regression[J]. Information Sciences, 2019, 473: 73-86.
[6] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.
[7] Chen C L P, Liu Z, Feng S. Universal approximation capability of broad learning system and its structural variations[J]. IEEE transactions on neural networks and learning systems, 2018, 30(4): 1191-1204.
[8] Kurtz, Stuart A, Hstad Johan. Computational limitations of small-depth circuits. ACM doctoral dissertation awards. The MIT Press, Cambridge, Mass. and London, 1987, xiii + 84 pp.[J]. Journal of Symbolic Logic, 1988, 53(04):84-1260.
[9] Jin J W, Chen C L P. Regularized robust broad learning system for uncertain data modeling[J]. Neurocomputing, 2018, 322: 58-69.
[10] Zhang L, Li J, Lu G, et al. Analysis and variants of broad learning system[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020.
[11] Huang J, Vong C M, Chen C L P, et al. Accurate and Efficient Large-Scale Multi-Label Learning With Reduced Feature Broad Learning System Using Label Correlation[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022.
[12] Jin J, Li Y, Chen C L P. Pattern Classification with Corrupted Labeling via Robust Broad Learning System[J]. IEEE Transactions on Knowledge and Data Engineering, 2021.
[13] Jin J, Li Y, Yang T, et al. Discriminative group-sparsity constrained broad learning system for visual recognition[J]. Information Sciences, 2021, 576: 800-818.
[14] Liu L, Cai L, Liu T, et al. Cauchy regularized broad learning system for noisy data regression[J]. Information Sciences, 2022, 603: 210-221.
[15] Shi W, Cao J, Zhang Q, et al. Edge computing: Vision and challenges[J]. IEEE internet of things journal, 2016, 3(5): 637-646.
[16] Mehmood T, Liland K H, Snipen L, et al. A review of variable selection methods in partial least squares regression[J]. Chemometrics and intelligent laboratory systems, 2012, 118: 62-69.
[17] Han S, Pool J, Tran J, et al. Learning both weights and connections for efficient neural network[J]. Advances in neural information processing systems, 2015, 28.
[18] Chen W, Wilson J, Tyree S,K. Weinberger, and Y. Chen. International Conference on Machine Learning. PMLR, 2015: 2285-2294.
[19] Carmon N, Ben-Dor E. A Spectral Assignment-Oriented Approach to Improve Interpretability and Accuracy of Proxy Spectral-Based Models[J]. IEEE Transactions on Geoscience and Remote Sensing, 2018, 57(6): 3221-3228.
[20] Zhu M, Gupta S. To prune, or not to prune: exploring the efficacy of pruning for model compression[J]. arXiv preprint arXiv:1710.01878, 2017.
[21] Luo J H, Wu J, Lin W. Thinet: A filter level pruning method for deep neural network compression[J]. IEEE International Conference on Computer Vision. 2017: 5058-5066.
[22] Zou H, Hastie T. Regularization and variable selection via the elastic net[J]. Journal of the royal statistical society: series B (statistical methodology), 2005, 67(2): 301-320.
[23] Hawkins D M, Maboudou-Tchao E M. Smoothed linear modeling for smooth spectral data[J]. International Journal of Spectroscopy, 2013, 2013.
[24] Tibshirani R, Saunders M, Rosset S, et al. Sparsity and smoothness via the fused lasso[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2005, 67(1): 91-108.
[25] Zdunek R, Cichocki A. Improved M-FOCUSS algorithm with overlapping blocks for locally smooth sparse signals[J]. IEEE Transactions on Signal Processing, 2008, 56(10): 4752-4761.
[26] Tibshirani R. Regression shrinkage and selection via the lasso[J]. Journal of the Royal Statistical Society: Series B (Methodological), 1996, 58(1): 267-288.
[27] Mahmood N. Sparse Ridge Fusion For Linear Regression[J]. 2013.
[28] Katano R, Endo T, Yamamoto A, et al. Estimation of sensitivity coefficient based on lasso-type penalized linear regression[J]. Journal of Nuclear Science and Technology, 2018, 55(10): 1099-1109.
[29] Lin Z, Chen M, Ma Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices[J]. arXiv preprint arXiv:1009.5055, 2010.
[30] Rockafellar R T. Augmented Lagrange multiplier functions and duality in nonconvex programming[J]. SIAM Journal on Control, 1974, 12(2): 268-285.
[31] Tseng P. Convergence of a block coordinate descent method for nondifferentiable minimization[J]. Journal of optimization theory and applications, 2001, 109(3): 475-494.
[32] Donoho D L. De-noising by soft-thresholding[J]. IEEE transactions on information theory, 1995, 41(3): 613-627.
[33] Asuncion A, Newman D. UCI machine learning repository[J]. 2007.

[34] Ye H, Cao F, Wang D. A hybrid regularization approach for random vector functional-link networks[J]. Expert Systems with Applications, 2020, 140: 112912.

[35] Chu F, Wang F, Wang X, et al. Performance modeling of centrifugal compressor using kernel partial least squares[J]. Applied Thermal Engineering, 2012, 44: 90-99.

First A. Fei Chu (M'17) received his B.Sc. degree in school of automation, Qingdao University, China, in 2007, received his M.Sc. and Ph.D. degrees in control theory and control engineering from Northeastern University, China, in 2009 and 2014 respectively. He currently is an associate professor in School of Information and Control Engineering, China University of Mining and Technology, Xuzhou; and also with the member of State Key Laboratory of Management and Control for Complex System, Institute of Automation, Chinese Academy of Sciences; and also with the member of Xuzhou Key Laboratory of Artificial Intelligence and Big Data, Xuzhou. He has authored over 30 papers in major international journals and conferences. His current research interests include modeling, control and optimization of complex industrial process, machine learning, big data mining, statistical process monitoring and operating state evaluation, etc.

Second B. Xinyu Lu received his B.Sc. degree in 2019 from Tongling University. He is currently pursuing the M.S. degree in School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China. His main research interests include modeling, control and optimization of complex industrial process.

Third C. Jiaming Su received his B.Sc. degree in 2018 from Xuhai College, China University of Mining and Technology, Xuzhou. He is currently pursuing the M.S. degree in School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China. His main research interests include modeling, control and optimization of complex industrial process.

Fourth D. Tao Liang received his B.Sc. degree in 2017 from Xuhai College, China University of Mining and Technology, Xuzhou. He is currently pursuing the M.S. degree in School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China. His main research interests include modeling, control and optimization of complex industrial process

Fifth E. C. L. Philip Chen (S'88–M'88–SM'94–F'07) received the Ph.D. degree from the University of Michigan at Ann Arbor, Ann Arbor, MI, USA. He is currently a Chair Professor with the school of Computer Science & Engineering, South China University of Technology, Guangzhou, Guangdong, China. He successfully architects the University of Macau's Engineering and Computer Science programs receiving accreditations from Washington/Seoul Accord through the Hong Kong Institution of Engineers (HKIE), Hong Kong, which is considered as his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty. His current research interests include systems, cybernetics, and computational intelligence. Dr. Chen is a fellow of The American Association for the Advancement of Science, The International Association for Pattern Recognition, the Chinese Association of Automation (CAA), and HKIE. He received the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS and an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the IEEE TRANSACTIONS ON CYBERNETICS. From 2012 to 2013, he was the IEEE Systems, Man, and Cybernetics Society President. He is currently the Vice President of CAA

Sixth F Xuesong Wang (M'15) received the Ph.D. degree in control science and engineering from the China University of Mining and Technology, Xuzhou, China, in 2002. She is currently a Professor with the School of Information and Control Engineering, China University of Mining and Technology, and also with the member of Xuzhou Key Laboratory of Artificial Intelligence and Big Data, Xuzhou. Her current research interests include machine learning, bioinformatics, and artificial intelligence.