# A Novel Robust Stacked Broad Learning System for Noisy Data Regression

Kai Zheng[1], Jie Liu[2]
Office of Academic Research, Moutai Institute, Renhuai, China[1]
Department of Brewing Engineering Automation, Moutai Institute, Renhuai, China[2]

***Abstract*—Robust broad learning system (RBLS) demonstrates the generalization and robustness for solving uncertain data regression tasks. To enhance representation ability of RBLS, this paper aims at developing a novel robust stacked broad learning system for solving noisy data regression problems, termed as RSBLS. In our work, we expand traditional BLS into a stacked broad learning system model with deep structure of feature nodes and enhancement nodes. Furthermore, $\ell_1$ norm loss function is employed to update the objective function of RSBLS for processing noisy data, we apply augmented Lagrange multiplier (ALM) to get the output weights of RSBLS which keeps the effectiveness and efficiency compared with weighted loss function. Simulation results over some regression datasets with outliers demonstrate that, the proposed RSBLS performs favorably with better robustness with respect to RVFL, BLS, Huber-WBLS, KDE-WBLS and RBLS.**

***Keywords—Robust; stacking; broad learning system; deep learning; neural networks***

## I. INTRODUCTION

Recently, inspired by random vector functional link neural networks (RVFL) [1, 2], Chen et al. proposed a novel randomized neural network architecture, broad learning system (BLS) [3-5]. Compared with the classical deep learning model, BLS adopts the broad structure which has the advantages of higher efficiency and fewer parameters. Due to its excellent learning ability, BLS has been widely concerned by scholars since it was proposed, and has developed rapidly in theoretical and applied research. To improve the interpretability of BLS, a novel broad neuro-fuzzy model, fuzzy broad learning system (FBLS) was presented, which reduces the number of fuzzy rules and improves the learning accuracy neuro-fuzzy model [6]. Subsequently, Guo et al. used FBLS to synthesize multi-view HDR images [7], Ali proposed a novel optic disk and cup segmentation method through FBLS for glaucoma screening [8]. Furthermore, compact FBLS (CFBLS) which has better interpretability and fewer fuzzy rules was designed to balance the algorithm accuracy and complexity [9]. For improve the representation of BLS, type-2 fuzzy BLS was given in [10]. For solving sequential data, recurrent broad learning system and structured manifold broad learning system (SM-BLS) were presented respectively [11, 12]. To process the data with less label, semi-supervised broad learning system (SS-BLS) by introducing manifold regularization method to BLS [13], and some other SS-BLS algorithms had been used in semi-supervised classification tasks [14, 15]. Otherwise, BLS and its variants had been extensive used in various engineering fields, such as traffic forecasting [16], image classification [17], EEG signals classification [18-20], sentiment analysis [21, 22].

In practical engineering applications, sensors are susceptible to equipment failure, human interference, working environment and other factors, and there are different degrees of noises and outliers in the collected data, thus reducing the generalization of the learning model. To solve the uncertain data regression problem effectively, Chu et al. proposed weighted broad learning system (WBLS) framework for tackling industrial noisy data [23]. Then, Zheng et al. designed a broad learning system based on maximum correntropy criterion (BLS-MCC) which used maximum correntropy criterion to calculate weights of training samples [24]. In addition, Liu et al. adopted Cauchy loss function to process the noisy data [25]. Meanwhile, $\ell_1$ norm cost function and $\ell_2$ regularization method were used in robust broad learning system (RBLS) [26], then elastic-net regularization approach replaced $\ell_2$ regularization method in RBLS [27]. Moreover, robust manifold broad learning system (RM-BLS) was used to predict large-scale noisy chaotic time series [28]. In addition, for online sequential learning, Guo et al. presented online robust echo state broad learning system (OR-ESBLS) [29]. However, the above models improved the robustness of BLS, the shallow models still lack feature representation capability.

Now-a-days, deep neural networks with multi-layer have powerful representation capability, BLS was also been expanded with multi-layers [30-32]. Therefore, to improve the noisy data processing performance of RBLS, we demonstrate a novel robust stacked broad learning system (RSBLS) for solving outlier data regression, which adopts deep structure of feature nodes and enhancement nodes through stacking deep model, $\ell_1$ norm loss function and $\ell_2$ regularization method ensure the learning accuracy and efficiency.

In brief, the highlights of RSBLS are listed as follows:

- A novel robust stacked broad learning system structure is demonstrated, we presented the model architecture and algorithm description of RSBLS in detail.
- $\ell_1$ norm loss function and $\ell_2$ regularization method are adopted to enhance the robustness of RSBLS.
- Experiments on the benchmark datasets with different noise ratios present the superiority of RSBLS.

The other sections of our manuscript are given as follows: Section II introduces the basic algorithm description of BLS. Section III presents the architecture and optimization method of the proposed RSBLS. Section IV demonstrates the uncertain data regression results on some datasets with different percentage of outliers. Finally Section V concludes the paper.

## II. Related Works

BLS is novel random neural network model with effective and efficient performance, proposed by Chen et al., which has the architecture as Fig. 1 [3-5].

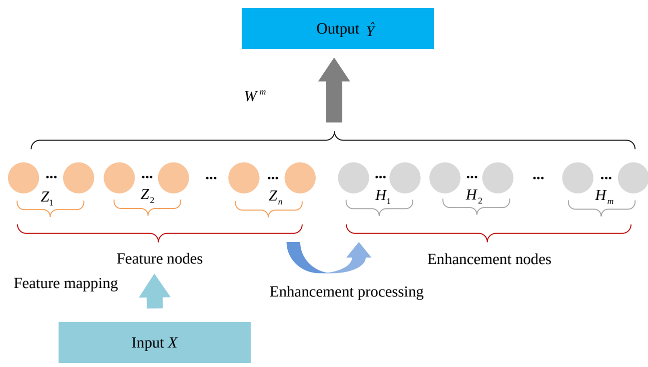


Fig. 1. The structure of BLS [3-5].

The modeling process of BLS is given as follows [3-5]:

Given a training data $\{X,Y\}$, $X=\{x_1,x_2,\ldots,x_N\}$ and $Y=\{y_1,y_2,\ldots,y_N\}$ express the feature and label. Among them, $x_i=\{x_{i,1},x_{i,2},\ldots,x_{i,d}\}\in\mathbb{R}^d$, $d$ is denoted as the number of feature dimension; $i=1,2,\ldots,N$, $N$ indicates the number of training samples.

### A. Feature Nodes Generation

The feature nodes are generated through Eq. (1), then $n$ groups of mapping nodes can be combined according to Eq. (2), each group has $L_e$ feature nodes.

$$Z_p=\phi(XW_{ep}+\beta_{ep}) \tag{1}$$

$$Z^n\equiv[Z_1,Z_2,\ldots,Z_n] \tag{2}$$

Among them, $\phi(\cdot)$ indicates the activation function of feature mapping; $W_{ep}$ and $\beta_{ep}$ represent random weights and biases respectively; $p=1,2,\ldots,n$. In particularly, the authors of BLS use sparse autoencoder to tune the initial parameters for obtaining better features [3-5].

### B. Enhancement Nodes Generation

All the feature nodes in Eq. (2) are enhanced by using Eq. (3), each enhancement processing generate $L_h$ nodes. Then the enhancement nodes can be combined according to Eq. (4).

$$H_q=\xi(Z^nW_{hq}+\beta_{hq}) \tag{3}$$

$$H^m\equiv[H_1,H_2,\ldots,H_m] \tag{4}$$

where, $\xi(\cdot)$ is denoted as the activation function, it can be set as the same as $\phi(\cdot)$; $W_{hq}$ and $\beta_{hq}$ are generated randomly; $q=1,2,\ldots,m$.

### C. Output $Y$ determination

The output $\hat{Y}$ of BLS can be calculated according to Eq. (5).

$$Y=[Z^n\,|\,H^m]W \tag{5}$$

$$W=[Z^n\,|\,H^m]^+Y \tag{6}$$

where, $W$ represents the output weight of BLS, $[Z^n\,|\,H^m]^+$ is determined by the ridge regression approximation as Eq. (7).

$$\begin{aligned}&[Z^n\,|\,H^m]^+\\&=\lim_{\lambda\to 0}(\lambda I+[Z^n\,|\,H^m][Z^n\,|\,H^m]^T)^{-1}[Z^n\,|\,H^m]^T\end{aligned} \tag{7}$$

## III. Robust Regularized Hierarchical Broad Learning System

### A. The Structure of RSBLS

BLS, as a randomized learning algorithm, has an effective and efficient structure. Although the novel structure can reduce the computational burden and enhance learning accuracy, BLS with multi-layer structure can extract deep representation information [30-33]. Meanwhile, the noisy and outliers in the training data affect the accuracy and generalization performance of BLS seriously. Therefore, we propose a novel RSBLS to solve noisy data regression problems. It is different from traditional BLS, RSBLS has multi-layer structure of feature nodes and enhancement nodes, the feature nodes and enhancement nodes of each layer are used as the input of next layer, only the feature nodes and enhancement nodes of the final layer are fully connected with the output.

In addition, due to the outliers usually take up a fraction of training data, the noisy data can be understood as having sparsity, $\ell_1$ norm function is not only more robust to solving the sparsity data, but also ensures a faster learning efficiency, it is especially suitable for solving large-scale data and deep models [34, 35]. The RSBLS adopts $\ell_1$ norm cost function and $\ell_2$ regularization method to solve the noisy data regression. Fig. 2 demonstrates the model structure of RSBLS; the RSBLS algorithm is described as follows:

*1) RSBLS parameters initialization:* To simplify the RSBLS model, the layer number is set as U, the feature mapping times of each layer are set to nu, the number of neurons in each feature mapping is $L_{ue}$, the enhancement processing times of each layer are set to mu, and the number of neurons in each enhancement processing is $L_{uh}$.

*a) Feature nodes generation:* The original data X is transformed into feature nodes by using Eq. (8), then all the nodes in the feature layer are combined through Eq. (9) and Eq. (10).

$$Z_p^u=\phi(X^uW_{ep}+\beta_{ep}) \tag{8}$$

$$Z^u=[Z_1^u,Z_2^u,\ldots,Z_{nu}^u] \tag{9}$$

where, $X^1 = X$; $X^u = S^{u-1}$; $\phi(\cdot)$ indicates the activation function of feature mapping; $W_{ep}$ and $\beta_{ep}$ are random values of feature nodes; $p = 1, 2, \ldots, nu$; $u = 1, 2, \ldots, U$.

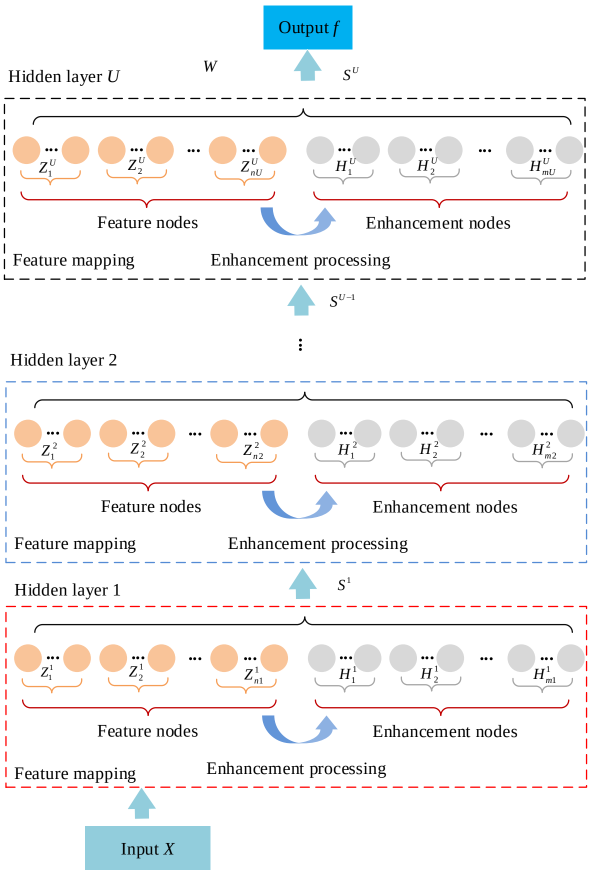


Fig. 2. The structure of RSBLS.

*b) Enhancement nodes generation:* All the feature nodes in Eq. (9) are enhanced as enhancement nodes through Eq. (10), then all the enhancement nodes are connected by Eq. (11).

$$H_q^u = \xi([Z^u W_{hq} + \beta_{hq}) \tag{10}$$

$$H^u = [H_1^u, H_2^u, \ldots, H_{mu}^u] \tag{11}$$

where, $\xi(\cdot)$ expresses the activation function of enhancement processing; $W_{hq}$ and $\beta_{hq}$ indicate the random parameters of enhancement nodes; $q = 1, 2, \ldots, mu$; $u = 1, 2, \ldots, U$.

In addition, we combine all the nodes of layer *u* through Eq. (12) as the input of next layer.

$$S^u = [Z^u, H^u] \tag{12}$$

*c) Target output matrix $Y$ determination:* To reduce the computational burden of RSBLS, we only connect the feature nodes and enhancement nodes of layer U in Eq. (13), then we use the $\ell_1$ norm cost function and $\ell_2$ regularization method to calculate the model output weights as Eq. (14).

$$S^U = [Z^U, H^U] \tag{13}$$

$$\beta = \arg\min_{\beta} \| S^U \beta - Y \|_1 + C \| \beta \|_2^2 \tag{14}$$

where, $C$ express as the $\ell_2$ regularization parameter.

The outputs of RSBLS can be calculated by Eq. (15).

$$Y = S^U \beta \tag{15}$$

## B. The Optimization of RSBLS

Hence Eq. (14) can be considered as a constrained convex optimization problem, we use augmented Lagrange multiplier (ALM) approach to solve this problem, and Eq. (14) can be transformed as Eq. (16).

$$\begin{aligned} L_\mu(e, \beta, \gamma) &= \| e \|_1 + \frac{1}{C} \| \beta \|_2^2 \\ &+ \upsilon^T (Y - H\beta - e) + \frac{\mu}{2} \| Y - H\beta - e \|_2^2 \end{aligned} \tag{16}$$

where, $e = \beta H - Y$; $\upsilon$ is denoted as the vector of Lagrange multiplier; $\mu = 2N / \| Y \|_1$.

The optimal $(e, \beta)$ and the Lagrange multiplier $\upsilon$ can be optimized by ALM method iteratively by using Eq. (17).

$$\begin{cases} (e_{\rho+1}, \beta_{\rho+1}) = \arg\min_{e,\beta} L_\mu(e, \beta, \upsilon_\rho) \\ \upsilon_{\rho+1} = \upsilon_\rho + \mu(Y - H\beta_{\rho+1} - e_{\rho+1}) \end{cases} \tag{17}$$

Moreover, we transform Eq. (17) as Eq. (18), then $\beta_{\rho+1}$ and $e_{\rho+1}$ are expressed as Eq. (19) and Eq. (20) respectively.

$$\begin{cases} \beta_{\rho+1} = \arg\min_{\beta} L_\mu(e_\rho, \beta, \upsilon_\rho) \\ e_{\rho+1} = \arg\min_{e} L_\mu(e, \beta_{\rho+1}, \upsilon_\rho) \\ \upsilon_{\rho+1} = \upsilon_\rho + \mu(Y - H\beta_{\rho+1} - e_{\rho+1}) \end{cases} \tag{18}$$

$$\beta_{\rho+1} = (H^T H + 2 / C\mu I)^{-1} H^T (Y - e_\rho + \upsilon_\rho / \mu) \tag{19}$$

$$\begin{aligned} e_{\rho+1} &= \text{shrink}(Y - H\beta_{\rho+1} + \upsilon_\rho / \mu, 1/\mu) \\ &\triangleq \max\{| Y - H\beta_{\rho+1} + \upsilon_\rho / \mu - 1/\mu, 0 |\} \\ &\circ \text{sign}(Y - H\beta_{\rho+1} + \upsilon_\rho / \mu) \end{aligned} \tag{20}$$

where, "○" denotes the element-wise multiplication.

The algorithm flow of RSBLS is described in Algorithm 1.

**Algorithm 1:** RSBLS
**Dataset:** $X = \{x_1, x_2, \ldots, x_N\}$, $x_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,d}\} \in \mathbb{R}^d$; $Y = \{y_1, y_2, \ldots, y_N\}$.
**Parameters:** the number of feature layer *V*, the feature mapping times of each layer *n*, the number of neurons in each feature mapping $L_e$; the number of enhancement layer *U*, the enhancement processing times of each layer *m*, the number of neurons in each enhancement processing $L_h$.
**Output:** $Y$.

1. Initialize $p=1$, $q=1$, $u=1$
   **Phase 1: feature nodes generation (Step 2-8)**
2. **for** $u \le U$, **do**
3.   **for** $p \le nu$, **do**
4.     Generate $W_{ep}$ and $\beta_{ep}$ randomly;
5.     Calculate $Z_p^u$ using Eq. (8);
6.   **end for**
7. **end for**
8. Set the feature mapping group $Z^u$ using Eq. (9);
   **Phase 2: enhancement nodes generation (Step 9-15)**
9. **for** $u \le U$, **do**
10.   **for** $q \le mu$, **do**
11.     Generate $W_{hq}$ and $\beta_{hq}$ randomly;
12.     Calculate $H_q^u$ using Eq. (10);
13.   **end for**
14. **end for**
15. Set the enhancement group $H^u$ using Eq. (11);
    **Phase 3: target output matrix $\hat{Y}$ determination (Step 16-21)**
16.   **for** $\rho = 1, 2, \ldots, P_{\max}$, **do**
17.     $\mu = 2N / \|Y\|_1$, $e_1 = 0$, $\upsilon_1 = 0$
18.     Compute $\beta$ by using Eq. (16)- Eq. (20);
19.   **end for**
20. Compute output $Y$ by using Eq. (13)- Eq. (15);
21. **return** $Y$.

## IV. NUMERICAL EXPERIMENTS

The related experiments of our paper are programmed based on MATLAB 2019b.

### A. Experimental Datasets

In this part, six benchmark datasets, including Concrete, Abalone, Stock, Mortgage, Treasury, and Compactiv from KEEL (http://www.keel.es/) are selected to demonstrate the feasibility of the RSBLS. Table I gives the corresponding information of these datasets.

In addition, to verify the robustness of RSBLS, the datasets are preprocessed as follows: we first carry out normalization processing, the features and corresponding labels are normalized in the range of [0, 1]. Moreover, 75% samples of the original datasets are selected as the training datasets randomly, the rest 25% samples are determined as the test datasets. In the last, 10%, 20%, 30%, 40% and 50% outliers with uniform distributed are inserted into the training datasets as Eq. (21).

$$y_{noise} = y + \mathrm{V} y_{outlier}, -0.5 \le \mathrm{V} y_{outlier} \le 0.5 \tag{21}$$

where, $y_{noise}$ expresses the contaminated training label in the range of [-0.5, 1.5]; $\mathrm{V} y_{outlier}$ means the random outlier.

TABLE I. THE ATTRIBUTES INFORMATION OF EXPERIMENTAL DATASETS

| Datasets | Features | Instances |
|---|---|---|
| Concrete | 8 | 1030 |
| Abalone | 8 | 4177 |
| Stock | 9 | 950 |
| Mortgage | 15 | 1049 |
| Treasury | 15 | 1049 |
| Compactiv | 21 | 8192 |

### B. Evaluation Indexes

To present the robustness of RSBLS, we carry out all the related algorithms 50 times independently, the average root mean square error (RMSE) (see Eq. (22)) of experimental results are recorded as the evaluation indexes.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - f_i\right)^2} \tag{22}$$

where, $y_i$ indicates the actual value of sample *i*; $f_i$ represents the output results of sample *i*; *N* denotes the number of samples.

### C. Experimental Results

*1) Parameters settings:* To illustrate the proposed RSBLS, RVFL [1, 2], BLS [3-5], Huber-WBLS [23], KED-WBLS [23] and RBLS [24] are chosen as the contrast algorithms, among them, we use the sigmoidal function (see Eq. (23)) as activation function of all the compared models.

$$S(x) = \frac{1}{1+e^{-x}} \tag{23}$$

Some key parameters of RSBLS, RVFL, BLS, Huber-WBLS, KED-WBLS and RBLS are listed as follows:

RVFL: the hidden layer nodes are selected from {50, 100, 150, 200}, the weights and biases are generated randomly in the range of [-1,1] and [0,1] respectively.

BLS, Huber-WBLS, KED-WBLS and RBLS: the feature mappings and feature nodes are chosen from {5, 10, …, 45, 50}, and the enhancement nodes are selected from {50, 100, 150, 200}. Moreover, these models adopt $\ell_2$ regularization technique and the regularization parameter *C* are chosen from $\{2^{-10}, 2^{-5}, 0, \ldots, 2^{15}, 2^{20}\}$. Some other models are set as the same as the original references.

RSBLS: the number of layers is set as 2; the feature mappings and feature nodes of each layer are chosen from {5, 10, …, 45, 50}, and the enhancement nodes of each layer are selected from {50, 100, 150, 200}. Moreover, these models adopt $\ell_2$ regularization technique and the regularization parameter $C$ are chosen from $\{2^{-10}, 2^{-5}, 0, \dots, 2^{15}, 2^{20}\}$.

*2) Results and discussion:* In this section, we give the performance evaluation of RSBLS, Table II to Table VII show the average test RMSE results of RSBLS compared with different models on six regression problems with uniform distributed outliers. As it can be seen from Table II to Table VII, with the increase of contamination rates, the performance of RVFL and BLS gradually become worse, Huber-WBLS, KED-WBLS and RBLS can improve the robustness of BLS.

TABLE II. PERFORMANCE COMPARISON OF DIFFERENT MODELS ON CONCRETE DATASET

| Models | Test Performance (RMSE) at different contamination rates | | | |
|---|---|---|---|---|
| | 10% | 20% | 30% | 40% |
| RVFL | 0.0934 | 0.0970 | 0.1062 | 0.1065 |
| BLS | 0.0957 | 0.0981 | 0.0983 | 0.0948 |
| Huber-WBLS | 0.0890 | 0.0893 | 0.0938 | 0.0914 |
| KED-WBLS | 0.0936 | 0.0883 | 0.0933 | 0.0914 |
| RBLS | 0.0889 | 0.0841 | 0.0957 | 0.0953 |
| RSBLS | **0.0781** | **0.0783** | **0.0834** | **0.0890** |

TABLE III. PERFORMANCE COMPARISON OF DIFFERENT MODELS ON ABALONE DATASET

| Models | Test Performance (RMSE) at different contamination rates | | | |
|---|---|---|---|---|
| | 10% | 20% | 30% | 40% |
| RVFL | 0.0745 | 0.0743 | 0.0810 | 0.0782 |
| BLS | 0.0742 | 0.0749 | 0.0781 | 0.0756 |
| Huber-WBLS | 0.0723 | 0.0736 | 0.0777 | 0.0732 |
| KED-WBLS | 0.0736 | 0.0747 | 0.0776 | 0.0744 |
| RBLS | 0.0727 | 0.0738 | 0.0778 | 0.0736 |
| **RSBLS** | **0.0719** | **0.0732** | **0.0764** | **0.0720** |

TABLE IV. PERFORMANCE COMPARISON OF DIFFERENT MODELS ON STOCK DATASET

| Models | Test Performance (RMSE) at different contamination rates | | | |
|---|---|---|---|---|
| | 10% | 20% | 30% | 40% |
| RVFL | 0.0456 | 0.0549 | 0.0584 | 0.0633 |
| BLS | 0.0425 | 0.0520 | 0.0567 | 0.0577 |
| Huber-WBLS | 0.0311 | 0.0349 | 0.0392 | 0.0460 |
| KED-WBLS | 0.0355 | 0.0358 | 0.0416 | 0.0428 |
| RBLS | 0.0320 | 0.0380 | 0.0394 | 0.0409 |
| **RSBLS** | **0.0301** | **0.0340** | **0.0332** | **0.0342** |

TABLE V. PERFORMANCE COMPARISON OF DIFFERENT MODELS ON MORTGAGE DATASET

| Models | Test Performance (RMSE) at different contamination rates | | | |
|---|---|---|---|---|
| | 10% | 20% | 30% | 40% |
| RVFL | 0.0275 | 0.0351 | 0.0527 | 0.0580 |
| BLS | 0.0293 | 0.0416 | 0.0518 | 0.0622 |
| Huber-WBLS | 0.0088 | 0.0124 | 0.0236 | 0.0333 |
| KED-WBLS | 0.0096 | 0.0111 | 0.0147 | 0.0188 |
| RBLS | 0.0052 | 0.0070 | 0.0057 | 0.0063 |
| **RSBLS** | **0.0050** | **0.0059** | **0.0062** | **0.0062** |

TABLE VI. PERFORMANCE COMPARISON OF DIFFERENT MODELS ON TREASURY DATASET

| Models | Test Performance (RMSE) at different contamination rates | | | |
|---|---|---|---|---|
| | 10% | 20% | 30% | 40% |
| RVFL | 0.0248 | 0.0373 | 0.0427 | 0.0507 |
| BLS | 0.0280 | 0.0310 | 0.0484 | 0.0538 |
| Huber-WBLS | 0.0126 | 0.0125 | 0.0195 | 0.0293 |
| KED-WBLS | 0.0125 | 0.0121 | 0.0184 | 0.0217 |
| RBLS | 0.0119 | 0.0114 | 0.0117 | 0.0100 |
| **RSBLS** | **0.0114** | **0.0103** | **0.0109** | **0.0090** |

TABLE VII. PERFORMANCE COMPARISON OF DIFFERENT MODELS ON COMPACTIV DATASET

| Models | Test Performance (RMSE) at different contamination rates | | | |
|---|---|---|---|---|
| | 10% | 20% | 30% | 40% |
| RVFL | 0.0404 | 0.0421 | 0.0451 | 0.0435 |
| BLS | 0.0287 | 0.0287 | 0.0291 | 0.0312 |
| Huber-WBLS | 0.0256 | 0.0250 | 0.0252 | 0.0278 |
| KED-WBLS | 0.0270 | 0.0265 | 0.0262 | 0.0290 |
| RBLS | 0.0249 | 0.0250 | 0.0252 | 0.0279 |
| **RSBLS** | **0.0242** | **0.0243** | **0.0238** | **0.0248** |

At the same time, it is obviously that RSBLS with 2 hidden layers has a better robustness compared with Huber-WBLS, KED-WBLS and RBLS. Our proposed model, RSBLS with $\ell_1$-norm loss function gains the best mean RMSE on 6 datasets, which demonstrates the effectiveness of regularization method. In addition, the uncertain data regression performance of those models on different datasets with different contamination rates indicates the strong robustness of RSBLS, at different levels of outliers; the RSBLS shows the best consistency.

In summary, RSBLS with $\ell_1$ norm loss function can solve uncertain data regression with uniform distributed outliers effectively; the stacked deep model is helpful to enhance the robustness of BLS well.

## V. Conclusion

In the paper, we propose a novel robust stacked broad learning system model with multi-layers for solving uncertain data regression problem, named as RSBLS. In the proposed RSBLS, we expand BLS into a stacked deep model with multi-layer of feature nodes and enhancement nodes, which can helpful to extract deep representation information. In addition, $\ell 1$-norm function is introduced to calculate the output weights of RSBLS, which can process noisy data and ensure learning efficiency of hierarchical model. Experimental results on some regression datasets with different ratios of noisy shows that, RSBLS has better robustness compared with RVFL, BLS, Huber-WBLS, KDE-WBLS and RBLS.

In the future, as some parameters should be selected by grid search which limits the search scope, some of the latest swarm intelligence algorithms can be used to choose the parameter of RBLS.

## Acknowledgment

Science and Technology Foundation of Guizhou Province (QKHZK [2023] YB450); Zunyi Technology and Big Data Bureau, Moutai Institute Joint Science and Technology Research and Development Project (ZSKHHZ [2021] 316).

## References

[1] Y. Pao and Y. Takefuji, “Functional-link net computing: Theory, system architecture, and functionalities,” Computer, vol. 25, no. 5, pp. 76-79, May 1992.

[2] Y. Pao, G. Park and D. J. Sobajic, “Learning and generalization characteristics of the random vector functional-link net,” Neurocomputing, vol. 6, no. 2, pp. 163-180, April 1994.

[3] C. L. P. Chen and Z. Liu, “Broad learning system: An effective and efficient incremental learning system without the need for deep architecture,” IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 1, pp. 10-24, January 2018.

[4] C. L. P. Chen, Z. Liu and S. Feng, “Universal approximation capability of broad learning system and its structural variations,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 4, pp. 1191-1204, April 2019.

[5] C. L. P. Chen and Z. Liu, “Broad learning system: A new learning paradigm and system without going deep,” 2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Hefei, China, 2017, pp. 1271-1276.

[6] S. Feng and C. L. P. Chen, “Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification,” IEEE Transactions on Cybernetics, vol. 50, no. 2, pp. 414-424, February 2020.

[7] H. Guo, B. Sheng, P. Li and C. L. P. Chen, “Multiview high dynamic range image synthesis using fuzzy broad learning system,” IEEE Transactions on Cybernetics, vol. 51, no. 5, pp. 2735-2747, May 2021.

[8] R. Ali, B. Sheng, P. Li, Y. Chen, H. Li, P. Yang, et al., “Optic disk and cup segmentation through fuzzy broad learning system for glaucoma screening,” IEEE Transactions on Industrial Informatics, vol. 17, no. 4, pp. 2476-2487, April 2021.

[9] S. Feng, C. L. P. Chen, L. Xu and Z. Liu, “On the accuracy–complexity tradeoff of fuzzy broad learning system,” IEEE Transactions on Fuzzy Systems, vol. 29, no. 10, pp. 2963-2974, October. 2021.

[10] H. Han, Z. Liu, H. Liu, J. Qiao and C. L. P. Chen, “Type-2 fuzzy broad learning system,” IEEE Transactions on Cybernetics, vol. 52, no. 10, pp. 10352-10363, October. 2022.

[11] M. Xu, M. Han, C. L. P. Chen and T. Qiu, “Recurrent broad learning systems for time series prediction,” IEEE Transactions on Cybernetics, vol. 50, no. 4, pp. 1405-1417, April 2020.

[12] M. Han, S. Feng, C. L. P. Chen, M. Xu and T. Qiu, “Structured manifold broad learning system: A manifold perspective for large-scale chaotic time series analysis and prediction,” IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 9, pp. 1809-1821, September. 2019.

[13] H. Zhao, J. Zheng, W. Deng and Y. Song, “Semi-supervised broad learning system based on manifold regularization and broad network,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 3, pp. 983-994, March 2020.

[14] S. Huang, Z. Liu, W. Jin and Y. Mu, “Broad learning system with manifold regularized sparse features for semi-supervised classification,” Neurocomputing, vol. 463, pp. 133-143, November 2021.

[15] L. Xu, C. L. P. Chen, R. Han, Graph-based sparse bayesian broad learning system for semi-supervised learning, Information Sciences, vol. 597, pp. 193-210, June 2022.

[16] D. Liu, S. Baldi, W. Yu, J. Cao and W. Huang, “On training traffic predictors via broad learning structures: A benchmark study,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 52, no. 2, pp. 749-758, February 2022.

[17] Y. Chu, H. Lin, L. Yang, S. Sun, Y. Diao, C. Min, et al, “Hyperspectral image classification with discriminative manifold broad learning system,” Neurocomputing, vol. 442, pp. 236-248, June 2021.

[18] Y. Yang, Z. Gao, Y. Li, Q. Cai, N. Marwan and J. Kurths, A complex network-based broad learning system for detecting driver fatigue from EEG signals, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 9, pp. 5800-5808, September. 2021.

[19] Z. Gao, W. Dang, M. Liu, W. Guo, K. Ma and G. Chen, Classification of EEG signals on VEP-based BCI systems with broad learning, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 11, pp. 7143-7151, November 2021.

[20] S. Issa, Q. Peng and X. You, “Emotion classification using EEG brain signals and the broad learning system,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 12, pp. 7382-7391, December 2021.

[21] T. Zhang, X. Gong and C. L. P. Chen, “BMT-Net: Broad multitask transformer network for sentiment analysis,” IEEE Transactions on Cybernetics, vol. 52, no. 7, pp. 6232-6243, July 2022.

[22] T. Zhang, X. Wang, X. Xu and C. L. P. Chen, “GCB-Net: Graph convolutional broad network and its application in emotion recognition,” IEEE Transactions on Affective Computing, vol. 13, no. 1, pp. 379-388, January-March 2022.

[23] F. Chu, T. Liang, C. L. P. Chen, X. Wang and X. Ma, “Weighted broad learning system and its application in nonlinear industrial process modeling,” IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 8, pp. 3017-3031, August. 2020.

[24] Y. Zheng, B. Chen, S. Wang and W. Wang, “Broad learning system based on maximum correntropy criterion,” IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 7, pp. 3083-3097, July 2021.

[25] L. Liu, L. Cai, T. Liu, C. L. P. Chen, X. Tang, “Cauchy regularized broad learning system for noisy data regression,” Information Sciences, vol. 603, pp. 210-221, July 2022.

[26] J. Jin, C. L. P. Chen and Y. Li, “Robust Broad Learning System for Uncertain Data Modeling,” 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 2018, pp. 3524-3529.

[27] W. Jin and C. L. P. Chen, “Regularized robust broad learning system for uncertain data modeling,” Neurocomputing, vol. 322, pp. 58-69, December 2018.

[28] S. Feng, W. Ren, M. Han and Y. Chen, “Robust manifold broad learning system for large-scale noisy chaotic time series prediction: A perturbation perspective,” Neural Networks, vol. 117, pp. 179-190, September 2019.

[29] Y. Guo, X. Yang, Y. Wang, F. Wang and B Chen, “Online robust echo state broad learning system,” Neurocomputing, vol. 464, pp. 438-449, November 2021.

[30] Z. Liu, C. L. P. Chen, S. Feng, Q. Feng and T. Zhang, “Stacked broad learning system: From incremental flatted structure to deep model,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 1, pp. 209-222, January 2021.

[31] Q. Zhou and X. He, “Broad learning model based on enhanced features learning,” IEEE Access, vol. 7, pp. 42536-42550, March 2019.

[32] V. Chauhan and A. Tiwari, “On the construction of hierarchical broad learning neural network: An alternative way of deep learning,” 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 2018, pp. 182-188.

[33] C. Zhang, S. Ding, L. Guo, J. Zhang, “Broad learning system based ensemble deep model,” Soft Computing, vol. 26, no. 7029-7041, August 2022.

[34] S. Cai, L. Zhang, W. Zuo and X. Feng, “A probabilistic collaborative representation based approach for pattern classification,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2950-2959.

[35] F. Cao, H. Ye and D. Wang, “A probabilistic learning algorithm for robust modeling using neural networks with random weights,” Information Sciences, vol. 313, pp. 62-78, August 2015.