Efficient Ensemble Broad Learning System Based on Dropout and Dropconnect

Yiwan Cao, Fei Chu, Shuai Li, C. L. Philip Chen, and Jun Fu

Abstract—**Broad learning system (BLS) is an emerging neural network characterized by its rapid processing and robust generalization capabilities. However, determining the appropriate structure for broad learning system is also a challenge. In addition, broad learning system may perform overfitting due to the dependence between nodes in processing fully connected network. To deal with these problems, an efficient ensemble broad learning system based on Dropout and Dropconnect is proposed in this paper. The proposed Dropout ensemble broad learning system randomly discards hidden nodes to improve diversity between individuals and reduce the synergy between nodes to improve prediction stability. The Dropconnect ensemble broad learning system randomly drops connection weights to generate more complementary models by adding input attribute disturbance. The experimental results on the UCI datasets confirm that the method proposed in this paper can solve the problem of model overfitting caused by the strong dependence between the nodes of ensemble broad learning system. The proposed algorithm outperforms the original BLS in terms of prediction stability and classification accuracy.**

Index Terms—**Broad learning system, ensemble learning, Dropout, Dropconnect**

I. Introduction

Broad learning system (BLS) is an efficient algorithm for training single hidden layer feedforward neural network, whose output layer connection weight is determined by analysis method [1]. BLS has emerged as a prominent area in the field of machine learning and has been successfully applied to the fields of prediction, classification, and regression due to its strong generalization ability [2–8].

To get better generalization performance, ensemble learning is introduced into BLS [9, 10]. Ensemble broad learning system (ENBLS) has been studied by many scholars. To address the shortcomings of aviation engine wear fault diagnosis based on oil analysis, Wang et al. [11] proposed a bagging-BLS which was established by combining BLS and bagging through the ensemble learning. Fan and Zhang [12] proposed the lncRNA-protein interactions with a broad learning system (LPI-BLS), which used a stacked ensemble broad learning system classifier with logistic regression model to predict the interaction between lncRNA and proteins. Zhu et al. [13] proposed the broad learning system with ensemble and classification (BLS-EC) algorithm, which combined BLS with ensemble learning to predict multi-step forward wind speed and enhance the stability of the network. Yu et al. [14] proposed an ensemble BLS based on progressive kernels, which reduced the impact of noise on the model by combining multiple randomly mapping feature spaces with kernel spaces. An incremental weighted ensemble broad learning system was proposed by Yang et al. [15], which utilized a progressive mechanism to improve the stability and robustness of the system. However, these ensemble methods use all models and each submodel is fully connected with the output layer, which may increase the synergy between nodes and the dependence of the model on special local features. This leads to redundancy of the whole ensemble model, which may reduce the generalization of model.

Deep networks have strong learning ability due to their large number of complex parameters, but they are easy to produce overfitting, and the training of large and complex network is quite time-consuming [16, 17]. Dropout is a new technology to solve this problem proposed by Hinton et al. [18]. Its core concept is to randomly delete some hidden layer nodes and their connections from the neural network in the network training process. Its motivation is to prevent overfitting caused by mutual adaptation between network hidden layer nodes. Dropconnect is also a new regularization method for fully connected network [19]. Unlike Dropout, it deletes weights instead of nodes. It is also an efficient method to reduce model overfitting through model averaging.

The existing research results indicate that Dropout can improve the learning performance of neural networks in many domains, such as computer vision, speech recognition, text classification, and so on [20]. Starting from Bayesian theory, Gal and Ghahramani [21] explained Dropout as Bayesian approximate inference in a deep Gaussian process. DeVries and Taylor [20] proposed the cutout method, which mainly applied Dropout directly to the input image, essentially removing the visual features with high activation values in the subsequent layers of convolutional neural network (CNN). This method of applying the mask to the input image can achieve the same performance with lower execution costs. Yang et al. [22] proposed a training method of deep

Manuscript received: 3 November 2023; revised: 15 March 2024; accepted: 24 April 2024. *(Corresponding author: Fei Chu.)*

Citation: Y. Cao, F. Chu, S. Li, C. L. P. Chen, and J. Fu, Efficient ensemble broad learning system based on Dropout and Dropconnect, *IJICS*, 2024, 29(2), 79–87.

Yiwan Cao, Fei Chu, and Shuai Li are with School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China (e-mail: caoyiwan613@163.com; chufei@cumt.edu.cn; lishuai19970910@163.com).

C. L. Philip Chen is with School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: philip.chen@ieee.org).

Jun Fu is with State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110004, China (e-mail: junfu@mail.neu.edu.cn).

Digital Object Identifier 10.62678/IJICS202403.10110

convolution neural network for large datasets by applying the idea of Dropout to train samples and applied it to Chinese character recognition. Excitation Dropout was proposed to encourage plasticity by dropping high-saliency neurons in deep neural networks [23]. Zhao et al. [24] proposed adaptive drop algorithms by adaptively adjusting the retention probability for training spiking neural network (SNN). Achille and Soatto [25] proposed to minimize the loss function by using information Dropout, which was a generalization of Dropout that made better use of limited capacity and adapted to the data. Biased Dropout was proposed and used in extreme learning machine to prevent overfitting by utilizing the difference of neurons and connection weights to keep more information [26]. Visser and Zumel [27] compared the differences in model regularization between Dropout and Dropconnect, and evaluated their regularization effects on shallow networks. Biased Dropconnect was proposed, which divided connection weights into high and low groups by setting a threshold and provided different Dropout probabilities to enhance sparsity and reduce complexity [26]. Structured Dropconnect was proposed to model the output distribution of a network for uncertainty inference in image classification [28]. Mobiny et al. [29] used Monte Carlo (MC)-Dropconnect to represent the model uncertainty by imposing a Bernoulli distribution on the model weights.

Inspired by the Dropout and Dropconnect techniques, Dropout ensemble broad learning system (DEBLS) and Dropconnect ensemble broad learning system (DCEBLS) are proposed for data classification in this paper. Unlike the use of Dropout in Ref. [30], the Dropout technique is used to generate efficient DEBLS classifier by Dropout hidden nodes to weaken even eliminate the joint adaptability between neuron nodes and enhance the generalization ability. However, DEBLS directly discards nodes, which may cause information loss. To further enhance the performance of model, DCEBLS introduces Dropconnect to randomly remove weights, which can retain the weights of some important nodes. Different from DEBLS, DCEBLS randomly drops the weights during training, resulting in sparse connections and obtaining subnetworks with different connection weights. The potential implication is to randomly set the input weights of the hidden layer to 0 instead of the output of this layer. In summary, the main contributions of the proposed methods are as follows:

(1) DEBLS is proposed to discard nodes randomly and ensemble submodels with different hidden layer structures in the training process. DEBLS can weaken the dependence between nodes and reduce the overfitting risk of the final model.

(2) DCEBLS is proposed to drop the connection weights rather than nodes to retain important features in the training process. DCEBLS generates different individuals from different perspectives of the sample by adding input attribute perturbations, and increases the generalization ability of the model.

The rest of this article is organized as follows. Section II briefly reviews the preliminary information related to our work. Section III introduces the proposed algorithm. Statistical and analytical analyses of experimental results are detailed in Section IV. Finally, Section V provides a summary of this article.

II. Related Work

A. Brief Review of Broad Learning System

Broad learning system is a new variant of random vector function link neural network (RVFLNN) [31], an effectively simpler network structure without deep architecture [32]. The structure of BLS includes four layers: input layer, feature layer, enhancement layer, and output layer. The input layer includes the input data $X \in \mathbb{R}^{N\times M}$ and the output layer includes the output matrix $Y \in \mathbb{R}^{N\times C}$, where N is the number of samples and M and C denote the dimensions, respectively.

The feature layer is composed of mapping features by projecting the data as follows [1]

$$Z_i = \phi(XW_{ei} + \beta_{ei}),\ i = 1,2,...,n \tag{1}$$

where Z_i is the i-th group of feature nodes, W_{ei} is the feature weight, β_{ei} is the bias of feature nodes, and they are randomly obtained. The function $\phi(\cdot)$ is sigmoid mapping function as $\phi(x) = (1+\mathrm{e}^{-x})^{-1}$, where e represents the natural constant. Denote all the feature nodes groups as $Z^n \equiv [Z_1, Z_2, ..., Z_n]$. There are m groups of enhancement nodes and each group has r nodes. The j-th group of enhancement nodes H_j is denoted as

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

where W_{hj} and β_{hj} are random enhancement weight and bias of enhancement nodes with proper dimensions, respectively. The function $\xi(\cdot)$ is the activation function of enhancement nodes. It can be various, but generally uses the same form. All enhancement node groups are denoted as $H^m \equiv [H_1, H_2, \ldots, H_m]$. The final form of the broad model can be considered as

$$Y = [Z^n | H^m] W^m = AW^m = AW \tag{3}$$

where A is the expanded input matrix, W^m is the output connecting weight and is calculated by the ridge regression of the objective function of this model. W^m can be abbreviated as W. The calculation equation is as follows

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

where $\|Y - AW\|_2^2$ is the error term and $\|W\|_2^2$ is the L2-norm regularization used to avoid overfitting. The positive λ is the regularization constraint value. The closed-form solution can be compressed by the ridge regression as follows

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

where I is the identity matrix. More details of the original BLS can be found in Ref. [1].

B. Dropout

This section briefly introduces the basic idea of Dropout through simple linear networks whose output is the weighted sum of the inputs [33, 34]. Given the input vector $\hat{I} = (\hat{I}_1, \hat{I}_2, ..., \hat{I}_s)$, the output of the linear neuron is

$O(\hat{I}) = \sum_{i=1}^{s} w_i \hat{I}_i$, where s represents the dimension of the input vector $\hat{I}$ and w_i is the i-th output weight of the neural network. The error can be expressed as $E = \left(t - \sum_{i=1}^{s} w_i \hat{I}_i\right)^2$, where t represents the actual output value. By introducing Dropout, there are 2^s possibilities to get a subnet (including empty network) by deleting $\hat{I}_i$ $(1 \leqslant i \leqslant s)$ with equal probability 0.5 [30]. The average output of all subnetworks is [35]

$$E(O) = \frac{1}{2^s} \sum_{S} O(\hat{I}) \tag{6}$$

where S is the number of subnetworks. The error of S modes is expressed as $E_s = \left(t - \sum_{i=1}^{s} w_i \delta_i \hat{I}_i\right)^2$, where δ_i $(1 \leqslant i \leqslant s)$ is a random variable that obeys the Bernoulli distribution and is independent of each other. p_i is the probability of a binary vector from 0 to 1 appearing as 1, $p_i = p(\delta_i = 1)$ and q_i is the probability of a binary vector from 0 to 1 appearing as 0, $q_i = p(\delta_i = 0) = 1 - p_i$.

According to the linear property of mathematical expectation, Eq. (7) can be obtained

$$E(O) = \sum_{i=1}^{s} w_i E(\delta_i) \hat{I}_i = \sum_{i=1}^{s} w_i p_i \hat{I}_i \tag{7}$$

Applied to neural networks, the feature nodes of a neural network can be expressed as $H = \hat{m} \cdot f(WX + \beta)$, where β is the bias, $\hat{m}$ is a binary mask vector with each element being equal to 1 with probability p, and $f(\cdot)$ represents the activation function, which can be expressed in different forms. More details of the Dropout can be found in Ref. [30].

C. Dropconnect

Similar to Dropout, the introduction of dynamic sparsity during neural network training is a characteristic of Dropconnect, with the only difference being that the sparsity of Dropconnect acts on weights rather than nodes in the hidden layer [28]. The main idea of Dropconnect can be expressed as follows [36]

$$H = f(X(W \circ \hat{M}) + \beta) \tag{8}$$

where $\circ$ denotes the Hadamard (elements-wise) product, X is the input, W is the traditional connecting weight, and β is the bias, respectively. $\hat{M}$ is a binary mask matrix with each element being equal to 1 with probability p, which is drawn independently from the Bernoulli distribution.

In the testing phase, the weight of each input (each hidden layer node is connected with multiple inputs) is sampled by Gaussian distribution [19]. Of course, the mean and variance of the Gaussian distribution are related to the previous probability value p. The input weight that follows the Gaussian distribution is $\hat{w} \sim N(\mu, \sigma^2)$, where $\mu = pWX$ and $\sigma^2 = p(1-p)(W \circ W)(X \circ X)$. More details of the Dropconnect can be found in Ref. [19].

D. Combination Strategy

Ensemble learning method improves generalization performance by combining a set of base learners instead of selecting the best one [37]. This shows that an appropriate combination method used in ensemble learning is crucial. A good combination method can reduce the risk of false selection of hypothesis, reduce the risk of false selection of local optimal solution assumption, and obtain a more accurate approximation of the real unknown hypothesis. In this paper, the absolute majority voting method is used as the combination strategy of the basic learner. In this method, each classifier votes for a category label, and the final output category is marked as the label with more than half of the votes. If all labels get less than half of the votes, the prediction is rejected. The method of obtaining the output result $\hat{Y}$ can be expressed as [38]

$$\hat{Y} = \begin{cases} c_j, \text{ if } \sum_{i}^{L} Y_i^j > 0.5 \times \sum_{\hat{f}=1}^{C} \sum_{i=1}^{L} Y_i^{\hat{f}}; \\ \text{reject, otherwise} \end{cases} \tag{9}$$

where $Y_i^{\hat{f}}$ represents the output predicted by the i-th submodel of the $\hat{f}$-th category, Y_i^j represents the output of Y_i on category mark c_j, and "reject" indicates that the number of votes cast is less than half and the model rejects the prediction result. In other words, if the prediction results of L classifiers for class j are greater than half of the total voting results, they are predicted as class j, otherwise, they are rejected. More details of the combination strategy can be found in Ref. [38].

III. Dropout and Dropconnect Ensemble Broad Learning Systems

A. Dropout Ensemble Broad Learning System

This section introduces the proposed method of Dropout ensemble broad learning system. For a given practical problem, it can select appropriate network structure and improve the accuracy of data classification. Based on a large redundant BLS network, this method trains multiple BLSs by discarding some hidden layer nodes, and uses them as the basic classifier of ensemble. The absolute majority voting method is used in DEBLS to combine the results of multiple BLS submodels for data classification [39]. The basic idea of the algorithm is shown in Fig. 1, and the pseudo code of the algorithm is shown in Algorithm 1.

The generation process of feature nodes and enhancement nodes is the same as before, which is obtained by random weight mapping. The corresponding hidden layer node matrix can be obtained as follows

$$A = [Z^n | H^m] \tag{10}$$

After obtaining A, l groups of vectors p_i conforming to Bernoulli distribution are randomly generated, where $p_i \sim \text{Bernoulli}(p)$.

To reduce the co-adaptability between nodes, different hidden layer node matrices are obtained by

$$A_i = A \circ p_i, \ \ 1 < i < l \tag{11}$$

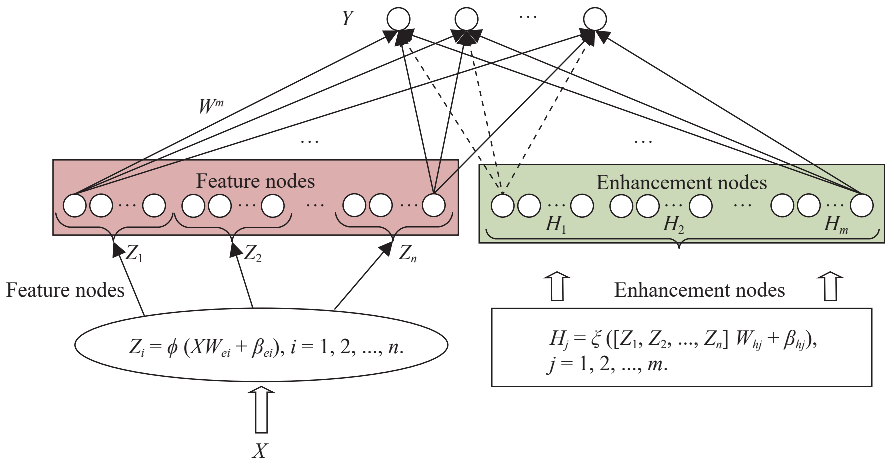


Figure 1 Structure of Dropout ensemble broad learning system.

Algorithm 1 Dropout ensemble broad learning system

Input: training samples (X, Y);

Output: output weights $W_1, W_2, ..., W_l$;

Initialization parameters: n groups of feature nodes, m enhancement nodes, the number of individual l, and nodes retention probability p_{n};

1: Calculate the mapping feature $Z^n \equiv [Z_1, Z_2, ..., Z_n]$ and the enhancement feature $H^m \equiv [H_1, H_2, ..., H_m]$;

2: Calculate l binary mask vectors with retention probability p_{n};

3: Calculate the corresponding extended matrix $A_i = A \circ p_i$;

4: Calculate output weights $W_1, W_2, ..., W_l$ of DEBLS by Eq. (12);

5: Calculate final prediction $Y = \max\limits_{1 \leqslant i \leqslant l}\{Y_i\}$, where $Y_i = W_i A_i$.

---

where l is the number of individual. The output weights can be expressed as

$$W_i = \left(A_i{}^{\mathrm{T}} A_i + \lambda I\right)^{-1} A_i{}^{\mathrm{T}} Y \tag{12}$$

The final ensemble model output is $Y = \max\limits_{1 \leqslant i \leqslant l}\{Y_i\}$, where $Y_i = W_i A_i$.

B. Dropconnect Ensemble Broad Learning System

Dropconnect ensemble broad learning system is proposed by randomly droping connecting weights to make sure that each neuron is not linked to the same weights, so that model can get better performance. In DCEBLS, Dropconnect is applied to the process of randomly generating feature nodes and enhancement nodes, generating different hidden layer nodes with implicit diversity through different connection weights.

After applying Dropconnect, the equation of the i-th group of feature nodes is

$$Z_i = \phi\left(X(W_{ei} \circ \hat{M}_{ei}) + \beta_{ei}\right), \quad i = 1, 2, ..., n \tag{13}$$

where W_{ei} is the feature weight, $\hat{M}_{ei}$ is a binary mask matrix with each element being equal to 1 with probability p and equal to 0 with probability $1-p$, β is the bias, and $\phi(\cdot)$ is a mapping function for feature nodes. $\hat{M}_{ei}$ is different for each feature nodes group, so that $Z^n = [Z_1, Z_2, ..., Z_n]$ is different for each submodel and Z_i is different in Z^n. The j-th group of enhancement nodes is denoted as

$$H_j = \xi\left(Z^n(W_{hj} \circ \hat{M}_{hj}) + \beta_{hj}\right), \quad j = 1, 2, ..., m \tag{14}$$

where W_{hj} is the enhancement connecting weight, $\hat{M}_{hj}$ is a random binary mask matrix with keeped probability p, β_{hj} is random bias of enhancement nodes with proper dimensions, and $\xi(\cdot)$ is the activation function of enhancement nodes. It can be various, but usually uses the same form. Hidden layer nodes groups can be represented as $A = [Z^n | H^m]$. In each submodel, Z^n and H^m are different, so there is implicit diversity among individuals. The output weights of each model are as follows

$$W_i = \left(A_i{}^{\mathrm{T}} A_i + \lambda I\right)^{-1} A_i Y \tag{15}$$

The final output is $Y = \max\limits_{1 \leqslant i \leqslant l}\{Y_i\}$, where $Y_i = W_i A_i$. The basic idea of DCEBLS can be represented in Fig. 2, and the pseudo code of the algorithm is shown in Algorithm 2.

IV. Experiment

To verify the performance of the proposed methods (DEBLS and DCEBLS), some experiments are designed. In the experiments, some public datasets are used to prove the effectiveness by comparing the classification ability of the proposed methods with the standard BLS and the ensemble broad learning system. The datasets come from UCI machine learning library [40], and the details are shown in Table 1. The simulation is carried out in MATLAB R2021b, running on the same Windows 10 machine with 8 GB memory and i5-8400 (2.80 GHz).

A. Evaluation Metric

To evaluate the proposed algorithms comprehensively, the following evaluation indices are adopted. The four indicators are accuracy (ACC), precision (PRE), recall (REC), and F1 score [41]. The accuracy is the ratio of the number of correctly classified samples to the total number of samples. Precision and recall are two commonly used measures in the field of statistical classification. F1 value is the harmonic average value based on precision and recall. A higher F1 value represents a better comprehensive performance of the

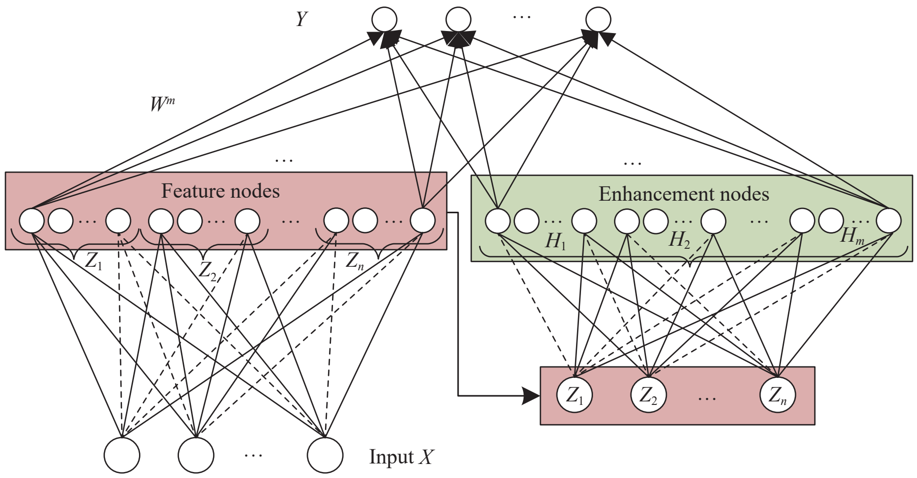


Figure 2 Structure of Dropconnect ensemble broad learning system.

Algorithm 2 Dropconnect ensemble broad learning system

Input: training samples (X, Y);

Output: output weights $W_1, W_2, ..., W_l$;

Initialization parameters: n groups of feature nodes, m enhancement nodes, the number of individual l, and weights retention probability p_{w};

1: Calculate binary mask matrix with probability p_{w} for feature and enhancement nodes groups;

2: Calculate the mapping feature $Z^n \equiv [Z_1, Z_2, ..., Z_n]$ by Eq. (13);

3: Calculate the enhancement feature $H^m \equiv [H_1, H_2, ..., H_m]$ by Eq. (14);

4: Calculate the corresponding extended matrix A_i;

5: Calculate output weights $W_1, W_2, ..., W_l$ of DCEBLS by Eq. (15);

6: Calculate final prediction $Y = \max_{1 \leqslant i \leqslant l} \{Y_i\}$, where $Y_i = W_i A_i$.

Table 1 UCI dataset description.

Dataset	Sample	Attribute
Australian	690	14
Default	30,000	24
Heart	270	13
Sonar	208	60
Spectf	267	22
Zoo	101	16

algorithm. These four metrics are chosen to evaluate the proposed approaches and their definitions are as follows

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{16}$$

$$\mathrm{PRE} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{17}$$

$$\mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{18}$$

$$\mathrm{F1} = \frac{2 \times \mathrm{PRE} \times \mathrm{REC}}{\mathrm{PRE} + \mathrm{REC}} \tag{19}$$

where TP and TN are the numbers of true predicted positive samples and true predicted negative samples, FP and FN are the numbers of false predicted positive samples and false predicted negative samples, respectively.

B. Result and Discussion

In the experiments, the standard BLS and ENBLS are used to compare with DEBLS and DCEBLS. To better illustrate the effectiveness of the algorithms, the above four performance indices and training time are compared. For structure parameters, including the numbers of feature groups, the feature nodes in each group, and the enhancement nodes, the search range is $[1, 10] \times [1, 40] \times [1, 400]$. The regularization parameters are selected in the range of $\left[2^{10}, 2^5, 2^4, ..., 2^{-4}, 2^{-5}, 2^{-10}, 2^{-20}, 2^{-30}\right]$. The number of individuals in all ensembles is set as $l = 10$. In addition, the corresponding weights and bias are randomly generated from the uniform interval $[-1, 1]$.

As an important parameter of DEBLS and DCEBLS, the retention probability has a great impact on the generalization performance. To clarify the impact of retention probability on DEBLS and DCEBLS, on Australian and Heart datasets, the changed retention probability p is used to observe the change of classification accuracy. In this experiment, the feature node groups, the number of feature nodes in each group, and the number of enhancement nodes are fixed to 10, 10, and 100, respectively. The regularization parameter is fixed as 2^{-2}, and the selection range of retention probability is [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. The rest setting including activation function is the same as that of standard BLS.

The influence of retention probability on the accuracy of DEBLS and DCEBLS is shown in Fig. 3. As shown in Fig. 3, different retention probabilities make the accuracy of the model change greatly. In both datasets, DEBLS and DCEBLS achieve the optimal accuracy when $p = 0.5$. It is more obvious from Fig. 3 that the model performance is poor at $p = 0.1$ due to too little retained information. With the increase of retention probability, the model accuracy first increases and then decreases. To achieve better generalization performance of the model, in the next experiment, the retention probability p is fixed to 0.5.

To better demonstrate the best results of several algorithms, we display them in bold and analyze the performance of each algorithm based on the experimental information in Table 2. Overall, the proposed algorithms perform well on all datasets. Compared with BLS, DEBLS has the best accuracy improvement in Sonar dataset, with an increase of 8.22% and an increase of 1.37% compared with ENBLS. At the same time, compared with BLS, the comprehensive index F1 value also increases by 5.63%. Other indicators also have different degrees of improvement on different datasets. For instance, on Spectf dataset, the accuracy of DEBLS is 0.7125, which is 2.50% and 1.25% higher than that of BLS and ENBLS,

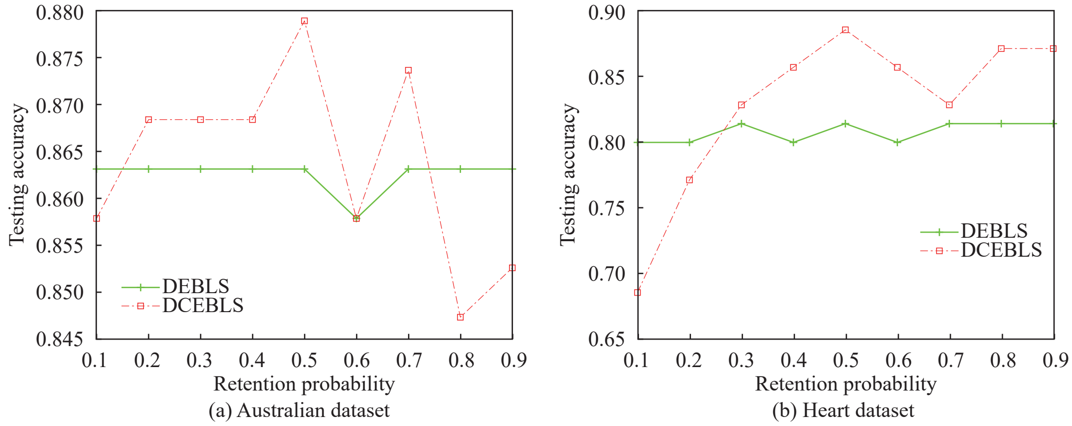


Figure 3 Variation of the classification accuracy of DEBLS and DCEBLS with different retention probabilities.

Table 2 Performance of different algorithms on selected UCI datasets. Bold indicates the best results.

Dataset	Method	ACC	PRE	REC	F1	Training time (s)
Australian	BLS	0.8789	0.8836	0.8748	0.8792	0.0116
	ENBLS	0.8947	0.9008	0.8910	0.8959	0.3646
	DEBLS	0.8895	0.8926	0.8850	0.8888	0.2217
	DCEBLS	**0.9053**	**0.9098**	**0.9010**	**0.9054**	0.3588
Default	BLS	0.8169	0.5963	0.7620	0.6690	0.1198
	ENBLS	0.8249	0.6190	**0.7767**	0.6889	12.5469
	DEBLS	0.8297	0.6412	0.7744	0.7015	10.2048
	DCEBLS	**0.8323**	**0.6579**	0.7700	**0.7095**	10.5523
Heart	BLS	0.8429	0.8424	0.8405	0.8414	0.0101
	ENBLS	0.8714	0.8681	0.8708	0.8695	0.3939
	DEBLS	0.8714	0.8681	0.8708	0.8695	0.1160
	DCEBLS	**0.9000**	**0.8904**	**0.9126**	**0.9014**	0.2349
Sonar	BLS	0.7671	0.8191	0.8023	0.8106	0.0073
	ENBLS	0.8356	0.8637	0.8337	0.8485	0.2416
	DEBLS	0.8493	0.8830	0.8514	0.8669	0.0835
	DCEBLS	**0.9041**	**0.9255**	**0.8939**	**0.9095**	0.2285
Spectf	BLS	0.6875	0.6875	0.7421	0.7138	0.0222
	ENBLS	0.7000	0.7000	0.7381	0.7185	0.2106
	DEBLS	0.7125	0.7125	0.7473	0.7295	0.1234
	DCEBLS	**0.8000**	**0.8000**	**0.8030**	**0.8015**	0.4508
Zoo	BLS	0.8710	0.8786	0.8393	0.8585	0.0108
	ENBLS	0.8710	0.8786	0.8429	0.8603	0.2174
	DEBLS	0.9032	0.9000	0.8571	0.8780	0.0567
	DCEBLS	**0.9677**	**0.9643**	**0.9762**	**0.9702**	0.1175

respectively. The recall rate is 0.52% and 0.92% higher than that of BLS and ENBLS, respectively.

The performance of DCEBLS is also shown in Table 2. Compared with the other three methods, DCEBLS has a certain degree of improvement in each index. As can be seen from Table 2, the accuracy of DCEBLS is increased by 6.45% and the F1 value is increased by 9.22% compared with DEBLS in Zoo dataset. On Australian dataset, the accuracy of DEBLS is worse than that of ENBLS, while the accuracy of DCEBLS is 1.06% higher than that of ENBLS. The main reason is that DEBLS directly discards nodes, so important feature information cannot be retained, while DCEBLS only discards weights, which can better retain the feature information.

It can be seen from Fig. 4 that the performances of DEBLS and DCEBLS are basically better than those of ENBLS and BLS in the ensemble process. As shown in Fig. 4, the accuracy of ENBLS reached 0.90 and then began to decline and fluctuate, while the accuracy of DCEBLS is 0.88 when the four submodels were integrated, and then the accuracy began to rise steadily. As can be seen from Fig. 5, the training time of DEBLS is improved compared with that of ENBLS. Due to the addition of matrix operation, the training time of DCEBLS is increased compared with DEBLS, but it is also slightly lower than ENBLS. As shown in Fig. 5, in Heart dataset, the training time of ENBLS is 3.40 and 1.68 times that of DEBLS and DCEBLS, respectively. The training time gap between DEBLS and ENBLS is increasing with the increase of the number of submodels.

In general, DEBLS is better than ENBLS in 4 out of 6 datasets and DCEBLS performs the best in all datasets in this experiment. Taking all evaluation indicators into account, standard BLS has the worst performance, which shows that ensemble learning under Dropout and Dropconnect framework introduced into BLS can effectively improve its unstable performance. DEBLS reduces the synergy between nodes through Dropout technology to prevent model overfitting. Moreover, DEBLS implicitly increases the diversity through different model structures to get better generalization and drops in training time for modeling. DCEBLS provides different input subspaces and reduces the dependence of the model on local features, which generates more complementary submodels to improve the performance of the ensemble model.

V. Conclusion

In this paper, Dropout ensemble broad learning system and Dropconnect ensemble broad learning system are proposed to improve the performance of models by reducing the similarity between submodels, so as to reduce the model overfitting. DEBLS retains the original advantages of starting training of BLS and increases the diversity between submodels by randomly discarding some nodes in the training process, reducing the synergy between nodes. In order to retain some important features and add input attribute disturbance, DCEBLS changes discarding nodes to randomly discard connection weights. Compared with DEBLS, DCEBLS can generate more complementary models, but the training time is slightly increased. In addition, many experiments are conducted to verify the feasibility of the proposed method. For several datasets proposed in the experiment, the experiment results show that the methods proposed in this paper can solve the problem of model overfitting caused by the strong dependence between the nodes of ensemble broad learning system and improve prediction stability.

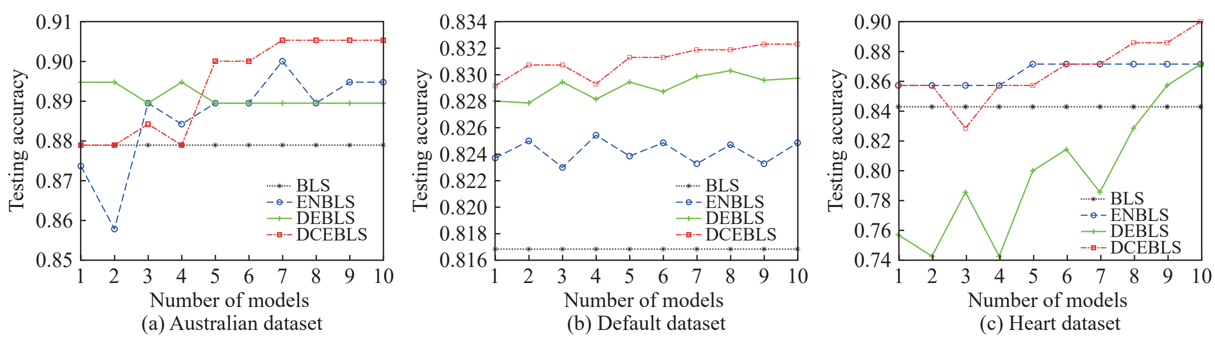


Figure 4 Comparison of different methods in terms of accuracy.

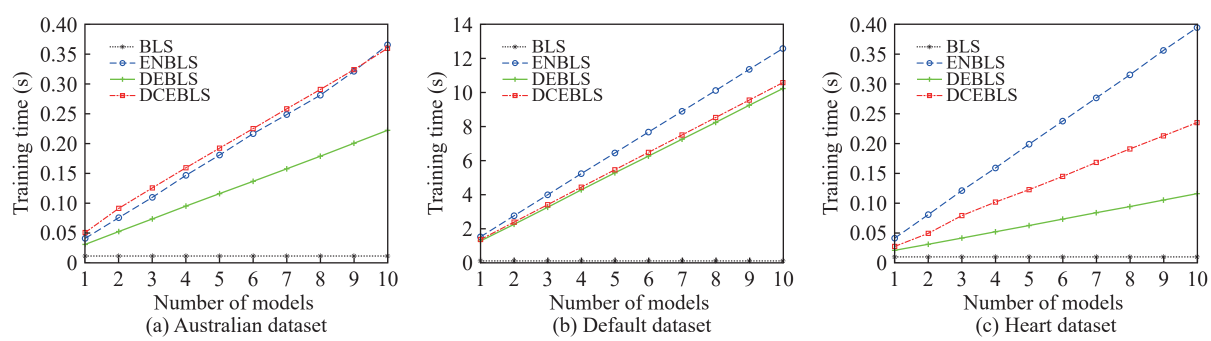


Figure 5 Comparison of different methods in terms of training time.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Nos. 61973304, 61873049, and 62073060), the Open Foundation of State Key Laboratory of Intelligent Optimized Manufacturing in Mining & Metallurgy Process (No. BGRIMM-KZSKL-2023-5), the Selection and Training Project of High Level Talents in the Sixteenth/Six Talent Peak of Jiangsu Province (No. DZXX-045), and the "Double First-Class" Construction to Enhance Independent Innovation Capability Project (No. 2022ZZCX01K01).

REFERENCES

[1] C. L. P. Chen and Z. Liu, Broad learning system: An effective and efficient incremental learning system without the need for deep architecture, *IEEE Trans. Neural Netw. Learn. Syst*., 2018, 29(1), 10–24.

[2] Y. Zheng, S. Wang, and B. Chen, Generalized multikernel correntropy based broad learning system for robust regression, *Inf. Sci*., 2024, 678, 121026.

[3] J. Jin, B. Geng, Y. Li, J. Liang, Y. Xiao, and C. L. P. Chen, Flexible label-induced manifold broad learning system for multiclass recognition, *IEEE Trans. Neural Netw. Learn. Syst*., to be published.

[4] X. Xu, S. Hu, P. Shi, H. Shao, R. Li, and Z. Li, Natural phase space reconstruction-based broad learning system for short-term wind speed prediction: Case studies of an offshore wind farm, *Energy*, 2023, 262, 125342.

[5] M. Han, W. Li, S. Feng, T. Qiu, and C. L. P. Chen, Maximum information exploitation using broad learning system for large-scale chaotic time-series prediction, *IEEE Trans. Neural Netw. Learn. Syst*., 2021, 32(6), 2320–2329.

[6] H. Guo, B. Sheng, P. Li, and C. L. P. Chen, Multiview high dynamic range image synthesis using fuzzy broad learning system, *IEEE Trans. Cybern*., 2021, 51(5), 2735–2747.

[7] W. Yu and C. Zhao, Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability, *IEEE Trans. Ind. Electron*., 2020, 67(6), 5081–5091.

[8] L. Liu, T. Liu, C. L. P. Chen, and Y. Wang, Modal-regression-based broad learning system for robust regression and classification, *IEEE Trans. Neural Netw. Learn. Syst*., to be published.

[9] C. Zhan, W. Jiang, F. Lin, S. Zhang, and B. Li, A Decomposition-ensemble broad learning system for AQI forecasting, *Neural Comput. Appl*., 2022, 34(21), 18461–18472.

[10] L. Guo, R. Li, and B. Jiang, An ensemble broad learning scheme for semisupervised vehicle type classification, *IEEE Trans. Neural Netw. Learn. Syst*., 2021, 32(12), 5287–5297.

[11] M. Wang, Q. Ge, H. Jiang, and G. Yao, Wear fault diagnosis of aeroengines based on broad learning system and ensemble learning, *Energies*, 2019, 12(24), 4750.

[12] X.-N. Fan and S.-W. Zhang, LPI-BLS: Predicting lncRNA–protein interactions with a broad learning system-based stacked ensemble classifier, *Neurocomputing*, 2019, 370, 88–93.

[13] L. Zhu, C. Lian, Z. Zeng, and Y. Su, A broad learning system with ensemble and classification methods for multi-step-ahead wind speed prediction, *Cognit. Comput*., 2020, 12(3), 654–666.

[14] Z. Yu, K. Lan, Z. Liu, and G. Han, Progressive ensemble kernel-based broad learning system for noisy data classification, *IEEE Trans. Cybern*., 2022, 52(9), 9656–9669.

[15] K. Yang, Z. Yu, C. L. P. Chen, W. Cao, J. You, and H. S. Wong, Incremental weighted ensemble broad learning system for imbalanced data, *IEEE Trans. Knowl. Data Eng*., 2022, 34(12), 5809–5824.

[16] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature*, 2015, 521(7553), 436–444.

[17] P. N. Suganthan, Letter: On non-iterative learning algorithms with closed-form solution, *Appl. Soft Comput*., 2018, 70, 1078–1082.

[18] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv: 1207.0580, 2012.

[19] L. Wan, M. D. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, Regularization of neural networks using Dropconnect, in *Proc. 30th International Conference on Machine Learning*, Atlanta, GA, USA, 2013, 1058–1066.

[20] T. DeVries and G. W. Taylor, Improved regularization of convolutional neural networks with cutout, arXiv preprint arXiv: 1708.04552, 2017.

[21] Y. Gal and Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in *Proc. 33rd International Conference on Machine Learning*, New York, NY, USA, 2016, 1050–1059.

[22] W. Yang, L. Jin, D. Tao, Z. Xie, and Z. Feng, DropSample: A new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten Chinese character recognition, *Pattern Recognit*., 2016, 58, 190–203.

[23] A. Zunino, S. A. Bargal, P. Morerio, J. Zhang, S. Sclaroff, and V. Murino, Excitation Dropout: Encouraging plasticity in deep neural networks, *Int. J. Comput. Vis*., 2021, 129(4), 1139–1152.

[24] J. Zhao, J. Yang, J. Wang, and W. Wu, Spiking neural network regularization with fixed and adaptive drop-keep probabilities, *IEEE Trans. Neural Netw. Learn. Syst*., 2022, 33(8), 4096–4109.

[25] A. Achille and S. Soatto, Information Dropout: Learning optimal representations through noisy computation, *IEEE Trans. Pattern Anal. Mach. Intell*., 2018, 40(12), 2897–2905.

[26] J. Lai, X. Wang, R. Li, Y. Song, and L. Lei, BD-ELM: A regularized extreme learning machine using biased Dropconnect and biased Dropout, *Math. Probl. Eng*., 2020, 2020, 3604579.

[27] L. Visser and A. A. Zumel, Comparison between the Dropout and Dropconnect regularization schemes, in *Proc. 17th SC@RUG 2019–2020*, Groningen, the Netherlands, 2020, 88–93.

[28] W. Zheng, J. Xie, X. Sun, and Z. Ma, Structured Dropconnect for uncertainty inference in image classification, in *Proc. IEEE International Conference on Image Processing*, Bordeaux, France, 2022, 366–370.

[29] A. Mobiny, P. Yuan, S. K. Moulik, N. Garg, C. C. Wu, and H. Van Nguyen, Dropconnect is effective in modeling uncertainty of Bayesian deep networks, *Sci. Rep*., 2021, 11(1), 5458.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res*., 2014, 15(1), 1929–1958.

[31] Y.-H. Pao and Y. Takefuji, Functional-link net computing: Theory, system architecture, and functionalities, *Computer*, 1992, 25(5), 76–79.

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

[33] Y.-C. Hu, Bankruptcy prediction using ELECTRE-based single-layer perceptron, *Neurocomputing*, 2009, 72(13–15), 3150–3157.

[34] Y.-C. Hu and J.-F. Tsai, Evaluating classification performances of single-layer perceptron with a Choquet fuzzy integral-based neuron, *Expert Syst. Appl*., 2009, 36(2), 1793–1800.

[35] A. Labach and S. Valaee, Regularizing neural networks by stochastically training layer ensembles, in *Proc. 30th International Workshop on Machine Learning for Signal Processing* (*MlSP*), Espoo, Finland, 2020, 1–6.

[36] E. H. Hssayni, N. E. Joudar, and M. Ettaouil, An adaptive drop method for deep neural networks regularization: Estimation of Dropconnect hyperparameter using generalization gap, *Knowl.-Based Syst*., 2022, 253, 109567.

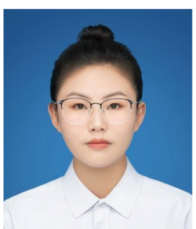
[37] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, Ensemble deep learning: A review, *Eng. Appl. Artif. Intell*., 2022, 115, 105151.

[38] J. Kazmaier and J. H. van Vuuren, The power of ensemble learning in sentiment analysis, *Expert Syst. Appl*., 2022, 187, 115819.

[39] A. Dogan and D. Birant, A weighted majority voting ensemble approach for classification, in *Proc. 4th International Conference on Computer Science and Engineering* (*UBMK*), Samsun, Turkey, 2019, 366–371.

[40] D. Dua and C. Graff, UCI machine learning repository [Online], http://archive.ics.uci.edu/ml, 3 October 2023.

[41] S. Shi, F. Nie, R. Wang, and X. Li, When multi-view classification meets ensemble learning, *Neurocomputing*, 2022, 490, 17–29.



Yiwan Cao received the BS degree from Henan Polytechnic University, China, in 2020. She is currently pursuing the MS degree at School of Information and Control Engineering, China University of Mining and Technology, China. Her research interests include modeling, control, and optimization of complex industrial processes.

Fei Chu received the BS degree from School of Automation, Qingdao University, China, in 2007, the MS and PhD degrees in control theory and control engineering from Northeastern University, China, in 2009 and 2014, respectively. He is currently an associate professor at School of Information and Control Engineering, China University of Mining and Technology, China, a member of State Key Laboratory of Management and Control for Complex System, Institute of Automation, Chinese Academy of Sciences, China, and a member of Xuzhou Key Laboratory of Artificial Intelligence and Big Data, China. He has authored over 30 papers in major international journals and conferences. His research interests include modeling, control, and optimization of complex industrial processes, machine learning, big data mining, statistical process monitoring, and operating state evaluation.

Shuai Li received the BS degree from Jiangnan University, China, in 2019. He is currently pursuing the MS degree at School of Information and Control Engineering, China University of Mining and Technology, China. His research interests include modeling, control, and optimization of complex industrial processes.

C. L. Philip Chen received the PhD degree from University of Michigan, Ann Arbor, MI, USA, in 1988. He is currently a chair professor at School of Computer Science and Engineering, South China University of Technology, China. He successfully architects the University of Macau's engineering and computer science programs, receiving accreditations from the Washington/Seoul Accord through The Hong Kong Institution of Engineers (HKIE), China, which is considered as his utmost contribution in engineering/computer science education for Macao as the former dean of the faculty. His research interests include systems, cybernetics, and computational intelligence. He is a fellow of American Association for the Advancement of Science, The International Association for Pattern Recognition, Chinese Association of Automation (CAA), and HKIE. He received the 2016 Outstanding Electrical and Computer Engineers Award from Purdue University. He is the editor-in-chief of *IEEE Transactions on Systems, Man, and Cybernetics: Systems* and an associate editor of *IEEE Transactions on Fuzzy Systems* and *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.

Jun Fu received the PhD degree in mechanical engineering from Concordia University, Canada, in 2009. He was a postdoctoral researcher at Department of Mechanical Engineering, Massachusetts Institute of Technology (MIT), USA, from 2010 to 2014. He received the 2018 Young Scientist Award in Science issued by the Ministry of Education of China (the first awardee in Chinese control community). He is currently a professor at Northeastern University, China. He is also an associate editor for *Control Engineering Practice*, *IEEE Transactions on Industrial Informatics*, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, and *IEEE Transactions on Neural Networks and Learning Systems*. His research interests include dynamic optimization, optimal control, switched systems, and their applications.