BLSHF: Broad Learning System with Hybrid Features

Weipeng Cao[1,2], Dachuan Li[3], Xingjian Zhang[1], Meikang Qiu[4], and Ye Liu[2](✉)

[1] CAAC Key Laboratory of Civil Aviation Wide Surveillance and Safety Operation Management and Control Technology, Civil Aviation University of China, Tianjin, China
caoweipeng@szu.edu.cn

[2] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
ly@szu.edu.cn

[3] Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
lidc3@mail.sustech.edu.cn

[4] Department of Computer Science, Texas A&M University-Commerce, Commerce, TX, USA

Abstract. Broad Learning System (BLS), a type of neural network with a non-iterative training mechanism and adaptive network structure, has attracted much attention in recent years. In BLS, since the mapped features are obtained by mapping the training data based on a set of random weights, their quality is unstable, which in turn leads to the instability of the generalization ability of the model. To improve the diversity and stability of mapped features in BLS, we propose the BLS with Hybrid Features (BLSHF) algorithm in this study. Unlike original BLS, which uses a single uniform distribution to assign random values for the input weights of mapped feature nodes, BLSHF uses different distributions to initialize the mapped feature nodes in each group, thereby increasing the diversity of mapped features. This method enables BLSHF to extract high-level features from the original data better than the original BLS and further improves the feature extraction effect of the subsequent enhancement layer. Diverse features are beneficial to algorithms that use non-iterative training mechanisms, so BLSHF can achieve better generalization ability than BLS. We apply BLSHF to solve the problem of air quality evaluation, and the relevant experimental results empirically prove the effectiveness of this method. The learning mechanism of BLSHF can be easily applied to BLS and its variants to improve their generalization ability, which makes it have good application value.

Keywords: Broad Learning System · Air quality prediction · Neural networks

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13369, pp. 655–666, 2022.
https://doi.org/10.1007/978-3-031-10986-7_53

1 Introduction

In recent years, deep learning technology has made breakthroughs in many fields [7,19]. Here we mainly focus on deep neural networks in deep learning. The complex connection between neurons and hidden layers enables the related model to extract multi-levels of feature information from the original data. Based on the extracted feature information, the model can construct the relationship between the feature description of training samples and their labels for predicting the labels of new samples. Traditional neural networks generally use the iterative training mechanism. Specifically, they first calculate the prediction error of the model based on the initialization parameters and training samples and then use the error back-propagation method to iteratively fine-tune all the weights in the neural network to make the prediction error of the model reach an acceptable threshold. This process is time-consuming, and the demand for hardware computing resources is huge. For example, GPU is often necessary for deep model training. This makes it difficult to train and deploy related models on-site in many scenarios with limited computing power.

To alleviate the defects of traditional neural networks, neural networks using non-iterative training mechanisms have attracted more and more attention in recent years [2,17]. This kind of neural network has one thing in common, that is, some parameters in the neural network remain unchanged in the subsequent model training process after initialization, and the parameters to be calculated only need to be solved at one time. Compared with traditional neural networks, this training method undoubtedly greatly improves the learning efficiency of the model. Relevant representative algorithms include Random Vector Functional Link network (RVFL) [16], Pseudo-Inverse Learning (PIL) [8], Extreme Learning Machine (ELM) [9], Stochastic Configuration Network (SCN) [14], and Broad Learning System (BLS) [3]. This study mainly focuses on BLS.

BLS is an upgraded version of RVFL proposed by Chen et al. in 2017 [3] which has two obvious characteristics. First, BLS uses a non-iterative training mechanism, which makes it very efficient in training. Second, the network structure of BLS has flexible scalability. Specifically, the number of its mapped feature nodes and enhancement nodes can be dynamically increased or pruned according to different tasks. After two feature mappings of the mapped feature layer and the enhancement layer, the different classes of the original data can be linearly separable in a high-dimensional space, and then the parameters of the model can be solved based on the ridge region theory. The universal approximation capability of BLS has been proved in [4]. Thanks to the excellent performance of BLS in multiple scenarios and a relatively complete theoretical foundation, it quickly attracted the attention of researchers. Improved BLS-based algorithms and applications are constantly being proposed. Related representative works include fuzzy BLS [6], recurrent BLS [15], weighted BLS [5], BLS with Proportional-Integral-Differential Gradient Descent [21], dense BLS [20], multi-view BLS [13], semi-supervised BLS [18], etc.

Although the above-mentioned BLS and its variants have shown great potential in many scenarios, their model performance is still expected to be further improved. Because we found that their input parameters are randomly generated based on a uniform distribution. BLS grouped the mapped feature nodes at the beginning of the design, but few studies use different distributions to initialize different groups of mapped feature nodes. In the early research of Cao et al., they found that for RVFL, using different distribution functions to initialize the input weights will have different effects on the performance of the model [1]. Moreover, using uniform distribution to initialize the input weights cannot always guarantee that the model has good generalization ability. Later, they further studied and found that in ensemble learning scenarios, using multiple distributions to initialize the sub-models can effectively improve the generalization ability of the final model [12].

Inspired by the above work, in this study, we propose to use multiple distribution functions to initialize different groups of mapped feature nodes, thereby enhancing the diversity of mapped features. We call this method: BLS with Hybrid Features (BLSHF). For many machine learning algorithms, the more diverse the features, the better the generalization ability of the model. BLSHF uses multiple distributions to initialize input weights in the mapped feature layer, which can obtain multi-level feature abstractions from the original data, and indirectly improves the diversity of enhancement features. Abundant features help the model to better mine the mapping relationship between the original feature description of the training samples and their labels, thereby improving the generalization ability of the model.

The contributions of this study can be summarized as follows.

- A novel BLS algorithm with hybrid features (i.e., BLSHF) was proposed in this study, which can greatly improve the diversity of the mapped features and enhancement features.
- The idea of BLSHF can be easily transferred to other BLS algorithms to further improve the performance of related models.
- To verify the effectiveness of BLSHF, we applied it to build an air quality prediction model. Extensive experimental results on two public air quality evaluation data sets show that BLSHF can achieve better generalization ability than BLS. The air quality prediction model based on BLSHF also provides a feasible solution for real-world related scenarios.

The remainder of this study is organized as follows: Sect. 2 briefly reviews the learning mechanism of BLS and related work. Details of the proposed BLSHF are presented in Sect. 3, followed by experimental results and analysis in Sect. 4. We conclude this study in Sect. 5.

2 Related Work

In this section, we review the training mechanism of BLS and the existing literature related to this study. As mentioned in Sect. 1, BLS is a feedforward neural network that uses a non-iterative training mechanism, and its basic network structure is shown in Fig. 1. Note that here we use the version that does not directly connect the input and output layers.

It can be observed from Fig. 1 that BLS is a four-layer neural network: input layer, mapped feature layer, enhancement layer, and output layer. In this study, we denote the input weights between the input layer and the mapped feature layer as W_input, the weights between the mapped feature layer and the enhancement layer as $W\_enhance$, and the output weights between the mapped feature layer and the enhancement layer and the output layer as W_output.

The mapped feature layer performs the first feature mapping on the original data transmitted from the input layer by groups and then concatenates the features extracted from all groups as input and transmits them to the enhancement layer for the second feature mapping. Then, the mapped feature and enhancement feature are connected to obtain the final feature matrix, which will be used to calculate the output weights (i.e., W_output) based on the ridge regression theory. Different from traditional neural networks, the training process of BLS is completed at one time, so it is called a neural network with a non-iterative training mechanism.

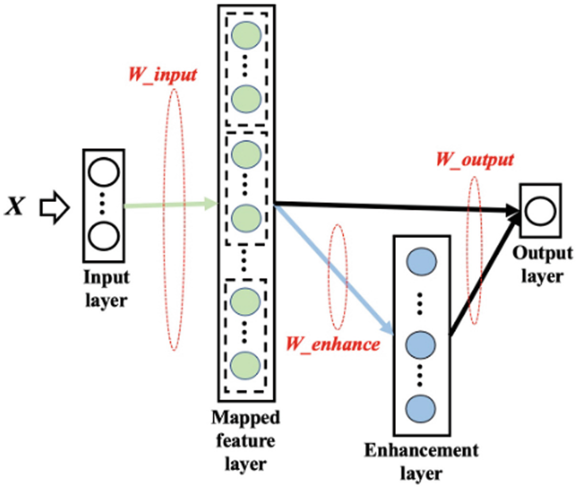

Fig. 1. Network structure of BLS.

In the above model training process, the input weights between the input layer and the mapped feature layer (i.e., W_input) are randomly generated according to a uniform distribution, and these parameters will remain unchanged in the subsequent model training process.

According to [1], using different distribution functions to initialize input weights will have different effects on the performance of non-iterative neural networks. BLS groups different mapped feature nodes in the mapped feature layer but does not use diversified strategies to initialize different groups of mapped feature nodes. Therefore, the original BLS model may still face the problem of failing to achieve optimal performance in some specific scenarios.

Liu et al. found that if multiple distributions are used to initialize the submodels separately, and then the ensemble learning mechanism is used to integrate their prediction results, the prediction ability of the final model can be effectively improved [12].

Inspired by this idea, we try to use different distributions to initialize different groups of mapped feature nodes in BLS to get a model with better generalization ability.

3 The Proposed Method

The core idea of the method proposed in this study is to initialize different groups of mapped feature nodes in the BLS with different distributions to obtain multi-levels of feature extraction. Diversified feature expression and fusion can allow the model to better mine the relationship between the original features of training samples and their label, and then obtain a model with better generalization ability. We call the proposed method: BLS with Hybrid Features (BLSHF).

The network structure of BLSHF is shown in Fig. 2. Except that the initialization of mapped feature nodes is different from the original BLS, other parts are the same. Note that for the consideration of control variables, we do not group enhancement nodes here. In other words, in the enhancement layer, we use a uniform distribution to generate random parameters for enhancement nodes in accordance with BLS. According to Fig. 2, we can easily implement the BLSHF algorithm. Its pseudo-code is shown in Algorithm 1.

It can be observed from Algorithm 1 that the difference between our proposed BLSHF algorithm and BLS is that we use different distribution functions to initialize mapped feature nodes to obtain more diverse features.

Universal Approximation Property of BLSHF: As mentioned above, if we reduce the number of the distribution function that initializes random parameters to 1 (i.e., the Uniform distribution), BLSHF will degenerate into BLS. In other words, BLSHF only improves the generalization ability of the model by increasing the diversity of mapped features and enhancement features, and does not significantly modify the network architecture or non-iterative training mechanism of the original BLS. The universal approximation property of BLS has been proven in [4]. Therefore, one can infer that the proposed BLSHF also has the same approximation property.

In the next section, we will empirically prove the effectiveness of this approach through experiments on air quality assessment.

4 Experimental Settings and Results

In this section, we evaluate the performance of the proposed BLSHF algorithm on two real-world air quality index prediction problems.

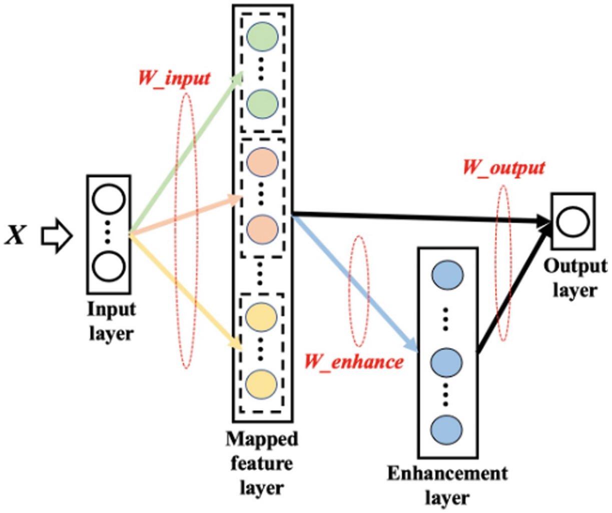

Fig. 2. Network structure of BLSHF.

These two data sets describe the urban air pollution in Beijing and Oslo in a specific period, respectively. As shown in Table 1, the Beijing PM 2.5 data set contains 41,757 samples, and each sample has 11 attributes, which depict the values of related indicators that may lead to a specific PM 2.5 value. Similarly, the Oslo PM 10 data set contains 500 samples, and each sample has 7 attributes, which depict the values of related indicators that may lead to a specific PM 10 value. Information about the specific meaning of each attribute of the sample can be viewed from the data source: Beijing PM 2.5 [11] and Oslo PM 10[1]. This study focuses on the modeling and performance evaluation of the proposed method on these two data sets.

In our experiment, we set the number of mapped feature nodes to 6, and the number of feature window to 3, which corresponds to three different distribution functions, namely: Uniform distribution, Gaussian distribution, and Gamma distribution. The number of enhancement nodes is set to 41. The activation functions of BLS and BLSHF are uniformly selected as the *Sigmoid* function.

For each data set, we split it into a training set and a testing set according to 7:3. Root Mean Square Error ($RMSE$) and Normalized Mean Absolute Error ($NMAE$) are chosen as the indicator to evaluate the performance of the model. The smaller these two indicators are, the better the model performance is. They can be calculated according to the following equations.

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(y_i^* - y_i)^2}{N}} \tag{1}$$

[1] http://lib.stat.cmu.edu/datasets/.

Algorithm 1: BLSHF Algorithm

```
Input: Training data $X$, activation functions $\phi 1(.)$ and $\phi 2(.)$ for the mapped
       feature layer and the enhancement layer, respectively.
Output: All parameters of the BLSHF model.
Randomly assign the values of $W\_input\_i$ for the $i$-th group of mapped feature nodes under the $i$-th distribution.
Use $\phi 1(.)$ to project and get the mapped feature of the $i$-th group of mapped feature nodes $Z\_i$. Specifically, $Z\_i = \phi 1(X * W\_input + \beta\_input)$, where $\beta\_input$ is the thresholds of mapped feature nodes.
Concatenate the mapped features of all groups: $Z = [Z\_1, \ldots, Z\_k]$, where $k$ is the number of the groups.
Randomly assign the values of $W\_enhance$ for the enhancement nodes under the uniform distribution.
Use $\phi 2(.)$ to enhance the mapped feature by the enhancement layer:
$H = \phi 2(Z * W_enhance + \beta\_enhance)$, where $\beta\_enhance$ is the thresholds of enhancement nodes.
Calculate the output weights $W\_output = [Z|H]^{+}Y$, where $Y$ is the real label matrix of the training samples.
Return: Input weights of mapped feature nodes in each group (i.e., $W\_input$), input weights of enhancement feature nodes (i.e., $W\_enhance$), and the output weights $W\_output$.
```

Table 1. Details of experimental datasets

Dataset	Number of attributes	Number of samples
Beijing PM 2.5	11	41757
Oslo PM 10	7	500

$$NMAE = \frac{MAE(y_i^*, y_i)}{\frac{1}{N}\sum_{i=1}^{N}|y_i|} \tag{2}$$

where y_i^* is the predicted label of the model for the i-th sample, y_i is the real label of the i-th sample, and N is the number of samples. $MAE(.)$ means the mean absolute error.

The experimental results of BLS and BLSHF on two air quality prediction data sets are shown in Table 2. For ease of comparison, we have bolded better metrics.

It can be observed from Table 2 that the BLSHF model can achieve lower prediction errors on all data sets than the BLS model, which implies that the proposed BLSHF model has better generalization ability than the original BLS model. This experimental phenomenon verifies a consensus in the field of machine learning: the diversification of data features helps the model to better learn the implicit patterns in the data.

In addition, we can also observe an interesting experimental phenomenon, that is, the training time of the BLSHF model is shorter than that of the BLS. This phenomenon may be because the sampling efficiency of the partial distribution function is higher than that of the Uniform distribution, thus improving the training efficiency of the overall model. However, this is only a speculation, and we will analyze this in more depth from a mathematical point of view in the future.

To show our experimental results more clearly, we visualized them separately, namely: Figs. 3–8. From these visualized figures, it can be intuitively found that our proposed BLSHF algorithm can not only achieve lower prediction errors than BLS, but also have faster training efficiency.

For the above experimental phenomenon, a speculative explanation is given here: using different distribution functions to initialize mapped feature nodes can allow the model to extract more diverse features, which is beneficial to improve the generalization ability of the model. Therefore, the prediction error of the BLSHF model is lower than that of the BLS.

Table 2. Details of experimental results

Dataset	Performance	BLS	The proposed BLSHF
Beijing PM 2.5	Training RMSE	0.0731	**0.0716**
	Testing RMSE	0.0741	**0.0726**
	Training NMAE	0.5349	**0.5275**
	Testing NMAE	0.5378	**0.5286**
	Training time	0.0669	**0.0434**
Oslo PM 10	Training RMSE	0.1506	**0.1471**
	Testing RMSE	0.1768	**0.1585**
	Training NMAE	0.2173	**0.2127**
	Testing NMAE	0.2408	**0.2252**
	Training Time	0.0493	**0.0401**

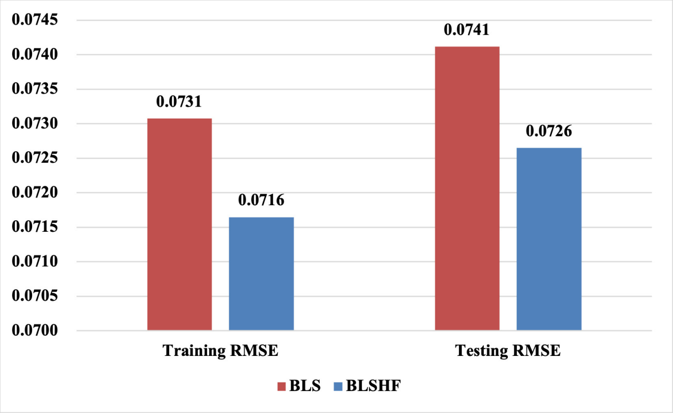

Fig. 3. RMSE of BLS and BLSHF models on Beijing PM 2.5 dataset.

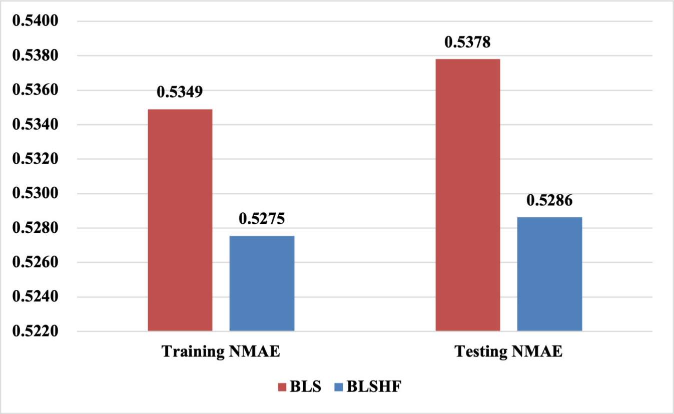

Fig. 4. NMAE of BLS and BLSHF models on Beijing PM 2.5 dataset.

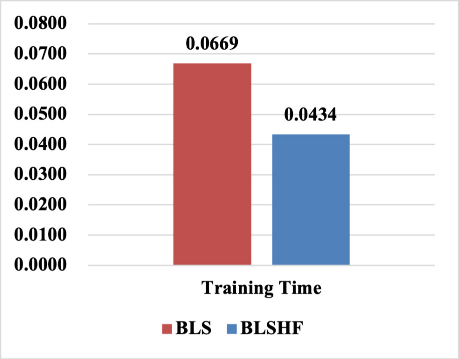

Fig. 5. Training time of BLS and BLSHF models on Beijing PM 2.5 dataset.

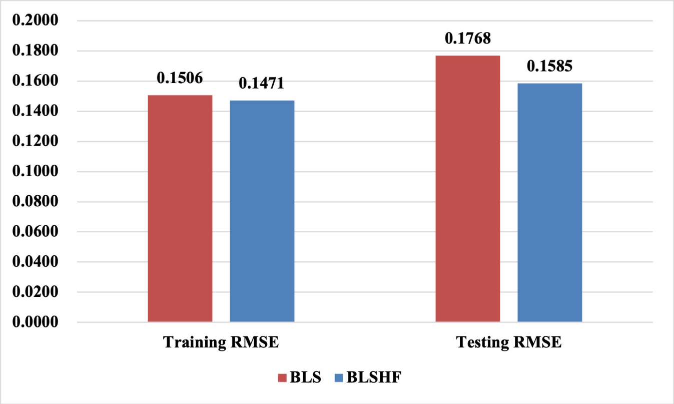

Fig. 6. RMSE of BLS and BLSHF models on Oslo PM 10 dataset.

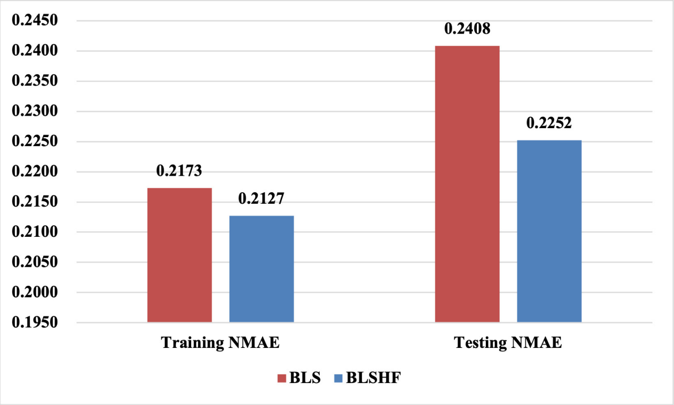

Fig. 7. NMAE of BLS and BLSHF models on Oslo PM 10 dataset.

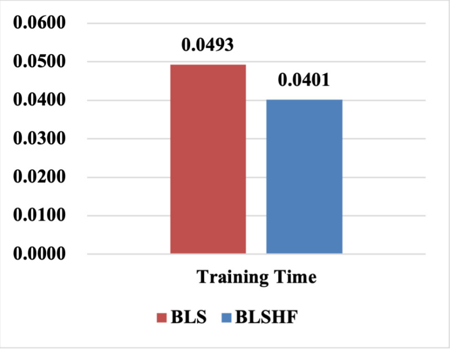

Fig. 8. Training time of BLS and BLSHF models on Oslo PM 10 dataset.

5 Conclusions

To improve the feature extraction capability of BLS, we innovatively design an improved BLS algorithm called BLSHF. Different from the original BLS, BLSHF uses multiple initialization strategies for each group of mapped feature nodes, which enables them to provide more diverse features for model learning. The diversity of mapped features further improves the diversity of enhancement features. The diversity of features helps the algorithm to better mine the internal patterns of the data, resulting in a model with better generalization ability.

BLSHF also inherits the non-iterative training mechanism of BLS, so it has the advantages of extremely fast training speed and low hardware computing power requirements. We apply BLSHF to model two real-world air pollution assessment problems, and the experimental results show that it can achieve better generalization ability than the BLS model. The air pollution perception model based on BLSHF also provides a new idea for real-world air quality monitoring research.

However, the current version of the BLSHF algorithm is still a shallow feedforward neural network. Even if we have improved the feature extraction capability of the original BLS, the existing solution may still be stretched in the face of complex datasets such as ImageNet [10]. In the future, we will consider using BLSHF as a stacking unit to build a more complex neural network to solve the modeling problem of complex scenarios.

Acknowledgment. This work was supported by National Natural Science Foundation of China (Grant No. 62106150), CAAC Key Laboratory of Civil Aviation Wide Surveillance and Safety Operation Management and Control Technology (Grant No. 202102), and CCF-NSFOCUS (Grant No. 2021001).

References

1. Cao, W., Patwary, M.J., Yang, P., Wang, X., Ming, Z.: An initial study on the relationship between meta features of dataset and the initialization of nnrw. In: 2019 International Joint Conference on Neural Networks, IJCNN, pp. 1–8. IEEE (2019)
2. Cao, W., Wang, X., Ming, Z., Gao, J.: A review on neural networks with random weights. Neurocomputing **275**, 278–287 (2018)
3. Chen, C.P., Liu, Z.: Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. IEEE Trans. Neural Netw. Learn. Syst. **29**(1), 10–24 (2017)
4. Chen, C.P., Liu, Z., Feng, S.: Universal approximation capability of broad learning system and its structural variations. IEEE Trans. Neural Net. Learn. Syst. **30**(4), 1191–1204 (2018)
5. Chu, F., Liang, T., Chen, C.P., Wang, X., Ma, X.: Weighted broad learning system and its application in nonlinear industrial process modeling. IEEE Trans. Neural Netw. Learn. Syst. **31**(8), 3017–3031 (2019)
6. Feng, S., Chen, C.P.: Fuzzy broad learning system: a novel neuro-fuzzy model for regression and classification. IEEE Trans. Cybern. **50**(2), 414–424 (2018)
7. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT Press (2016)
8. Guo, P., Chen, C.P., Sun, Y.: An exact supervised learning for a three-layer supervised neural network. In: 1995 International Conference on Neural Information Processing, ICNIP, pp. 1041–1044 (1995)
9. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE International Joint Conference on Neural Networks, IJCNN, vol. 2, pp. 985–990. IEEE (2004)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
11. Liang, X., et al.: Assessing beijing's pm2. 5 pollution: severity, weather impact, apec and winter heating. Proc. Royal Soc. A Math. Phys. Eng. Sci. **471**(2182), 20150257 (2015)
12. Liu, Y., Cao, W., Ming, Z., Wang, Q., Zhang, J., Xu, Z.: Ensemble neural networks with random weights for classification problems. In: 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence, ACAI, pp. 1–5 (2020)

13. Shi, Z., Chen, X., Zhao, C., He, H., Stuphorn, V., Wu, D.: Multi-view broad learning system for primate oculomotor decision decoding. IEEE Trans. Neural Syst. Rehabil. Eng. **28**(9), 1908–1920 (2020)
14. Wang, D., Li, M.: Stochastic configuration networks: fundamentals and algorithms. IEEE Trans. Cybern. **47**(10), 3466–3479 (2017)
15. Xu, M., Han, M., Chen, C.P., Qiu, T.: Recurrent broad learning systems for time series prediction. IEEE Trans. Cybern. **50**(4), 1405–1417 (2018)
16. Zhang, L., Suganthan, P.N.: A comprehensive evaluation of random vector functional link networks. Inf. Sci. **367**, 1094–1105 (2016)
17. Zhang, L., Suganthan, P.N.: A survey of randomized algorithms for training neural networks. Inf. Sci. **364**, 146–155 (2016)
18. Zhao, H., Zheng, J., Deng, W., Song, Y.: Semi-supervised broad learning system based on manifold regularization and broad network. IEEE Trans. Circuits Syst. I Regul. Pap. **67**(3), 983–994 (2020)
19. Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. **30**(11), 3212–3232 (2019)
20. Zou, W., Xia, Y., Cao, W.: Dense broad learning system based on conjugate gradient. In: 2020 International Joint Conference on Neural Networks, IJCNN, pp. 1–6. IEEE (2020)
21. Zou, W., Xia, Y., Cao, W., Ming, Z.: Broad learning system with proportional-integral-differential gradient descent. In: Qiu, M. (ed.) ICA3PP 2020. LNCS, vol. 12452, pp. 219–231. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60245-1_15