Research Review for Broad Learning System: Algorithms, Theory, and Applications

Xinrong Gong, Tong Zhang, *Member, IEEE*, C. L. Philip Chen, *Fellow, IEEE*, and Zhulin Liu, *Member, IEEE*

***Abstract*—In recent years, the appearance of the broad learning system (BLS) is poised to revolutionize conventional artificial intelligence methods. It represents a step toward building more efficient and effective machine-learning methods that can be extended to a broader range of necessary research fields. In this survey, we provide a comprehensive overview of the BLS in data mining and neural networks for the first time, focusing on summarizing various BLS methods from the aspects of its algorithms, theories, applications, and future open research questions. First, we introduce the basic pattern of BLS manifestation, the universal approximation capability, and essence from the theoretical perspective. Furthermore, we focus on BLS's various improvements based on the current state of the theoretical research, which further improves its flexibility, stability, and accuracy under general or specific conditions, including classification, regression, semisupervised, and unsupervised tasks. Due to its remarkable efficiency, impressive generalization performance, and easy extendibility, BLS has been applied in different domains. Next, we illustrate BLS's practical advances, such as computer vision, biomedical engineering, control, and natural language processing. Finally, the future open research problems and promising directions for BLSs are pointed out.**

***Index Terms*—Broad learning system (BLS), classification, feature learning, regression, research review, semisupervised learning (SSL), unsupervised learning.**

Manuscript received 9 July 2020; revised 23 October 2020 and 14 January 2021; accepted 17 February 2021. Date of publication 17 March 2021; date of current version 18 August 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFA0706200 and Grant 2019YFB1703600; in part by the National Natural Science Foundation of China under Grant 62076102, Grant U1813203, and Grant U1801262; in part by the National Natural Science Foundation of Guangdong for Distinguished Young Scholar under Grant 2020B1515020041; in part by the Science and Technology Major Project of Guangzhou under Grant 202007030006; in part by the Science and Technology Program of Guangzhou under Grant 202002030250; and in part by the Guangdong–Hong Kong–Macao Greater Bay Area Center for Brain Science and Brain-Inspired Intelligence Fund under Grant 2019016. This article was recommended by Associate Editor E. E. Herrera-Viedma. *(Corresponding author: Tong Zhang.)*

Xinrong Gong and Zhulin Liu are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.

Tong Zhang is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Pazhou Lab, Guangzhou 510335, China (e-mail: tony@scut.edu.cn).

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, also with the Pazhou Lab, Guangzhou 510335, China, and also with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3061094.

Digital Object Identifier 10.1109/TCYB.2021.3061094

I. Introduction

HIGH-DIMENSIONAL and large-scale data have gradually become a major challenge for the existing machine-learning methods due to their complex knowledge and interdependence between objects. Deep-structure neural networks and learning are generally composed of multiple layers to learn data features with multiple levels of abstraction by mining complex knowledge from simpler concepts, which have been applied in many fields and have achieved breakthrough successes [1]. These deep structures learn knowledge with the feature representation, classification, and pattern recognition of data through nonlinear information processing and abstraction [2], such as deep belief networks (DBN) [3], [4]; the deep Boltzmann machines (DBM) [5]; and convolutional neural networks (CNNs) [6]–[9]. However, these deep structures suffer from a mass of iterative training processes that consume time and computational resources due to their large number of parameters and complicated hand-designed structures. What is more, it is not easy to rapidly update its deep architecture parameters because it involves the entire network when facing the newly added samples.

In this case, some flat networks have notably emerged to increase neural networks' efficiency and performance by adding neural nodes laterally. The single-layer feedforward network (SLFN) [10]–[12] has been strictly proved from a theoretical point of view that it has universal approximation capabilities [13], making this simple but powerful network applicable to many fields. However, this conventional slow gradient-based learning algorithm is susceptible to parameter settings [14], which leads to its slow convergence and easy to trap in a local minimum. The random vector functional-link neural network (RVFL) [15], [16] is another kind of flat neural network with a different theory from SFLN. The weights between input data and the hidden layer of RVFL are randomly generated for saving the training time and calculated amount. RVFL can combine the input data with generated nonlinear features and directly connect to the output. However, with the advent of big data, directly connecting input data to the output layer may prevent the system from maintaining its accessibility anymore.

Until very recently, the broad learning system (BLS)[1] is proposed by Chen *et al.* [17]–[19] and became one of the most popular networks recently due to its outstanding performance in machine-learning tasks. Unlike RVFL, BLS can map the

[1] https://broadlearning.ai/bls/

input samples to a more suitable feature space to handle high-volume and time-variety data. More important, BLS retains the powerful mechanism for randomly generating hidden-layer node weights according to any continuous probability distribution. Only the weights from the hidden layer to the output layer need to be trained through the pseudoinverse algorithm. Specifically, the system can be updated incrementally without reconstructing the entire network from the beginning when facing the newly added samples and hidden nodes.

In this article, we propose a comprehensive survey of its principal advances and critical breakthroughs from the past few years of BLS for the first time, focusing on summarizing various BLS methods from the perspective of its algorithms, theories, applications, and future open research directions. We hope that this article can provide practical help for many novices who are about to participate in this field of BLS, obtaining an overall picture of the recent latest research, technology, and applications of BLS, and guide them for their future research. We also hope that some experts who would like to compare BLS models can find useful information.

Our article makes notable contributions summarized as follows.

1) We made the most detailed summary of the basic forms and learning methods of BLS. More important, BLS's essence has been further explained, which can provide readers with another way of understanding the algorithm of BLS from a macro perspective.
2) We provide the most comprehensive survey of BLS technology. For each type of BLS neural network, we provide detailed descriptions of representative models. In addition, we also make the necessary comparisons and summarize the corresponding algorithms.
3) We analyzed some crucial advantages of BLS compared to conventional learning algorithms. Simultaneously, based on these advantages, some opportunities and promising research directions are provided, including hardware, theory, and application of BLS.

The remainder of this survey is organized as follows. Section II will give the preliminary work of RVFL and its incremental learning algorithm. In Section III, we introduce the formulations of classical BLS, including the architecture, essence, incremental, and regularized forms. In Section IV, we report the universal approximation capability of BLS. We then introduce the extensions and improvements of BLS for classification and regression, semisupervised learning (SSL), unsupervised learning, and feature representation and selection in Sections V–VIII, respectively. The applications of BLSs are summarized in Section IX and Section X points out the future open research problems and promising direction for BLSs. Section XI concludes this article.

II. Preliminary Work

A. Random Vector Functional-Link Neural Networks

RVFL [20] is a randomized version of the functional-link neural network. The input data are first transformed into enhanced features through nonlinear mapping. In this process, all weights and bias are randomly generated. Finally, the representation of the input layer and the enhancement layer will be fed into the subsequent process, and the weight of the output layer will be obtained through the pseudoinverse operation. This flat structure can make the input data and output data directly related.

Furthermore, by enhancing the nonlinear mapping of nodes, it can increase the fitting ability and stability of the entire network. Pao *et al.* [21] discussed the learning and generalization characteristics of RVFL. Also, Zhang and Suganthan [22] investigated a comprehensive evaluation of RVFL by using 121 UCI datasets. Good performance in classification, regression tasks, and control systems shows the powerful capabilities of RVFL.

B. Incremental Learning of Flat Networks

Incremental learning refers to a learning system that can continuously learn new knowledge from new samples and can save most of the knowledge that has been previously gained [23], [24]. An incremental learning algorithm should also have the following characteristics: 1) it can continuously add new data samples and learn new knowledge; 2) data that have been processed before do not need to be treated repeatedly; 3) it can learn new knowledge and save most of the previous learned knowledge at the same time; and 4) when faced with new data, there is no need to reconstruct the model. These learning mechanisms are more in line with human thinking.

The functional-link neural networks usually assume that all input data can be used in the training stage when solving the weight matrix. It is not efficient at all to update the entire parameter by running a complete training cycle. Chen *et al.* [25], [26] proposed a fast-learning algorithm that makes it easier to update the weights instantly for newly added data or newly added enhancement nodes. With the proposed approach, the flat networks become very attractive in terms of learning speed.

III. Classical BLS

This section will introduce the classical BLS in detail, including the mapping feature and enhancement feature nodes, basic manifestation, incremental forms, essence, and regularized BLS.

A. BLS Mapping Feature and Enhancement Feature Nodes

BLS contains three essential parts: 1) mapping feature nodes; 2) enhancement feature nodes; and 3) output matrix. Training a BLS includes two main stages: 1) randomly generating the weight of both mapping feature and enhancement feature and 2) computing the weight between the hidden layer and output layer. In other words, the weights of the BLS random layer are randomly generated, and only the weight of the link between the last output layer and the hidden layer needs to be obtained by training. The architecture of the BLS is shown in Fig. 1.

In the first stage, BLS's original input data are transmitted as mapping features and then the structure can be expanded based

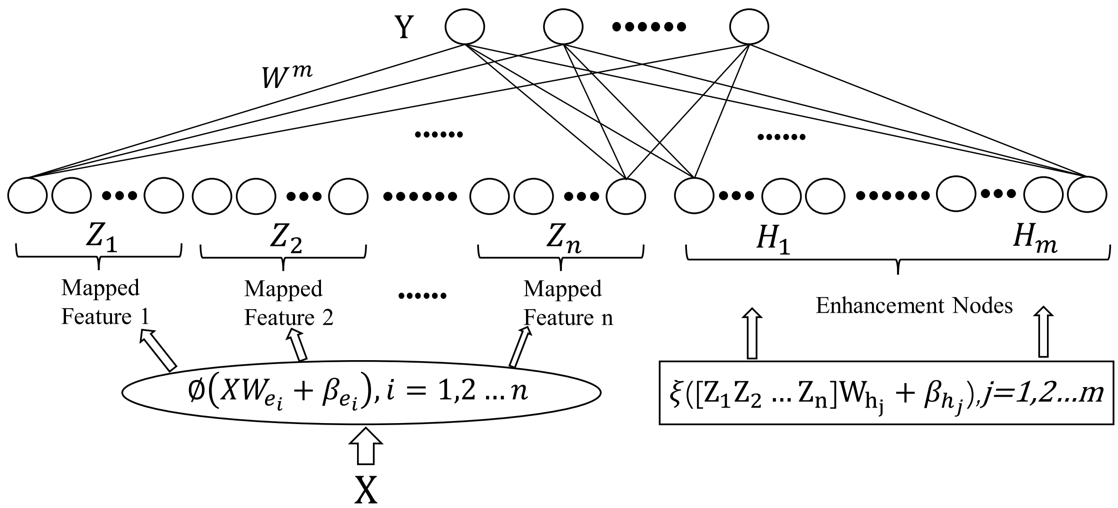


Fig. 1. Architecture of BLS.

on the enhancement nodes. The random feature mapping stage differentiates BLS from other existing learning methods such as SLFN, which updates its parameter by the gradient descent-based algorithm or support vector machine (SVM) [27], [28], which uses kernel functions for feature mapping.

Assume the training dataset matrix of a supervised task as $\{(\mathbf{X}, \mathbf{Y})|\mathbf{X} \in \mathbb{R}^{N\times M}, \mathbf{Y} \in \mathbb{R}^{N\times C}\}$ from C classes, where N is the number of data samples and M is the dimension of features. As we know, the BLS network has n groups of mapping feature nodes with each group containing k nodes. It has m groups of enhancement nodes and each group contains q nodes. The output of the ith group of the mapping feature nodes can be represented as the equation of the form

$$\mathbf{Z}_i = \phi_i\big(\mathbf{X}\mathbf{W}_{ei} + \boldsymbol{\beta}_{ei}\big), \quad i = 1, 2, \ldots, n \tag{1}$$

where weighting matrix $\mathbf{W}_{e_i}$ and bias term $\boldsymbol{\beta}_{e_i}$ are randomly initialized with the proper dimensions. Next, these randomly generated mapping feature groups will be collected together, which can be denoted as $\mathbf{Z}^n = [\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n]$. BLS will use $\mathbf{Z}_n$ composed of these groups of mapping features to expand the enhancement layer through a nonlinear activation function

$$\mathbf{H}_j = \xi_j\Big(\mathbf{Z}^n\mathbf{W}_{h_j} + \boldsymbol{\beta}_{h_j}\Big), \quad j = 1, 2, 3, \ldots, m \tag{2}$$

where $\xi_j(\cdot)$ is the nonlinear activation function. In Section IV, it was proved that any absolutely integerable activation function can be used in BLS. It can be seen from the above description that BLS retains the random weight mechanism of RVFL for quickly mapping input samples into a more suitable space. This mechanism can explain why BLS can train neural networks more effectively than traditional back propagation neural networks. Simultaneously, it is more suitable to process massive high-dimensional and time-varying data by mapping the input data into the feature space through a function than directly connecting the input data to the output layer.

Some commonly used nonlinear activation functions are summarized in Table I. Xu and Chen [29] made a detailed comparison of 20 popular activation functions on different datasets in classification and regression. Experiments show that different activation functions enable the model to obtain different nonlinear expression capabilities. In addition, in recent years, many researchers have made some improvements to the ways for generated random mapping feature nodes, hoping to adapt to different fields' requirements. In [17], Chen and Liu used the autoencoder [30], [31] to fine tune the random initial $\mathbf{W}_{e_i}$ to a set of sparse and compact features for obtaining better feature mapping parameters. Furthermore, Feng and Chen [32], [33] explored the Takagi–Sugeno (TS) fuzzy system [34], [35] for replacing the mapping feature nodes with a group of TS fuzzy subsystems. Lin *et al.* [36]–[39] proposed the wavelet BLS, which used the wavelet function to obtain the mapping feature nodes. Liu *et al.* [40] replaced the linear mapping function with nonlinear multikernel functions.

TABLE I
COMMONLY USED ACTIVATION FUNCTION FOR GENERATING THE ENHANCEMENT FEATURES BASED ON MAPPING FEATURES

Sigmoid	$\mathbf{H}(\mathbf{W}, \boldsymbol{\beta}, \mathbf{Z}^n) = \frac{1}{1+\exp(-(\mathbf{W}\cdot\mathbf{Z}^n+\boldsymbol{\beta}))}$
Tansigmoid	$\mathbf{H}(\mathbf{W}, \boldsymbol{\beta}, \mathbf{Z}^n) = \frac{1-\exp(2\cdot(-(\mathbf{W}\cdot\mathbf{Z}^n+\boldsymbol{\beta})))}{1+\exp(2\cdot(-(\mathbf{W}\cdot\mathbf{Z}^n+\boldsymbol{\beta})))}$
Gaussian	$\mathbf{H}(\mathbf{W}, \boldsymbol{\beta}, \mathbf{Z}^n) = \exp(-\boldsymbol{\beta}\Vert\mathbf{Z}^n - \mathbf{W}\Vert)$
Tanh	$\mathbf{H}(\mathbf{W}, \boldsymbol{\beta}, \mathbf{Z}^n) = \frac{1-\exp(-(\mathbf{W}\cdot\mathbf{Z}^n+\boldsymbol{\beta}))}{1+\exp(-(\mathbf{W}\cdot\mathbf{Z}^n+\boldsymbol{\beta}))}$
Tribas	$\mathbf{H}(\mathbf{W}, \boldsymbol{\beta}, \mathbf{Z}^n) = max(1 - \lvert\mathbf{W}\cdot\mathbf{Z}^n + \boldsymbol{\beta}\rvert, 0)$
ReLU	$\mathbf{H}(\mathbf{W}, \boldsymbol{\beta}, \mathbf{Z}^\mathbf{n}) = max(0, \mathbf{W}\cdot\mathbf{Z}^n + \boldsymbol{\beta})$

B. Basic BLS

In the second stage of BLS learning, the weights between the hidden and output layers will be calculated. The outputs of the enhancement nodes can be denoted as $\mathbf{H}^m = [\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_m]$. The mapping feature nodes and enhancement nodes will be concatenated into a matrix, and the final BLS could be denoted by

$$\begin{aligned}\mathbf{Y} &= [\mathbf{Z}_1, \ldots, \mathbf{Z}_n|\mathbf{H}_1, \ldots, \mathbf{H}_m]\mathbf{W}^m \\ &= \big[\mathbf{Z}^n|\mathbf{H}^m\big]\mathbf{W}^m.\end{aligned} \tag{3}$$

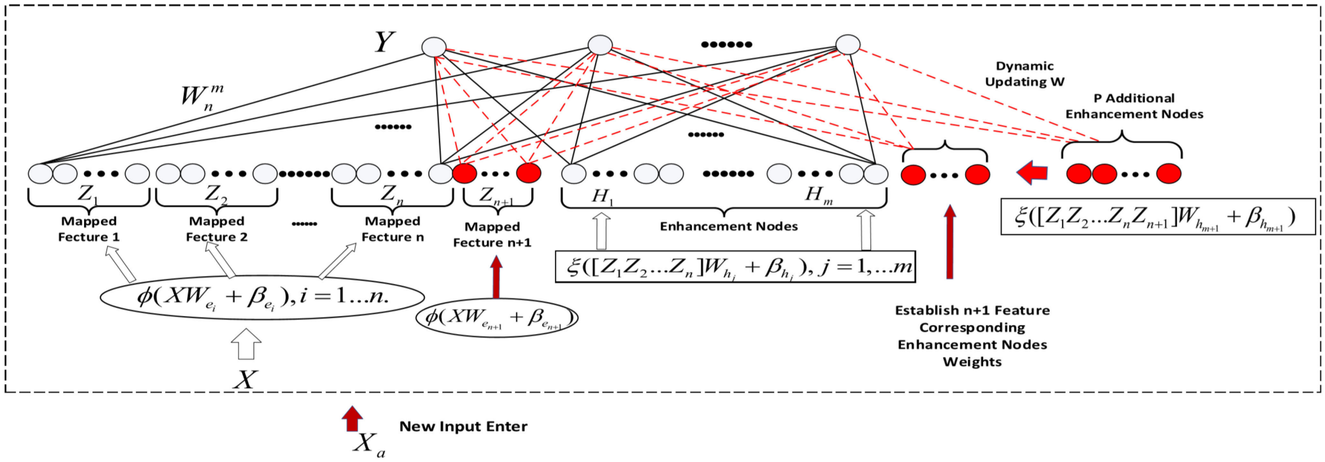


Fig. 2. Dynamic expansion of BLS, including the increment of additional enhancement nodes, feature mapping nodes, and input samples.

As we know, the pseudoinverse algorithm can be considered as a convenient method to obtain the output-layer weights in a random-weight flatted neural network. So, the output-layer weights can be computed rapidly as

$$\mathbf{W}^m = \left[\mathbf{Z}^n|\mathbf{H}^m\right]^+\mathbf{Y}. \tag{4}$$

However, it is too expensive to calculate this generalized inverse duo to the dimension and velocity in training data through some standard methods, such as orthogonal projection, iterative, and SVD. There is an alternative way to solve the pseudoinverse

$$\arg\min_{\mathbf{W}} : \left\|\mathbf{AW}-\mathbf{Y}\right\|_v^{\sigma_1} + \lambda\left\|\mathbf{W}\right\|_u^{\sigma_2} \tag{5}$$

where $\sigma_1 > 0$ and $\sigma_2 > 0$, and v and u are norm regularization. Adding a constraint term λ to the original least squares estimate $\mathbf{A} = [\mathbf{Z}^n|\mathbf{H}^m]$ to make it possible to find the pseudoinverse when the original generalized inverse is under the ill condition. The inverse problem degenerates into the least square problem when $\lambda = 0$ and leads the solution to original pseudoinverse. By taking $\sigma_1 = \sigma_2 = u = v = 2$, the above optimal problem is setting as ridge regression learning Algorithms. Ridge regression is a biased estimate of the regression parameter $\mathbf{A}$. Its result is to make the residuals larger, but it will make the model more generalized. So, the connection weights of the BLS can be approximated as

$$\mathbf{W}^m = \left(\lambda\mathbf{I} + \mathbf{AA}^T\right)^{-1}\mathbf{A}^T\mathbf{Y}. \tag{6}$$

We specifically have

$$\mathbf{A}^+ = \lim_{\lambda\to 0}\left(\lambda\mathbf{I} + \mathbf{AA}^T\right)^{-1}\mathbf{A}^T. \tag{7}$$

In addition, the algorithm of BLS is shown in Algorithm 1.

Algorithm 1: Basic BLS [17]

Input: training samples $\mathbf{X}$; number of feature maps n; number of groups of enhancement nodes m;
Output: $\mathbf{W}$

```
for i = 1; i ≤ n do
    Random W_{e_i}, β_{e_i};
    Calculate Z_i = φ_i(X W_{e_i} + β_{e_i});
end
Set the feature mapping group Z^n = [Z_1, Z_2, ..., Z_n];
for j = 1; j ≤ m do
    Random W_{h_i}, β_{h_i};
    Calculate H_j = ξ_j(Z^n W_{h_j} + β_{h_j});
end
Set the enhancement nodes group
    H^m = [H_1, H_2, ..., H_m];
Set A and calculate A^+ with Eq. (7);
Calculate W^m with Eq. (6);
Set W = W^m.
```

C. Incremental BLS

Based on the dynamic stepwise updating algorithm for functional-link neural networks, BLS can take advantage of it to develop various incremental learning for fast remodeling without suffering from the complex retraining process. Three different scenarios, including the incremental enhancement nodes, mapping feature nodes, and new input data, can satisfy the various requirements and applications. The dynamic version for incremental learning of BLS is shown in Fig. 2. Next, we will introduce three different incremental BLSs.

1) Increment of Additional Enhancement Nodes: In many scenarios, the trained model cannot achieve satisfactory results. At this time, BLS can dynamically increase the number of hidden layer nodes of enhancement to achieve better learning capabilities. This incremental mechanism eliminates the need to retrain the entire model and only needs to calculate the weight of newly additional enhancement nodes to measure up the purpose of the rapid reconstruction for the model.

Assume that q enhancement nodes are added to BLS for increasing the performance. Denote $\mathbf{A} = [\mathbf{Z}^n|\mathbf{H}^m]$, so that the input matrix with new q enhancement nodes $\mathbf{A}^{m+1}$ can be represented as

$$\mathbf{A}^{m+1} = \left[\mathbf{A}^m|\xi\left(\mathbf{Z}^n\mathbf{W}_{h_{m+1}} + \boldsymbol{\beta}_{h_{m+1}}\right)\right] \tag{8}$$

where $W_{h_{m+1}}, \boldsymbol{\beta}_{h_{m+1}}$ are randomly generated weight and biases for q additional enhancement nodes. Therefore, the pseudoinverse of this incremental BLS matrix could be deduced as

$$\left(\mathbf{A}^{m+1}\right)^{+} = \begin{bmatrix} (\mathbf{A}^{m})^{+} - \mathbf{D}\mathbf{B}^{T} \\ \mathbf{B}^{T} \end{bmatrix} \tag{9}$$

where

$$\mathbf{D} = \left(\mathbf{A}^{m}\right)^{+} \xi\left(\mathbf{Z}^{n}\mathbf{W}_{h_{m+1}} + \boldsymbol{\beta}_{h_{m+1}}\right) \tag{10}$$

$$\mathbf{B}^{T} = \begin{cases} \mathbf{C}^{+} & \text{if } \mathbf{C} \neq 0 \\ \left(\mathbf{1} + \mathbf{D}^{T}\mathbf{D}\right)^{-1}\mathbf{D}^{T}(\mathbf{A}^{m})^{+} & \text{if } \mathbf{C} = 0 \end{cases} \tag{11}$$

and

$$\mathbf{C} = \xi\left(\mathbf{Z}^{n}\mathbf{W}_{h_{m+1}} + \boldsymbol{\beta}_{h_{m+1}}\right) - \mathbf{A}_{n}^{m}\mathbf{D}. \tag{12}$$

Finally, the new weights and biases connected with the hidden layer and output layer can be calculated by

$$\mathbf{W}^{m+1} \triangleq \left(\mathbf{A}^{m+1}\right)^{+}\mathbf{Y} = \begin{bmatrix} \mathbf{W}^{m} - \mathbf{D}\mathbf{B}^{T}\mathbf{Y} \\ \mathbf{B}^{T}\mathbf{Y} \end{bmatrix}. \tag{13}$$

We notice that this incremental version algorithm of BLS can reconstitute the model only need to compute the pseudoinverse of the new additional enhancement instead of the entire weights.

2) Increment of the Feature Mapping Nodes: Some discriminant models may be affected by problems, such as insufficient features or poor feature mapping. The feature node or mapping method failed to select enough underlying variation factors and was unable to transfer the input data to a better feature space. Some neural networks, especially deep learning models, retrain the entire model by modifying the number of layers or adding the number of filters or layers to achieve enhanced model expression [41]–[43]. However, these methods require manual experience and the entire reconstruction process, which may consume much time. It does not effectively improve the capability of a network to find suitable features. BLS can quickly increase the mapping feature nodes for containing the representation of the input data. It will successfully transfer the input data to a more suitable feature space. At the same time, adding new feature mapping nodes can use an incremental algorithm similar to the above, adding the enhancement nodes, which can easily update the weights without retraining the entire network.

Here, let us consider the algorithm for newly incremental mapping feature nodes. Following Section III-B, we denote the number of mapping feature group and enhancement group as n and m in basic BLS, respectively. Assume that the $(n+1)$th group of mapping feature nodes are added and denoted by

$$\mathbf{Z}_{n+1} = \phi_{n+1}\left(\mathbf{X}\mathbf{W}_{e_{n+1}} + \boldsymbol{\beta}_{e_{n+1}}\right) \tag{14}$$

naturally, the output of related enhancement nodes can be calculated as

$$\mathbf{H}_{ex_m} = \left[\xi\left(\mathbf{Z}_{n+1}\mathbf{W}_{ex_1} + \boldsymbol{\beta}_{ex_1}\right), \ldots, \xi\left(\mathbf{Z}_{n+1}\mathbf{W}_{ex_m} + \boldsymbol{\beta}_{ex_m}\right)\right] \tag{15}$$

where $\mathbf{W}_{ex_i}$ and ξ_{ex_i} are randomly generated for computing the output of enhancement nodes when newly group of feature nodes is added into BLS. Then, denote $\mathbf{A}_{n+1}^{m} = [\mathbf{A}_{n}^{m}|\mathbf{Z}_{n+1}|\mathbf{H}_{ex_m}]$. Based on these, the pseudoinverse matrix could be computed as

$$\left(\mathbf{A}_{n+1}^{m}\right)^{+} = \begin{bmatrix} \left(\mathbf{A}_{n}^{m}\right)^{+} - \mathbf{D}\mathbf{B}^{T} \\ \mathbf{B}^{T} \end{bmatrix} \tag{16}$$

where

$$\mathbf{D} = \left(\mathbf{A}_{n}^{m}\right)^{+}\left[\mathbf{Z}_{n+1}|\mathbf{H}_{ex_m}\right] \tag{17}$$

$$\mathbf{B}^{T} = \begin{cases} (\mathbf{C})^{+} & \text{if } \mathbf{C} \neq 0 \\ \left(\mathbf{1} + \mathbf{D}^{T}\mathbf{D}\right)^{-1}\mathbf{D}^{T}\left(\mathbf{A}_{n}^{m}\right)^{+} & \text{if } \mathbf{C} = 0 \end{cases} \tag{18}$$

and

$$\mathbf{C} = \left[\mathbf{Z}_{n+1}|\mathbf{H}_{ex_m}\right] - \mathbf{A}^{m}\mathbf{D}. \tag{19}$$

Again, the new weights with new additional mapping feature nodes can be calculated by

$$\mathbf{W}_{n+1}^{m} \triangleq \left(\mathbf{A}_{n+1}^{m}\right)^{+}\mathbf{Y} = \begin{bmatrix} \mathbf{W}^{m} - \mathbf{D}\mathbf{B}^{T}\mathbf{Y} \\ \mathbf{B}^{T}\mathbf{Y} \end{bmatrix}. \tag{20}$$

3) Increment of Input Data: In actual applications, the amount of data tends to increase gradually. When the model is faced with new data, an excellent learning method should be able to make some simple changes to the trained system to learn the knowledge contained in the new data [44], [45]. At the same time, incremental learning requires that the time cost of modifying a trained model is usually lower than the cost of retraining a system. BLS can conveniently handle the situation of the input training samples that keep entering.

Denote $\{\mathbf{X_a}, \mathbf{Y_a}\}$ as the new inputs added into the model. In addition, the initial BLS consists of n groups of mapping feature nodes and m groups of enhancement nodes. $\mathbf{A}_{n}^{m}$ represents as the related expanded input matrix. So, we can first obtain the mapping feature that can be denoted as

$$\mathbf{Z}_{\mathbf{a}}^{n} = \left[\phi\left(\mathbf{X_a}\mathbf{W}_{e_1} + \boldsymbol{\beta}_{e_1}\right), \ldots, \phi\left(\mathbf{X_a}\mathbf{W}_{e_n} + \boldsymbol{\beta}_{e_n}\right)\right] \tag{21}$$

then the matrix of output that collects the mapping feature and enhancement nodes for a new additional training sample is denoted as follows:

$$\mathbf{A}_{x} = \left[\mathbf{Z}_{\mathbf{a}}^{n}|\xi\left(\mathbf{Z}_{x}^{n}\mathbf{W}_{h_1} + \boldsymbol{\beta}_{h_1}\right), \ldots, \xi\left(\mathbf{Z}_{x}^{n}\mathbf{W}_{h_m} + \boldsymbol{\beta}_{h_m}\right)\right]. \tag{22}$$

Hence, the input matrix of the newly additional training sample could be calculated as

$${}^{x}\mathbf{A}_{n}^{m} = \begin{bmatrix} \mathbf{A}_{n}^{m} \\ \mathbf{A}_{x}^{T} \end{bmatrix}. \tag{23}$$

The associated pseudoinverse of ${}^{x}\mathbf{A}_{n}^{m}$ can be could be deduced as

$$\left({}^{x}\mathbf{A}_{n}^{m}\right)^{+} = \left[\left(\mathbf{A}_{n}^{m}\right)^{+} - \mathbf{B}\mathbf{D}^{T}\middle|\mathbf{B}\right] \tag{24}$$

where

$$\mathbf{D}^{T} = \mathbf{A}_{x}^{T}\left(\mathbf{A}_{n}^{m}\right)^{+} \tag{25}$$

$$\mathbf{B}^{T} = \begin{cases} (\mathbf{C})^{+} & \text{if } \mathbf{C} \neq 0 \\ \left(\mathbf{1} + \mathbf{D}^{T}\mathbf{D}\right)^{-1}\left(\mathbf{A}_{n}^{m}\right)^{+}\mathbf{D}^{T} & \text{if } \mathbf{C} = 0 \end{cases} \tag{26}$$

and

$$\mathbf{C} = \mathbf{A}_{x}^{T} - \mathbf{D}^{T}\mathbf{A}_{n}^{m}. \tag{27}$$

Finally, the new connection weight between the hidden layer and output layer for the newly additional training data can be calculated as

$$ {}^{x}\mathbf{W}_n^m = \mathbf{W}_n^m + \left(\mathbf{Y}_a^T - \mathbf{A}_x^T \mathbf{W}_n^m\right)\mathbf{B} \tag{28} $$

where $\mathbf{Y_a}$ is the related labels of newly added training data. As mentioned before, BLS does not need to rebuild all knowledge bases when faced with new training input samples. Instead, based on the original knowledge base, it only makes updates due to new data, which is more in line with human thinking principles. At the same time, BLS can quickly calculate new weights without rebuilding the entire neural network.

Moreover, Zhang *et al.* [46] gave an analysis of the principle of fast incremental learning ability of BLS. Four lemmas about the pseudoinverse of matrix $\mathbf{A}$ were listed that can help to prove the incremental learning of BLS.

D. Essences of BLS

From a learning point of view, the essence of the original BLS is aimed at satisfying several salient targets simultaneously [17], [18], which is different from the traditional learning algorithm, such as the perceptron [47], feedforward neural networks [48], backpropagation algorithm [14].

1) *Discriminative Feature Space:* Traditional neural-network learning principles need to adjust the weights of the hidden layer during the training process. This black-box program is generally not known to users and lacks theoretical interpretability. However, BLS can map the original samples into a discriminative feature space spanned by these vectors by utilizing random hidden layer weights. The hidden layer nodes' parameters can be randomly generated according to any continuous probability distribution, which has been proved theoretically. This random mechanism can quickly provide BLS with a fast training process.
2) *Universal Approximation Capability:* Usually, the neural-network architecture and its corresponding algorithms may not consider the universal approximation capability. However, BLS has theoretically analyzed its universal approximation capability at the beginning of the design, which has also been strictly theoretically proved.
3) *Unified Learning Theory:* There should be a unified learning architecture for the designed "generalized" network of BLS. This network can also have corresponding variants based on the original structure, including matrix multiplication nodes, fuzzy rules, manifold embedding, ridge polynomials, multikernel functions, wavelet function, etc.
4) *Generalization Performance:* Traditional SLFN are trained through gradient-descent learning algorithms. The SLFN usually converge slowly and trap in a local minimum. Therefore, their generalization performance is more sensitive to parameter settings, such as the learning rate. BLS aims to reach better generalization performance by minimizing the smallest training error and norm of output weight based on (5). In addition, other regularization methods also can be applied in this loss function, such as sparse and graph regularization.
5) *Incremental Learning Without Reconstructing the Entire Model:* Once the neural-network weights based on the BP algorithm are trained over, in the face of newly added samples or hidden nodes, the entire model needs to be retrained. It will take much time, which suffered from lacking flexibility. However, BLS can be quickly reconstructed through three different incremental methods, including the increment of additional enhancement nodes, the increment of the feature mapping nodes, and the increment of input data. These novel incremental learning methods can enable BLS to remodel without updating the entire network's parameters quickly.

E. Regularized BLS

Generally speaking, a good machine-learning algorithm needs to satisfy two demands: it can not only learn the knowledge of training data well but also have good generalization ability [49]–[51], which can be explained from the perspective of Occam's razor principle [52]. There will always be some noise in the dataset, so the model may take the noise into account when fitting the data, resulting in a reduction in the generalization ability of the model. How to use regularization to avoid the overfitting of the model during the learning process and improve the generalization ability of the model has always been the focus of research.

In standard BLS, the weight connected with the output layer and hidden layer can be solved by ridge regression approximation with $u = v = \sigma_1 = \sigma_2 = 2$

$$ \arg\min_{\mathbf{W}}: \ \|\mathbf{AW} - \mathbf{Y}\|_2^2 + \lambda\|\mathbf{W}\|_2^2 \tag{29} $$

where the first term denotes the training errors and the second term is a regularization term. λ is the tradeoff regularization parameter.

Jin and Chen [53] Jin *et al.* [54] primarily studied the stability and generalization performance of BLS and proposed the novel regularized BLS to control the complexity and improve the performance of the original BLS. Since some outliers do not follow the Gaussian distribution in most practical applications, regularized BLS introduces the widely used Laplacian distribution information to measure errors and can be denoted as

$$ \arg\min_{\mathbf{W}}: \ \|\mathbf{AW} - \mathbf{Y}\|_1 + \lambda_1\|\mathbf{W}\|_2^2 \tag{30} $$

where $\|\cdot\|_1$ is the l_1-norm [55]. The l_2 regularizer encourages the weight to be smooth. In addition, the l_1 regularizer has been took into consideration for sparsity when $\mathbf{W}$ is measured by the Laplacian function

$$ \arg\min_{\mathbf{W}}: \ \|\mathbf{AW} - \mathbf{Y}\|_1 + \lambda_2\|\mathbf{W}\|_1. \tag{31} $$

In order to take full advantage of l_1 and l_2 regularization, Jin *et al.* further proposed an elastic regularized BLS, which can be built in following formulation:

$$ \arg\min_{\mathbf{W}}: \ \|\mathbf{AW} - \mathbf{Y}\|_1 + \lambda_1\|\mathbf{W}\|_1 + \lambda_2\|\mathbf{W}\|_2 \tag{32} $$

intuitively, the objective function cannot be calculated directly and can be tackled by an iterative optimization, called the augmented lagrange multiplier (ALM) method [56], [57], which can be denoted as

$$\mathcal{L}(\mathbf{E}, \mathbf{Q}, \mathbf{W}, \mathbf{C}_1, \mathbf{C}_2) = \|\mathbf{E}\|_1 + \lambda_1\|\mathbf{Q}\|_1 + \lambda_2\|\mathbf{W}\|_2^2 + \frac{\mu}{2}\left\|\mathbf{Y} - \mathbf{AW} - \mathbf{E} + \frac{\mathbf{C}_1}{\mu}\right\|_2^2 + \frac{\mu}{2}\left\|\mathbf{Q} - \mathbf{W} + \frac{\mathbf{C}_2}{\mu}\right\|_2^2 \tag{33}$$

where $\mathbf{E} = \mathbf{Y} - \mathbf{AW}$ and $\mathbf{Q} = \mathbf{W}$. $\mathbf{C}_1$ and $\mathbf{C}_2$ are the Lagrange multipliers and $\mu \geq 0$ is also the penalty term. These hyper parameters can be optimized through iteration.

The above-mentioned regularized BLS is designed under the condition that the feature mapping nodes and the enhancement nodes remain unchanged. Gan *et al.* [58] designed a method based on weighted generalized cross-validation to find the regularization parameters when the nodes change dynamically.

IV. UNIVERSAL APPROXIMATION CAPABILITY

The universal approximation capability of SLFN's has been well proved [59]. But, its network parameters need to be adjusted continuously during training, and its assumption that the activation function must be continuous and differentiable. In addition, Igelnik and Pao [60] have well studied the universal approximation capability of RVFL. It has been shown that RVFL can eliminate the disadvantage of a long training process compared to gradient-descent-based SLFN and also its universal approximation capability has been proved for continuous functions on compact sets [13].

Inspired by RVFL, Chen *et al.* [18] gave a detailed proof for the universal approximation capability of BLS. It is shown that BLS is a universal learner for continuous functions on compact sets even with randomly generated mapping features nodes and enhancement features nodes.

The universal approximation ability of BLS is in the form of probabilistic convergence. Give a continuous function $f \in C(\mathbf{I}^d)$, where f is defined on the standard hypercube $\mathbf{I}^d = [0; 1]^d \subset \mathbb{R}^d$. As we can find in [60], an RVLF with one hidden-layer network can be denoted as

$$f_n(x) = \sum_{k=1}^{n} a_k g(w_k x + b_k) \tag{34}$$

where $a_k, b_k \in R, w_k \in R^d$, and $x \in I^d$. There are two functions when BLS generates the mapping feature nodes and enhancement feature nodes, where $\phi(\cdot)$ is a nonconstant-bounded mapping feature function and $\zeta(\cdot)$ is the activation function. With these two functions, BLS can equivalently be denoted as

$$f_{\mathbf{w}_{m,n}}(\mathbf{x}) = \sum_{i=1}^{n*k} w_i\phi\big(\mathbf{x}\mathbf{w}_{e_i} + \beta_{e_i}\big) + \sum_{j=1}^{m*q} w_{nk+j}\xi\big(\mathbf{z}\mathbf{w}_{h_j} + \beta_{h_j}\big)$$
$$= \sum_{i=1}^{n*k} w_i\phi\big(\mathbf{x}\mathbf{w}_{e_i} + \beta_{e_i}\big) + \sum_{j=1}^{m*q} w_{nk+j}\xi\big(\mathbf{x}; \{\phi, \mathbf{w}_{h_j}, \beta_{h_j}\}\big) \tag{35}$$

where $\mathbf{z} = [\phi(\mathbf{x}\mathbf{w}_{e_1} + \beta_{e_1}), \ldots, \phi(\mathbf{x}\mathbf{w}_{e_{nk}} + \beta_{e_{nk}})]$ is the all feature nodes. In addition, $\mathbf{w}_{m,n} = (n, m, w_1, \ldots, w_{nk+mq}, \mathbf{w}_{e_1}, \ldots, \mathbf{w}_{e_{nk}}, \mathbf{w}_{h_1}, \ldots, \mathbf{w}_{h_{mq}}, \beta_{e_1}, \ldots, \beta_{e_{nk}}, \beta_{h_1}, \ldots, \beta_{h_{mq}})$ is an overall parameter of the functional-link neural network. In addition, the weight and bias between input data and one hidden layer, including mapping feature and enhancement feature nodes, are randomly generated which is denoted as $\lambda_{m,n} = (\mathbf{w}_{e_1}, \ldots, \mathbf{w}_{e_{nk}}, \mathbf{w}_{h_1}, \ldots, \mathbf{w}_{h_{mq}}, \beta_{e_1}, \ldots, \beta_{e_{nk}}. \ .\beta_{h_1}, \ldots, \beta_{h_{mq}})$. Assume that $\lambda_{m,n}$ is defined on the probability measure $\mu_{m,n}$ and E is a symbol of expectation with respect to $\mu_{m,n}$. From this, the distance between the continuous function $f(\cdot)$ and the approximation function structured by BLS $f_{\mathbf{w}_{m,n}}$ on any compact set $\mathbf{K}$ can be denoted as

$$\rho_{\mathbf{K}}\big(f, f_{w_{m,n}}\big) = \sqrt{E\left[\int_{\mathbf{K}}\big(f(\mathbf{x}) - f_{w_{m,n}}(\mathbf{x})\big)^2 d\mathbf{x}\right]} \tag{36}$$

where $\mathbf{K} \subset \mathbf{I}^d$. The above shows that the universal approximation ability of BLS based on the BVFL net is proved by solving the distance between the approximate function and the continuous function. So, the main result will be described as follows.

Theorem 1: For any compact $\mathbf{K}$, $\mathbf{K} \subset \mathbf{I}^d$, $\mathbf{K} \neq \mathbf{I}^d$ and any continuous function $f, f \in C(\mathbf{I}^d)$. There exists a sequence of BLS $f_{\mathbf{w}_{m,n}}$, which is constructed by nonconstant bounded mapping feature function ϕ and absolutely integrable activation function ζ, such that

$$\int_{\mathbb{R}^d} \xi^2(\mathbf{x}) d\mathbf{x} < \infty \tag{37}$$

and a respective sequence of probability measures $\mu_{m,n}$. In addition, the parameters $\lambda_{m,n}$ that is generated randomly are sampled from the distributions of probability measures $\mu_{m,n}$ such that

$$\lim_{m,n\to\infty} \rho_{\mathbf{K}}\big(f, f_{\mathbf{w}_{m,n}}\big) = 0. \tag{38}$$

The detailed information of the proof process can be found in [18]. From the above theorem, it is stated that BLS is a universal nonlinear function approximator for any continuous function on any compact set $\mathbf{K}$, $\mathbf{K} \subset \mathbf{I}^d$ and $\mathbf{K} \neq \mathbf{I}^d$. Furthermore, it should be known that the random parameters sampled from the distributions of probability measures $\mu_{m,n}$ are simple in generating. Next, it will give a corollary to extend the range of BLS's activation function to Sigmoid.

Corollary 1: For any compact $\mathbf{K}$, $\mathbf{K} \subset \mathbf{I}^d$ and $\mathbf{K} \neq \mathbf{I}^d$ and any continuous function $f, f \in C(\mathbf{I}^d)$. There exists a sequence of BLS $f_{\mathbf{w}_{m,n}}$, which is constructed by nonconstant bounded mapping feature function ϕ and any differentiable activation function ζ, such that

$$\int_R \big[g'(x)\big]^2 dx < \infty \tag{39}$$

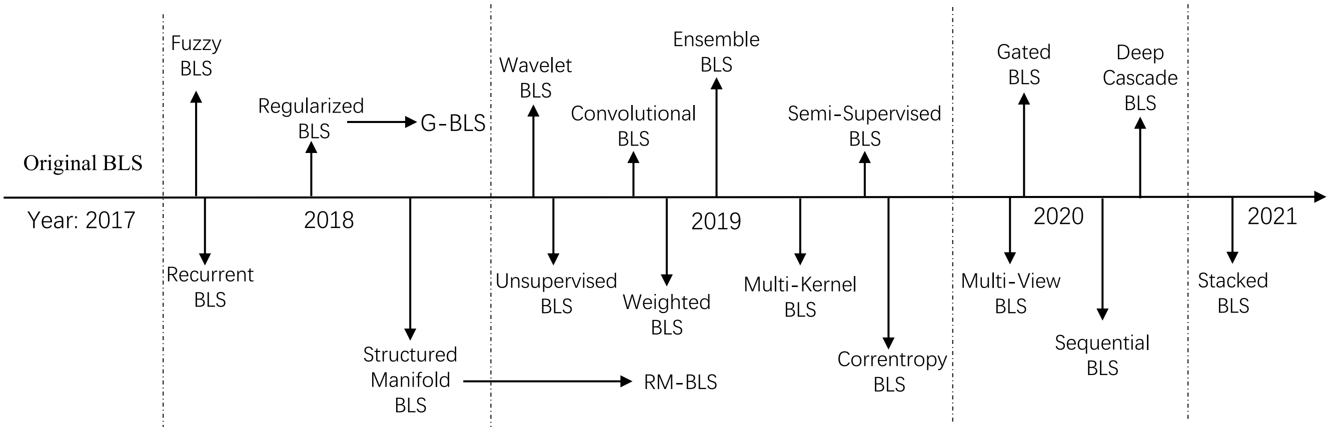


Fig. 3. Road map of BLSs. Milestone variants are shown in this figure.

and a respective sequence of probability measures $\mu_{m,n}$. In addition, the parameters $\lambda_{m,n}$, which are generated randomly, are sampled from the distributions of probability measures $\mu_{m,n}$ such that

$$\lim_{m,n\to\infty} \rho_{\mathbf{K}}\big(f, f_{\mathbf{w}_{m,n}}\big) = 0. \tag{40}$$

From Corollary 1, we can see that any activation function that can be used empirically in the neural-network computing can be used in BLS as well, such as kernel mappings, nonlinear transformations, or convolutional functions.

Corollary 2: For any compact $\mathbf{K}$, $\mathbf{K} \subset \mathbf{I}^d$, and $\mathbf{K} \neq \mathbf{I}^d$ and any measurable function f, $f \in C(\mathbf{I}^d)$. There exists a sequence of BLS $f_{\mathbf{w}_{m,n}}$, which is constructed by nonconstant bounded mapping feature function ϕ and absolutely integrable activation function ζ, such that

$$\int_{\mathbb{R}^d} \xi^2(\mathbf{x}) d\mathbf{x} < \infty \tag{41}$$

and a respective sequence of probability measures $\mu_{m,n}$. In addition, the parameters $\lambda_{m,n}$, which are generated randomly, are sampled from the distributions of probability measures $\mu_{m,n}$ such that

$$\lim_{m,n\to\infty} \rho_{\mathbf{K}}\big(f, f_{\mathbf{w}_{m,n}}\big) = 0. \tag{42}$$

As can be found in [60], there exists a continuous function g for any $f \in L^2(\mathbf{K})$, such that

$$||f - g|| < \epsilon. \tag{43}$$

The above theory shows that BLS is a universal function approximator. The pseudoinverse algorithm can calculate the output layer of BLS, and the weights of the layers other than the output layer are randomly generated. Proving the universal approximation ability of this functional-link neural network is not a simple matter. Moreover, what kind of distribution is used to generate weights has always been a complicated problem, and it is a direction for future research.

V. BLS for Classification and Regression

Since the birth of BLS, it has attracted significant interest and attention of many researchers. This section summarizes various extensions of classical BLS for classification and regression problems. In addition, there are many BLSs' variants, and milestone ones are shown in Fig. 3. Due to space limitations, we only show a limited number of variants.

A. Improving the Compactness of BLS

Due to the characteristics of its structure, BLS usually achieves the purpose of increasing the performance of the network by mainly increasing the number of hidden-layer nodes compared to conventional architectures. This way of randomly generating the hidden-layer nodes and weights makes BLS face the risk of a relatively large network scale and increasing the response time in some sensitive scenarios. Therefore, many studies have focused on improving the compactness of the BLS framework in recent years, which provides reliable suggestions for reducing the number of parameters and calculation time for BLS.

One intuitive way is to prune or replace hidden neurons during the training stage. The hidden neurons of BLS have different information. Only neurons with sufficient information can be added to the network. Reasonably reducing the number of irrelevant neurons can sufficiently reduce the network size and the number of parameters. Thus, Huang and Chen [61] proposed a novel bidirectional BLS (B-BLS) which can reduce the network parameters of BLS. As the B-BLS network grows, some hidden nodes will be effectively incrementally updated. B-BLS can improve learning efficiency by reducing the number of nodes. Experiments show that in some applications, B-BLS can significantly reduce the training time is compared to the standard BLS.

Furthermore, Zhang *et al.* [46] proposed a novel modified BLS that has a cascade of enhancement nodes with dense connections (CEBLS-dense). Fig. 4 shows the architecture of the CEBLS-dense. CEBLS-dense can modularize the enhancement node. The input of the first enhancement node of each module is composed of all functional nodes and the last enhancement node of the previous module. At the same time, the last node of each module is fed into the output, which can effectively avoid the risk of redundant information and overfitting. Thanks to this dense structure, CEBLS-dense

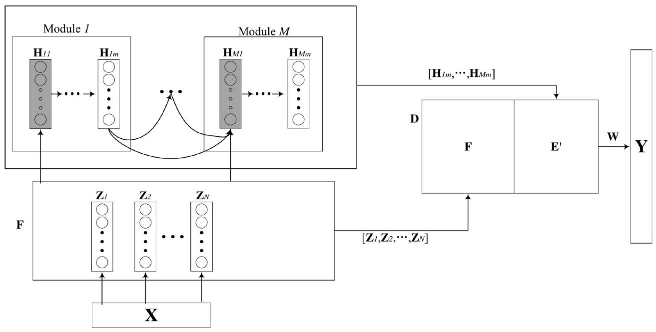


Fig. 4. Structure of CEBLS-dense: cascade of enhancement nodes with dense connections proposed by Zhang *et al.* [46].

can reduce training time by 43% and 48% in the MNIST and Fashion-MNIST datasets compared to the original BLS, respectively.

The above description illustrates the truth that reducing parameters and response time has essential meanings and applications. We will explain in detail in Section X.

B. Recurrent and Gated BLS for Sequential Data

Due to the existence of hidden state units in its structure, the recurrent neural network (RNN) can remember the information of the previous input data [62]. This inborn structural advantage makes the RNN more effective for processing sequence data, including time-series signals and text sequence, which can mine timing information and semantic information.

Chen *et al.* [18] initially considered two possible improvements, which have capable of processing sequence signals based on BLS. First, the cascade of feature map nodes can change it into a structure similar to a recurrent system, called recurrent feature nodes. It can learn sequential knowledge from data and be denoted as

$$\mathbf{Z}_k = \phi\big(\mathbf{Z}_{k-1}\mathbf{W}_{e_k} + \mathbf{X}\mathbf{W}_{z_k} + \boldsymbol{\beta}_{e_i}\big),\ \ k = 1, 2, \ldots, n \quad (44)$$

where matrices $\mathbf{W}_{z_k}$, $\mathbf{W}_{e_k}$, and $\boldsymbol{\beta}_{e_k}$ are randomly generated. Specifically, in this model, with recursive feature mapping nodes, the value of each $\mathbf{Z}_k$ will be affected by the previous value $\mathbf{Z}_{k-1}$ and the input sample $\mathbf{X}$ simultaneously.

Based on this variant, Du *et al.* [63] proposed a novel recurrent BLS (Recurrent-BLS) shown in Fig. 5(a). Furthermore, it has been expanded to LSTM-BLS by adding corresponding gatings (Gated-BLS) which are shown in Fig. 5(b). Recurrent-BLS and Gated-BLS are used in real-world experiments for text classification in the domain of natural language processing. Experiments show that the well-designed framework can learn sequence information and word importance at the same time, which has better results in terms of accuracy and training time.

As we can see that Recurrent-BLS can store word sequence information by computing recursive mapping nodes based on (44)

$$\mathbf{Z}_k = f\big(\mathbf{X}_k\boldsymbol{\alpha}^z + \mathbf{Z}_{k-1}\mathbf{U} + \boldsymbol{\beta}^z\big) \quad (45)$$

where the weights $\boldsymbol{\alpha}^z$ and $\mathbf{U}$ and bias $\boldsymbol{\beta}^z$ are randomly generated. Following, the enhancement nodes can be used to capture

Algorithm 2: Recurrent-BLS [63]

Input: The matrix representation of pth word in all N training samples: $\mathbf{X}_p$, $p = 1$ to k; and the label matrix for all N training samples: $\mathbf{Y}$;

Output: $\mathbf{W}$

Random α^z, $\mathbf{U}$ and β^z;
Set $\mathbf{Z}_0$ to all zeros;
for $p = 0; p \le k$ **do**
  Calculate $\mathbf{Z}_p = f(\mathbf{X}_p\boldsymbol{\alpha}^z + \mathbf{Z}_{p-1}\mathbf{U} + \boldsymbol{\beta}^z)$;
end
Obtain the final $\mathbf{Z}_k$;
Artificially set l for enhancement node $\mathbf{H}_p$;
Random α^h and β^h;
for $p = 1; p \le k$ **do**
  Calculate $\mathbf{H}_p = g(\mathbf{X}_p\boldsymbol{\alpha}^h + \boldsymbol{\beta}^h)$;
end
Calculate connecting weights:
$\mathbf{W} = \big([\mathbf{Z}_k|\mathbf{H}_1, \cdots, \mathbf{H}_k]^T[\mathbf{Z}_k|\mathbf{H}_1, \cdots, \mathbf{H}_k]\big)^{-1}$ $[\mathbf{Z}_k|\mathbf{H}_1, \cdots, \mathbf{H}_k]^T\mathbf{Y}$

the information of word importance by taking the original words $\mathbf{X}_p$ as input rather than the mapping feature nodes. So, each enhancement node $\mathbf{H}_p$ can be calculated by

$$\mathbf{H}_p = g\Big(\mathbf{X}_p\boldsymbol{\alpha}^h + \boldsymbol{\beta}^h\Big) \quad (46)$$

where $g(\cdot)$ is the activation function. Then, the last state of the mapping feature for learning sequence information and enhancement nodes for learning the word importance will be simultaneously collected together as

$$\mathbf{A} = [\mathbf{Z}_k|\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_k]. \quad (47)$$

Finally, weight $\mathbf{W} = [\mathbf{W}_z|\mathbf{W}_h]$ connected the output layer and hidden layer can be formulated through ridge regression approximation and obtained by

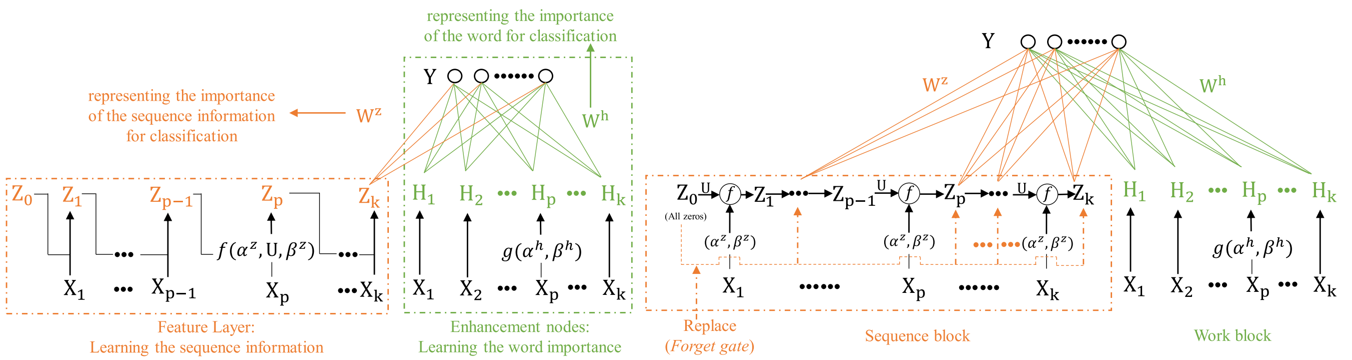
$$\begin{aligned}\mathbf{W} &= [\mathbf{Z}_k|\mathbf{H}_1, \ldots, \mathbf{H}_k]^{+}\mathbf{Y} \\ &= \big([\mathbf{Z}_k|\mathbf{H}_1, \ldots, \mathbf{H}_k]^T[\mathbf{Z}_k|\mathbf{H}_1, \ldots, \mathbf{H}_k]\big)^{-1} \\ &\quad \times [\mathbf{Z}_k|\mathbf{H}_1, \ldots, \mathbf{H}_k]^T\mathbf{Y}. \end{aligned} \quad (48)$$

The process of Recurrent-BLS is shown in Algorithm 2.

The forget gate is designed in Gated-BLS to forget the unrelated information, which is shown in Fig. 5(b). BLS divides the input sentence into several fixed-length short sentences by introducing an additional parameter of c. By adjusting the weights $\mathbf{W}^z$ and c simultaneously, the effectiveness of the control information flow is improved. This gating approach is similar to LSTM. In detail, $\mathbf{Z}_{p+1}$ can be calculated when we obtain $\mathbf{Z}_p$

$$\mathbf{Z}_{p+1} = \begin{cases} f\big(\mathbf{X}_{p+1}\boldsymbol{\alpha}^z + \mathbf{Z}_0\mathbf{U} + \boldsymbol{\beta}^z\big), & \text{if } p \bmod c = 0 \\ f\big(\mathbf{X}_{p+1}\boldsymbol{\alpha}^z + \mathbf{Z}_p\mathbf{U} + \boldsymbol{\beta}^z\big), & \text{else.} \end{cases} \quad (49)$$

Consequently, the sequence information for G-BLS is represented by $\mathbf{Q} = [\mathbf{Z}_p]$. For example, given $c = 2$ and $k = 8$, $\mathbf{Q} = [\mathbf{Z}_2, \mathbf{Z}_4, \mathbf{Z}_6, \mathbf{Z}_8]$. Simply, the process of Gated-BLS is shown in Algorithm 3.



(a) (b)

Fig. 5. BLS structure with recurrent feature nodes proposed by Du *et al.* [63]. (a) Recurrent-BLS. (b) Long short-term memory (LSTM)-like architecture: gated-BLS.

Algorithm 3: LSTM-Like Architectures: Gated-BLS [63]

Input: The matrix representation of pth word in all N training samples: $\mathbf{X}_p$, $p = 1$ to k; and the label matrix for all N training samples: $\mathbf{Y}$;

Output: $\mathbf{W}$

Random α^z, $\mathbf{U}$ and β^z;

Set $\mathbf{Z}_0$ to all zeros;

Artificially set a certain step size c;

for $p = 0; p \leq k$ **do**

  Calculate $\mathbf{Z}_p = f(\mathbf{X}_p\boldsymbol{\alpha}^z + \mathbf{Z}_{p-1}\mathbf{U} + \boldsymbol{\beta}^z)$;

  Calculate:

$$\mathbf{Z}_{p+1} = \begin{cases} f\left(\mathbf{X}_{p+1}\boldsymbol{\alpha}^z + \mathbf{Z}_0\mathbf{U} + \boldsymbol{\beta}^z\right), & \text{if } p \bmod c = 0 \\ f\left(\mathbf{X}_{p+1}\boldsymbol{\alpha}^z + \mathbf{Z}_p\mathbf{U} + \boldsymbol{\beta}^z\right), & \text{else} \end{cases}$$

end

Obtain $\mathbf{Q} = \left[\mathbf{Z}_p\right]$, where p mod c=0;

Artificially set l for enhancement node $\mathbf{H}_p$;

Random α^h and β^h;

for $p = 1; p \leq k$ **do**

  Calculate $\mathbf{H}_p = g(\mathbf{X}_p\boldsymbol{\alpha}^h + \boldsymbol{\beta}^h)$;

end

Calculate connecting weights:

$\mathbf{W} = \left([\mathbf{Q}_k|\mathbf{H}_1, \cdots, \mathbf{H}_k]^T[\mathbf{Q}_k|\mathbf{H}_1, \cdots, \mathbf{H}_k]\right)^{-1} [\mathbf{Q}_k|\mathbf{H}_1, \cdots, \mathbf{H}_k]^T\mathbf{Y}$

Another change is to transform the cascading enhancement nodes into a recursive form, which can capture the dynamic characteristics of the data, which is called recurrent enhancement nodes and can formulate as

$$\mathbf{H}_k = \xi(\mathbf{H}_{j-1}\mathbf{W}_{h_j} + \mathbf{Z}^n\mathbf{W}_{z_j} + \boldsymbol{\beta}_{h_i}),\ \ j = 1, 2, \ldots, m \quad (50)$$

where $\mathbf{W}_{h_j}$ and $\mathbf{W}_{z_j}$ is randomly generated for the last state of enhancement nodes $\mathbf{H}_{j-1}$ and the collected features $\mathbf{Z}^n$.

Based on this variant, Xu *et al.* [64] connected the enhanced nodes in each group cyclically, called R-BLS. As shown in Fig. 6, this recurrent structure can capture the dynamic characteristics of the time series. Two typical chaotic systems and a real-world air-quality dataset [65] were used to evaluate the performance of R-BLS, which shows its powerful capability of time-series prediction.

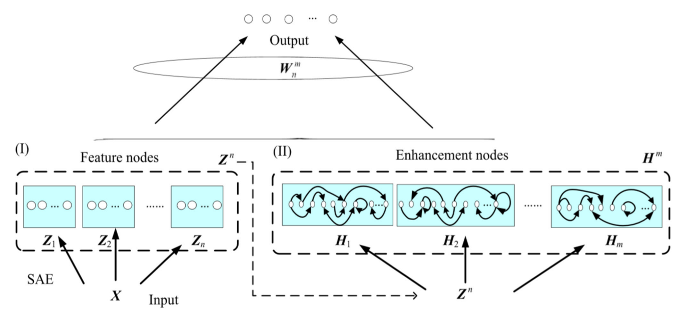


Fig. 6. Structure of an R-BLS proposed by Xu *et al.* [64].

In addition, Kuok and Yuen [66] proposed another novel approach for sequential data reconstruction, called sequential BLS (SBL). SBL can reconstruct the missing signal of damaged sensors executed sequentially with moving time windows.

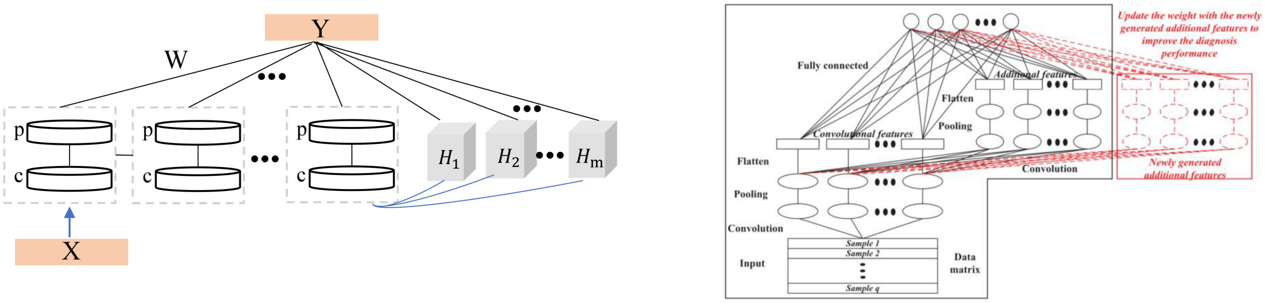
C. Convolutional BLS

CNN is the most commonly used feedforward artificial neural-network method for feature extraction in the image field, which has been applied to various pattern recognition problems [67], [68] and dramatically improves the accuracy of data classification [69]. The raw CNN consists of three structures: 1) convolution; 2) activation; and 3) pooling [70].

Chen *et al.* [18] constructed a preliminary method of combining BLS with convolution kernels, which provided *a priori* knowledge for subsequent research. Fig. 7(a) shows the architecture that replaces the mapping feature nodes in the original BLS with convolution and pooling operators and designs a cascade of convolution functions. First, the weights of convolutional filters are randomly sampled under a given distribution, and the feature mapping nodes can be defined as

$$\begin{aligned}\mathbf{Z}_k &= \phi\left(\mathbf{Z}_{k-1}; \{\mathbf{W}_{e_k}, \boldsymbol{\beta}_{e_k}\}\right) \\ &\triangleq \theta\left(P\left(\mathbf{Z}_{k-1} \otimes \mathbf{W}_{e_k} + \boldsymbol{\beta}_{e_k}\right)\right), \qquad \text{for } k = 1, \ldots, n \quad (51)\end{aligned}$$

where $\otimes$ is the convolutional operator. The functions $P(\cdot)$ and $\theta(\cdot)$ are the pooling operator and activation function,



(a) (b)

Fig. 7. Two different architectures of convolutional BLS. (a) Cascade of convolution feature mapping nodes (CCFBLS). (b) Broad convolutional neural network (BCNN).

respectively. Then, the enhancement nodes can be generated by $\mathbf{H}_j = \xi_j(\mathbf{X}\mathbf{W}_{h_j} + \boldsymbol{\beta}_{h_j})$, $j = 1, 2, 3, \ldots, m$, where $\mathbf{Z}^n = [\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n]$. Finally, $\mathbf{Z}^n$ and $\mathbf{H}^m$ are connected directly to calculated the output weight. It achieved a better testing accuracy and less time in face recognition images selected from the MS-Celeb-1M [71] database compared with Resnet-34.

Furthermore, Li *et al.* [72] applied this version of convolutional BLS with the Adam algorithm to image classification. Yang [73] proposed a CNN-based BLS, which treats the input image as mapping feature nodes with principal component analysis (PCA), considering the hidden state after convolution and pooling layers as enhancement features. We can see that it is natural to combine the BLS and CNN to handle some more complex image data.

In addition, Yu and Zhao [74] adopted a similar structure. They designed a novel convolutional BLS, called the broad convolutional neural network (BCNN), for industrial process fault diagnosis, which is shown in Fig. 7(b). BCNN can express the existing sample by lagging samples which allows it to learn the interactive information between each sample. At the same time, its incremental function enables timely feedback on new faults and does not need to retrain the model entirely.

In BCNN, each sample x_i will be represented as a data matrix $\bar{\boldsymbol{x}}_i$ by $q-1$ lagging data

$$\bar{\boldsymbol{x}}_i = \begin{bmatrix} x_i^1, & x_i^2, & \ldots & x_i^p \\ x_{i-1}^1, & x_{i-1}^2, & \ldots & x_{i-1}^p \\ \vdots & \vdots & & \vdots \\ x_{i-q+1}^1, & x_{i-q+1}^2, & \ldots & x_{i-q+1}^p \end{bmatrix} \tag{52}$$

where p is the number of variables and $\boldsymbol{x}_i^j$ indicates the jth variable of the sample $\boldsymbol{x}_i$. Then, a group of convolutional mapping feature can be generated through the randomly generated convolution filters $\mathbf{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots\}$, pooling operators, and nonlinear activation function $\hbar(\cdot)$

$$\begin{aligned} \boldsymbol{c}^c &= \hbar(\mathrm{Conv}(\bar{\boldsymbol{x}}, \boldsymbol{f})) \\ \boldsymbol{c}^p(i,j) &= \max\left\{\boldsymbol{c}^c\left(i', j'\right)\middle| i \le i' \le i+u, j \le j' \le j+v\right\}. \end{aligned} \tag{53}$$

Analogously, the additional features can be generated through another filters $\mathbf{F}^A = \{\boldsymbol{f}_1^a, \boldsymbol{f}_2^a, \ldots\}$ by the same way

$$\boldsymbol{a}^c = \hbar\left(\mathrm{Conv}\left(\boldsymbol{c}^p, \boldsymbol{f}^a\right)\right)$$

$$\boldsymbol{a}^p(i,j) = \max\left\{\boldsymbol{a}^c\left(i', j'\right)\middle| i \le i' \le i+u, j \le j' \le j+v\right\}. \tag{54}$$

Finally, it will combine convolutional feature with additional feature $\mathbf{H} = [\mathbf{C}_f, \mathbf{A}_f]$ and calculate the weight $\mathbf{W}$ by ridge regression based on (58) and (59). In addition, two incremental learning with new training samples and new fault classes were introduced for dynamic actual application without remodeling the entire network.

D. Weighted BLS for Data With Outliers

In some practical applications, the sensor may face failures and human disturbances when collecting data. It will inevitably produce some noise and outliers [75], [76]. Chu *et al.* [77] proposed a novel weighted BLS to deal with the problem of outliers in these sample data. Unlike classical BLS that treats all training data points as equal, weighted BLS constrains each sample's contribution to modeling by adding a variable of the penalty factor. It allows normal samples to obtain higher weights to increase their contribution, while lower weights are assigned to suspicious outliers to reduce their contribution.

In weighted BLS, the input data were directly connected with the enhancement nodes through random input weights, which can be denoted as

$$\mathbf{H}_j = \xi_j\left(\mathbf{X}\mathbf{W}_{h_j} + \boldsymbol{\beta}_{h_j}\right), \quad j = 1, 2, 3, \ldots, m. \tag{55}$$

Therefore, the weighted BLS can be wrote as

$$\begin{aligned} \mathbf{Y} &= \left[\mathbf{X}|\mathbf{H}^m\right]\mathbf{W}_m \\ &= \mathbf{A}\mathbf{W}_m \end{aligned} \tag{56}$$

where $\mathbf{A} = [\mathbf{X}|\mathbf{H}^m]$. The connection weight between the hidden layer and output layer can be calculated by weighted ridge regression

$$\underset{\mathbf{W}}{\arg\min}: \ \|\theta\mathbf{A}\mathbf{W} - \theta\mathbf{Y}\|_2^2 + \lambda\|\mathbf{W}\|_2^2 \tag{57}$$

where θ is the weighted penalty factor, measuring the training data weight $\theta = [\theta_1, \theta_2, \ldots, \theta_n]^T$. Similar to (6), the weight parameters can be calculated as

$$\mathbf{W}^m = \left(\lambda\mathbf{I} + \mathbf{A}^T\theta^2\mathbf{A}\right)^{-1}\mathbf{A}^T\theta^2\mathbf{Y} \tag{58}$$

the pseudoinverse of $\mathbf{A}$ can be denoted as

$$\mathbf{A}^+ = \lim_{\lambda \to 0}\left(\lambda\mathbf{I} + \mathbf{A}^T\theta^2\mathbf{A}\right)^{-1}\mathbf{A}^T\theta. \tag{59}$$

Algorithm 4: Weighted BLS [77]

Input: training samples $\mathbf{X}$,$\mathbf{Y}$;
Output: $\mathbf{W}$
for $j = 1; j \leq m$ **do**
  Random $\mathbf{W}_{h_i}$, $\boldsymbol{\beta}_{z_i}$;
  Calculate $\mathbf{H}_j = \xi_j(\mathbf{X}\mathbf{W}_{h_j} + \beta_{h_j})$;
end
Set $\mathbf{H}_j = [\mathbf{H}_1, \ldots, \mathbf{H}_m]$;
Set $\mathbf{A}^m$ and calculate residual error;
Let initial θ be an identity matrix;
if *stopping iterative criterion is not satisfied;* **then**
  Update θ;
  Calculate $\mathbf{W}_m$, $\mathbf{A}_m^+$ with Eq. (58), (59);
end
Set $\mathbf{W} = \mathbf{W}_m$

In practice, the weights and penalty factors will be calculated iteratively

$$\mathbf{MAX} = \max\left(\left|\mathbf{W}_m^{(l)} - \mathbf{W}_m^{l-1}\right|\right), \quad l = 1, 2, 3, \ldots \tag{60}$$

where l and $\mathbf{W}_m^{(l)}$ denote the number of iteration and the lth iteration solution, respectively. Algorithm 4 of weighted BLS is summarized as follows.

In addition, two different weighted incremental learning algorithms for the increment of data and additional enhancement nodes were also proposed to handle newly added data points. Due to limited space, we are not able to cover these two algorithms. Intuitively, weighted BLS can add different training coefficients that are used for different classes in training samples, which can reduce the overfitting of the model. We believe that this efficient weighted BLS can not only deal with outliers and noises during system modeling but also the imbalanced data [78].

In addition, Zheng *et al.* [79] proposed a method, called correntropy BLS (CBLS), which dealt with outliers and noise to enhance the robustness of BLS. CBLS uses the maximum entropy criterion (MCC) to train the output weights and derives three incremental learning methods from the weighted regularized least-squares solution.

E. Fuzzy BLS

The neuro-fuzzy approach has a wide range of applications in the fields of nonlinear system identification and control due to its advantages of the interpretability of IF-THEN rules and powerful universal approximators for the neural network. Nevertheless, when dealing with high-dimensional data, it often faces the problems of "rule explosion" and extended training time.

Feng *et al.* [32], [33], [80]–[82] proposed a novel neuro-fuzzy model, called fuzzy BLS, by integrating BLS and TS fuzzy systems. Fuzzy BLS replaces feature nodes with some fuzzy subsystems. At the same time, the k-means algorithm is used to determine the number of fuzzy rules and the center of the Gaussian membership function to solve the rule explosion problem.

Assume that the input data are $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$, where $\mathbf{x}_s = (x_{s1}, x_{s2}, \ldots, x_{sM})$, $s = 1, 2, \ldots, N$. The output vector of the ith fuzzy subsystem for the sth training sample $\mathbf{x}_s$ is calculated as

$$\boldsymbol{Z}_{si} = \left(\omega_{s1}^i z_{s1}^i, \omega_{s2}^i z_{s2}^i, \ldots, \omega_{sK_i}^i z_{sK_i}^i\right) \tag{61}$$

where z_{sk}^i can be calculated with α_{kt}^i through first-order TS fuzzy system. Then, the intermediate output $\mathbf{Z}$ of n fuzzy subsystems can be denoted as

$$\mathbf{Z}^n = (\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n) \in \mathbb{R}^{N \times (K_1 + K_2 + \cdots + K_n)} \tag{62}$$

where $\mathbf{Z}_i = (\mathbf{Z}_{1i}, \mathbf{Z}_{2i}, \ldots, \mathbf{Z}_{Ni})$. In the ith fuzzy subsystem, it is easy to obtain the defuzzification output for traning sample x_s

$$\begin{aligned} \boldsymbol{F}_{si} &= \sum_{k=1}^{K_i} \lambda_{sk}^i \left(\sum_{t=1}^{M} w_k^i \alpha_{kt}^i x_{st}\right) \\ &= \sum_{t=1}^{M} \alpha_{kt}^i x_{st} \left(\lambda_{s1}^i, \ldots, \lambda_{sK_i}^i\right) \begin{pmatrix} w_1^i \\ \vdots \\ w_{K_i}^i \end{pmatrix} \end{aligned} \tag{63}$$

where λ_{sk}^i is the weight fire strength for the kth fuzzy rule and w_k^i is a parameter. The output matrix of the ith fuzzy subsystem for all training sample $\mathbf{X}$ is expressed as

$$\boldsymbol{F}_i = (\boldsymbol{F}_{1i}, \boldsymbol{F}_{2i}, \ldots, \boldsymbol{F}_{Ni})^T \triangleq \boldsymbol{D}\Omega^i \boldsymbol{\delta}^i \tag{64}$$

where $\boldsymbol{D} = \operatorname{diag}\{\sum_{t=1}^{M} \alpha_{kt}^i x_{1t}, \ldots, \sum_{t=1}^{M} \alpha_{kt}^i x_{Nt}\}$ and

$$\Omega^i = \begin{pmatrix} \omega_{11}^i & \cdots & \omega_{1K_i}^i \\ \vdots & & \vdots \\ \omega_{N1}^i & \cdots & \omega_{NK_i}^i \end{pmatrix}, \quad \delta^i = \begin{pmatrix} \delta_{11}^i & \cdots & \delta_{1C}^i \\ \vdots & & \vdots \\ \delta_{K_i 1}^i & \cdots & \delta_{K_i C}^i \end{pmatrix}.$$

Then, we can obtain the aggregative output of n fuzzy subsystems for the top layer

$$\begin{aligned} \boldsymbol{F}^n &= \sum_{i=1}^{n} \boldsymbol{F}_i = \sum_{i=1}^{n} \boldsymbol{D}\Omega^i \boldsymbol{\delta}^i = \boldsymbol{D}\left(\Omega^1, \ldots, \Omega^n\right) \begin{pmatrix} \boldsymbol{\delta}^1 \\ \vdots \\ \boldsymbol{\delta}^n \end{pmatrix} \\ &\triangleq \boldsymbol{D\Omega\Delta} \in \mathbb{R}^{N \times C} \end{aligned} \tag{65}$$

where Ω and δ is the matrix for fire strength and parameter, respectively. In addition, the enhancement nodes can be generated based in $\mathbf{Z}^n$ and be denoted as

$$\mathbf{H}^m = (\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_m) \in \mathbb{R}^{N \times (L_1 + L_2 + \cdots + L_m)} \tag{66}$$

where $\mathbf{H}_j = \xi_j(\mathbf{Z}^n \mathbf{W}_{h_j} + \beta_{h_j}), j = 1, 2, 3, \ldots, m$ is the output of jth enhancement node groups. Therefore, the final output of a fuzzy BLS is

$$\begin{aligned} \hat{\mathbf{Y}} &= \mathbf{F}^n + \mathbf{H}^m \mathbf{W}_e \\ &= \mathbf{D}\Omega\Delta + \mathbf{H}^m \mathbf{W}_e \\ &= \left(\mathbf{D}\Omega, \mathbf{H}^m\right) \begin{pmatrix} \Delta \\ \mathbf{W}_e \end{pmatrix} \\ &\triangleq \left(\mathbf{D}\Omega, \mathbf{H}^m\right) \mathbf{W} \end{aligned} \tag{67}$$

so the final weight can be calculated rapidly through pseudoinverse

$$\mathbf{W} = \left(\mathbf{D}\Omega, \mathbf{H}^m\right)^+ \mathbf{Y} \tag{68}$$

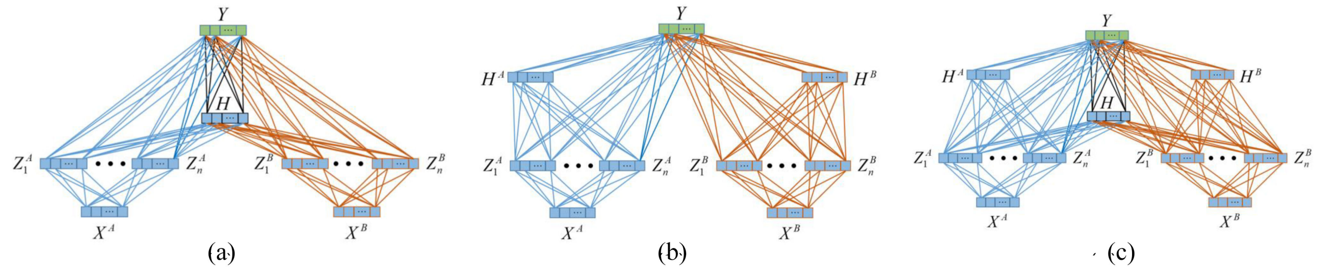


Fig. 8. Three different architectures of multiview BLS. (a) MvBLS with shared enhancement features. (b) MvBLS2 with respective enhancement features. (c) MvBLS3 with both shared and respective enhancement features.

where

$(\mathbf{D}\Omega, \mathbf{H}^m)^+ = ((\mathbf{D}\Omega, \mathbf{H}^m)^T(\mathbf{D}\Omega, \mathbf{H}^m))^{-1}(\mathbf{D}\Omega, \mathbf{H}^m)^T$.

In some popular regression and classification benchmark tasks, fuzzy BLS can obtain higher accuracy and less running time, which shows its advantages.

At the same time, fuzzy BLS has also attracted the attention of many scholars, and it has achieved good results in its application. Guo *et al.* [83] applied Fuzzy BLS to the field of multiview high-dynamic-range image (HDR) synthesis based on a set of multiview low-dynamic-range (LDR) images. Lin *et al.* [84] proposed a novel method combined with the merits of two-phase processing methods and the Fuzzy BLS for rain removal from a single image. In addition, Huang *et al.* [85] proposed a control strategy on the broad fuzzy neural network (BFNN) by using impedance learning when facing an unknown dynamic environment.

F. Multiview BLS

In some practical problems, the same thing can be described in many different ways or from different perspectives. Multiview learning can make full use of the complementary information reflected from different views to describe things in more detail [86]. Shi *et al.* [87] expanded the BLS to a multiview BLS, which treats two signals of medial frontal local potential field (LPFs) [88] and action potential (spikes) [89] came from the monkey brain as two different views for decoding the oculomotor decision [90].

Three novel types of multiview BLS were proposed with different connections of enhancement layers and mapping feature layers, which have been shown in Fig. 8. We will introduce MvBLS as an example. MvBLS first constructs the mapping feature nodes of each view separately and optimizes them through sparse autoencoders. Then, connect the optimized mapping features of each view to obtain nonlinear enhancement features.

Let the two views be A and B and the corresponding data matrices be $\mathbf{X}_A$ and $\mathbf{X}_B$. Then, generate the mapping feature nodes $\mathbf{Z}^A = [\mathbf{Z}_1^A, \ldots, \mathbf{Z}_n^A]$ for view A and $\mathbf{Z}^B = [\mathbf{Z}_1^B, \ldots, \mathbf{Z}_n^B]$ for view B. The enhancement nodes $\mathbf{H}$ can be generated based on both views $[\mathbf{Z}^A|\mathbf{Z}^B]$. Let $\mathbf{Z}' = [\mathbf{Z}^A|\mathbf{Z}^B|\mathbf{H}]$, so the weight $\mathbf{W}_o$ connected hidden layers and output layers can be calculated through (6)

$$\mathbf{W}_o = \arg\min_{\mathbf{W}}\left(\left\|\mathbf{Z}'\mathbf{W} - \mathbf{Y}\right\|_F^2 + \lambda\|\mathbf{W}\|_F^2\right)$$
$$= \left(\lambda_2\mathbf{I} + \mathbf{Z}'^T\mathbf{Z}'\right)^{-1}\mathbf{Z}'^T\mathbf{Y} \quad (69)$$

where λ is the $l-2$ regularization coefficient. Fig. 8 also gives two other multiview BLS structures. These general multiview BLS models can learn the internal connections contained in different views, which is more helpful for understanding the information reflected by the same thing.

G. BLS With Manifold Learning

In recent years, manifold learning, especially the Laplacian eigenmaps [91], has been used widely to intuitively discover hidden information in structured data and utilize the geometric relationships between variables of nonlinear multivariate systems. Manifold learning can maintain the local relationship of adjacent data points in the manifold and redraw it to a low-dimensional space, which means that it can reduce the dimension by graph embedding [92].

Basic BLS may ignore the structure information through the least mean square error method, so many efforts are spent on how to let BLS learn the structure information concealed in data.

Jin *et al.* [93] incorporated the manifold learning into the objective function of the standard BLS, called graph regularized BLS (GBLS), which assumed that similar images data may share similar properties. GBLS has two different graph structures: 1) intrinsic graph BLS (IGBLS) will consider the intrinsic graph that retains the label consistency information of the data and 2) intrinsic and penalty graph BLS (IPGBLS) will consider the intrinsic and penalty graph simultaneously. In many image recognition tasks, the GBLS model can possess superior performance for image recognition compared with the standard BLS and several state-of-the-art recognition methods. Wang *et al.* [94] proposed a novel structure, called domain-adaptation broad learning (DABL), for hyperspectral image classification, which adds the domain adaptation and manifold regularization to the output layer of it.

Furthermore, Han *et al.* [95] proposed a novel unified framework, called structured manifold BLS (SM-BLS), which does well in nonuniform embedding, dynamical system revealing, and time-series prediction. SM-BLS benefits from a nonuniform embedding strategy, which can compress the manifold information into low-dimensional embedding and discover the intrinsic structure information in time series. This nonuniform strategy based on a structured manifold learning method can automatically discover complicated relationships from multivariate time series.

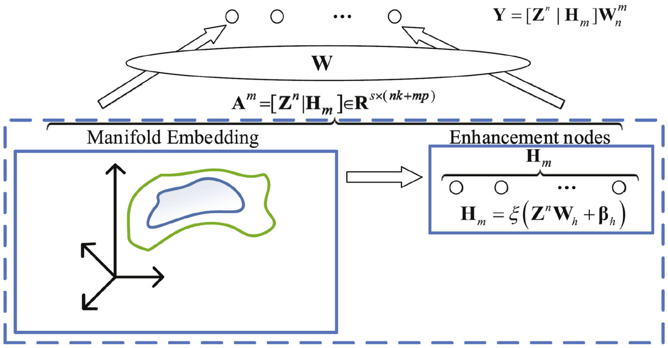


Fig. 9. Structure of an robust manifold BLS (RM-BLS) proposed by Feng *et al.* [96].

Feng *et al.* [96] also proposed a novel algorithm for large-scale noisy, chaotic time-series prediction, called robust manifold BLS (RM-BLS), which is shown in Fig. 9. Different from SM-BLS, RM-BLS can extract the manifold representation and intrinsic structural information of the dynamical system through the manifold feature selection with random perturbation approximation. A nonconvex and nonsmooth objective function can solve the Stiefel optimization for manifold embedding. Finally, the manifold embedding and enhancement will be collected together to calculate the output weight of RM-BLS.

Meanwhile, many researchers have jointed manifold learning with BLS to solve semisupervised and unsupervised tasks. Regarding these different tasks, we will introduce them in detail in the following sections (Sections VI and VII).

H. BLS With Ensemble Learning

Ensemble learning [97] means combining the capabilities of multiple individuals and trains multiple classifiers, and joins these classifiers to reduce the risk of overfitting at the same time and achieve better prediction performance. Common ensemble learning methods include Boosting [98], Bagging [99], and Stacking [100].

Ensemble learning has also been studied in the BLS literature. Fan and Zhang [101] proposed a method of constructing a stacked ensemble classifier based on a BLS to predict the interaction between lncRNA and protein, which is called LPI-BLS. LPI-BLS connects related features extracted from k-mer and pseudo nucleotide/amino acid composition extracted from lncRNA sequence and protein sequence, respectively, to form a feature vector of lncRNA-protein pairs [102], [103]. Building five BLS classifiers on these foundations and using the stacked ensemble classifier to concatenate the output from the above five BLS classifiers can effectively predict the interaction between lncRNA and protein.

In addition, Wang *et al.* [104] proposed a method, called bagging-BLS, with 100 sub-BLS, which has a more stable and reliable application in wear fault diagnosis of aero-engines. Zhu *et al.* [105] applied BLS with ensemble and classification, called BLS-EC, to predict multistep-ahead wind speed. In order to overcome the randomness and instability of a single BLS, BLS-EC can improve the generalization and stability of the network.

I. Other Variants of BLS

In recent years, BLS has developed different valuable variants for different application scenarios due to its flexibility. Limited by space, we have summarized other important variants as follows.

Wavelet BLS: Lin *et al.* [36]–[39] replaced the mapping function with wavelet for forecasting and canceling the physiological tremor in teleoperation.

Deep Cascade BLS: Ye *et al.* [106] designed an adaptive deep cascading BLS (DCBLS), which modified feature nodes and mapping nodes into n and m depth cascade structures, respectively.

Stacked BLS: The Stacked BLS [107] superimposes multiple BLS modules through residual connections, which increases the depth of the flat network and improves the learning ability of the network. At the same time, the basic pseudoinverse training mechanism is used in each module, so as to retain the efficiency and effectiveness of BLS. This novel variant allows the shallow network to improve performance when faced with complicated image recognition and classification problems.

VI. BLS for Semisupervised Learning

SSL can leverage both labeled data and unlabeled data to learn knowledge and do pattern recognition tasks [108], [109]. Therefore, SSL has an extraordinary application value [110]. The SSL assumes that both labeled and unlabeled input data come from the same marginal distribution [111]. Naturally, unlabeled data can provide useful information for exploring the data structure in the input space.

In recent years, the construction of semisupervised algorithms based on BLS has aroused the interest of many researchers. Zhao *et al.* [112] extended BLS to SSL, called SS-BLS, by introducing manifold learning [113] and achieved good results. Manifold regularization assumes that the input data follows certain clusters or manifold structures in the input space, and it can build a Laplacian matrix to propagate the information of labeled data to unlabeled data. Its essence is the label propagation based on the manifold hypothesis, which can effectively improve the classification efficiency of unscratched data.

The training set was divided into two parts: 1) the labeled data $\{\mathbf{X}_l, \mathbf{Y}_l\} = \{X_i, Y_i\}_{i=1}^{l}$ and 2) unlabeled data $\mathbf{X}_u = \{x_i\}_{i=1}^{u}$, where l and u are the number of labeled data and unlabeled data, respectively. The collected nodes from the mapping feature and enhancement $\mathbf{A} = [\mathbf{Z}^n|\mathbf{H}^m]$ were rerepresented as

$$\mathbf{A} = \left[\mathbf{a}(x_1)^T, \ldots, \mathbf{a}(x_i)^T, \ldots, \mathbf{a}(x_{l+u})^T\right]^T. \tag{70}$$

So, the formulation of semisupervised BLS (SS-BLS) with manifold regularization and BLS is given by

$$\begin{aligned} \min_{\beta \in \mathbb{R}^{n_h \times n_o}} \quad & \frac{\theta}{2}\left\|\boldsymbol{\beta}^*\right\|^2 + \frac{1}{2}\sum_{i=1}^{l} C_i \|\mathbf{e}_i\|^2 + \frac{\lambda}{2}\operatorname{Tr}\left(\mathbf{F}^T \mathbf{L} \mathbf{F}\right) \\ \text{s.t.} \quad & \mathbf{e}_i^T = \mathbf{a}(x_i)\boldsymbol{\beta}^* - \mathbf{y}_i^T, \quad i = 1, \ldots, l \\ & \mathbf{f}_i(x_i) = \mathbf{a}(x_i)\boldsymbol{\beta}^*, \quad i = 1, \ldots, l+u \end{aligned} \tag{71}$$

where $\mathbf{L}$ is the graph Laplacian matrix built based on both labeled and unlabeled data, $\mathbf{F} \in \mathbb{R}^{(l+u)\times(n*k+m*k)}$ is the BLS output with its ith row equal to $\mathbf{f}(x_i)$, and $\operatorname{Tr}(\cdot)$ denotes the trace of a matrix. In addition, λ and ξ are the tradeoff parameter associated with the manifold regularization term and the further constraints with weight $\boldsymbol{\beta}^*$, respectively. In addition, $\mathbf{C}_i$ denotes the penalty coefficient, which gives the sample different weights. Similar to Weight BLS discussed in Section V-D, different categories of samples should be weighted to alleviate the problem of overfitting and poor generalization in categories with a small number of samples. To optimize (71), the objective function was transformed into

$$\begin{aligned} \min_{\beta \in \mathbb{R}^{n_h \times n_o}} \quad & \frac{\theta}{2}\left\|\boldsymbol{\beta}^*\right\|^2 + \frac{1}{2}\sum_{i=1}^{l} \left\| \mathbf{C}^{\frac{1}{2}}\left(\mathbf{A}\boldsymbol{\beta}^* - \tilde{\mathbf{Y}}\right)\right\|^2 \\ & + \frac{\lambda}{2}\operatorname{Tr}\left(\boldsymbol{\beta}^{*T}\mathbf{A}^T\mathbf{L}\mathbf{A}\boldsymbol{\beta}^*\right) \end{aligned} \tag{72}$$

where $\tilde{\mathbf{Y}} \in \mathbb{R}^{(l+u)\times n_o}$ is the training target with both labeled and unlabeled data, its first l row is equal to $\mathbf{Y}_l$, the rest is equal to 0, $\mathbf{C} \in \mathbb{R}^{(l\times u)\times(l\times u)}$ is the diagonal matrix, the first l diagonal elements is equal to $\mathbf{C}_i$, and the remaining diagonal elements are 0.

The optimal solution to the final weight of SS-BLS can be expressed as

$$\boldsymbol{\beta}^* = \left(\boldsymbol{\theta}\mathbf{I}_{n_h} + \mathbf{A}^T\mathbf{C}\mathbf{A} + \lambda\mathbf{A}^T\mathbf{L}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{C}\tilde{\mathbf{Y}} \tag{73}$$

where n_h is the sum of the number of enhancement nodes and feature nodes, and $\mathbf{I}_{n_h}$ is the identity matrix of dimension n_h.

If the number of labeled data is fewer than the number of hidden neurons, the solution can be alternative as

$$\boldsymbol{\beta}^* = \mathbf{A}^T\left(\boldsymbol{\theta}\mathbf{I}_{l+u} + \mathbf{C}\mathbf{A}\mathbf{A}^T + \lambda\mathbf{L}\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{C}\tilde{\mathbf{Y}}. \tag{74}$$

The SS-BLS is summarized in Algorithm 5.

Algorithm 5: SS-BLS [112]

Input: Labeled samples: $\{\mathbf{X}_l, \mathbf{Y}_l\} = \{X_i, Y_i\}_{i=1}^{l}$;
Unlabeled samples: $\mathbf{X}_u = \{x_i\}_{i=1}^{u}$.
Output: The mapping function of SS-BLS: $\mathbf{f}:\mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_o}$
Construct the graph Laplacian matrix $\mathbf{L}$ with $\mathbf{X}_l$ and $\mathbf{X}_u$;
Construct the hidden nodes of mapping features and enhancement features $\mathbf{A} = [\mathbf{Z}^n|\mathbf{H}^m] \in \mathbb{R}^{(l+u)\times(nk+m)}$;
Select tradeoff parameters $\mathbf{C}_0$, λ, θ;
calculate the output $\boldsymbol{\beta}^*$ by manifold regularization framework;
if $n_k \geq N$ **then**
  $\boldsymbol{\beta}^* = \mathbf{A}^T\left(\boldsymbol{\theta}\mathbf{I}_{l+u} + \mathbf{C}\mathbf{A}\mathbf{A}^T + \lambda\mathbf{L}\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{C}\tilde{\mathbf{Y}}$
else
  $\boldsymbol{\beta}^* = \left(\boldsymbol{\theta}\mathbf{I}_{n_h} + \mathbf{A}^T\mathbf{C}\mathbf{A} + \lambda\mathbf{A}^T\mathbf{L}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{C}\tilde{\mathbf{Y}}$
end

Meanwhile, other researchers also provide some exciting methods for SS-BLS. Kong *et al.* [114] proposed an SS-BLS method that can be applied to hyperspectral imagery classification (HIC). The class probability structure is incorporated into BLS to expand for the semisupervised tasks. By making full use of the information of limited labeled samples and large unlabeled samples, SS-BLS can obtain the spectral-space representation and achieve good results in the HIC task (HSI) [115]. In addition, Liu *et al.* [116] introduced incremental learning to SS-BLS based on manifold regularization for reducing the learning time and the storage of historical data. She *et al.* [117] designed a graph-based SS-BLS (GSS-BLS) for electroencephalogram (EEG) signal classification. Three motor imagery EEG datasets of brain–computer interaction (BCI) competitions [118] were used to verify the validity and practicability of the GSS-BLS. Zheng *et al.* [119] introduced Fick's law-assisted propagation (FLAP) into BLS to deal with collected actual label and unlabeled data, which is called FLAP-BLS.

VII. BLS FOR UNSUPERVISED LEARNING

For some tasks, the data received from the sensor are difficult to label the category manually or the cost of manual category labeling is too high [120]. Therefore, unsupervised learning [121] has always been a challenging task. BLS was formerly proposed for supervised tasks, such as classification and regression, and rarely considered unsupervised learning problems. In recent years, there is some valuable work that extends BLS to the field of unsupervised learning. Liu *et al.* [122] developed a combined structure based on the K-means clustering and BLS. Feng *et al.* [123] proposed a broad graph-based robust continuous clustering (RCC) algorithm to upgrade the RCC.

What is more, Kong *et al.* [124] proposed a novel unsupervised BLS (UBL) with graph regularization for hyperspectral image clustering. UBL can adjust the weights of its mapping feature nodes, and enhancement nodes through graph regularized sparse autoencoders (GRSAE) to maintain the manifold structured information of hyperspectral input images preprocessed with hierarchical guidance filtering (HGF) [125]. This fine-tuning method for weights of mapping feature nodes and enhancement nodes is different from the original BLS that randomly generates weights. Besides, UBL can construct a novel objective function consisting of the l_2-norm and the items of graph regularization, which can solve the problem by selecting the eigenvectors corresponding to the smallest eigenvalues.

Given the HGF-based HSI samples $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where n is the number of samples. According to the manifold assumption, two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ closed to each other in the original space should also be closed to each other in the corresponding mapping feature and enhancement feature space. So, the process of generating the mapping feature can be fine tuned by GRSAE

$$\begin{aligned}\arg\min_{\mathbf{W}_i^M}&\left\|\mathbf{X}\mathbf{W}_i^M-\mathbf{E}_i\right\|_2^2+\lambda\left\|\mathbf{W}_i^M\right\|_1+\alpha\sum_{i=1}^{n}\sum_{j=1}^{n}w_{ij}\left\|\mathbf{e}_i-\mathbf{e}_j\right\|_2^2\\&=\arg\min_{\mathbf{W}_i^M}\left\|\mathbf{X}\mathbf{W}_i^M-\mathbf{E}_i\right\|_2^2+\lambda\left\|\mathbf{W}_i^M\right\|_1+\alpha\,\mathrm{tr}\left(\mathbf{E}_i^T\mathbf{L}\mathbf{E}_i\right)\end{aligned}\tag{75}$$

where $\mathbf{E}_i$ and $\mathbf{W}_i^M$ represent the ith group of mapping feature nodes and desired weights, respectively. λ is the tradeoff parameter and α is the graph regularization parameter of GRSAE, tr$(\cdot)$ represents the trace operation. $\mathbf{L}$ is the Laplace matrix that can be obtained by constructing a k-nearest neighbor graph. As we know, (75) can be solved by the alternating direction method of multipliers (ADMM) [126]. So, (75) can be expressed as

$$\begin{aligned}J=\arg\min_{W_i^M}&\left\|\mathbf{X}\mathbf{W}_i^M-\mathbf{E}_i\right\|_2^2+\lambda\|\boldsymbol{O}\|_1+\alpha\,\mathrm{tr}\left(\mathbf{E}_i^T\mathbf{L}\mathbf{E}_i\right)\\&+\rho\mathbf{u}^T\left(\mathbf{W}_i^M-\mathbf{O}\right)+\frac{\rho}{2}\left\|\mathbf{W}_i^M-\mathbf{O}\right\|_2^2\end{aligned}\tag{76}$$

where $\rho>0$ is a constant. $\mathbf{W}_i^M$, $\mathbf{O}$, and $\mathbf{u}$ are updated alternatively while keeping the other variables fixed based on the ADMM optimization method.

Next, the weights for generating the enhancement feature based on mapping feature nodes can also be fine tuned with the GRSAE method. The mapping feature nodes will be collected together $\mathbf{E}=[\mathbf{E}_1,\mathbf{E}_2,\ldots,\mathbf{E}_{GM}]$. So, the feature of enhancement nodes $\mathbf{F}$ are denoted as

$$\mathbf{F}=\sigma\left(\mathbf{E}\mathbf{W}^E\right)\tag{77}$$

where $\sigma(\cdot)$ is the nonlinear function. In UBL, the l_2-norm term and manifold regularization term need to be added when constructing the objective function of UBL. So, the model of UBL can be expressed as

$$\begin{aligned}\arg\min_{W^m}\ &\frac{1}{2}\left(\left\|\mathbf{W}^m\right\|_2^2+\zeta\,\mathrm{Tr}\left([\mathbf{E}|\mathbf{F}]^T\left(\mathbf{W}^m\right)^T\mathbf{L}^m\mathbf{W}^m[\mathbf{E}|\mathbf{F}]\right)\right)\\ \text{s.t.}\ &\left(\mathbf{W}^m\right)^T\mathbf{W}^m=\mathbf{I}\end{aligned}\tag{78}$$

where $\zeta(\cdot)$ is the graph regularization parameter of UBL, and $\mathbf{L}^m$ is the Laplace matrix constructed by $[\mathbf{E}|\mathbf{F}]$. $\mathbf{I}$ is the constraint term for avoiding the rank deficit phenomenon of generalized eigenvalue decomposition. So, (78) can be calculated as

$$\left(\mathbf{I}+\zeta[\mathbf{E}|\mathbf{F}]^T\mathbf{L}^m[\mathbf{E}|\mathbf{F}]\right)\mathbf{W}^m=\xi[\mathbf{E}|\mathbf{F}]^T[\mathbf{E}|\mathbf{F}]\mathbf{W}^m\tag{79}$$

then, the output of UBL can be obtained with $\mathbf{W}^m$ by

$$\mathbf{Y}=[\mathbf{E}|\mathbf{F}]\mathbf{W}^m.\tag{80}$$

Finally, the clustering result $\mathbf{Y}^c$ can be obtained by applying spectral clustering on $\mathbf{Y}$. Compared with several common clustering methods based on fuzzy C-means (FCM) [127], spectral-spatial clustering methods (SSC) [128], and unsupervised extreme learning machine (UELM) [129], the UBL can obtain higher clustering accuracy.

VIII. BLS FOR FEATURE ENGINEERING

In this section, we will briefly introduce the application of BLS in feature engineering, including feature representation and feature selection.

A. BLS for Feature Representation

Learning powerful feature representation is always a core issue in many research fields, which has a very significant impact on the performance of the model in subsequent tasks, especially for some data with small sample sizes and small dimensions [130], [131]. In recent years, due to its advantage of the flexible framework and robust capacity, BLS has been applied to feature representation in corresponding research fields.

Tang *et al.* [132] applied BLS to generate robust representations for machine-learning models, especially when dealing with small datasets or small data dimensions. In addition, the random label-based autoencoder (RLA) was introduced to generate more different features. Given a two-layer encoder structure, the input fed into a machine-learning model is as follows:

$$\begin{aligned}\text{input}&=\left[\mathbf{Z}^n|\mathbf{H}^m\right]\\&=[\mathbf{Z}_1,\ldots,\mathbf{Z}_n|\mathbf{H}_1,\ldots,\mathbf{H}_m]\\&=\begin{bmatrix}\sigma(w_{11}\sigma(w_{12}x+b_{12})+b_{11}),\ldots\\\sigma(w_{n1}\sigma(w_{n2}x+b_{n2})+b_{n1})|\\\sigma\left(w'_{11}\sigma\left(w'_{12}Z^n+b'_{12}\right)+b'_{11}\right),\ldots\\\sigma\left(w'_{m1}\sigma\left(w'_{m2}Z^n+b'_{m2}\right)+b'_{m1}\right)\end{bmatrix}\end{aligned}\tag{81}$$

where w and b are weights and bias, respectively, and $\sigma(\cdot)$ is the activation function. Compared with other feature extraction methods, this framework can build a more robust and powerful representation, especially in the case of small datasets.

Furthermore, it also provides a flexible way to develop further the representation, which means that more robust feature representations can be searched in the broad space. Zhang *et al.* [133] proposed a rich feature combination for BLS, called Rich-BLS. Its learning process includes unsupervised group-by-group feature extraction and supervised learning with the $l2$-norm. The broadly fused feature representations in the proposed model exploit the potential of BLS. Zhang *et al.* [134] designed a novel BMT-Net that learns the representation of sentiment analysis tasks through the block of a multitask transformer, and uses BLS to expand it on the concept of width to find sentence-level robustness representation. In addition, Dang *et al.* [135] proposed a novel deep-wide network (DWnet) to recognize human actions. DWnet can learn spatial–temporal features then improving them with BLS. Zhang *et al.* [136] proposed a novel EEG emotion classification model, called GCB-net. It used BLS to improve the feature representation of EEG data. In addition, Gong *et al.* [137] combined the Generalized Autoregressive Pretraining language model (XLNet) and BLS to capture sentence-level representation in feature spaces of depth and width. Shen *et al.* [138] proposed a landmark recognition architecture, called MobileNetV3-BLS, which takes the output features from the last convolution layer as the mapped features of BLS.

TABLE II
SOME RECENT APPLICATIONS OF BLSs DISCUSSED IN SECTION IX

Field	Subfield	Method
Computer vision and image processing	Face recognition	Zhang et al. [144], P-BLS [145], ICBL [146] vision-brain [147], Zhang et al. [148], Luan et al. [149]
	Action recognition	DW-Net [136], Fisher-BLS [150], Sun et al. [151] Wavelet BLS [152], Lei et al. [153], Li et al. [154] FBLS-MD [155], Lin et al. [156], Luo et al. [157]
	Image classification	DABL [94], GBLS [93], Guo et al. [158], Cai et al. [159] AdaBoost-BLS [160], DBNet [161], Dang et al. [162] Distributed BLS [163], IWBLS [164], SS-BLS [115], UBL [125] BLSCF [165], Han et al. [166], EAF-BAP [167], Wang et al. [168]
	Image processing	Low-rank BLS [169], Guo et al. [83], DCBLS [107] Jin et al. [170], SID-BLS [84], Shrivastava et al. [171] SAE-BLS [172], Lei et al. [173], Chen et al. [174], Zhai et al. [175]
Medical/biomedical data analysis	Electroencephalography (EEG)	GCB-Net [137], CNBLS [176], GSS-BLS [118], TCBLS [177] MvBLS [87] Mixed-Norm BLS [178], BDGLS [168] LPVG-based BLS [179], CTW-BLS [180], Zou et al. [181]
	Medical data analysis	LPI-BLS [101], Luo et al. [182], MEKM-BLS [183] MISSIM [184], FNT [185], Chew et al. [186], CFCEBLS [187]
	Medical image segmentation	Liu et al. [188], Atlas-BLS [189]
System modeling and prediction	Time series prediction	MIE-BLS [190], R-BLS [64], SM-BLS [95], EMD-BLS [191] BLS-EC [105], Spatio-Temporal BLS [192], Zhou et al. [193] RM-BLS [96], SPCA-BLS [194], Yang et al. [195]
	System modeling	Weight BLS [77], CCBLS [196], Kuok et al. [197] ProBLS [198], Sei-BLS [199], RDBLS [200], BCRBM [201] Regularized-BLS [53], MRBL [202], Lai et al. [203]
Internet and communication engineering	Network and information security	RBF-BLS [204]–[206], Ma et al. [207] Zhong et al. [208], BL-TDA [209], ASSD [210]
	Internet and smart city	Wei et al. [211], BRL [212] Broad Forest [213], Xu et al. [214]
Control and robotics	Robotics	BLS-FDDEIF [215], BLS-SMC [216], BLS-FONTSM [217]
	System control	Adaptive-BLS [218], RIPF [219], BFNN [85] DT-BLS [220], Feng et al. [80], Yuan et al. [221] WBLAF [36], [39], QBLS [38], TDFW-BLS [37]
Fault dectect and diagnosis	Motors fault diagnosis	BCNN [74], PABSFD [222], bagging-BLS [104] Wang et al. [223], Jiang et al. [224], transfer-BLS [225] Multi-stage BLS [226], FIBL [227], Han et al. [228], OBLS [229]
Nature language processing	Text classification	Gate-BLS [63], BMT-Net [135], BroXLNet [138]

B. BLS for Feature Extraction and Selection

Recently, Starzyk *et al.* [139] analyzed the properties of wide neural networks regarding feature selection and their significance. The quality of features and data will directly affect the performance and response time of the model [140].

Liu *et al.* [141] proposed a feature selection algorithm for orthogonal BLS based on a preset threshold and the mutual information between the feature and the output, which can select mapping weights from orthogonal basis sets and ensure that the extracted features are independent of each other. The weights of mapping feature and enhancement nodes are orthogonal

$$\begin{aligned}\mathbf{I}\big(\mathbf{Z}_i;\mathbf{Z}_j\big) &= 0 \quad \forall i,j \in \{1,\ldots,n\}\\ \mathbf{I}\big(\mathbf{H}_i;\mathbf{H}_j\big) &= 0 \quad \forall i,j \in \{1,\ldots,m\}\end{aligned} \tag{82}$$

where $\mathbf{I}(\cdot)$ is the mutual information. It can guarantee the orthogonality between each mapping features and enhancement features. We want to contain the good features when $\mathbf{I}(\mathbf{Y};\phi_i(\mathbf{X}\mathbf{W}_{e_i}^N+\beta_{e_i})) >= \varepsilon_e$ and $\mathbf{I}(\mathbf{Y};\xi_j(\mathbf{Z}\mathbf{W}_{h_j}^m+\beta_{h_j})) >= \varepsilon_h$, where ε_e and ε_h are preset thresholds for feature selection.

Moreover, Zhou and He [142] proposed a different BLS with multilayer enhancement. The features extracted by this improved model are the fusion of randomly mapping features and more in-depth enhancement features, which is more helpful for discovering primary attributes hidden in the data.

IX. APPLICATIONS

As mentioned earlier, BLS has both linear mapping and nonlinear mapping capabilities and can quickly fit the required functions and systems. Its structure is very concise due to the lack of layer-to-layer coupling. In some cases, it can achieve competitive performance with a short calculation time compared to deep learning. In addition, BLS can quickly remodel the system by adding enhancement nodes, feature-mapping nodes, or new input through incremental learning and does not need to retrain the entire network when the network needs to be expanded. These advantages enable BLS to be widely used in many fields, such as image processing and computer vision, medical and biomedical data analysis, control engineering, fault detection, system modeling, and prediction. Table II gives an overview of the applications of BLS.

A. BLS for Image Processing and Computer Vision

Computer vision is a multidisciplinary study covering a wide range of fields. Due to the limited understanding of biological vision and the complexity of visual perception in the dynamic world, more theories and technologies try to obtain information from images or multidimensional data for understanding the meaning of images. As mentioned earlier, BLS can provide new ideas and methods for computer vision and image processing. In recent years, many researchers have studied the application of the BLS and its variants in the field of computer vision and image processing, such as image classification, human action recognition, facial recognition, and image processing.

BLS can quickly process face data due to its powerful ability to map input data with feature space. Zhang *et al.* [143] applied BLS to facial expression recognition (FER) and signally achieve an accuracy of 99.15% on the Extended CohnKanade Dataset (CK+) [229]. In addition, Han *et al.* [144] took emotional information entropy as a mapping feature through the transfer learning model. They designed a Personalized BLS to recognize the personalized expression under the condition of a small sample size. Besides, Zhang *et al.* [147] introduced the concept of feature blocks to BLS for processing face data, which can mildly alleviate the effects of occlusion and lighting issues during face recognition.

BLSs have also achieved good results in human motion recognition tasks. Dang *et al.* [135] proposed a novel DWnet to recognize human actions based on 3-D skeleton. It extracts spatial–temporal features and then feeds into BLS to recognize relevant human action. For 2-D human activity recognition, Yao *et al.* [154] incorporated multilevel fused features and used BLS based on matrix decomposition to enhance the classification results. Lin *et al.* [155] applied incremental BLS to human activity recognition. In addition, many researchers focus on student action recognition based on BLS [149], [150], [152].

Furthermore, BLS achieved good results in image classification by leveraging the strong ability of function fitting [160], [230], such as container number identification [165], event-based object classification [231], crack and noncrack damage classification [157], Chinese herbal medicine classification [158], and robotic material recognition [167].

In addition, BLS can capture a large amount of information on spectral and spatial that implied in the hyperspectral image to distinguish surface objects. Kong *et al.* [114], [124] proposed an SS-BLS and a UBL to HIC and clustering, which can significantly reduce a lot of parameters compared to some deep learning architecture. Chu *et al.* [232] proposed a novel discriminative locality preserving BLS (DPBLS) to obtain discriminative information and local manifold structure of samples for HIC tasks.

Moreover, the number of applicable directions of BLSs in image processing and computer vision is still growing. It includes zero-shot image classification [166], object detection [159], [168], image synthesis [83], image denoising [106], image colorization [169], etc., [151], [161], [163], [172], [173].

B. BLS for Medical/Biomedical Application

Medical and biomedical data, such as protein sequences, gene fragments, transcranial ultrasound data, and EEG, are usually with high-dimensional features or hard to capture its structured and unstructured high-level information. These physiological data that are positively related to human health can provide humans with critical implicit information in the process of disease prevention and diagnosis, patient rehabilitation, health monitoring, etc. So, it is interesting to find the flexible application of BLSs in this area.

EEG records the changes of the radio waves when the brain is active, which is the reflection of brain nerve cells on the surface of the cerebral cortex or scalp [233]. In recent years, EEG has been widely used in the fields of emotion recognition and BCI. For instance, Zhang *et al.* [136], Wang *et al.* [234] proposed a significant architecture, called the graph convolutional broad network (GCB-Net), based on graph CNN and BLS to explore the high-level graph topology information for EEG emotion recognition, which achieved the state-of-the-art results on the SJTU emotion EEG dataset (SEED) [235] and DREAMER dataset [236]. Similarly, Yang *et al.* [175] designed a novel complex network based on BLS for detecting driver fatigue from EEG. Issa *et al.* [179] selected only one EEG channel to extract the grayscale image (GSI) [237] feature by continuous wavelet transform (CWT) and then feed into BLS for EEG emotion classification. Other related methods with BLS applied in EEG signal analysis can be found in [117], [177], [178], and [180].

Long noncoding RNA (lncRNA) is considered to be an important role in regulating biological processes, such as cell differentiation and proliferation, and chromatin modification [238]–[240]. Recently, Fan and Zhang [101] proposed a novel LPI-BLS structure using a stacked ensemble with a logistical regression model and BLS to predict lncRNA-protein interactions for preventing cellular dysfunction and disease [241]. Xu *et al.* [184] designed a deep-and-wide network to predict the lncRNA subcellular locations. Zheng *et al.* [183] proposed a model that can look for sequence similarity of microRNA (miRNAs) with modified BLS to predict miRNA-disease associations. Furthermore, Luo and Zhao [181] applied BLS to blood glucose prediction of new subjects with type 1 diabetes. In addition, BLSs can search for appropriate features in broad space and excavate the high-quality information on magnetic resonance images (MRI), such as segmentation tasks [187], [188], [242].

C. BLS for System Modeling and Prediction

Traditional neural networks have achieved excellent results in the field of system modeling and prediction. They are used to solve practical problems and are more widely used in military science, space science, weather forecasting, and industrial automation. BLS has excellent potential for matching the nonlinear and nonstationary system data, such as time-series prediction and system modeling.

As for time-series prediction, Xu *et al.* [64] developed a great recurrent BLS (R-BLS) to learn the dynamic characteristics of chaotic time series. RBLS can recurrently connect its enhancement nodes and update the parameters by conjugate gradient methods [243]. Spatio temporal broad learning network (ST-BLN) is designed to forecast the traffic speed to ensure public safety and improve traffic management [191]. Liu *et al.* [244] compared BLS with a variety of different algorithms in detail in the traffic prediction task and found that the training process of BLS has two-three orders of magnitude faster. Zhu and Lian [190] developed a method combine with BLS and empirical mode decomposition (EMD) [245] to forecast nonlinear wind speed time series. Furthermore, Guo *et al.* [246] applied BLS to predicting the initial and final setting times of cement, which provided an efficient measurement technology in the construction schedule and building quality.

In addition, Yang *et al.* [194] applied BLS to model the data of port cargo throughput. Kuok and Yuen [196] used BLS to modeling the 2008 Wenchuan earthquake strong ground motion records data [247] for learning the complicated relationship of unstructured spatial data. In summary, BLS is easy to modify and expand and has other related applications in system modeling and time-series prediction, which can be found in [96], [105], [189], [193], [195], [198], [248], and [249].

D. BLS for Fault Detect and Diagnosis

Some equipment systems are prone to many faults due to their complex conditions and continuously changing working state, such as high-speed trains, electrical equipment, and electric motors [250]. If these system faults are not detected in time, the consequences will be hazardous. Thanks to its flat structure and easy scalability, BLS's calculation speed can meet the real-time requirements of the fault detection system.

For example, Zhao *et al.* [221] proposed a novel fault diagnosis method, called PABSFD, based on BLS and PCA to efficiently accomplish the fault classification of the rotor system. Jiang *et al.* [223], [226] applied BLS to detect the fault in induction motors and three-phase induction motors, which achieved quicker and more accurate results. Wang *et al.* [104] proposed a novel wear fault diagnosis method for aircraft engines based on BLS and ensemble learning that is called bagging-BLS. Bagging-BLS can obtain more stable and reliable results compared to the random forest models and single BLS, which have great significance for ensuring aircraft safety. Furthermore, Chen and Jiang [251] applied BLS to fault detection and diagnosis for the traction system in high-speed trains [252]. Wang and Zhao [222] proposed a model based on BLS for fault diagnosis using infrared thermography, which has considered the spatial thermography distribution and temporal temperature features.

E. BLS for Internet and Communication Engineering

The emergence of a huge Internet and communication ecosystem is changing the way of human life. Today, big data and the Internet complement each other and play an important role. In recent years, BLS has been widely used in the Internet and communication fields, such as network security [253], multiple-input–multiple-output (MIMO) hybrid beamforming [254], Internet of Things (IoT) [255], [256], and Internet protocol television [257] (IPTV) prediction.

Network security means that the system is not damaged or leaked due to accidental or malicious reasons, the system runs continuously, and network services are not interrupted. Rios *et al.* [203] applied BLS to the detection of denial of service cyberattacks in communication networks. Li *et al.* [204], [205] used BLS and its extensions to evaluate the performance of detecting network intrusions. In addition, Ma *et al.* [206] proposed a novel optimized BLS architecture with graph theory for leakage detection and universal localization for pipeline networks, which can moderate the indeterminate performance caused by original BLS. Zhong *et al.* developed a model to identify fraud phone calls in the communication industry field. It used the TF-IDF [258] to transform the first 15-s speech signal into text data and then feed into BLS to recognize whether the call is fraudulent or not. Another key thing to know, Long *et al.* [259] proposed an online analog beam selection method based on increment BLS and SSL for millimeter-wave (mm-wave) MIMO systems hybrid beamforming [260], [261]. It shows the unique ability of BLs on incremental learning. In order to establish a kind of fast autonomous IoT system, Wei *et al.* [211] proposed an efficient algorithm based on deep reinforcement learning (DRL) and BLS, called broad reinforcement learning (BRL), to overcome the disadvantage of time-consuming modeling and training procedures compared to traditional reinforcement learning [262], [263]. Other related applications with BLS can be found in [208], [210], [212], [212], [213], and [264], such as understanding the user quality of experience (QoE) and android malware detection, etc.

F. BLS for Control

In recent years, neural networks and their variants have been successfully used in control due to the universal function approximation capability [265]–[267]. These neural-network-based controllers have achieved good results in adaptive nonlinear control [268], [269], sliding-mode control [270] and other scenarios. However, designing a neural network with a lightweight network structure without losing accuracy still attracts many interests. Contrary to some popular deep neural networks, which suffer from a time-consuming training process, BLS can provide a much faster method with competitive accuracy.

Inspired by the neural motor behaviors, Huang *et al.* [217] proposed a novel framework of adaptive neural control based on BLS and deterministic learning [271], which is used to estimate the unknown model of manipulator dynamic. Thanks to the advantage of the flexible increment of BLS, this closed-loop system can remain globally stable. Lin *et al.* [36]–[39] designed a different and great wavelet BLS for estimating the physiological tremor in teleoperation. Chen *et al.* [218] proposed a BLS-based adaptive dynamic surface control method for autonomous surface vehicles, called relation-invariable persistent formation (RIPF) control, to solve the mission trajectory tracking problems. Furthermore, many researchers combine fuzzy with BLS for nonlinear system control. Huang *et al.* [85], [272] proposed a broad fuzzy neural control framework that solved the problem of robot-environment interaction based on BLS and fuzzy system. Feng *et al.* proposed an algorithm using the gradient descent method, which applies the BLS to the control of nonlinear dynamic systems. Other related methods can be found in [214]–[216], [219], [220], and [273].

TABLE III
ADVANTAGES OF BLS COMPARED TO CONVENTIONAL LEARNING METHODS (FOR EXAMPLE, BP). WE HOPE THIS INFORMATION ON ADVANTAGES CAN PROVIDE USEFUL GUIDANCE FOR FUTURE RESEARCH

Conventional learning methods (e.g., BP)	Broad learning system (BLS)
Time-consuming in each train processing	Fast learning through Pseudo-Inverse Algorithm
Difficult for parallel operation	flexible for parallel implementation
Need retraining the whole network when a newly input	Rapidly update without remodeling by incremental learning
Very sensitive to the number of neuron and layer	Stable in a wide range of hidden nodes
Complex process of response and computational complexity	Low energy consumption, low computational complexity, real-time response
Sensitive to task-specified parameters	Least human intervention
Very sensitive to hand-designed architecture	One main network type is useful for different applications

Due to its flexible network structure and strong universal approximation capability, it can better manage the changing and complicated situations in many applications. In recent years, we have witnessed the remarkable success of BLS. We summarize some of the latest applications of BLS and we have reason to believe that BLS can obtain more achievements in more potential fields in the future. In Section X, we will give a more detailed introduction about opportunities and hopeful future directions.

X. OPPORTUNITIES AND FUTURE DIRECTIONS

As we all know, BLS has made excellent achievements in many different fields, including computer vision, medical/biomedical data analysis, and system control. However, to search for BLS's more inherent capability and better realize its function, some unresolved problems are worth further discussion.

This section will give some directions that may be interesting and worthy of research in the future from the technical and application levels. The ultimate purpose is to promote BLS's development and innovation for diversity, high performance, robust scalability, and other requirements. It can make it beneficial for developing science, engineering, medical, education, and other fields. Table III shows a brief description of the advantages of BLS compared to traditional convolutional learning methods (e.g., BP). We hope that these descriptions can help readers understand the advantages of BLSs in detail and may obtain some help in future research.

A. Hardware Implementation of BLS

Neural-network hardware refers to a hardware system that supports the simulated neural-network model's scale and the speed of neural calculation. How to implant complex and diverse neural-network models into hardware systems has always attracted many researchers' attention. In recent years, the implementation of reconfigurable hardware is more popular as it has the advantage of lower noise sensitivity, higher accuracy, better repeatability, and compatibility with other types of preprocessors [274]–[276].

The complexity of the model and the number of parameters will directly affect the implementation at the hardware level. Decherchi *et al.* [277] introduced a l_1-regularized cost function for extreme learning machines (ELM) [278], [279], which can preferentially adopt a sparse solution and, therefore, favors pruning. They applied this simplified model to implement on two reconfigurable digital hardware, including field-programmable gate array devices (FPGAs) and low-cost low-performance devices such as complex programmable logic devices (CPLD). Meanwhile, Xie and Wang [280] proposed a D&BLS by stacking multiple lightweight BLS subsystems, which can avoid overfitting and reducing the size of the system structure at the same time. It enables fast training and good versatility to maintain powerful capability. In the future, reducing the computational complexity and the number of parameters of the BLS model will be an interesting research direction.

Until very recently, a chip specifically designed for artificial intelligence products or services has emerged, called the application-specific integrated circuit (ASIC), which focuses on accelerating machine learning, especially neural networks. At the same time, Pei *et al.* [281] introduced a smart chip, called Tianjic. Tianjic chip adopts a many-core architecture, reconfigurable building blocks, and a streamlined dataflow with hybrid coding schemes. This chip architecture can accommodate multiple machine-learning algorithms and provide a clear idea for implementing BLS hardware. How to implant BLS into a similar artificial intelligence chip is still an open problem.

B. Tradeoff Between Depth and Width in Neural Network

Expanding the number of connection layers between neural networks can improve nonlinear expression and better find convenient features. It is also one of the most important reasons for the success of the traditional BP algorithm [1], [6], [14]. However, Lu *et al.* [282] indicated the importance of fully understanding the tradeoff between the depth and width of neural networks. At the same time, they provided a proof that a deep fully connected ReLU NN with a width less or equal to $(n + 4)$ can approximate any

Lebesgue integrable function in *n*-dimensional $\mathbb{R}$ space with arbitrary precision. They also pointed out the importance of demonstrating the upper and lower bounds of network-fitting capabilities.

Chen *et al.* [18] designed some composite BLS structures under different constraints. One of them briefly explores the architecture combination of width and depth that can be regarded as a form of multilayer stacking of the enhancement nodes. These redrawn models are not only "width" but also "depth." Note that the width mentioned here is not the number of convolutional neural-network channels [283], [284]. In future work, how to explore the tradeoff and effective combination of width and depth in BLS will become valuable research content.

C. Diving Deeper With the Random Weight Mechanism

Random feature weight is the fundamental idea of the RVLF-based algorithm for BLS. This random weight mechanism makes the BLS training method and the way to find the optimal global solution inherently different from neural networks based on the BP algorithm. The random weight mechanism can intuitively transfer input data to a more efficient space by mapping features and enhancement feature nodes. This method can have universal approximation capability and good robustness so that the model can avoid the risk of overfitting [285]. First, the probability distribution functions used to generate hidden nodes, such as mapping feature nodes and enhancement nodes, may have an implicit effect on the general performance of networks in practical applications [286]–[288]. Therefore, what kind of probability distribution function is used to generate hidden layer random weights is worthy of further exploration. In addition, too strong randomness is unpredictable and requires the guidance of prior knowledge. Excessive reliance on random weight mechanisms sometimes results in lower efficiency, and the mapping way of input data may not necessarily meet the requirements of some unpredictable practical application scenarios. In the original BLS version, Chen and Liu [17] adopted the advantages of sparse autoencoders [289], [290]. It can fine tune random features to sparse and compact features, which solves the disadvantage of blindly generated features. Lan *et al.* [291] designed a novel kernel function-based BLS, which can map feature mapping nodes and enhancement nodes into a new set of additional features through the Gaussian kernel functions. These features will be used with the hidden layer nodes to calculate the output. Therefore, how to propose some more effective methods for random mapping of feature nodes and enhancement nodes is also an exciting research direction.

D. Brain-Like Intelligence and Brain-Inspired System

The two fields of neuroscience and artificial intelligence have always been intersecting. A better understanding of the biological brain will have significant guidance on the construction of intelligent machines [292]. Neurobiology focuses on the biological mechanism of brain tissues, such as neurons and synapses [293]. However, computational neurology mainly uses the biological mechanism to mathematically model the structure of neurons and synapses, simulating the environment to find its characteristics similar to the biological brain. The human brain directly or indirectly inspires all human-made sciences, technologies, and systems. It imitates the brain as an autonomous intelligent system in the environment of humans, animals, machines, and society [294].

Although the emergence of deep learning and big data in recent years has made some models surpass humans in some tasks, it is powerless to deal with the complex problems that the human brain can handle. In addition, it requires a large number of computing resources and data resources as support. On the contrary, the human brain is an incredibly optimized system and its low power consumption makes the human brain an absolute advantage in handling complex problems [295], [296]. Therefore, people hope to imitate the way the human brain works to process information.

Brain-inspired algorithms have attracted the attention of researchers in more countries in recent years. The earliest neuron network was a Perceptron proposed by American computer scientist Rosenblatt [297]. The Perceptron can only have a single layer and can only complete linear classification and regression tasks. Later, a well-known backpropagation algorithm was born [14], [298], [299], which inspired the brain to modify synapses to improve the behavior during the learning process. These algorithms are developed in a data science background, abstracting neurons' input and output into vectors and matrices. The function of neurons is mainly to perform matrix multiplication. The processing of information by biological neurons is not a matrix of analog and digital, but biological electrical signals that appear in pulses [300]–[302].

At present, there are still many gaps in the study of the cognitive mechanism of the brain [303]. In addition, the characteristics of the biological brain's operating mechanism have not been completely discovered. To some extent, BLS seems to fill the gaps in traditional machine learning and biological learning mechanisms. BLS can be stable within a range of active neurons, and it is easy to implement due to its flat structure concurrently. It reveals that BLS is different from some algorithms that are very sensitive to artificial prefabricated parameters and does not require too much human intervention. At the same time, a network type can meet the requirements of different applications. Its incremental learning form is more in line with the characteristics of biological learning. The advantage of fast computing provided by the pseudoinverse algorithm also makes it easier for BLS to meet actual applications' real-time characteristics. The above-mentioned shows that BLS embodies the mechanism of biological learning in many aspects. Wang *et al.* [146] assumed the vision to determine the robot's attention, and this "vision-brain" cognition is very robust. They integrated BLS and vision-brain to solve the problem of FER. Only a small part of computational neurology methods uses biological methods to simulate the connection of the human brain. There are even fewer mechanisms used in brain-like computing. Brain-like computing algorithms will have great research value [304]–[306]. Therefore, how to apply BLS to brain-like computing is also a promising direction in the future.

E. Edge Computing

Edge computing allows application developers and service providers to implement cloud and Internet services on the network's edge. Its goal is to provide computing, storage, and network bandwidth near input data or users. Different application scenarios, such as the IoT [255], [256], [307]; smart homes [308], [309]; military operations [310], [311]; and autonomous vehicles [312] all create massive connections and the resulting massive data to be processed. If these huge amounts of data are uploaded to the cloud computing center in real time and make decisions, they may generate many difficult computing power challenges and bandwidth limits. The problem of immediate response due to delay is more obvious than before. Edge computing has shown its advantages when faced with such a situation. Because it is deployed near the device side, related devices can instantly feedback and filter most of the lower value data through algorithms. It can effectively reduce the load on the cloud and make massive links and massive data processing possible.

Edge intelligence is also becoming the focus of attention [313], [314]. It enables the combination of edge-end devices and artificial intelligence technology to mine more valuable data on the side and make higher quality decisions. However, current AI technologies are relatively time-consuming and insufficient to support edge devices' resource limitation conditions. Moreover, when new data are generated, they cannot be updated in real time.

Based on the above analysis, it is natural to think that many advantages of BLS seem to meet edge computing or edge intelligence requirements. Edge devices hope to minimize the number of agent parameters and storage space [315]. BLS is a flat network, and increasing the network's width grows the number of parameters much less than depth. Its storage space required can handle a variety of applications. Second, edge computing requires that edge-end agents can reduce computational complexity and energy consumption. Because the weight connects input data, the hidden layer of BLS is randomly generated. Only the weights between the hidden layer and output layer are learned through a fast pseudoinverse algorithm. This advantage can handle it and also make the corresponding time of the edge to achieve the characteristics of real time [316]. Finally, BLS's incremental learning method enables edge agents to update timely when faced with newly added data without the need to reconstruct the entire network architecture. Peng *et al.* proposed a novel edge computing-based traffic analysis system using broad learning. It shows that BLS can incrementally train the traffic data, which is more suitable for edge computing [317]. Therefore, considering the match between BLS's characteristics and the actual application of edge computing, how to deploy it in edge computing is also a direction that is worth studying in the future.

XI. CONCLUSION

In this article, we conducted a comprehensive survey of the BLS and its latest developments in many fields in recent years. First, BLS's fundamental representation was introduced in detail, including the feature-space mapping method of its data and the pseudoinverse calculation method of the output layer weights. The activation functions of some commonly used enhancement layers were listed. We also focused on the essence of BLS. This neural network of random hidden-layer neurons without a deep architecture makes it with the advantage of discriminative feature space, universal approximation capability, unified learning theory, generalization performance, and fast increment.

Since the birth of BLS, people have become more and more interested in this research topic. They have proposed many necessary theoretical foundations and useful variants involving classification, regression, SSL, unsupervised learning, and other different tasks. We also gave a roadmap with some milestone variants, which helps readers understand the development of BLSs from a macroscopic perspective.

Besides, it summarizes all the BLS application fields so far, including computer vision and image processing, medical data analysis, system modeling and prediction, fault detection, system control, etc. Empirical results in many fields indicated that BLS and its variants have significant advantages, such as high efficiency, high accuracy, and rapid increment. Finally, by comparing BLS and traditional learning algorithms, many advantages were obtained that are conducive to future research on BLSs. At the same time, five promising and promising future directions were proposed, including: 1) hardware implementation; 2) the tradeoff between depth and width in the neural network; 3) random weight mechanism; 4) brain-like intelligence and brain-inspired system; and 5) edge computing. In the future, we intend to give a comprehensive evaluation of BLSs under a specific quantitative context, including the effect of the bias in the output neuron, theoretical proof of stacked BLS, the effect of range for randomly generated parameters in hidden neurons, etc. We hope that this comprehensive survey will help readers understand BLS research and help their future research.

ACKNOWLEDGMENT

The authors would like to thank the support of the Data Science and Artificial Intelligence Laboratory (D-Sail) of the South China University of Technology for participating in the study of BLS projects. The authors also would like to thank the helpful discussions with group members of D-sail and the anonymous reviewers for their valuable comments.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, 2014.

[3] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[4] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[5] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," in *Proc. 12th Int. Conf. Artif. Intell. Stat.*, 2009, pp. 448–455.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2012, pp. 1097–1105.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: arXiv:1409.1556.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[9] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 1–9.

[10] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Netw.*, vol. 6, no. 6, pp. 861–867, 1993.

[11] J.-X. Peng, K. Li, and G. W. Irwin, "A new Jacobian matrix for optimal learning of single-layer neural networks," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 119–129, Jan. 2008.

[12] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Netw.*, vol. 2, no. 6, pp. 459–473, 1989.

[13] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: Theory, system architecture, and functionalities," *Computer*, vol. 25, no. 5, pp. 76–79, May 1992.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[15] Y. Ren, P. N. Suganthan, N. Srikanth, and G. Amaratunga, "Random vector functional link network for short-term electricity load demand forecasting," *Inf. Sci.*, vols. 367–368, pp. 1078–1093, Nov. 2016.

[16] P. K. Dash, H. P. Satpathy, A. C. Liew, and S. Rahman, "A real-time short-term load forecasting system using functional link network," *IEEE Trans. Power Syst.*, vol. 12, no. 2, pp. 675–680, May 1997.

[17] 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.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[18] C. L. P. Chen, Z. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1191–1204, Apr. 2019.

[19] C. L. P. Chen and Z. Liu, "Broad learning system: A new learning paradigm and system without going deep," in *Proc. IEEE 32nd Youth Acad. Annu. Conf. Chin. Assoc. Autom. (YAC)*, Hefei, China, 2017, pp. 1271–1276.

[20] Y.-H. Pao, S. M. Phillips, and D. J. Sobajic, "Neural-net computing and the intelligent control of systems," *Int. J. Control*, vol. 56, no. 2, pp. 263–289, 1992.

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

[22] L. Zhang and P. N. Suganthan, "A comprehensive evaluation of random vector functional link networks," *Inf. Sci.*, vols. 367–368, pp. 1094–1105, Nov. 2016.

[23] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[24] S. Ruping, "Incremental learning with support vector machines," in *Proc. IEEE Int. Conf. Data Mining*, San Jose, CA, USA, 2001, pp. 641–642.

[25] C. L. P. Chen and J. Z. Wan, "A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 1, pp. 62–72, Feb. 1999.

[26] C. L. P. Chen, "A rapid supervised learning neural network for function interpolation and approximation," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1220–1230, Sep. 1996.

[27] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[28] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, Mar. 2002.

[29] L. Xu and C. L. P. Chen, "Comparison and combination of activation functions in broad learning system," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Toronto, ON, Canada, 2020, pp. 3537–3542.

[30] J. Xu *et al.*, "Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images," *IEEE Trans. Med. Imag.*, vol. 35, no. 1, pp. 119–130, Jan. 2016.

[31] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, pp. 171–178, Jul. 2016.

[32] S. Feng and C. L. P. Chen, "Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 414–424, Feb. 2020.

[33] S. Feng and C. L. P. Chen, "Nonlinear system identification using a simplified fuzzy broad learning system: Stability analysis and a comparative study," *Neurocomputing*, vol. 337, pp. 274–286, Apr. 2019.

[34] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.

[35] P. Bergsten, R. Palm, and D. Driankov, "Observers for Takagi-Sugeno fuzzy systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 1, pp. 114–121, Feb. 2002.

[36] J. Lin, Z. Liu, and J. Lai, "Wavelet broad learning filter: A novel adaptive filter for estimating the physiological tremor in teleoperation," in *Proc. IEEE Int. Conf. Security Pattern Anal. Cybern. (SPAC)*, Jinan, China, 2018, pp. 354–359.

[37] J. Lin, Z. Liu, C. L. P. Chen, and Y. Zhang, "Three-domain fuzzy wavelet broad learning system for tremor estimation," *Knowl. Based Syst.*, vol. 192, Mar. 2020, Art. no. 105295.

[38] J.-T. Lin, Z. Liu, C. L. P. Chen, and Y. Zhang, "Quaternion broad learning system: A novel multi-dimensional filter for estimation and elimination tremor in teleoperation," *Neurocomputing*, vol. 380, pp. 78–86, Mar. 2020.

[39] J. Lin, Z. Liu, C. L. P. Chen, and Y. Zhang, "A wavelet broad learning adaptive filter for forecasting and cancelling the physiological tremor in teleoperation," *Neurocomputing*, vol. 356, pp. 170–183, Sep. 2019.

[40] Z. Liu, C. L. P. Chen, T. Zhang, and J. Zhou, "Multi-kernel broad learning systems based on random features: A novel expansion for nonlinear feature nodes," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Bari, Italy, 2019, pp. 193–197.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 4700–4708.

[43] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2015, pp. 649–657.

[44] S. Furao and O. Hasegawa, "An incremental network for on-line unsupervised classification and topology learning," *Neural Netw.*, vol. 19, no. 1, pp. 90–106, 2006.

[45] T. Zhang, G. Su, C. Qing, X. Xu, B. Cai, and X. Xing, "Hierarchical lifelong learning by sharing representations and integrating hypothesis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 2, pp. 1004–1014, Feb. 2021.

[46] L. Zhang *et al.*, "Analysis and variants of broad learning system," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Jun. 2, 2020, doi: 10.1109/TSMC.2020.2995205.

[47] H.-D. Block, "The perceptron: A model for brain functioning. I," *Rev. Mod. Phys.*, vol. 34, no. 1, p. 123, 1962.

[48] W. F. Schmidt, M. A. Kraaijveld, and R. P. W. Duin, "Feedforward neural networks with random weights," in *Proc. Int. Conf. Pattern Recognit. Vol. II Conf. B, Pattern Recognit. Methodol. Syst.*, Hague, The Netherlands, 1992, pp. 1–4.

[49] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, vol. 375. Dordrecht, The Netherlands: Springer, 1996,

[50] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Comput.*, vol. 7, no. 2, pp. 219–269, Mar. 1995.

[51] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning Theory and Kernel Machines*. Heidelberg, Germany: Springer, 2003, pp. 144–158.

[52] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Occam's razor," *Inf. Process. Lett.*, vol. 24, no. 6, pp. 377–380, 1987.

[53] J.-W. Jin and C. L. P. Chen, "Regularized robust broad learning system for uncertain data modeling," *Neurocomputing*, vol. 322, pp. 58–69, Dec. 2018.

[54] J. Jin, C. L. P. Chen, and Y. Li, "Robust broad learning system for uncertain data modeling," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Miyazaki, Japan, 2018, pp. 3524–3529.

[55] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Regularized robust coding for face recognition," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1753–1766, May 2013.

[56] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2010. [Online]. Available: arXiv:1009.5055.

[57] X. Fang, Y. Xu, X. Li, Z. Lai, W. K. Wong, and B. Fang, "Regularized label relaxation linear regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1006–1018, Apr. 2018.

[58] M. Gan, H.-T. Zhu, G.-Y. Chen, and C. L. P. Chen, "Weighted generalized cross-validation-based regularization for broad learning system," *IEEE Trans. Cybern.*, early access, Sep. 9, 2020, doi: 10.1109/TCYB.2020.3015749.

[59] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, no. 9, pp. 1481–1497, Sep. 1990.

[60] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1320–1329, Nov. 1995.

[61] P. Huang and B. Chen, "Bidirectional broad learning system," in *Proc. IEEE 7th Int. Conf. Ind. Eng. Appl. (ICIEA)*, Bangkok, Thailand, 2020, pp. 963–968.

[62] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[63] J. Du, C.-M. Vong, and C. L. P. Chen, "Novel efficient RNN and LSTM-like architectures: Recurrent and gated broad learning systems and their applications for text classification," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1586–1597, Mar. 2021.

[64] M. Xu, M. Han, C. L. P. Chen, and T. Qiu, "Recurrent broad learning systems for time series prediction," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1405–1417, Apr. 2020.

[65] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sens. Actuat. B, Chem.*, vol. 129, no. 2, pp. 750–757, 2008.

[66] S.-C. Kuok and K.-V. Yuen, "Model-free data reconstruction of structural response and excitation via sequential broad learning," *Mech. Syst. Signal Process.*, vol. 141, Jul. 2020, Art. no. 106738.

[67] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, 2009, pp. 248–255.

[68] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[69] P. N. Druzhkov and V. D. Kustikova, "A survey of deep learning methods and software tools for image classification and object detection," *Pattern Recognit. Image Anal.*, vol. 26, no. 1, pp. 9–15, 2016.

[70] L. Zhang and P. N. Suganthan, "Visual tracking with convolutional random vector functional link network," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3243–3253, Oct. 2017.

[71] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-celeb-1M: A dataset and benchmark for large-scale face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 87–102.

[72] T. Li, B. Fang, J. Qian, and X. Wu, "CNN-based broad learning system," in *Proc. IEEE 4th Int. Conf. Signal Image Process. (ICSIP)*, Wuxi, China, 2019, pp. 132–136.

[73] F. Yang, "A CNN-based broad learning system," in *Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, 2018, pp. 2105–2109.

[74] W. Yu and C. Zhao, "Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability," *IEEE Trans. Ind. Electron.*, vol. 67, no. 6, pp. 5081–5091, Jun. 2020.

[75] W. Dai, Q. Liu, and T. Chai, "Particle size estimate of grinding processes using random vector functional link networks with improved robustness," *Neurocomputing*, vol. 169, pp. 361–372, Dec. 2015.

[76] R. Gribonval, R. Jenatton, and F. Bach, "Sparse and spurious: Dictionary learning with noise and outliers," *IEEE Trans. Inf. Theory*, vol. 61, no. 11, pp. 6298–6319, Nov. 2015.

[77] 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 Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3017–3031, Aug. 2020.

[78] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[79] Y. Zheng, B. Chen, S. Wang, and W. Wang, "Broad learning system based on maximum correntropy criterion," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 24, 2020, doi: 10.1109/TNNLS.2020.3009417.

[80] S. Feng and C. L. P. Chen, "Broad learning system for control of nonlinear dynamic systems," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Miyazaki, Japan, 2018, pp. 2230–2235.

[81] S. Feng and C. L. P. Chen, "Performance analysis of fuzzy BLS using different cluster methods for classification," in *Diabetes*, vol. 75. New Delhi, India: Jaypee Health Sci. Publ., 2018, pp. 25–81.

[82] S. Feng, C. L. P. Chen, L. Xu, and Z. Liu, "On the accuracy-complexity trade-off of fuzzy broad learning system," *IEEE Trans. Fuzzy Syst.*, early access, Jul. 16, 2020, doi: 10.1109/TFUZZ.2020.3009757.

[83] 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.*, early access, Aug. 30, 2019, doi: 10.1109/TCYB.2019.2934823.

[84] X. Lin, L. Ma, B. Sheng, Z.-J. Wang, and W. Chen, "Utilizing two-phase processing with FBLS for single image deraining," *IEEE Trans. Multimedia*, vol. 23, pp. 664–676, Apr. 2021. doi: 10.1109/TMM.2020.2987703.

[85] H. Huang, C. Yang, Z. Ju, Y. Yuan, and Z. Li, "Broad fuzzy neural control using impedance learning," in *Proc. IEEE 4th Int. Conf. Adv. Robot. Mechatronics (ICARM)*, Toyonaka, Japan, 2019, pp. 173–178.

[86] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Inf. Fusion*, vol. 38, pp. 43–54, Nov. 2017.

[87] Z. Shi, X. Chen, C. Zhao, H. He, V. Stuphorn, and D. Wu, "Multi-view broad learning system for primate oculomotor decision decoding," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 9, pp. 1908–1920, Sep. 2020.

[88] D. C. Conner, A. A. Rizzi, and H. Choset, "Composition of local potential functions for global robot control and navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 4. Las Vegas, NV, USA, 2003, pp. 3546–3551.

[89] B. P. Bean, "The action potential in mammalian central neurons," *Nat. Rev. Neurosci.*, vol. 8, no. 6, pp. 451–465, 2007.

[90] S. D. Stavisky, J. C. Kao, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy, "A high performing brain–machine interface driven by low-frequency local field potentials alone and together with spikes," *J. Neural Eng.*, vol. 12, no. 3, 2015, Art. no. 036009.

[91] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[92] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[93] J. Jin, Z. Liu, and C. L. P. Chen, "Discriminative graph regularized broad learning system for image recognition," *Sci. China Inf. Sci.*, vol. 61, no. 11, 2018, Art. no. 112209.

[94] H. Wang, X. Wang, C. L. P. Chen, and Y. Cheng, "Hyperspectral image classification based on domain adaptation broad learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3006–3018, Jun. 2020. doi: 10.1109/JSTARS.2020.3001198.

[95] 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 Trans. Knowl. Data Eng.*, vol. 31, no. 9, pp. 1809–1821, Sep. 2019.

[96] S. Feng, W. Ren, M. Han, and Y. W. Chen, "Robust manifold broad learning system for large-scale noisy chaotic time series prediction: A perturbation perspective," *Neural Netw.*, vol. 117, pp. 179–190, Sep. 2019.

[97] R. Polikar, "Ensemble learning," in *Ensemble Machine Learning*. Boston, MA, USA: Springer, 2012, pp. 1–34.

[98] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

[99] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[100] Y. Wang, D. Wang, N. Geng, Y. Wang, Y. Yin, and Y. Jin, "Stacking-based ensemble learning of decision trees for interpretable prostate cancer detection," *Appl. Soft Comput.*, vol. 77, pp. 188–204, Apr. 2019.

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

[102] X. Pan, Y.-X. Fan, J. Yan, and H.-B. Shen, "IPMiner: Hidden ncRNA-protein interaction sequential pattern mining with stacked autoencoder for accurate computational prediction," *BMC Genom.*, vol. 17, no. 1, p. 582, 2016.

[103] H. M. Berman *et al.*, "The protein data bank," *Nucl. Acids Res.*, vol. 28, no. 1, pp. 235–242, 2000.

[104] M. Wang, Q. Ge, H. Jiang, and G. Yao, "Wear fault diagnosis of aeroengines based on broad learning system and ensemble learning," *Energies*, vol. 12, no. 24, p. 4750, 2019.

[105] 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," *Cogn. Comput.*, vol. 12, pp. 654–666, Dec. 2019.

[106] H. Ye, H. Li, and C. L. P. Chen, "Adaptive deep cascade broad learning system and its application in image denoising," *IEEE Trans. Cybern.*, early access, Mar. 23, 2020, doi: 10.1109/TCYB.2020.2978500.

[107] 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 Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 209–222, Jan. 2021.

[108] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, San Rafael, CA, USA: Morgan & Claypool Publishers, 2009, pp. 1–130.

[109] X. J. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. of Wisconsin-Madison, Madison, WI, USA, Rep. TR 1530, 2005.

[110] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 5070–5079.

[111] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2271–2282, May 2010.

[112] H. Zhao, J. Zheng, W. Deng, and Y. Song, "Semi-supervised broad learning system based on manifold regularization and broad network," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 3, pp. 983–994, Mar. 2020.

[113] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Dec. 2006.

[114] Y. Kong, X. Wang, Y. Cheng, and C. L. P. Chen, "Hyperspectral imagery classification based on semi-supervised broad learning system," *Remote Sens.*, vol. 10, no. 5, p. 685, 2018.

[115] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.

[116] X. Liu, T. Qiu, C. Chen, H. Ning, and N. Chen, "An incremental broad learning approach for semi-supervised classification," in *Proc. IEEE Int. Conf. Depend. Auton. Secure Comput. Int. Conf. Pervasive Intell. Comput. Int. Conf. Cloud Big Data Comput. Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, Fukuoka, Japan, 2019, pp. 250–254.

[117] Q. She, Y. Zhou, H. Gan, Y. Ma, and Z. Luo, "Decoding EEG in motor imagery tasks with graph semi-supervised broad learning," *Electronics*, vol. 8, no. 11, p. 1273, 2019.

[118] F. Lotte and C. Guan, "Regularizing common spatial patterns to improve BCI designs: Unified theory and new algorithms," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 2, pp. 355–362, Feb. 2011.

[119] J. Zheng, Y. Yuan, H. Zhao, and W. Deng, "A novel broad learning model-based semi-supervised image classification method," *IEEE Access*, vol. 8, pp. 116756–116765, 2020.

[120] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.

[121] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, pp. 845–889, Dec. 2004.

[122] Z. Liu, J. Zhou, and C. L. P. Chen, "Broad learning system: Feature extraction based on k-means clustering algorithm," in *Proc. IEEE 4th Int. Conf. Inf. Cybern. Comput. Soc. Syst. (ICCSS)*, Dalian, China, 2017, pp. 683–687.

[123] Q. Feng, C. L. P. Chen, and L. Chen, "Broad graph-based non-negative robust continuous clustering," *IEEE Access*, vol. 8, pp. 121693–121704, 2020.

[124] Y. Kong, Y. Cheng, C. L. P. Chen, and X. Wang, "Hyperspectral image clustering based on unsupervised broad learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1741–1745, Nov. 2019.

[125] B. Pan, Z. Shi, and X. Xu, "Hierarchical guidance filtering-based ensemble classification for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 4177–4189, Jul. 2017.

[126] J. F. Mota, J. M. Xavier, P. M. Q. Aguiar, and M. Püschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2718–2723, May 2013.

[127] A. G. sS. Filho *et al.*, "Hyperspectral images clustering on reconfigurable hardware using the *k*-means algorithm," in *Proc. IEEE 16th Symp. Integr. Circuits Syst. Design (SBCCI)*, Sao Paulo, Brazil, 2003, pp. 99–104.

[128] H. Zhang, H. Zhai, L. Zhang, and P. Li, "Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3672–3684, Jun. 2016.

[129] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.

[130] G. Hu, X. Peng, Y. Yang, T. M. Hospedales, and J. Verbeek, "Frankenstein: Learning deep face representations using small data," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 293–303, Jan. 2018.

[131] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2017, pp. 4077–4087.

[132] H. Tang, P. Dong, and Y. Shi, "A construction of robust representations for small data sets using broad learning system," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Dec. 24, 2019, doi: 10.1109/TSMC.2019.2957818.

[133] T.-L. Zhang, R. Chen, X. Yang, and S. Guo, "Rich feature combination for cost-based broad learning system," *IEEE Access*, vol. 7, pp. 160–172, 2018.

[134] T. Zhang, X. Gong, and C. L. P. Chen, "BMT-Net: Broad multitask transformer network for sentiment analysis," *IEEE Trans. Cybern.*, early access, Mar. 4, 2021, doi: 10.1109/TCYB.2021.3050508.

[135] Y. Dang, F. Yang, and J. Yin, "DWNet: Deep-wide network for 3D action recognition," *Robot. Auton. Syst.*, vol. 126, Apr. 2020, Art. no. 103441.

[136] T. Zhang, X. Wang, X. Xu, and C. L. P. Chen, "GCB-Net: Graph convolutional broad network and its application in emotion recognition," *IEEE Trans. Affect. Comput.*, early access, Aug. 27, 2019, doi: 10.1109/TAFFC.2019.2937768.

[137] X.-R. Gong, J.-X. Jin, and T. Zhang, "Sentiment analysis using autoregressive language modeling and broad learning system," in *Proc. IEEE Int. Conf. Bioinformat. Biomed. (BIBM)*, San Diego, CA, USA, 2019, pp. 1130–1134.

[138] K. Shen, Y. Zhuang, and Y. Zhu, "Incremental learning-based land mark recognition for mirco-UAV autonomous landing," in *Proc. IEEE 39th Chin. Control Conf. (CCC)*, Shenyang, China, 2020, pp. 6786–6791.

[139] J. A. Starzyk, R. Niemiec, and A. Horzyk, "Feature significance in wide neural networks," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, Xiamen, China, 2019, pp. 909–916.

[140] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[141] Z. Liu, B. Chen, B. Xie, H. Qiang, and Z. Zhu, "Feature selection for orthogonal broad learning system based on mutual information," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, 2019, pp. 1–8.

[142] Q. Zhou and X. He, "Broad learning model based on enhanced features learning," *IEEE Access*, vol. 7, pp. 42536–42550, 2019.

[143] T. Zhang, Z.-L. Liu, X.-H. Wang, X.-F. Xing, C. L. P. Chen, and E. Chen, "Facial expression recognition via broad learning system," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Miyazaki, Japan, 2018, pp. 1898–1902.

[144] J. Han, L. Xie, J. Liu, and X. Li, "Personalized broad learning system for facial expression," *Multimedia Tools Appl.*, vol. 79, pp. 16627–16644, Jul. 2019.

[145] C. Liu, L. Feng, S. Guo, H. Wang, S. Liu, and H. Qiao, "An incrementally cascaded broad learning framework to facial landmark tracking," *Neurocomputing*, vol. 410, pp. 125–137, Oct. 2020.

[146] W. Wang, X. Deng, L. Ding, and L. Zhang, "The vision–brain hypothesis," in *Brain-Inspired Intelligence and Visual Perception*. Singapore: Springer, 2020, pp. 17–39.

[147] D. Zhang, H. Yang, P. Chen, and T. Li, "A face recognition method based on broad learning of feature block," in *Proc. IEEE 9th Annu. Int. Conf. CYBER Technol. Autom. Control Intell. Syst. (CYBER)*, Suzhou, China, 2019, pp. 307–310.

[148] B. Luan, X. Li, and W. Wang, "Broad recognition of fake faces," in *Proc. 5th Int. Conf. Adv. Robot. Mechatronics (ICARM)*, Shenzhen, China, 2020, pp. 21–24.

[149] Y. Shi *et al.*, "Student body gesture recognition based on fisher broad learning system," *Int. J. Wavelets Multiresolution Inf. Process.*, vol. 17, no. 01, 2019, Art. no. 1950001.

[150] W. Sun, Z. Zhou, and H. Li, "Sitting posture recognition in real-time combined with index map and BLS," in *Proc. 3rd Int. Conf. Innovat. Artif. Intell.*, 2019, pp. 101–105.

[151] J. Lin, Z. Liu, and J. Lai, "sEMG gestures recognition based on wavelet broad learning system," in *Proc. Chin. Autom. Congr. (CAC)*, Hangzhou, China, 2019, pp. 1658–1662.

[152] F. Lei, Y. Wei, J. Hu, H. Yao, W. Deng, and Y. Lu, "Student action recognition based on multiple features," in *Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Soc. Comput. (CPSCom) IEEE Smart Data (SmartData)*, Atlanta, GA, USA, 2019, pp. 428–432.

[153] J. Li, J. Zhong, F. Chen, and C. Yang, "An incremental learning framework for skeletal-based hand gesture recognition with leap motion," in *Proc. IEEE 9th Annu. Int. Conf. CYBER Technol. Autom. Control Intell. Syst. (CYBER)*, Suzhou, China, 2019, pp. 13–18.

[154] H. Yao, M. Yang, T. Chen, Y. Wei, and Y. Zhang, "Depth-based human activity recognition via multi-level fused features and fast broad learning system," *Int. J. Distrib. Sens. Netw.*, vol. 16, no. 2, 2020, Art. no. 1550147720907830.

[155] Z. Lin, H. Chen, Q. Yang, and X. Hong, "A flexible approach for human activity recognition based on broad learning system," in *Proc. 11th Int. Conf. Mach. Learn. Comput.*, 2019, pp. 368–373.

[156] S. Luo, C. Zhao, and Y. Fu, "An intelligent human activity recognition method with incremental learning capability for bedridden patients," in *Proc. IEEE 15th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, 2018, pp. 1284–1289.

[157] L. Guo, R. Li, X. Shen, and B. Jiang, "Crack and noncrack damage automatic classification from concrete surface images using broad network architecture," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Nanchang, China, 2019, pp. 1966–1971.

[158] C. Cai *et al.*, "Classification of chinese herbal medicine using combination of broad learning system and convolutional neural network," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Bari, Italy, 2019, pp. 3907–3912.

[159] B. Wang and C. L. P. Chen, "License plate character segmentation using key character location and projection analysis," in *Proc. Int. Conf. Security Pattern Anal. Cybern. (SPAC)*, Jinan, China, 2018, pp. 510–514.

[160] Y. Dang, F. Yang, B. Su, J. Yin, and J. Liu, "DBNet: A new generalized structure efficient for classification," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dali, China, 2019, pp. 1–6.

[161] Y. Dang, L. Wang, J. Yin, X. Zhu, Z. Feng, and J. Guo, "Estimating cement compressive strength from microstructure images using broad learning system," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Miyazaki, Japan, 2018, pp. 4230–4235.

[162] Y. Zhai and Y. Liu, "Distributed broad learning system," in *Proc. 12th Int. Conf. Mach. Learn. Comput.*, 2020, pp. 567–573.

[163] J. Fan, X. Wang, X. Wang, J. Zhao, and X. Liu, "Incremental wishart broad learning system for fast polsar image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1854–1858, Dec. 2019.

[164] P. Chen, "Hyperspectral image classification based on broad learning system with composite feature," in *Proc. IEEE Int. Conf. Power Intell. Comput. Syst. (ICPICS)*, Shenyang, China, 2020, pp. 842–846.

[165] Y. Han, T. Li, Y. Zuo, Y. Tian, Y. Cao, and C. L. P. Chen, "Application of broad learning system for container number identification," in *Proc. IEEE Int. Conf. Security Pattern Anal. Cybern. (SPAC)*, Jinan, China, 2018, pp. 332–336.

[166] J. Zhang, X. Wang, and Y. Cheng, "Broad attribute prediction model with enhanced attribute and feature," *IEEE Access*, vol. 7, pp. 124606–124620, 2019.

[167] Z. Wang, H. H. Liu, and X. Xu, "Broad learning for robotic material recognition," in *Proc. IEEE Chin. Autom. Congr. (CAC)*, Hangzhou, China, 2019, pp. 1917–1922.

[168] Q. Zheng, P. Zhang, X. You, F. Wang, and Z. Liu, "Hierarchical learning for salient object detection," in *Proc. IEEE Int. Conf. Security Pattern Anal. Cybern. (SPAC)*, Shenzhen, China, 2017, pp. 192–197.

[169] Y. Jin, B. Sheng, P. Li, and C. L. P. Chen, "Broad colorization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 2, 2020, doi: 10.1109/TNNLS.2020.3004634.

[170] S. Shrivastava, A. Aggarwal, and P. Chattopadhyay, "Broad neural network for change detection in aerial images," in *Emerging Technology in Modelling and Graphics*. Singapore: Springer, 2020, pp. 327–339.

[171] C. L. P. Chen and B. Wang, "Random-positioned license plate recognition using hybrid broad learning system and convolutional networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Aug. 4, 2020, doi: 10.1109/TITS.2020.3011937.

[172] M. Lei, Z. Rao, M. Li, X. Yu, and L. Zou, "Identification of coal geographical origin using near infrared sensor based on broad learning," *Appl. Sci.*, vol. 9, no. 6, p. 1111, 2019.

[173] Z. Chen, T. Gao, B. Sheng, P. Li, and C. L. P. Chen, "Outdoor shadow estimating using multiclass geometric decomposition based on BLS," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 2152–2165, May 2020.

[174] Y. Zhai *et al.*, "Facial beauty prediction via local feature fusion and broad learning system," *IEEE Access*, vol. 8, pp. 218444–218457, 2020.

[175] 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 Trans. Syst., Man, Cybern., Syst.*, early access, Dec. 6, 2019, doi: 10.1109/TSMC.2019.2956022.

[176] X. Jia *et al.*, "Multi-channel EEG based emotion recognition using temporal convolutional network and broad learning system," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Toronto, ON, Canada, 2020, pp. 2452–2457.

[177] Y. Zheng, X. Qin, Z. Xi, and B. Chen, "Mixed-norm based broad learning system for EEG classification," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Berlin, Germany, 2019, pp. 7092–7095.

[178] 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 Trans. Syst., Man, Cybern., Syst.*, early access, Feb. 4, 2020, doi: 10.1109/TSMC.2020.2964684.

[179] S. Issa, Q. Peng, and X. You, "Emotion classification using EEG brain signals and the broad learning system," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Feb. 13, 2020, doi: 10.1109/TSMC.2020.2969686.

[180] J. Zou, Q. She, F. Gao, and M. Meng, "Multi-task motor imagery EEG classification using broad learning and common spatial pattern," in *Proc. Int. Conf. Intell. Sci.*, 2018, pp. 3–10.

[181] S. Luo and C. Zhao, "Transfer and incremental learning method for blood glucose prediction of new subjects with type 1 diabetes," in *Proc. 12th Asian Control Conf. (ASCC)*, Kitakyushu, Japan, 2019, pp. 73–78.

[182] L. Shen *et al.*, "Multiple empirical kernel mapping based broad learning system for classification of Parkinson's disease with transcranial sonography," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, 2018, pp. 3132–3135.

[183] K. Zheng, Z.-H. You, L. Wang, Y.-R. Li, Y.-B. Wang, and H.-J. Jiang, "MISSIM: Improved miRNA-disease association prediction model based on chaos game representation and broad learning system," in *Proc. Int. Conf. Intell. Comput.*, 2019, pp. 392–398.

[184] J. Xu, P. Wu, Y. Chen, H. Dawood, and D. Wang, "The wide and deep flexible neural tree and its ensemble in predicting long non-coding RNA subcellular localization," in *Proc. Int. Conf. Intell. Comput.*, 2018, pp. 515–525.

[185] M. Chew, P. Wu, and Y. Chen, "Operon prediction using broad learning system," in *Proc. Int. Conf. Elect. Commun. Comput. Eng. (ICECCE)*, Istanbul, Turkey, 2020, pp. 1–5.

[186] R. Han, C. L. P. Chen, and Z. Liu, "A novel convolutional variation of broad learning system for Alzheimer's disease diagnosis by using mri images," *IEEE Access*, vol. 8, pp. 214646–214657, 2020.

[187] H. Liu, A. Pi, and V. Chaudhary, "Broad learning-based intervertebral discs localization and segmentation," in *Proc. 3rd Int. Symp. Image Comput. Digit. Med.*, 2019, pp. 263–268.

[188] Y. Liu, C. Wang, and Y. Wei, "Hippocampus segmentation in MR image based on atlas registration and broad learning," in *Proc. IEEE Chin. Control Decis. Conf. (CCDC)*, Nanchang, China, 2019, pp. 2707–2711.

[189] 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.*, early access, Jul. 22, 2020, doi: 10.1109/TNNLS.2020.3004253.

[190] L. Zhu and C. Lian, "Wind speed forecasting based on a hybrid EMD-BLS method," in *Proc. IEEE Chin. Autom. Congr. (CAC)*, Hangzhou, China, 2019, pp. 2191–2195.

[191] Z. Cui and C. Zhao, "Spatio-temporal broad learning networks for traffic speed prediction," in *Proc. IEEE 12th Asian Control Conf. (ASCC)*, Kitakyushu, Japan, 2019, pp. 1536–1541.

[192] N. Zhou, J. Lu, Z. Yan, and X. Xu, "Short-term load forecasting using broad learning system," in *Proc. IEEE 3rd Conf. Energy Internet Energy Syst. Integr. (EI2)*, Changsha, China, 2019, pp. 497–501.

[193] W. Li, M. Han, and S. Feng, "Multivariate chaotic time series prediction: Broad learning system based on sparse PCA," in *Proc. Int. Conf. Neural Inf. Process.*, 2018, pp. 56–66.

[194] J. Yang *et al.*, "Forecast application of time series model based on BLS in port Cargo throughput," in *Proc. IEEE Int. Conf. Security Pattern Anal. Cybern. (SPAC)*, Jinan, China, 2018, pp. 249–254.

[195] F. Chu, J. Su, T. Liang, X. Ma, and F. Wang, "Modeling method for performance prediction of centrifugal compressor based on broad learning system," in *Proc. IEEE Chin. Control Decis. Conf. (CCDC)*, Nanchang, China, 2019, pp. 5223–5227.

[196] S.-C. Kuok and K.-V. Yuen, "Broad learning for nonparametric spatial modeling with application to seismic attenuation," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 35, no. 3, pp. 203–218, 2020.

[197] S.-C. Kuok, K.-V. Yuen, S. Roberts, and M. A. Girolami, "Propagative broad learning for nonparametric modeling of ambient effects on structural health indicators," *Struct. Health Monitoring*, to be published.

[198] S.-C. Kuok and K.-V. Yuen, "Broad learning system for nonparametric modeling of clay parameters," *ASCE-ASME J. Risk Uncertainty Eng. Syst. A, Civil Eng.*, vol. 6, no. 2, 2020, Art. no. 04020024.

[199] B. Sheng, P. Li, Y. Zhang, L. Mao, and C. L. P. Chen, "GreenSea: Visual soccer analysis using broad learning system," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1463–1477, Mar. 2021.

[200] C.-Y. Zhang, Y.-Y. Xiao, and J.-Q. Pu, "A multifunctional and robust learning approach for human motion modelling," in *Proc. IEEE 6th Int. Conf. Inf. Cybern. Comput. Soc. Syst. (ICCSS)*, Chongqing, China, 2019, pp. 138–145.

[201] S.-C. Kuok and K.-V. Yuen, "Multi-resolution broad learning for model updating using incomplete modal data," *Struct. Control Health Monitor.*, vol. 27, no. 8, p. e2571, 2020.

[202] S. Lai, Y. Zhao, and Y. Yang, "Broad learning system for predicting student dropout in massive open online courses," in *Proc. 8th Int. Conf. Inf. Educ. Technol.*, 2020, pp. 12–17.

[203] A. L. G. Rios, Z. Li, K. Bekshentayeva, and L. Trajković, "Detection of denial of service attacks in communication networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seville, Spain, 2020, pp. 1–5.

[204] Z. Li, A. L. G. Rios, G. Xu, and L. Trajković, "Machine learning techniques for classifying network anomalies and intrusions," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Sapporo, Japan, 2019, pp. 1–5.

[205] Z. Li, P. Batta, and L. Trajkovic, "Comparison of machine learning algorithms for detection of network intrusions," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Miyazaki, Japan, 2018, pp. 4248–4253.

[206] D. Ma, J. Wang, Q. Sun, and X. Hu, "A novel broad learning system based leakage detection and universal localization method for pipeline networks," *IEEE Access*, vol. 7, pp. 42343–42353, 2019.

[207] R. Zhong, X. Dong, R. Lin, and H. Zou, "An incremental identification method for fraud phone calls based on broad learning system," in *Proc. IEEE 19th Int. Conf. Commun. Technol. (ICCT)*, Xi'an, China, 2019, pp. 1306–1310.

[208] W. Yuan, Y. Jiang, H. Li, and M. Cai, "A lightweight on-device detection method for Android malware," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Dec. 27, 2019, doi: 10.1109/TSMC.2019.2958382.

[209] T. Qiu, X. Liu, X. Zhou, W. Qu, Z. Ning, and C. L. P. Chen, "An adaptive social spammer detection model with semi-supervised broad learning," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 29, 2020, doi: 10.1109/TKDE.2020.3047857.

[210] X. Wei, Q. Duan, and L. Zhou, "A QoE-driven tactile Internet architecture for smart city," *IEEE Netw.*, vol. 34, no. 1, pp. 130–136, Jan./Feb. 2020.

[211] X. Wei, J. Zhao, L. Zhou, and Y. Qian, "Broad reinforcement learning for supporting fast autonomous IoT," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7010–7020, Aug. 2020.

[212] X. Wei, H. Xia, and L. Zhou, "Broad forest: A non-neural network style broad model for streaming video QoE evaluation," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Feb. 18, 2020, doi: 10.1109/TCSVT.2020.2974811.

[213] L. Xu, X. Wei, Y. Gao, and J. Mao, "IPTV user QoE prediction based on broad learning system," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Changchun, China, 2019, pp. 584–589.

[214] C.-C. Tsai, C.-F. Hsu, C.-W. Wu, and F.-C. Tai, "Cooperative localization using fuzzy ddeif and broad learning system for uncertain heterogeneous omnidirectional multi-robots," *Int. J. Fuzzy Syst.*, vol. 21, no. 8, pp. 2542–2555, 2019.

[215] C.-C. Tsai, B.-Y. Chen, and F.-C. Tai, "Sliding-mode control augmented with broad learning system for self-balancing inverse-atlas ball-riding robots with uncertainties," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Bari, Italy, 2019, pp. 941–946.

[216] C.-C. Tsai, C.-C. Yu, and C.-W. Wu, "Adaptive distributed bls-fontsm formation control for uncertain networking heterogeneous omnidirectional mobile multirobots," *J. Chin. Inst. Eng.*, vol. 43, no. 2, pp. 171–185, 2020.

[217] H. Huang, T. Zhang, C. Yang, and C. L. P. Chen, "Motor learning and generalization using broad learning adaptive neural control," *IEEE Trans. Ind. Electron.*, vol. 67, no. 10, pp. 8608–8617, Oct. 2020.

[218] C. P. Chen, D. Yu, and L. Liu, "Automatic leader-follower persistent formation control for autonomous surface vehicles," *IEEE Access*, vol. 7, pp. 12146–12155, 2018.

[219] B. Liu, X. Zeng, F. Tian, S. Zhang, and L. Zhao, "Domain transfer broad learning system for long-term drift compensation in electronic nose systems," *IEEE Access*, vol. 7, pp. 143947–143959, 2019.

[220] L. Yuan, T. Li, C. L. P. Chen, Q. Shan, and M. Han, "Broad learning system-based learning controller for course control of marine vessels," in *Proc. IEEE 10th Int. Conf. Intell Control Inf. Process. (ICICIP)*, Marrakesh, Morocco, 2019, pp. 133–136.

[221] H. Zhao, J. Zheng, J. Xu, and W. Deng, "Fault diagnosis method based on principal component analysis and broad learning system," *IEEE Access*, vol. 7, pp. 99263–99272, 2019.

[222] J. Wang and C. Zhao, "Broad learning system based visual fault diagnosis for electrical equipment thermography images," in *Proc. IEEE Chin. Autom. Congr. (CAC)*, Xi'an, China, 2018, pp. 1632–1637.

[223] S. B. Jiang, P. K. Wong, R. Guan, Y. Liang, and J. Li, "An efficient fault diagnostic method for three-phase induction motors based on incremental broad learning and non-negative matrix factorization," *IEEE Access*, vol. 7, pp. 17780–17790, 2019.

[224] Y. Wang, C. Wang, S. Kang, J. Xie, Q. Wang, and V. I. Mikulovich, "Network-combined broad learning and transfer learning: A new intelligent fault diagnosis method for rolling bearings," *Meas. Sci. Technol.*, vol. 31, no. 11, 2020, Art. no. 115013.

[225] C. Peng, R. Lu, O. Kang, and W. Kai, "Batch process fault detection for multi-stage broad learning system," *Neural Netw.*, vol. 129, pp. 298–312, Sep. 2020.

[226] S. B. Jiang, P. K. Wong, and Y. C. Liang, "A fault diagnostic method for induction motors based on feature incremental broad learning and singular value decomposition," *IEEE Access*, vol. 7, pp. 157796–157806, 2019.

[227] R. Han, R. Wang, and G. Zeng, "Fault diagnosis method of power electronic converter based on broad learning," *Complexity*, vol. 2020, Jul. 2020, Art. no. 7463291.

[228] C. Peng and L. RuiWei, "Process monitoring of batch process based on overcomplete broad learning network," *Eng. Appl. Artif. Intell.*, vol. 99, Mar. 2021, Art. no. 104139.

[229] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, San Francisco, CA, USA, 2010, pp. 94–101.

[230] L. Yang, S. Song, and C. L. P. Chen, "Transductive transfer learning based on broad learning system," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Miyazaki, Japan, 2018, pp. 912–917.

[231] S. Gao, G. Guo, H. Huang, X. Cheng, and C. L. P. Chen, "An end-to-end broad learning system for event-based object classification," *IEEE Access*, vol. 8, pp. 45974–45984, 2020.

[232] Y. Chu *et al.*, "Hyperspectral image classification based on discriminative locality preserving broad learning system," *Knowl. Based Syst.*, vol. 206, Oct. 2020, Art. no. 106319.

[233] J. C. Henry, "Electroencephalography: Basic principles, clinical applications, and related fields," in *Neurology*, vol. 67. Philadelphia, PA, USA: Elsevier, 2006, p. 2092.

[234] X.-H. Wang, T. Zhang, X.-M. Xu, L. Chen, X.-F. Xing, and C. L. P. Chen, "EEG emotion recognition using dynamical graph convolutional neural networks and broad learning system," in *Proc. IEEE Int. Conf. Bioinformat. Biomed. (BIBM)*, Madrid, Spain, 2018, pp. 1240–1244.

[235] W.-L. Zheng and B.-L. Lu, "Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks," *IEEE Trans. Auton. Mental Develop.*, vol. 7, no. 3, pp. 162–175, Sep. 2015.

[236] S. Katsigiannis and N. Ramzan, "DREAMER: A database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 1, pp. 98–107, Jan. 2018.

[237] M. Shi, Y. Fujisawa, T. Wakabayashi, and F. Kimura, "Handwritten numeral recognition using gradient and curvature of gray scale image," *Pattern Recognit.*, vol. 35, no. 10, pp. 2051–2059, 2002.

[238] J. T. Y. Kung, D. Colognori, and J. T. Lee, "Long noncoding RNAs: Past, present, and future," *Genetics*, vol. 193, no. 3, pp. 651–669, 2013.

[239] K. C. Wang and H. Y. Chang, "Molecular mechanisms of long noncoding RNAs," *Mol. Cell*, vol. 43, no. 6, pp. 904–914, 2011.

[240] M. Guttman and J. L. Rinn, "Modular regulatory principles of large non-coding RNAs," *Nature*, vol. 482, no. 7385, pp. 339–346, 2012.

[241] M. Ramanathan, D. F. Porter, and P. A. Khavari, "Methods to study RNA–protein interactions," *Nat. Meth.*, vol. 16, no. 3, pp. 225–234, 2019.

[242] S. G. Ali *et al.*, "Cost-effective broad learning-based ultrasound biomicroscopy with 3D reconstruction for ocular anterior segmentation," *Multimedia Tools Appl.*, pp. 1–18, Aug. 2020. doi: 10.1007/s11042-020-09303-9.

[243] R. Fletcher, "Conjugate gradient methods for indefinite systems," in *Numerical Analysis*. Heidelberg, Germany: Springer, 1976, pp. 73–89.

[244] D. Liu, S. Baldi, W. Yu, J. Cao, and W. Huang, "On training traffic predictors via broad learning structures: A benchmark study," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Jul. 20, 2020, doi: 10.1109/TSMC.2020.3006124.

[245] T. Tanaka and D. P. Mandic, "Complex empirical mode decomposition," *IEEE Signal Process. Lett.*, vol. 14, no. 2, pp. 101–104, Feb. 2007.

[246] J. Guo, L. Wang, K. Fan, and B. Yang, "An efficient model for predicting setting time of cement based on broad learning system," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106698.

[247] P. Cui *et al.*, "The wenchuan earthquake (May 12, 2008), sichuan province, china, and resulting geohazards," *Nat. Hazards*, vol. 56, no. 1, pp. 19–36, 2011.

[248] X. Li, Y. Zhu, Y. Zuo, T. Li, and C. L. P. Chen, "Prediction of ship fuel consumption based on broad learning system," in *Proc. Int. Conf. Security Pattern Anal. Cybern. (SPAC)*, Guangzhou, China, 2019, pp. 54–58.

[249] Y. Ma, L. Wu, Y. Guan, and Z. Peng, "The capacity estimation and cycle life prediction of lithium-ion batteries using a new broad extreme learning machine approach," *J. Power Sources*, vol. 476, Nov. 2020, Art. no. 228581.

[250] S. Nandi, H. A. Toliyat, and X. Li, "Condition monitoring and fault diagnosis of electrical motors—A review," *IEEE Trans. Energy Convers.*, vol. 20, no. 4, pp. 719–729, Dec. 2005.

[251] H. Chen and B. Jiang, "A review of fault detection and diagnosis for the traction system in high-speed trains," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 450–465, Feb. 2020.

[252] J. Feng, J. Xu, W. Liao, and Y. Liu, "Review on the traction system sensor technology of a rail transit train," *Sensors*, vol. 17, no. 6, p. 1356, 2017.

[253] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: A survey," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 2, pp. 52–73, 2nd Quart., 2009.

[254] A. F. Molisch *et al.*, "Hybrid beamforming for massive MIMO: A survey," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 134–141, Sep. 2017.

[255] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[256] L. Tan and N. Wang, "Future Internet: The Internet of Things," in *Proc. IEEE 3rd Int Conf. Adv. Comput. Theory Eng. (ICACTE)*, vol. 5. Chengdu, China, 2010, pp. 356–380.

[257] Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani, "Internet protocol television (IPTV): The killer application for the next-generation Internet," *IEEE Commun. Mag.*, vol. 45, no. 11, pp. 126–134, Nov. 2007.

[258] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Interpreting TF-IDF term weights as making relevance decisions," *ACM Trans. Inf. Syst.*, vol. 26, no. 3, pp. 1–37, 2008.

[259] Y. Long, Z. Chen, and S. Murphy, "Broad learning based hybrid beamforming for mm-wave MIMO in time-varying environments," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 358–361, Feb. 2020.

[260] F. Sohrabi and Y. Wei, "Hybrid digital and analog beamforming design for large-scale antenna arrays," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 501–513, Apr. 2016.

[261] W. Y. Lee and S. J. Lee, "Capacity of millimetre-wave multiple-input multiple-output channels in a conference room," *IET Commun.*, vol. 6, no. 17, pp. 2879–2885, Nov. 2012.

[262] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, May 1996.

[263] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[264] M. Chen, X. Wei, Y. Gao, L. Huang, M. Chen, and B. Kang, "Deep-broad learning system for traffic flow prediction toward 5G cellular wireless network," in *Proc. IEEE Int. Wireless Commun. Mobile Comput. (IWCMC)*, Limassol, Cyprus, 2020, pp. 940–945.

[265] H. Yang and J. Liu, "An adaptive RBF neural network control method for a class of nonlinear systems," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 2, pp. 457–462, Mar. 2018.

[266] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[267] T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Trans. Neural Netw.*, vol. 6, no. 4, pp. 911–917, Jul. 1995.

[268] Y.-J. Liu, J. Li, S. Tong, and C. L. P. Chen, "Neural network control-based adaptive learning design for nonlinear systems with full-state constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1562–1571, Jul. 2016.

[269] C. L. P. Chen and Y.-H. Pao, "An integration of neural network and rule-based systems for design and planning of mechanical assemblies," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 5, pp. 1359–1371, Sep./Oct. 1993.

[270] G. Debbache and N. Goléa, "Neural network based adaptive sliding mode control of uncertain nonlinear systems," *J. Syst. Eng. Electron.*, vol. 23, no. 1, pp. 119–128, Feb. 2012.

[271] C. Wang and D. J. Hill, "Deterministic learning and rapid dynamical pattern recognition," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 617–630, May 2007.

[272] H. Huang, C. Yang, and C. L. P. Chen, "Optimal robot-environment interaction under broad fuzzy neural adaptive control," *IEEE Trans. Cybern.*, early access, Jun. 22, 2020, doi: 10.1109/TCYB.2020.2998984.

[273] C.-C. Tsai, C.-C. Chan, Y.-C. Li, and F.-C. Tai, "Intelligent adaptive PID control using fuzzy broad learning system: An application to tool-grinding servo control systems," *Int. J. Fuzzy Syst.*, vol. 22, pp. 2149–2162, Aug. 2020.

[274] E. Z. Mohammed, "Hardware implementation of artificial neural network using field programmable gate array," *Int. J. Comput. Theory Eng.*, vol. 5, no. 5, pp. 780–783, 2013.

[275] S. Sahin, Y. Becerikli, and S. Yazici, "Neural network implementation in hardware using FPGAs," in *Proc. Int. Conf. Neural Inf. Process.*, 2006, pp. 1105–1112.

[276] S. Muthuramalingam, S. Himavathi, and E. Srinivasan, "Neural network implementation using FPGA: Issues and application," *Int. J. Inf. Technol.*, vol. 4, no. 2, pp. 86–92, 2008.

[277] S. Decherchi, P. Gastaldo, A. Leoncini, and R. Zunino, "Efficient digital implementation of extreme learning machines for classification," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 8, pp. 496–500, Aug. 2012.

[278] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, Jan. 2015.

[279] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[280] R. Xie and S. Wang, "Downsizing and enhancing broad learning systems by feature augmentation and residuals boosting," *Complex Intell. Syst.*, vol. 6, pp. 411–429, Apr. 2020.

[281] J. Pei *et al.*, "Towards artificial general intelligence with hybrid tianjic chip architecture," *Nature*, vol. 572, no. 7767, pp. 106–111, 2019.

[282] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2017, pp. 6231–6239.

[283] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2014, pp. 2924–2932.

[284] A. Rahimi and B. Recht, "Random features for large-scale kernel machines,"(NIPS'07) in *Advances in Neural Information Processing Syst.*, 2008. Vancouver, BC, Canada: Curran Associates Inc., pp. 1177–1184, Red Hook, NY, USA.

[285] G.-B. Huang, M.-B. Li, L. Chen, and C.-K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, nos. 4–6, pp. 576–583, 2008.

[286] J. Y. F. Yam and T. W. S. Chow, "A weight initialization method for improving training speed in feedforward neural network," *Neurocomputing*, vol. 30, nos. 1–4, pp. 219–232, 2000.

[287] M. Fernández-Redondo and C. Hernández-Espinosa, "Weight initialization methods for multilayer feedforward," in *Proc. Eur. Symp. Artif. Neural Netw. (ESANN)*, 2001, pp. 119–124.

[288] G. P. Drago and S. Ridella, "Statistically controlled activation weight initialization (SCAWI)," *IEEE Trans. Neural Netw.*, vol. 3, no. 4, pp. 627–631, Jul. 1992.

[289] W. Luo, J. Li, J. Yang, W. Xu, and J. Zhang, "Convolutional sparse autoencoders for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3289–3294, Jul. 2018.

[290] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie, "Facial expression recognition via learning deep sparse autoencoders," *Neurocomputing*, vol. 273, pp. 643–649, Jan. 2018.

[291] K. Lan, K. Yang, Z. Yu, G. Han, J. You, and C. L. P. Chen, "Adaptive weighted broad learning system for software defect prediction," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, 2020, pp. 1–7.

[292] J. S. Albus, "Outline for a theory of intelligence," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 473–509, May/Jun. 1991.

[293] A. Novellino, P. D'Angelo, L. Cozzi, M. Chiappalone, V. Sanguineti, and S. Martinoia, "Connecting neurons to a mobile robot: An in vitro bidirectional neural interface," *Comput. Intell. Neurosci.*, vol. 2007, Jul. 2007, Art. no. 012725.

[294] Y. Wang, W. Kinsner, and D. Zhang, "Contemporary cybernetics and its facets of cognitive informatics and computational intelligence," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 4, pp. 823–833, Aug. 2009.

[295] M. I. Posner and S. E. Petersen, "The attention system of the human brain," *Annu. Rev. Neurosci.*, vol. 13, no. 1, pp. 25–42, 1990.

[296] S. E. Petersen and M. I. Posner, "The attention system of the human brain: 20 years after," *Annu. Rev. Neurosci.*, vol. 35, pp. 73–89, Jul. 2012.

[297] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408. Amsterdam, The Netherlands: Elsevier, 1958.

[298] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception*, Elsevier, 1992, pp. 65–93.

[299] G. Hinton, "How to do backpropagation in a brain," in *Proc. NIPS Deep Learn. Workshop*, vol. 656, 2007, pp. 1–16.

[300] T. P. Vogels and L. Abbott, "Gating multiple signals through detailed balance of excitation and inhibition in spiking networks," *Nat. Neurosci.*, vol. 12, no. 4, pp. 483–491, 2009.

[301] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Front. Neurosci.*, vol. 10, p. 508, Nov. 2016.

[302] U. Egert, D. Heck, and A. Aertsen, "Two-dimensional monitoring of spiking networks in acute brain slices," *Exp. Brain Res.*, vol. 142, no. 2, pp. 268–274, 2002.

[303] T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner, "Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks," *Science*, vol. 334, no. 6062, pp. 1569–1573, 2011.

[304] M.-M. Poo, J.-L. Du, N. Y. Ip, Z.-Q. Xiong, B. Xu, and T. Tan, "China brain project: basic neuroscience, brain diseases, and brain-inspired computing," *Neuron*, vol. 92, no. 3, pp. 591–596, 2016.

[305] Q. Xia and J. J. Yang, "Memristive crossbar arrays for brain-inspired computing," *Nat. Mater.*, vol. 18, no. 4, pp. 309–323, 2019.

[306] C. J. Wan *et al.*, "Proton-conducting graphene oxide-coupled neuron transistors for brain-inspired cognitive systems," *Adv. Mater.*, vol. 28, no. 18, pp. 3557–3563, 2016.

[307] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, 2012.

[308] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—Past, present, and future," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1190–1203, Nov. 2012.

[309] M. Chan, E. Campo, D. Estève, and J.-Y. Fourniols, "Smart homes—Current features and future perspectives," *Maturitas*, vol. 64, no. 2, pp. 90–97, 2009.

[310] N. K. Jaiswal, *Military Operations Research: Quantitative Decision Making*, vol. 5. New York, NY, USA: Springer, 2012.

[311] D. Singh, G. Tripathi, A. M. Alberti, and A. Jara, "Semantic edge computing and IoT architecture for military health services in battlefield," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, 2017, pp. 185–190.

[312] J.-F. Bonnefon, A. Shariff, and I. Rahwan, "The social dilemma of autonomous vehicles," *Science*, vol. 352, no. 6293, pp. 1573–1576, 2016.

[313] Y. Kang *et al.*, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGARCH Comput. Archit. News*, vol. 45, no. 1, pp. 615–629, 2017.

[314] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[315] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[316] B. Luo, S. Tan, Z. Yu, and W. Shi, "EdgeBox: Live edge video analytics for near real-time event detection," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Seattle, WA, USA, 2018, pp. 347–348.

[317] X. Peng, K. Ota, and M. Dong, "Edge computing based traffic analysis system using broad learning," in *Proc. Int. Conf. Artif. Intell. Commun. Netw.*, 2019, pp. 238–251.

Xinrong Gong received the B.S. degree from the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2018. He is currently pursuing the Ph.D. degree in computer science and engineering from South China University of Technology, Guangzhou, China.

His research interests include broad learning systems, affective computing, and machine learning.

Tong Zhang (Member, IEEE) received the B.S. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2009, the M.S. degree in applied mathematics and the Ph.D. degree in software engineering from the University of Macau, Macau, China, in 2011 and 2016, respectively.

He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, also with Pazhou Lab, Guangzhou, China. His research interests include affective computing, evolutionary computation, neural networks, and other machine-learning techniques and their applications. Dr. Zhang was working in publication matters for many IEEE conferences.

C. L. Philip Chen (Fellow, IEEE) received the M.S. degree in electrical and computer science from the University of Michigan at Ann Arbor, Ann Arbor, MI, USA, in 1985, and the Ph.D. degree in electrical and computer science from Purdue University, West Lafayette, IN, USA, in 1988.

He is the Chair Professor and the Dean with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, also with Pazhou Lab, Guangzhou, China. Being a Program Evaluator of the Accreditation Board of Engineering and Technology Education in the U.S., for computer engineering, electrical engineering, and software engineering programs, he successfully architects the University of Macau's Engineering and Computer Science programs, receiving accreditations from Washington/Seoul Accord through Hong Kong Institute of Engineers (HKIE), Hong Kong, of which is considered as his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty of Science and Technology. His current research interests include cybernetics, systems, and computational intelligence.

Dr. Chen was a recipient of the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University, in 1988. He received the IEEE Norbert Wiener Award in 2018 for his contribution toward systems and cybernetics, and machine learning. He is also a highly cited researcher by Clarivate Analytics in 2018, 2019, and 2020. He is currently the Editor-in-Chief of the IEEE Transactions on Cybernetics and an Associate Editor of the IEEE Transactions on Artificial Intelligence and IEEE Transactions on Fuzzy Systems. He was the IEEE Systems, Man, and Cybernetics Society President from 2012 to 2013, and the Editor-in-Chief of the IEEE Transactions on Systems, Man, and Cybernetics: Systems from 2014 to 2019. He was the Chair of TC 9.1 Economic and Business Systems of the International Federation of Automatic Control from 2015 to 2017. He is a Fellow of AAAS, IAPR, CAA, and HKIE; and a Member of Academia Europaea, European Academy of Sciences and Arts.

Zhulin Liu (Member, IEEE) received the bachelor's degree in mathematics from Shandong University, Jinan, China, in 2009, the M.S. degree in mathematics, and the Ph.D. degree in software engineering from the University of Macau, Macau, China, in 2011 and 2019, respectively.

She is currently a Postdoctoral Fellow with the South China University of Technology, Guangzhou, China. Her research area is in broad learning systems, computational intelligence, machine learning, and function approximation.