Data Mining and Machine Learning
https://www.sciltp.com/journals/dmml

*Review*

# Broad Learning System in Data Mining and Machine Learning

Fan Yun *, Zhiwen Yu and Kaixiang Yang

The School of Computer Science and Engineering, South China University of Technology, 382 Waihuan East Road, Xiaoguwei Street, Guangzhou 510006, China
* Correspondence: 820832107@qq.com

**How To Cite**: Yun, F.; Yu, Z.; Yang, K. Broad Learning System in Data Mining and Machine Learning. *Data Mining and Machine Learning* **2025**, *1*(1), 100002.

Received: 4 September 2025
Revised: 16 November 2025
Accepted: 16 December 2025
Published: 17 December 2025

**Abstract:** The paper presents a comprehensive survey of Broad Learning System (BLS) in data mining and machine learning. BLS is characterized as a lightweight, single-layer neural network that leverages a pseudo-inverse-based weight-update mechanism. BLS supports dynamic training through incremental learning and pruning strategies. In recent years, BLS has garnered increasing attention and given rise to numerous variants. To enhance BLS performance and robustness, researchers have integrated kernel methods, manifold learning, and ensemble learning techniques into the BLS framework. Finally, the paper outlines prospective directions for future BLS research.

**Keywords:** Broad Learning System; dynamic system; incremental learning; kernel learning; manifold learning; ensemble learning; data mining; machine learning

## 1. Introduction

Deep learning is a kind of popular neural network in data mining [1]. However, due to the characteristics of the backpropagation gradient update method and the trend to deepen the number of layers, Deep Neural Networks (DNN) [2] have several inevitable problems. The most serious problem is the complexity of the network. Due to the large number of hyperparameters and complex structures, the training of DNN requires a large amount of time and powerful hardware accelerators, which suffer from great computational power consumption. Secondly, to achieve better performance, DNN tends to increase the depth of the model and the number of parameters, which is prone to problems such as gradient disappearance, gradient explosion, slow convergence speed, and falling into local optimal solutions. When training data are updated or the number of network layers is deepened, deep learning methods such as generative adversarial network (GAN) [3], transformer [4] and convolutional neural networks [5] need to retrain the whole model to adapt to new data and improve performance, which is also expensive.

Due to the above problems, Broad Learning System (BLS) [6,7] with a simple structure, fast training speed, incremental characteristics, and strong function-fitting ability, can be used as an alternative to deep learning in data mining and machine learning. BLS is a single-hidden-layer neural network, which is different from multilayer DNN. Broad Learning System does not increase the number of network layers, but expands the network in a broad direction, as its name indicates, and extracts abstract and rich features by adding hidden-layer nodes.

In DNN, the network iteratively updates the weights layer by layer by gradient descent, while BLS directly obtains the weights through the pseudo-inverse calculation [8]. Therefore, BLS avoids the above DNN problems. The pseudo-inverse weight update method also enables the incremental characteristic of BLS, so there is no need to retrain the whole BLS model like DNN when new data arrive. BLS has also been proven to have a strong function-fitting ability [9].

Nowadays, BLS and its variants are widely applied in different fields such as data mining [10–14], industry [15–18], visual recognition [19,20], data stream [21–24], domain adaptation [25,26], bioinformatics [27,28], and different types of data such as noisy data [29–31] and imbalanced data [22,32–34]. Except for classification tasks, BLS variants can be used in clustering [35–37], and semi-supervised tasks [38–40].

The rest of the paper is divided into 5 sections. Section 2 introduces the structure and various objective functions of BLS in detail. Section 3 focuses on the dynamic system of BLS, including incremental learning and pruning methods. Section 4 presents several BLS variants in data mining and machine learning, including kernel

BLS, manifold BLS, and ensemble BLS. Section 5 looks at the future development of BLS. The last section gives the conclusion of the paper.

## 2. Broad Learning System

Broad Learning System is a single hidden-layer neural network that randomly generates weights and biases for hidden nodes and uses pseudo-inverse to calculate output weights, achieving fast incremental learning and high-precision modeling. This section introduces the network structure and training process of Broad Learning System. Different objective functions are also present in this section.

### *2.1. Network Structure*

Broad Learning System is proposed based on Random Vector Functional Link Neural Network (RVFLNN) [41]. RVFLNN is a type of single hidden layer feedforward neural network, which randomly generates fixed weights and biases from input data to the hidden layer, known as the enhancement layer.

Based on RVFLNN, BLS adds feature nodes in the hidden layer. As shown in Figure 1, the hidden layer of BLS consists of two parts: feature nodes and enhancement nodes. Original data is linearly mapped to obtain feature nodes, which are then transformed into enhancement nodes through nonlinear activation functions. This enables BLS to have both linear and nonlinear mappings, enhancing the network's expression ability. Finally, the pseudo-inverse operation is used for weight updating, which requires only a single calculation and leads to an efficient training process.

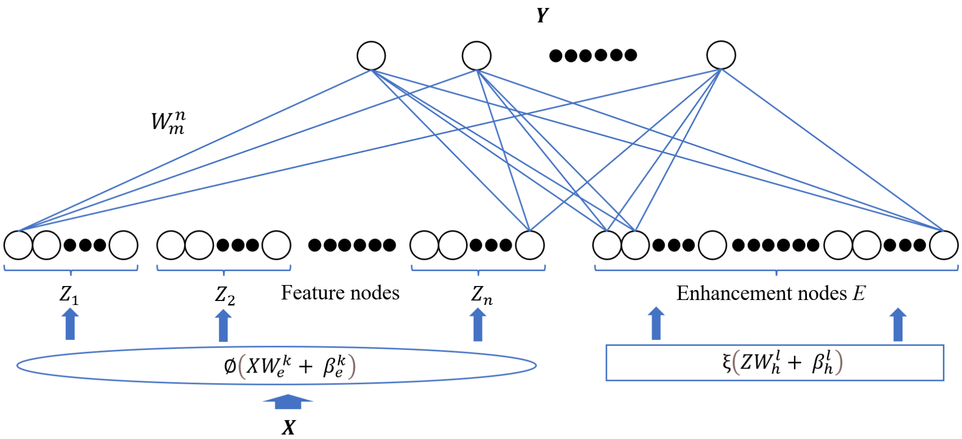


**Figure 1.** Broad Learning System structure.

#### 2.1.1. Feature Nodes Generation

BLS improves the hidden layer of RVFLNN [42] by adding feature nodes. The original data are not directly mapped into enhancement nodes, but through a linear feature transformation. Assume that there are $n$ sets of feature nodes, each with $i$ feature nodes. The calculation method for each feature mapping $Z$ is as follows:

$$Z_k = \phi\left(XW_e^k + \beta_e^k\right), k = 1, 2, ..., n \tag{1}$$

where $X$ denotes the input data, $\phi$ is the linear activation function, $W_e$ and $\beta_e$ are random-generated weight matrices and biases, which speed up the BLS training process. To overcome the randomness of the model and accelerate convergence, a sparse autoencoder (SAE) [43] is generally used for feature extraction. SAE fine-tuned $W_e$ and $\beta_e$ to reduce redundant features, making the representation of the feature layer more compact. By concatenating $n$ sets of feature nodes, we can obtain the feature node matrix $Z^N$.

$$Z^N = [Z_1|Z_2|...|Z_n] \tag{2}$$

#### 2.1.2. Enhancement Nodes Generation

The second step is to generate enhancement nodes. The calculation process for the enhancement nodes and the feature nodes is similar. The difference is that the enhancement nodes are generated based on all the feature nodes $Z$. Assume that the enhancement layer has $m$ sets of nodes, each with $j$ enhancement nodes. The calculation of enhancement nodes is:

$$E_l = \xi\left(ZW_h^l + \beta_h^l\right), l = 1, 2, ..., m \tag{3}$$

where $\xi$ is the nonlinear activation function, which brings the nonlinear expression ability to BLS. $W_h$ and $\beta_h$ are random-generated weight matrices and biases. The weight matrix $W_h$ is an orthogonally normalized random matrix, which can map the feature nodes to a subspace of higher dimension, since there may be linearly correlated terms in the randomly generated matrix. After orthogonal normalization, each term in the matrix is linearly independent, further improving the nonlinear expression ability of the enhancement nodes. By concatenating $m$ sets of enhancement nodes, the enhancement node matrix can be obtained as $E$.

$$E^M = [E_1|E_2|...|E_m] \tag{4}$$

Concatenate all feature nodes and enhancement nodes to obtain the hidden-layer matrix $H$ of BLS.

$$H = [Z^N|E^M] \tag{5}$$

2.1.3. Weight Update

For general linear regression, the objective function of BLS is to minimize the classification error:

$$f(x) = argmin_{W_m^n}(||Y - \hat{Y}||_2^2) \tag{6}$$

where $Y$ denotes the data label and $\hat{Y}$ is the output of BLS. By setting the partial derivative of Equation (6) to 0, the pseudo-inverse can be calculated to update the output weight $W_m^n$ of BLS with $n$ feature nodes and $m$ enhancement nodes:

$$W_m^n = \left(H^\top H + \lambda I\right)^{-1} H^\top Y \tag{7}$$

where $H$ is the hidden-layer node matrix. When encountering a pathological matrix, where $H^\top H$ is non-invertible, it is impossible to calculate $W_m^n$, and if $H^\top H$ tends towards 0, it will cause $W_m$ to infinity. Therefore, the objective function of BLS uses a ridge regression model and introduces an $L2$ norm penalty term, where $\lambda$ is the ridge regression coefficient. Therefore, the calculation formula for $W_m^n$ becomes this, where $I$ is the identity matrix and $(H^\top H + \lambda I)H^\top$ is the pseudo inverse of $H$, which we denote as $H^\dagger$. The addition of the penalty term for the $L2$ norm makes the estimation of the weight matrix $W_m^n$ no longer unbiased, which may cause a certain loss in the training accuracy of BLS, but also avoids the problem of overfitting and improves the generalization ability of BLS.

Finally, we can infer the output $\hat{Y}$ with the output weight $W_m^n$:

$$\hat{Y} = HW_m^n \tag{8}$$

The pseudo-code of BLS is as depicted in Algorithm 1.

### *2.2. Objective Function*

Broad learning system uses pseudo-inverse to solve the least squares problem, which is limited by noise and outliers. In order to solve these problems, some researchers proposed different objective functions and optimization methods.

Jin et al. [19] proposed Discriminative Group-sparsity constrained BLS (DG-BLS), whose objective function is (9). DG-BLS applies group sparsity constraints to class-specific features $C$ in Equation (10) to enlarge inter-class distances and ensure that the same class mapping features have the same sparse structure. Apply group sparsity constraints $L$ to the label error term in Equation (11), so that the mapping features have a sparse structure similar to the label space, better identify the label space, and improve the discriminative ability of the projection matrix.

$$f(x) = min_{W,E}(||Y + E - HW||_F^2 + \lambda_1 C + \lambda_2 L + \lambda_3 ||W||_F^2) \tag{9}$$

$$C = \sum_{i=1}^{N} ||E_i||_{2,1} \tag{10}$$

$$L=\sum_{i=1}^{N}||H_iW||_{2,1} \tag{11}$$

where $E$ represents the label error term.

**Algorithm 1** Broad Learning System

**Input**: the dataset $X$;
the number of feature nodes $n$;
the number of enhancement nodes $m$;
the regularization coefficient $\lambda$;

```
for k in 1, ..., n do
    Randomly generate the weight W_e^k and bias β_e^k of the feature nodes;
    Calculate the feature nodes Z_k from Equation (1);
end for
Concatenate all feature nodes Z^N from Equation (2);

for l in 1, ..., m do
    Randomly generate the weight W_h^l and bias β_h^l of the enhancement nodes;
    Calculate enhancement nodes E_l from Equation (3);

end for
Concate all the enhancement nodes E^M from Equation (4);
Get the hidden layer matrix H from Equation (5);
Calculate the weight between the hidden layer and the output W_m from Equation (7);
Calculate the output Ŷ from Equation (8);
```

**Output**: the output $\hat{Y}$.

In addition, some researchers used correlation entropy to suppress non-Gaussian noise. For example, Zheng et al. [44] proposed Correntropy BLS (CBLS), which dealt with outliers and noise to improve the robustness of BLS. CBLS uses the maximum entropy criterion to train the output weights and derives three incremental learning methods from the weighted regularized least-squares solution. Zhang et al. [45] proposed Information entropy to deal with non-Gaussian noise. A quantized minimum error entropy criterion is proposed to improve BLS discriminability and robustness with little computational cost. The loss function is as Equation (12):

$$L(\boldsymbol{w})=-\hat{I}_2^Q(e)+\lambda\left\|\boldsymbol{w}\right\|_2^2=-\frac{1}{N^2}\sum_{i=1}^{N}\left(\sum_{j=1}^{L}s_j\cdot G_\sigma(e_i-c_j)\right)+\lambda\left\|\boldsymbol{w}\right\|_2^2 \tag{12}$$

where $\hat{I}_2(e)$ indicates the empirical version of the quadratic information potential, $Q$ is the quantization operation, $N$ is the number of error samples, $s_j$ is $N$ quantized to the code $c_j$, and $\lambda$ denotes the regularization coefficient.

$$\begin{aligned} e_i &= y_i - a_i w \\ c_j &= y_{c_j} - a_{c_j} w \end{aligned} \tag{13}$$

where $e_i$ is the collection of error samples, $a_i$ denotes the $i$th row of hidden nodes $H$, $a_{c_j}$ and $y_{c_j}$ denotes $a$ and $y$ corresponding to the quantization code $c_j$ in Equation (13).

The weight $\boldsymbol{w}$ can be calculated as Equation (14):

$$\boldsymbol{w}=[R+2\lambda N^2\sigma^2 I]^{-1}P \tag{14}$$

$$\begin{aligned} P &= \textstyle\sum_{i=1}^{N}\left(\sum_{j=1}^{L}s_j\cdot G_\sigma(e_i-c_j)(y_i-y_{c_j})[a_i-a_{c_j}]^{\mathrm{T}}\right) \\ R &= \textstyle\sum_{i=1}^{N}\left(\sum_{j=1}^{L}s_j\cdot G_\sigma(e_i-c_j)[a_i-a_{c_j}]^{\mathrm{T}}[a_i-a_{c_j}]\right) \end{aligned} \tag{15}$$

where $P$ and $R$ in Equation (15) are related to $\boldsymbol{w}$, and $\sigma$ denotes the bandwidth of the Gaussian kernel function, $I$ is the identity matrix.

## 3. Dynamic System

The incremental learning mechanism enables Broad Learning System to have a natural dynamic structure, which can avoid network reconstruction and resource waste by adjusting the structure and parameters. Unlike static neural networks, dynamic neural networks [46] can adapt their structure and parameters based on input data during training or inference, improving the performance and efficiency of the model, enhancing interpretability and generality in data mining.

### *3.1. Incremental Learning*

When faced with the continuous increase in data, BLS can dynamically increase the number of nodes, retain old knowledge while learning new data information, thus improving model performance. Incremental learning includes four types: input data, feature nodes, enhancement nodes [47], and output categories [48]. When faced with poor performance with static data, it is possible to use only incremental node augmentation to improve the performance of BLS. However, the 4 kinds of incremental learning can be interconnected. For example, when new data arrive, the number of feature nodes and corresponding enhancement nodes can be increased, thereby enhancing BLS's ability to extract and express data features. And if the new data contains new categories, adding new output category nodes is necessary. Figure 2 shows the incremental learning of BLS.

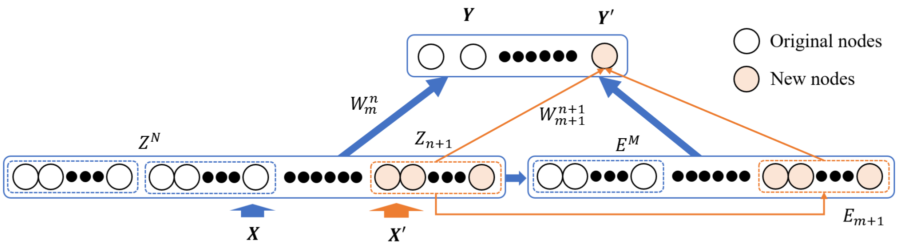


**Figure 2.** Incremental learning of Feature Nodes.

Initially, suppose that the original data $\{X, Y\}$ with the original classes $k$ are trained in the original BLS with $n$ feature nodes $Z^N$ and $m$ enhancement nodes $E^M$. When $q$ new data $\{X', Y'\}$ that belong to the $k+1$ class come. The new output can be denoted as:

$$Y' = \begin{bmatrix} Y & 0_{k\times q} \\ 0_{1\times q} & 1_{1\times q} \end{bmatrix} \tag{16}$$

where 1 is the unit matrix, and 0 is the 0 matrix. New $n+1$ feature nodes that generated from $\{X', Y'\}$ are represented as:

$$Z_{n+1} = \phi(X' W_e^{n+1} + \beta_e^{n+1}) \tag{17}$$

Since feature nodes are then mapped to enhancement nodes, the corresponding enhancement nodes will also increase during the process of incremental learning of feature nodes. The new enhancement nodes $E_{m+1}$ can be denoted as:

$$E_{m+1} = \xi\left(Z_{n+1} W_h^{m+1} + \beta_h^{m+1}\right) \tag{18}$$

BLS employs random generated $W_h^{m+1}$ and $\beta_h^{m+1}$ to calculate the new enhancement nodes. The new hidden layer can be represented as $H_{m+1}^{n+1} = [H_m^n | Z_{n+1} | E_{m+1}]$. On this basis, the pseudo-inverse matrix can be calculated as follows:

$$H_{m+1}^{n+1} = [(H_m^n)^{\dagger} - AC^{\top} | A] \tag{19}$$

$$A = \begin{cases} (B)^{\dagger} & \text{if } B \neq 0 \\ (H_m^n)^{\dagger} C \left(1 + C^{\top} C\right)^{-1} & \text{if } B = 0 \end{cases} \tag{20}$$

$$\begin{aligned} B &= [Z_{n+1} | E_{m+1}] - C^{\top} H_m^n \\ C^{\top} &= [Z_{n+1} | E_{m+1}] (H_{m+1}^{n+1})^{\dagger} \end{aligned} \tag{21}$$

where $B^\dagger$ in Equation (20) is the pseudo-inverse of $B$ in Equation (21).

Similarly, we can use the following formula to calculate the new weight of incremental feature nodes:

$$W_{m+1}^{n+1} = \begin{bmatrix} W_m^n - CA^\top Y \\ 1_{1\times q}A^\top Y \end{bmatrix} \tag{22}$$

From the above formulas, we can see that the incremental algorithms do not recalculate the whole model, but calculate a new pseudo-inverse of the hidden layer to reconstruct BLS.

The incremental BLS is suitable for stream learning. Stream learning is a learning paradigm that updates models online with infinite and real-time stream data. The goal of streaming learning is to continuously maintain the model performance under concept drift, data imbalance, and resource constraints. Concept Drift refers to the phenomenon in which the statistical relationship between target variables or features and labels undergoes systematic changes over time, resulting in a decline in the performance of the trained model. Researchers utilized the incremental nature of BLS to address the issue of online updates in streaming learning.

To address the issues of data distribution changes, low training efficiency, and model instability caused by concept drift, Bakhshi et al. [49] designed a novel ensemble method based on BLS. BELS uses mini blocks for incremental updates and employs the output layer ensemble to address the limitations of BLS and handle drift. Experiments demonstrated the adaptability of BELS to various types of drift. Zhong et al. [50] proposed the Robust Incremental BLS (RI-BLS) to address the issues of low accuracy and long training time in handling unstable data streams in BLS incremental learning. In incremental learning, RI-BLS updates two memory matrices storing information and effectively updates weights through pre-calculated ridge regression. RI-BLS improves the stability of the original BLS time complexity and reduces space complexity. The proposed weight update strategy is also applicable to other random neural networks.

Some researchers designed novel update rules for incremental BLS. Lin et al. [22] designed a new incremental update method using Gaussian kernel mapping to generate feature nodes to improve the robustness and performance of BLS. The experiments on drift datasets validated the advanced robustness and accuracy of eBLS-W. Dealing with data streams with concept drift, Lei et al. [51] proposed online BLS to address the issues of reduced accuracy and expensive update costs of the existing incremental BLS in online learning tasks. The weight estimation algorithm was designed to effectively improve performance by using Cholesky decomposition and forward-backward substitution. In addition, the proposed efficient online update strategy effectively speeds up the online update time. To address the problem of concept drift caused by auxiliary training data under semi-supervised tasks, Guo et al. [38] proposed a robust semi-supervised BLS guided by ensemble self-training, which recursively updates its data, structure, and parameters using a data-driven dynamic node mechanism guided by label purity and dynamic network structure adjustment. Guo et al. [39] also proposed semi-supervised incremental self-training BLS, and designed a dynamic neuron incremental mechanism based on the precision of the labeled data to effectively guide self-updating of network structures.

### *3.2. Pruning Method*

Broad learning system tends to grow wider to improve its generalization ability. However, too many hidden nodes will cause redundancy and longer processing time. In order to solve node redundancy, some researchers used pruning strategies to avoid redundant nodes. The hidden nodes with small contributions to BLS are considered to have little correlation with the result. Thus, deleting unnecessary nodes is beneficial for BLS simplification.

Some researchers designed different evaluation indicators to select important nodes in BLS. Shi et al. [52] selected hidden-layer neurons based on their importance in classification to reduce errors propagating from lower-layer neurons to upper-layer neurons. Liu et al. [53] proposed an adaptive evolutionary pruning BLS (ADP-BLS), in which enhanced nodes are kept based on sensitivity. Chu et al. [54] simplified BLS with complex structures through the contribution of nodes. Liu et al. [55] proposed an Echo State Network (BPESN), which removes redundant nodes with low contributions. The permutation is performed on the data row corresponding to the deleted node in the last row, and a permutation matrix is formed. The permutation matrix is triangulated by block through a unified transformation.

Other researchers proposed regularization methods to sparsify the BLS network. Li et al. [56] proposed a dynamic sparse training BLS, which introduces a regularization term to constrain the output weight threshold in the objective function. An output weight mask is generated based on the importance of the output weights. Through dynamic training, the performance of the model is improved while maintaining a sparse network structure. Zhang et al. [57] proposed Elastic-net Robust BLS (GER-BLS) to address the issues of outliers and dense networks. The regularization method improves robustness by treating all output residuals as a whole, and induces node-level sparsity by penalizing output weights connected to a single node, promoting network sparsity. They also designed DGER-BLS to process distributed data.

## 4. BLS Variants

This section introduces three types of popular variants of Broad Learning System in data mining and machine learning, including kernel-based BLS, manifold BLS, and ensemble BLS.

### *4.1. Kernel-Based BLS*

Kernel methods [58] are a class of machine learning algorithms that operate by mapping data into a higher-dimensional feature space, where linearly inseparable patterns become linearly separable. Typically, kernel regression [59] is a classical non-parametric regression method that uses the kernel function to locally weight average samples to fit non-linear relationships and estimate the target value, which is suitable for complex and non-linear distributed data. Therefore, instead of random projection, some researchers used kernel functions to generate hidden nodes to improve the nonlinear expression ability of BLS.

Some researchers used the single kernel function in BLS. Chen et al. [60] applied kernel mapping instead of random weight generalization in feature nodes. After mapping the data to Gaussian kernel space, the uncertainty caused by noise can be reduced, and the generalization ability of BLS can be improved.

To remove redundancy caused by random projection in BLS, Chen et al. further proposed DKWBLS [61] and DKCSBLS [62] with Gaussian kernel nodes for binary and multiclass classification tasks. Moreover, the kernel mapping fixes the number of feature nodes and enhancement nodes, so that the hyperparameter of BLS is reduced. In kernel-based BLS, authors use the auto-encoder to calculate the weights between data and feature nodes, feature nodes, and enhancement nodes. Yu et al. designed KBLS [29] to adopt the Gaussian function in enhancement mapping to improve the robustness of BLS. KBLS is used as the base learner in a progressive boosting framework. PEKB solves the loss of base learners using gradient and subgradient, and obtains the residual as the training label for the following base learners.

However, the structure of the feature space mapped by the single kernel method is entirely determined by the design of the kernel function, and searching for the proper kernel parameters is time-consuming. To solve these problems, researchers designed multi-kernel methods [63] based on BLS. Liu et al. [64] generated feature nodes with the multiscale kernel method. MKBLS used the Fourier kernel approximation method to combine the Gaussian kernel with different parameters. Yun et al. [27] designed multi-kernel fusion at the feature-level and the decision-level with ensemble BLS for disease diagnosis.

### *4.2. Manifold BLS*

Manifold Learning (ML) [65], also known as Nonlinear Dimensionality Reduction, is a set of methods to find low-dimensional structures in data. There are many high-dimensional datasets in the real world, and high-dimensional data are usually located on a low-dimensional manifold. Manifold learning preserves the data's inherent geometric structure by mapping it to a low-dimensional space, which helps BLS handle high-dimensional data.

Researchers have applied manifold learning to BLS, including modifying node mapping methods, optimizing weights, and designing new objective functions, and have widely applied it to various tasks such as classification, time series prediction, semi-supervised learning, clustering, and domain adaptation.

For image classification tasks, Jin et al. [66] incorporated manifold learning into the objective function of BLS, known as Graph Regularization BLS (GBLS), which assumes that similar images may have similar properties.

$$\arg\min_{W} L_{GBLS} = \|Y - HW\|_2^2 + \lambda_1 E_G + \lambda_2 \|W_m\|_2^2 \tag{23}$$

where $E_G$ is a graph regularization term used to reflect the local discriminative structure of the data. $\lambda_1$ and $\lambda_2$ are trade-off parameters that control the importance of the graph regularization term and the model complexity regularization term, respectively. The graph regularization term $E_G$ is constructed within the graph embedding framework, treating the training data as a set of vertices in an undirected graph and representing the similarity between vertices through a weight matrix. Graph regularization helps preserve the geometric structure and discriminative information of the data. GBLS can learn more discriminative output weights, thereby improving classification ability. In addition, the objective function of GBLS has a closed-form solution, so its optimization process is as efficient as standard BLS.

In addition, Wang et al. [26] added manifold regularization to the objective function of BLS to maintain the manifold structure of the Hyperspectral Image (HSI). Domain Adaptation Broad Learning added domain adaptation and manifold regularization to the output layer to reduce distribution differences and maintain the manifold structure. Chu et al. [67] introduced discriminative information and the local manifold structure of samples to enhance the discriminative ability of BLS, therefore improving its performance in hyperspectral image classification tasks. Lei et al. [68] proposed adaptive manifold convolutional BLS for emotion recognition. A label manifold information exploration module is designed to maintain the consistency of data labels.

Some researchers have designed manifold embedding strategies in BLS to express the intrinsic structure of chaotic time series, thereby improving the prediction performance of large-scale time series. Han et al. [69] proposed Structured Manifold-BLS, which introduces structured manifold learning for non-uniform embedding and unsupervised manifold learning, improving the dynamic discovery and feature extraction capabilities of SM-BLS for large-scale chaotic time series prediction tasks. Feng et al. [70] proposed robust manifold BLS for large-scale noisy chaotic time series prediction. RM-BLS uses the random perturbation approximation of manifold feature selection to remove features unrelated to low-dimensional manifold embedding, which makes the model robust and well-performed to noise and outliers in dynamic systems.

Traditional BLS is used to classify labeled data, but in reality there is massive unlabeled data. Therefore, researchers have added manifold regularization terms to BLS and proposed a series of semi-supervised and clustering BLS algorithms. Zhao et al. [71] improved the classification performance of unlabeled data by constructing Laplacian matrices and performing label propagation based on manifold assumptions. Liu et al. [72] designed an Incremental Semi-supervised BLS, which applies manifold regularization to explore the data distribution. ISBL effectively utilizes unlabeled data and improves the semi-supervised performance. Zhang et al. [73] integrated feature and sample manifold regularization into the fuzzy BLS, enhancing its ability to utilize potential structural relationships in data.

$$\min_W L = \|Y - D\|_2^2 + \lambda_1 Tr((D)^T L_s(D)) + \lambda_2 Tr((D)^T L_f(D)) + \alpha \|W\|_2^2 \tag{24}$$

where $L_s$ stands for the sample Laplacian matrix and $L_f$ is the feature Laplacian matrix, $D = HW$. DMR-FBLS improved the prediction accuracy of admission rates to the ICU.

For clustering tasks, Kong et al. [74] designed an unsupervised BLS to deal with unlabeled HSI. UBL uses the graph regularization sparse autoencoder to fine-tune hidden-layer node weights to maintain the intrinsic manifold structure of data, and adds the norm of output weights and graph regularization term to the loss function. Shi et al. [37] designed an ensemble clustering based on the manifold BLS. MBLS is trained to learn better consensus partitions, comprehensively and efficiently combining information from basic partitions and feature space to improve clustering performance.

$$\min_W L_{MBLS} = \|Y - HW\|_2^2 + \lambda_1 Tr((HW)^T L(HW)) + \lambda_2 \|W_m\|_2^2 \tag{25}$$

### *4.3. Ensemble BLS*

Most of the BLS variants are single machine learning learners. Generally speaking, the aggregated predictions of multiple models can often outperform a single model, especially when the base learners have high accuracy and diversity. Ensemble learning [75] is a machine learning technique that involves combining multiple individual models, known as base learners, to create a stronger and more accurate predictive model. Yun et al. [10] analyzed the efficiency and advantage of using BLS as base learners in ensemble learning and proposed a selective ensemble learning method based on incremental BLS. As the base learner, BLS has been applied in ensemble strategies such as Bagging, Boosting, and Stacking.

Stacking is an ensemble learning technique that combines predictions from multiple base models by training a meta-model to make the final prediction. Stacking allows the strengths of different base models to be captured, potentially improving overall predictive performance. In addition, to increase the diversity of base learners, resampling techniques such as Boostrap are often used in ensemble learning methods. Yan et al. [76] used BLS as the base learner in Bagging and Stacking to improve accuracy and stability. Wu et al. [77] created a subset of base BLS with Bootstrap, and integrated the results of primary learners by Stacking. Fan et al. [78] used base BLS to obtain diverse features in the lncRNA-protein pair, and to output the relationship between protein features with the stacking method.

Boosting is an ensemble learning technique that combines multiple base models to create a strong predictive model. Unlike other ensemble methods that focus on combining models in parallel, boosting emphasizes sequentially improving the performance of the base models by paying more attention to misclassified instances. Boosting algorithms iteratively train base models, with each subsequent model focusing on the samples that were misclassified by the previous models. In response to the challenge of outliers and noise in imbalanced samples, Yang et al. [47] presented a weighted BLS to assign and augment the weight of minority samples. To obtain the primary distribution of samples, AWBLS was proposed to incorporate a density-based weight generation mechanism, considering both inter-class and intra-class distances. Finally, IWEB is utilized to enhance the stability and robustness of AWBLS using a progressive Boosting method.

Some researchers applied the ensemble BLS in anomaly detection tasks. For example, Zhao et al. [79] proposed an AdaBoost Weighted BLS for transformer fault diagnosis, which uses a cost-sensitive matrix to deal with the imbalanced data. Lin et al. [80] proposed SMC-OCBLS with a boosting framework for imbalanced network traffic data. SMC-OCBLS constructs different feature spaces by randomly selected features to train multiple OCBLS, which improves the performance and generalization ability of BLS.

Besides using BLS as the base learner, some research used ensemble learning methods such as Bootstrap and Stacking in BLS. Gao et al. [81] used the Bootstrap and decision tree methods in generating feature nodes. Xie et al. [82] used the Boostrap strategy to generate enhancement nodes, which lightens BLS structure while improving its performance. By integrating the residuals, D&BLS stacks several lightweight BLS subsystems to ensure stronger feature representation capabilities and better classification performance. In [83], Xie et al. used a simple linear model to transmit shared feature nodes between adjacent BLS subsystems. Then, RST&BLS integrates these subsystems in a double stacking manner, which can make BLS subsystems more diverse through residual. Liu et al. [84] stacked several BLS blocks to form a multilayer BLS. Stacked BLS built the higher BLS block by fixing the lower block, and used the residual as its label.

## 5. Future Development

BLS has a simple structure, a fast training speed, an incremental characteristic, and a strong function-fitting ability. In addition to its advantages, BLS also has some limitations. As a single hidden-layer neural network with random generation weight mechanism, BLS has limited feature extraction ability and unstable performance. The feature extraction ability of BLS is not as good as DNN with deep structures, making it difficult to extract complex data features, especially image data. Therefore, how to combine the advantages of feature extraction in DNN with efficient training methods in BLS becomes a topic worthy of further research.

The incremental advantages of BLS have been widely utilized in stream data. However, computational efficiency and error accumulation problems exist in BLS incremental learning, making the model unstable. Novel BLS incremental learning algorithms can be studied in the future. Many people think that the wider the BLS, the better the performance. However, when the amount of data and hidden nodes is too large, the pseudo-inverse calculation may have problems such as excessive computational overhead, time-consuming, and insufficient memory, which makes it difficult to train. In the future, parallel BLS algorithms can be designed to deal with big data.

In addition, combining the idea of BLS and semi-supervised learning, with manifold learning or other methods, BLS can learn from both tagged and unlabeled data, which is of great significance for practical applications. In general, more novel BLS network architectures can be further researched in order to optimize the performance of BLS on different types of data and in different fields in the future.

## 6. Conclusions

The paper offers a systematic exposition of the Broad Learning System in the contexts of data mining and machine learning. We first elaborate on its architectural design and training paradigm, emphasizing the closed-form pseudo-inverse solution that underpins dynamic incremental learning. Subsequently, we survey BLS variants, including kernelized, manifold-regularized, and ensemble BLS, highlighting their respective theoretical contributions and empirical advances. The exposition ends with a forward-looking discussion of prospective research for BLS.

## Author Contributions

F.Y.: conceptualization, methodology, data curation, writing—original draft preparation; Z.Y.: supervision, writing—reviewing and editing; K.Y.: data curation, investigation. All authors have read and agreed to the published version of the manuscript.

## Funding

This work was supported in part by the National Natural Science Foundation of China No.62572199, 92467109, U21A20478, National Key R&D Program of China 2023YFA1011601 and the Major Key Project of PCL, China under Grant PCL2025AS11.

## Data Availability Statement

Not applicable.

## Conflicts of Interest

The authors declare no conflict of interest.

## Use of AI and AI-Assisted Technologies

No AI tools were utilized for this paper.

## References


1. Jiawei, H.; Micheline, K. Data Mining: Concepts and Techniques. *Data Min. Concepts Model. Methods Algorithms Second Ed.* **2006**, *5*, 1–18.
2. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
3. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; et al. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*; Curran: Red Hook, NY, USA, 2014; pp. 2672–2680.
4. Douillard, A.; Ramé, A.; Couairon, G.; et al. DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 9275–9285.
5. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Curran: Red Hook, NY, USA, 2012; pp. 1097–1105.
6. Chen, C.P.; Liu, Z. Broad learning system: A new learning paradigm and system without going deep. In Proceedings of the 2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Hefei, China, 19–21 May 2017; pp. 1271–1276.
7. Chen, C.P.; Liu, Z. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 10–24.
8. Golub, G.; Kahan, W. Calculating the Singular Values and Pseudo-Inverse of a Matrix. *J. Soc. Ind. Appl. Math.* **1965**, *2*, 205–224.
9. Chen, C.L.P.; Liu, Z.; Feng, S. Universal Approximation Capability of Broad Learning System and Its Structural Variations. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 1191–1204.
10. Yun, F.; Yu, Z.; Yang, K.; et al. AdaBoost-Stacking Based on Incremental Broad Learning System. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 7585–7599.
11. Hu, X.; Chen, C.L.P.; Zhang, T. Broad Metric Learning: A Fast and Efficient Discriminative Metric Learning Model. *IEEE Trans. Cybern.* **2025**, *11*, 4675–4688.
12. Yang, K.; Liu, Y.; Yu, Z.; et al. Extracting and Composing Robust Features With Broad Learning System. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 3885–3896.
13. Yang, K.; Wang, H.; Yang, Q.; et al. An Ensemble Broad Learning Scheme for Short-Term Load Forecasting. In Proceedings of the 2022 China Automation Congress (CAC), Xiamen, China, 25–27 November 2022; pp. 4901–4905.
14. Zhang, W.; Liu, Z.; Jiang, Y.; et al. Self-balancing Incremental Broad Learning System with privacy protection. *Neural Netw.* **2024**, *178*, 106436.
15. Yang, K.; Shi, Y.; Yu, Z.; et al. Stacked One-Class Broad Learning System for Intrusion Detection in Industry 4.0. *IEEE Trans. Ind. Inform.* **2023**, *19*, 251–260.
16. Zhu, C.; Pu, Y.; Lyu, Z.; et al. Multidimensional information fusion and broad learning system-based condition recognition for energy pipeline safety. *Knowl.-Based Syst.* **2024**, *300*, 112259.
17. Zhong, Z.; Yang, K.; Yu, Z.; et al. Towards Efficient Anomaly Detection Using Memory Broad Learning System. In Proceedings of the 2023 9th International Conference on Control Science and Systems Engineering (ICCSSE), Shenzhen, China, 16–18 June 2023; pp. 252–257.
18. Yang, K.; Chen, W.; Bi, J.; et al. Multi-view broad learning system for electricity theft detection. *Appl. Energy* **2023**, *352*, 121914.
19. Jin, J.; Li, Y.; Yang, T.; et al. Discriminative group-sparsity constrained broad learning system for visual recognition. *Inf. Sci.* **2021**, *576*, 800–818.
20. Lei, C.; Guo, J.; Chen, C.L.P. ConvBLS: An Effective and Efficient Incremental Convolutional Broad Learning System Combining Deep and Broad Representations. *IEEE Trans. Artif. Intell.* **2024**, *5*, 5075–5089.
21. Zhong, Z.; Yu, Z.; Fan, Z.; et al. Adaptive Memory Broad Learning System for Unsupervised Time Series Anomaly Detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 8331–8345.
22. Lin, M.; Yu, Z.; Yang, K.; et al. Dynamic Chunk-based Active Learning Based on Enhanced Broad Learning System for Imbalanced Drifting Data Streams. *IEEE Trans. Knowl. Data Eng.* **2025**, 1–13.

23. Lin, Y.; Yu, Z.; Yang, K.; et al. Ensemble Denoising Autoencoders Based on Broad Learning System for Time-Series Anomaly Detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 13913–13926.
24. Li, P.; Zhong, Z.; Zhang, T.; et al. A New Perspective on Time Series Anomaly Detection: Faster Patch-based Broad Learning System. *arXiv* **2024**, arXiv:2412.05498.
25. Zhang, L.; Yu, Z.; Yang, K.; et al. Transferable and discriminative broad network for unsupervised domain adaptation. *Knowl.-Based Syst.* **2025**, *315*, 113297.
26. Wang, H.; Wang, X.; Chen, C.L.P.; et al. Hyperspectral Image Classification Based On Domain Adaptation Broad Learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2020**, *13*, 3006–3018.
27. Yun, F.; Yu, Z.; Yang, K.; et al. Ensemble Multi-Kernel Broad Learning System for Disease Diagnosis. In Proceedings of the 2024 IEEE International Conference on Medical Artificial Intelligence (MedAI), Shanghai, China, 8–10 October 2024; pp. 97–104.
28. Tang, L.; Chen, W.; Yu, Z.; et al. Adaptive Fuzzy Weighted Broad Network for Imbalanced Medical Diagnosis. In Proceedings of the 2024 IEEE International Conference on Medical Artificial Intelligence (MedAI), Shanghai, China, 8–10 October 2024; pp. 145–152.
29. Yu, Z.; Lan, K.; Liu, Z.; et al. Progressive Ensemble Kernel-Based Broad Learning System for Noisy Data Classification. *IEEE Trans. Cybern.* **2022**, *52*, 9656–9669.
30. Zhang, Y.; Ke, S.; Li, J.; et al. DHR-BLS: A Huber-type robust broad learning system with its distributed version. *Knowl.-Based Syst.* **2025**, *314*, 113184.
31. Chen, W.; Yang, K.; Yu, Z.; et al. Adaptive Broad Network With Graph-Fuzzy Embedding for Imbalanced Noise Data. *IEEE Trans. Fuzzy Syst.* **2025**, *33*, 1949–1962.
32. Lin, Y.; Yu, Z.; Yang, K.; et al. Boosting Adaptive Weighted Broad Learning System for Multi-Label Learning. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 2204–2219.
33. Yang, K.; Chen, W.; Shi, Y.; et al. Simplified Kernel-Based Cost-Sensitive Broad Learning System for Imbalanced Fault Diagnosis. *IEEE Trans. Artif. Intell.* **2024**, *5*, 6629–6644.
34. Chen, W.; Yu, Z.; Yang, K.; et al. Minimum variance weighted broad cascade network structure for imbalanced classification. *Knowl.-Based Syst.* **2025**, *324*, 113803.
35. Yu, Z.; Zhong, Z.; Yang, K.; et al. Broad Learning Autoencoder With Graph Structure for Data Clustering. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 49–61.
36. Shi, Y.; Yang, K.; Yu, Z.; et al. Adaptive Ensemble Clustering With Boosting BLS-Based Autoencoder. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 12369–12383.
37. Shi, Y.; Chen, D.; Chen, L.; et al. Ensemble Clustering Based on Manifold Broad Learning System. In Proceedings of the 2022 China Automation Congress (CAC), Xiamen, China, 25–27 November 2022; pp. 2227–2232.
38. Guo, J.; Chen, C.L.P. A Robust Semi-Supervised Broad Learning System Guided by Ensemble-Based Self-Training. *IEEE Trans. Cybern.* **2024**, *54*, 6410–6422.
39. Guo, J.; Liu, Z.; Chen, C.L.P. An Incremental-Self-Training-Guided Semi-Supervised Broad Learning System. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *36*, 7196–7210.
40. Guo, L.; Li, R.; Jiang, B. An ensemble broad learning scheme for semisupervised vehicle type classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5287–5297.
41. Pao, Y.H.; Takefuji, Y. Functional-link net computing: theory, system architecture, and functionalities. *Computer* **1992**, *25*, 76–79.
42. Ren, Y.; Suganthan, P.N.; Srikanth, N.; et al. Random vector functional link network for short-term electricity load demand forecasting. *Inf. Sci. Int. J.* **2016**, *367/368*, 1078–1093.
43. Ng, A. *Sparse Autoencoder*; CS294A Lecture Notes; Stanford University: Stanford, CA, USA, 2011.
44. Zheng, Y.; Chen, B.; Wang, S.; et al. Broad learning system based on maximum correntropy criterion. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 3083–3097.
45. Zhang, S.; Liu, Z.; Chen, C.P. Broad learning system based on the quantized minimum error entropy criterion. *Sci. China Inf. Sci.* **2022**, *65*, 222203.
46. Guo, J.; Chen, C.L.P.; Liu, Z.; et al. Dynamic Neural Network Structure: A Review for Its Theories and Applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *36*, 4246–4266.
47. Yang, K.; Yu, Z.; Chen, C.P.; et al. Incremental weighted ensemble broad learning system for imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2021**, *34*, 5809–5824.
48. Liu, W.; Yang, H.; Sun, Y.; et al. A Broad Neural Network Structure for Class Incremental Learning. in *International Symposium on Neural Networks*; Springer International Publishing: Cham, Switzerland, 2018.
49. Bakhshi, S.; Ghahramanian, P.; Bonab, H.; et al. A Broad Ensemble Learning System for Drifting Stream Classification. *IEEE Access* **2023**, *11*, 89315–89330.
50. Zhong, L.; Philip Chen, C.L.; Guo, J.; et al. Robust Incremental Broad Learning System for Data Streams of Uncertain Scale. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 7580–7593.

51. Lei, C.; Chen, G.Z.; Chen, C.L.P.; et al. Online-BLS: An Accurate and Efficient Online Broad Learning System for Data Stream Classification. *arXiv* **2025**, arXiv:2501.16932.
52. Shi, Q.; Suganthan, P.N.; Katuwal, R. Weighting and Pruning based Ensemble Deep Random Vector Functional Link Network for Tabular Data Classification. *Pattern Recognit.* **2022**, *132*, 108879.
53. Liu, Y.; Yang, K.; Yu, Z.; et al. Pruning Broad Learning System based on Adaptive Feature Evolution. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–9.
54. Chu, F.; Su, J.; Liang, T.; et al. Least Angle Regression Adaptive Incremental Broad Learning System. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 7153–7157.
55. Liu, W.; Bai, Y.; Jin, X.; et al. Broad Echo State Network with Reservoir Pruning for Nonstationary Time Series Prediction. *Comput. Intell. Neurosci.* **2022**, *2022*, 3672905.
56. Haigang, L.I.; Sun, J.; Cao, Y.; et al. Research on broad learning systems based on dynamic sparse training. *Exp. Technol. Manag.* **2024**, *41*, 53–60.
57. Zhang, Y.; Dai, Y.; Ke, S.; et al. Generalized sparse and outlier-robust broad learning systems for multi-dimensional output problems. *Inf. Sci.* **2024**, *677*, 25.
58. Elisseeff, A.E.; Weston, J. A kernel method for multi-labelled classification. *Neural Inf. Process. Syst.* **2001**, *14*.
59. Takeda, H.; Farsiu, S.; Milanfar, P. Kernel Regression for Image Processing and Reconstruction. *IEEE Trans. Image Process.* **2007**, *16*, 349–366.
60. Chen, W.; Yang, K.; Shi, Y.; et al. Kernel-based Class-specific Broad Learning System for software defect prediction. In Proceedings of the 2021 8th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), Beijing, China, 10–12 December 2021; pp. 109–114.
61. Chen, W.; Yang, K.; Zhang, W.; et al. Double-kernelized weighted broad learning system for imbalanced data. *Neural Comput. Appl.* **2022**, *34*, 19923–19936.
62. Chen, W.; Yang, K.; Yu, Z.; et al. Double-kernel based class-specific broad learning system for multiclass imbalance learning. *Knowl.-Based Syst.* **2022**, *253*, 109535.
63. Lanckriet, G.R.G.; Cristianini, N.; Bartlett, P.L.; et al. Learning the Kernel Matrix with Semidefinite Programming. *J. Mach. Learn. Res.* **2002**, *5*, 27–72.
64. Liu, Z.; Chen, C.P.; Zhang, T.; et al. Multi-kernel broad learning systems based on random features: A novel expansion for nonlinear feature nodes. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; pp. 193–197.
65. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828.
66. Jin, J.; Geng, B.; Li, Y.; et al. Flexible Label-Induced Manifold Broad Learning System for Multiclass Recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 16076–16090.
67. Chu, Y.; Lin, H.; Yang, L.; et al. Hyperspectral image classification based on discriminative locality preserving broad learning system. *Knowl.-Based Syst.* **2020**, *206*, 106319.
68. Lei, C.; Chen, C.L.P.; Zhang, T. AM-ConvBLS: Adaptive Manifold Convolutional Broad Learning System for Cross-Session and Cross-Subject Emotion Recognition. *IEEE Trans. Affect. Comput.* **2025**, *16*, 2788–2802.
69. Min, H.; Shoubo, F.; Philip, C.C.L.; et al. Structured Manifold Broad Learning System: A Manifold Perspective for Large-Scale Chaotic Time Series Analysis and Prediction. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 1809–1821.
70. Feng, S.; Ren, W.; Han, M.; et al. Robust manifold broad learning system for large-scale noisy chaotic time series prediction: A perturbation perspective. *Neural Netw.* **2019**, *117*, 179–190.
71. Zhao, H.; Zheng, J.; Deng, W.; et al. Semi-Supervised Broad Learning System Based on Manifold Regularization and Broad Network. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 983–994.
72. Liu, X.; Qiu, T.; Chen, C.; et al. An Incremental Broad Learning Approach for Semi-Supervised Classification. In Proceedings of the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Fukuoka, Japan, 5–8 August 2019.
73. Zhang, X.; Nebot, A. Semi-Supervised Dual-Manifold Regularized Fuzzy Broad Learning for ICU Admission Prediction in Post-COVID Transplant Recipients. In Proceedings of the 2024 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Yokohama, Japan, 30 June 2024–5 July 2024
74. Kong, Y.; Cheng, Y.; Chen, C.L.P.; et al. Hyperspectral Image Clustering Based on Unsupervised Broad Learning. *IEEE Geosci. Remote. Sens. Lett.* **2019**, *16*, 1741–1745.
75. Dong, X.; Zhiwen, Y.U.; Cao, W.; et al. A survey on ensemble learning. *Front. Comput. Sci.* **2019**, *14*, 241–258.
76. Yicheng, Y.; Wei, G.; Liwen, W. Broad learning system based on ensemble learning. In Proceedings of the 2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), Xi'an, China, 28–30 May 2021; pp. 62–67.
77. Wu, C.; Qiu, T.; Zhang, C.; et al. Ensemble strategy utilizing a broad learning system for indoor fingerprint localization. *IEEE Internet Things J.* **2021**, *9*, 3011–3022.

78. Fan, X.N.; Zhang, S.W. LPI-BLS: Predicting lncRNA–protein interactions with a broad learning system-based stacked ensemble classifier. *Neurocomputing* **2019**, *370*, 88–93.
79. Zhao, B.; Yang, D.; Karimi, H.R.; et al. Filter-wrapper combined feature selection and adaboost-weighted broad learning system for transformer fault diagnosis under imbalanced samples. *Neurocomputing* **2023**, *560*, 1.1–1.12.
80. Lin, M.; Yang, K.; Yu, Z.; et al. Hybrid Ensemble Broad Learning System for Network Intrusion Detection. *IEEE Trans. Ind. Inform.* **2024**, *20*, 5622–5633.
81. Gao, Y.; Miao, H.; Chen, J.; et al. Explosive Cyber Security Threats During COVID-19 Pandemic and a Novel Tree-Based Broad Learning System to Overcome. *IEEE Trans. Intell. Transp. Syst.* **2022**, *25*, 786–795.
82. Xie, R.; Wang, S. Downsizing and enhancing broad learning systems by feature augmentation and residuals boosting. *Complex Intell. Syst.* **2020**, *6*, 411–429.
83. Xie, R.; Vong, C.M.; Chen, C.P.; et al. Dynamic Network Structure: Doubly Stacking Broad Learning Systems with Residuals and Simpler Linear Model Transmission. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, *6*, 1378–1395.
84. Liu, Z.; Chen, C.P.; Feng, S.; et al. Stacked broad learning system: From incremental flatted structure to deep model. *IEEE Trans. Syst. Man, Cybern. Syst.* **2020**, *51*, 209–222.

**Fan Yun** received the B.S. degree from the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China in 2022. She is currently pursuing the Ph.D. degree in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. Her research interests include machine learning and data mining.

**Zhiwen Yu** is a Professor in School of Computer Science and Engineering, South China University of Technology, China. He received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2008. Dr. Yu has authored or coauthored more than 170 refereed journal articles and international conference papers, including more than 60 articles in the journals of IEEE Transactions. He is an Associate Editor of the IEEE Transactions on systems, man, and cybernetics: systems. He is a senior member of IEEE and ACM, a Member of the Council of China Computer Federation (CCF).

**Kaixiang Yang** received the B.S. degree and M.S. degree from the University of Electronic Science and Technology of China and Harbin Institute of Technology, China, in 2012 and 2015, respectively, and the Ph.D. degree from the School of Computer Science and Engineering, South China University of Technology, China, in 2020.
He has been a Research Engineer with the 7th Research Institute, China Electronics Technology Group Corporation, Guangzhou, China, from 2015 to 2017, and has been a postdoctoral researcher with Zhejiang University from 2021 to 2023. He is currently an associate professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. His research interests include pattern recognition, machine learning, and industrial data intelligence.