SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

June 2026, Vol. 69, Iss. 6, 162106:1–162106:16
https://doi.org/10.1007/s11432-025-4957-6

Adaboost selective learning based on multi-kernel broad learning system

Fan YUN, Zhiwen YU*, Kaixiang YANG & C. L. Philip CHEN

School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China

Received 16 January 2025/Revised 1 February 2026/Accepted 3 March 2026/Published online 19 May 2026

Abstract To enhance the feature-extraction ability of kernel-based broad learning system (BLS) and thereby improve its overall performance, the paper proposes an Adaboost selective ensemble learning algorithm based on multi-kernel BLS (MKBLS). AS-MKBLS integrates multi-kernel learning at both the feature and decision levels, comprising three main components. First, recognizing the limited feature-mapping capabilities of traditional single-kernel BLS and the challenges posed by manual parameter tuning, a multi-kernel BLS algorithm is designed. By fusing global and local kernel functions to capture diverse feature information, MKBLS achieves composite feature mapping ability, enabling better data representation in high-dimensional feature space, thereby improving model prediction accuracy. Next, to overcome the limitations of feature-level fusion in multi-kernel functions, MKBLS serves as the base model for an ensemble algorithm, designed to fuse the multi-kernel functions at the decision level. Finally, addressing the issue of traditional selective ensemble methods relying on a single measurement metric, a multi-dimensional selective ensemble algorithm based on MKBLS is proposed. Combining performance metrics of accuracy and diversity helps select both accurate and complementary base MKBLS, thereby enhancing the overall performance of the selective ensemble model. Experiments demonstrate the superiority of the proposed method, highlighting its advancements over other kernel-based learners and ensemble methods.

Keywords Adaboost, broad learning system, kernel method, multi-kernel learning, selective ensemble learning

Citation Yun F, Yu Z W, Yang K X, et al. Adaboost selective learning based on multi-kernel broad learning system. Sci China Inf Sci, 2026, 69(6): 162106, https://doi.org/10.1007/s11432-025-4957-6

1 Introduction

Broad learning system (BLS) [1] is a lightweight neural network with a single hidden layer that has the universal approximation capability. BLS utilizes the pseudo-inverse computation for weight update, thereby avoiding problems such as vanishing and exploding gradients. While the random weight generation mechanism increases the computational speed of BLS, it also introduces issues like node redundancy and unstable performance. To address these challenges, researchers have incorporated kernel methods into BLS, replacing the random-weight mechanism with kernel mapping to generate hidden nodes, thus improving the nonlinear mapping ability and stability of BLS.

However, existing kernel-based BLS models [2] use Gaussian kernels as the kernel function. In contrast, multi-kernel learning (MKL) [3] provides greater flexibility by automatically learning the optimal combination of kernels and their parameters from the data. Different kernel functions offer unique characteristics and mapping abilities. The composite feature space formed by multiple kernels combines the distinct feature-mapping capabilities of each subspace to integrate data from diverse sources.

There are two main strategies for fusing multi-kernel learning: feature-level fusion and decision-level fusion. In feature-level fusion, multiple base kernels are first combined into a composite kernel based on a predefined strategy, followed by the application of a single-kernel learning method to the combined kernel. These approaches, however, are often characterized by complex optimization tasks, low efficiency, and scalability issues, limiting their applicability in large-scale scenarios. On the other hand, decision-level fusion in multi-kernel learning applies a single-kernel algorithm to each base classifier to obtain decision-level results. These results are then integrated using strategies such as ensemble learning, which combines the decision-level outputs of multiple kernels.

To enhance the efficiency and performance of multi-kernel fusion, this paper proposes an ensemble learning method for the multi-kernel broad learning system (MKBLS) that incorporates both feature-level and decision-level fusion from multi-kernel learning. The contributions of the paper are summarized as follows.

* Corresponding author (email: zhwyu@scut.edu.cn)

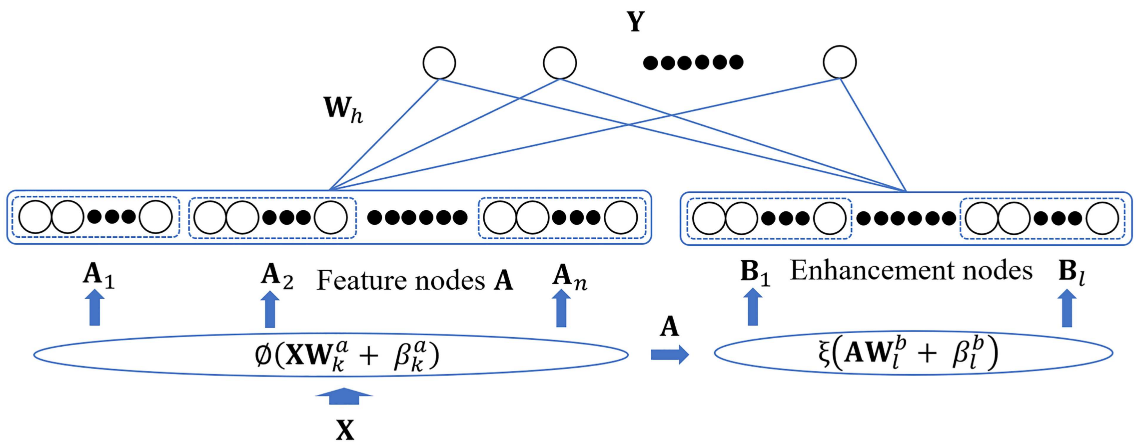

Figure 1 (Color online) Broad learning system structure.

(1) In response to the limitations of the feature mapping capacity of the single-kernel BLS and the necessity for manual parameter adjustment, an MKBLS algorithm that combines the advantages of global and local kernels is designed to capture richer features and elevate the performance of BLS.

(2) To expand the limited feature-level fusion methods of MKBLS, an ensemble algorithm based on MKBLS, named A-MKBLS, is designed to extend multi-kernel fusion to the decision level, thereby enabling a comprehensive and multi-view fusion of kernel BLS. Additionally, 3 diversity enhancement approaches are employed to enhance the diversity of ensemble learning further, thereby improving the performance of MKBLS ensemble algorithms.

(3) To address the limited selection capability of traditional selective ensemble methods based on the single measurement index, a multi-dimensional selective ensemble framework that incorporates precision and diversity is designed. More accurate and complementary base MKBLS are chosen to further select and optimize the multi-kernel feature space, thereby improving the overall performance of the selective ensemble MKBLS.

The rest of this article is organized into five sections. Section 2 provides a concise introduction to the background and research status of BLS and kernel learning. Section 3 introduces three BLS algorithms: MKBLS, A-MKBLS, and AS-MKBLS. The ablation experiment and comparison experiment are introduced in Section 4. The last section summarizes the key conclusions and contributions of the whole paper and forecasts future work.

2 Related work

This section provides a comprehensive review of the literature and research advancements in the fields of broad learning system, kernel learning, and kernel-based BLS.

2.1 Broad learning system

Originating from random vector functional link (RVFL) [4], BLS also has a single hidden layer. Due to its universal approximation ability, BLS has received widespread attention.

Figure 1 shows the structure of the broad learning system. In BLS, the learning process is streamlined and efficient, typically involving three main stages: feature node $\mathbf{A}$ generation with linear function ϕ, enhancement node $\mathbf{B}$ generation with nonlinear function ξ, and output weights $\mathbf{W}_h$ update.

BLS variants [5] can primarily be categorized into 3 main areas: objective functions [6, 7], feature extraction techniques [8, 9], and applications. Some researchers modify the objective function to adapt BLS to different tasks. For example, Zhang et al. [10] substituted the minimum error entropy for the minimum mean-square error of BLS. Zheng et al. [11] used the maximum correntropy criterion to train C-BLS. Jin et al. proposed label enhancement methods LDMBLS, REMBLS [12], and GLEBLS [13].

BLS has been applied to different kinds of data, including image data [14, 15], time series data [16, 17], biomedical data [18, 19], communication data [20], industrial data [21], and other different fields [22], such as semi-supervised learning [23, 24].

Table 1 A summary of different kernel BLS.

Category	Algorithm	Multi-kernel feature fusion	Multi-kernel decision fusion	Different kinds of kernels	Application
Single kernel method	CSKBLS [31]	×	×	×	Software defect prediction
	DKWBLS [32]	×	×	×	Imbalanced data classification
	DKCSBLS [33]	×	×	×	Imbalanced data classification
Multi-kernel method	MKBLS [34]	✓	×	×	General classification
	BGAT-MK [35]	✓	×	✓	Semantic segmentation
Ensemble kernel method	E-KBLS [2]	×	✓	×	Load forecasting
	PEKB [36]	×	✓	×	Noisy data classification
	Ours	✓	✓	✓	General classification

2.2 Kernel learning

Kernel learning methods [25] are a class of machine learning algorithms operated by mapping data into a higher-dimensional feature space, where linearly inseparable patterns become linearly separable. Kernel methods are particularly useful for solving complex, nonlinear problems while leveraging the computational efficiency of linear algorithms.

The choice of kernel function is crucial in kernel methods, as different kernels capture different types of relationships or similarities between data. Commonly used kernels include linear kernels, polynomial kernels, radial basis function (RBF) kernels, and sigmoid kernels. Selecting the appropriate kernel and tuning its parameters can significantly impact the performance of the algorithm.

Kernel learning methods can be divided into single-kernel learning and multi-kernel learning. The main task of single kernel learning is to select a kernel parameter suitable for data expression. Although single kernel learning is widely used, it cannot effectively handle data with heterogeneous features from multiple sources. In many practical scenarios, using a single predefined kernel function is usually not sufficient, because features of actual data are often not singular, but heterogeneous. If different features share the same kernel function, the model may not obtain the optimal mapping result. In addition, single kernel methods require selecting kernel functions and specifying parameters based on experience or experiments, which is inconvenient. This has sparked research in multi-kernel learning, which combines multiple kernel functions to achieve better capabilities and greater flexibility in solving real-world problems.

Real-world data often has multiple feature expressions, corresponding to kernel learning, where various features can be extracted by multiple kernels. Multi-kernel learning is a research hotspot in machine learning, with wide-ranging applications in pattern recognition [26], remote sensing [27], video analysis [28], and various fields. MKL focuses on how to utilize a set of base kernels to improve the feature expression ability. There are two kinds of kernel fusion methods in multi-kernel learning, feature-level fusion [29], and decision-level fusion [30]. Feature-level fusion first fuses a set of base kernels into a combination kernel according to a certain strategy, and then applies the single-kernel learning algorithm to the combination kernel. Decision-level fusion refers to implementing a single-kernel algorithm on each base kernel to obtain decision-level features, and then applying a certain strategy, such as ensemble methods, to fuse decision-level features.

Compared with single-kernel learning, multi-kernel learning can extract more feature expressions and has better fitting ability and robustness. Therefore, multi-kernel learning is designed and applied in our BLS algorithm.

2.3 Kernel-based broad learning system

The random weight mapping of the broad learning system ensures the model's efficiency, but also results in issues of instability and interpretability. The quality of node mapping determines the function-fitting performance of BLS. Thus, crafting a more potent and efficient approach for node mapping is an imminent challenge that requires resolution.

To tackle the problems of node redundancy and fluctuating performance in a broad learning system, the kernel-based BLS was introduced to enhance the node generation mechanism. This innovation aims to enhance both the capacity for nonlinear feature mapping and BLS overall stability.

Table 1 shows different kinds of kernel BLS compared with the proposed method. Existing kernel BLS variants, such as CSKBLS [31], DKWBLS [32], and DKCSBLS [33], replace random projection with a single Gaussian kernel to suppress noise and redundancy. However, these approaches are limited by one fixed kernel and therefore capture only one scale of feature information. On the one hand, MKBLS [34] partially addresses this by combining several Gaussian bandwidths in KBLS, which can be regarded as kernel fusion at the feature level. BGAT-MK [35] employs multiple kernel functions to map nodes, enriching the diverse feature representations of BLS hidden nodes in different kernel spaces. On the other hand, E-KBLS [2] and PEKB [36] embed Gaussian kernels in BLS and use KBLS in ensemble learning to improve model robustness, known as kernel fusion at the decision level.

In summary, existing implementations of kernel broad learning systems predominantly employ the Gaussian kernel function for kernel mapping, with many relying on a single kernel approach. These methods need the selection of appropriate kernel functions and parameter specifications, typically based on empirical knowledge or through trial-and-error experiments, which can be inefficient.

Moreover, most of the existing kernel BLS focus on specific domains. However, for general classification tasks, real-world data often exhibit heterogeneous features rather than being singular. The optimal kernel function to handle such diverse characteristics may differ significantly. Utilizing a uniform kernel function in BLS may not achieve the most effective mapping for the data.

Different kernel functions possess distinct attributes that enable them to map data into various spaces, thereby capturing a wide range of data features. In practical scenarios, the integration of multiple kernel functions can be advantageous in dealing with heterogeneous data from diverse sources. The ensemble of kernels can combine the unique strengths of each kernel to extract a richer set of features, leading to improved model performance and generalization. The shift towards using multiple kernel functions in BLS reflects a growing recognition of the complexity and variability in data and the need for more flexible and adaptive machine-learning approaches. By leveraging the complementary nature of different kernels, BLS can better capture the underlying structure and patterns in the data, enhancing its predictive accuracy and robustness.

Therefore, applying multi-kernel learning methods to the node mapping of BLS is a worthwhile research direction. According to the information we have, the proposed method is the first to use different kinds of kernels in BLS, with multi-kernel fusion at the feature level and decision level, aiming at general classification tasks.

3 The proposed multi-kernel broad learning system

To improve the feature mapping capability and performance of BLS, we design a multi-kernel BLS ensemble algorithm, which can be expressed in three parts. Firstly, MKBLS is designed to combine multiple global and local kernels on the feature level. Secondly, a boosting algorithm based on MKBLS is designed to complete the decision-level fusion of multiple kernels. Finally, a multi-dimensional selective ensemble method is built to better select base MKBLS with strong feature expression.

3.1 Problem definition

In this paper, we address the problem of multi-class classification on general datasets. The performance of the proposed algorithm is evaluated based on the precision metric, which measures the proportion of correctly classified instances among all instances assigned to a particular class. The paper aims to design a multi-kernel selective ensemble framework to improve the multi-kernel fusion on the feature level and decision level of BLS. Our goal is to enhance the classification accuracy and achieve superior performance compared to existing methods on various benchmark datasets.

The training process for the proposed AS-MKBLS is as follows. First, train a sequence of base MKBLS classifiers mapping from the input data $\mathbf{X}$ to the multi-kernel space, and then into the label space Y. Second, select the base MKBLS with the proposed multi-dimensional metric to get well-performed and diverse base learners. Finally, retrain the results of selected base MKBLS to form the final ensemble result $\hat{Y}$.

The classification performance evaluation index is accuracy, which is defined as follows:

$$accuracy = \frac{1}{m}\sum_{i}^{m}\mathbf{I}\left(\widehat{Y}_i = Y_i\right), \tag{1}$$

where m is the number of samples in the dataset, $\widehat{Y}$ denotes the prediction result and Y denotes the ground-truth label. $\mathbf{I}(\widehat{Y}_i = Y_i)$ is the indicator function. If the ith sample is classified correctly, $\mathbf{I}$ returns 1, otherwise returns 0.

3.2 MKBLS

A multi-kernel combination strategy is designed to construct MKBLS, which can improve the fitting ability and performance of the single-kernel BLS. The kernel method is particularly beneficial for improving the nonlinear feature

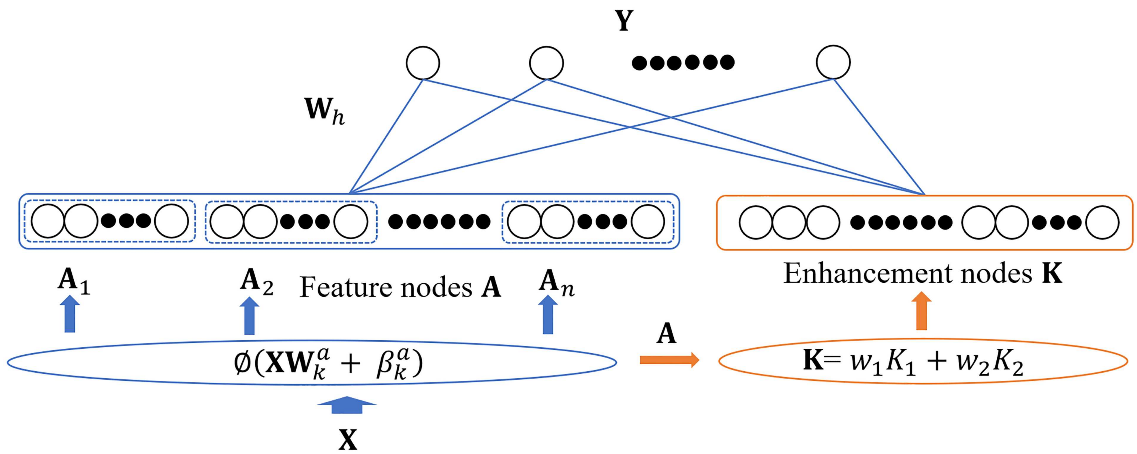

Figure 2 (Color online) Network structure of MKBLS.

expression capabilities of the BLS. By doing so, MKBLS can capture more complex patterns and relationships within the data, leading to better model predictions.

Global kernels have powerful generalization ability to extract global features, and local kernels have strong fitting ability to extract local features. To further improve the generalization and feature extraction abilities of the multi-kernels, the proposed MKBLS employs the local Gaussian kernel and the global polynomial kernel to gather the strengths of different kinds of kernels.

The structure of the proposed MKBLS is shown in Figure 2. The initial step of constructing MKBLS is using a linear function ϕ to generate n groups of feature nodes from input data $\mathbf{X}$:

$$\mathbf{A}_k = \phi\left(\mathbf{X}\mathbf{W}_k^a + \beta_k^a\right), \quad k = 1, 2, \dots, n. \tag{2}$$

By randomly generating the weights $\mathbf{W}_k^a$ and biases β_k^a, and then fine-tuning by a sparse autoencoder (SAE) [37] to obtain sparse features in BLS. SAE optimizes parameters by minimizing the reconstruction error and employs L1 regularization to limit the activation of weights, which removes redundant features, making feature representation more compact. Assume that n' is the number of feature nodes in each group, so the total number of feature nodes is $n_a = nn'$. m is the number of samples, so the dimension of $\mathbf{A}$ is (m, n_a).

MKBLS employs the multi-kernel mapping technique to generate enhancement nodes from feature nodes $\mathbf{A}$, as shown in the orange part of Figure 2. The proposed multi-kernel method uses a linear weighted combination of global and local multi-scale kernels to avoid complex optimization problems. The combination allows MKBLS to benefit from the strengths of different kernels: the local kernel K_1 for capturing fine-grained details and the global kernel K_2 for general trends. We use a linear combination to simplify the model, avoiding the need for complex computational overhead and time-consuming optimization algorithms. The kernel function of MKBLS is as follows:

$$\mathbf{K} = w_1 K_1 + w_2 K_2, \quad w_1, w_2 > 0, \tag{3}$$

where $w_1 + w_2 = 1$. It is simple to extract the critical feature information with the convex sum of different kernels. In the positive form, weights are intuitive for representing the contribution of different kernels, where a larger weight indicates a greater contribution of the corresponding kernel.

The proposed MKBLS uses different types of multi-scale kernels as candidate kernels, including local Gaussian kernels and global polynomial kernels.

The Gaussian kernel, also known as the radial basis function (RBF) kernel, maps all feature nodes into a space that is constrained within an invisible sphere in the high-dimensional feature space. This spherical constraint is due to the exponential decay of the kernel function with increasing distance between points, which emphasizes the locality property. The influence of a point on its neighbors diminishes with distance.

$$K_1 = K_{gauss}(a_1, a_2) = \mathrm{e}^{-\frac{\|a_1 - a_2\|^2}{2\sigma^2}}, \quad \sigma > 0, \tag{4}$$

where σ controls the scope of the Gaussian kernel. The larger the value is, the larger the local influence range of the Gaussian kernel function.

The polynomial kernel is another widely used kernel function in machine learning. In the polynomial kernel, samples can be mapped into the high-dimensional polynomial feature space to extract more complex relationships by setting kernel parameters. Therefore, the polynomial kernel is a kind of global kernel. The polynomial kernel can be defined as

$$K_2 = K_{poly}(a_1, a_2) = (\alpha \cdot a_1 \cdot a_2 + c)^d, \quad d = 1, 2, \ldots, N. \tag{5}$$

When the parameters α and c are set to large values, the polynomial kernel can map the data into a high-dimensional space. This property is called the global aspect of the polynomial kernel because it can capture a wide range of interactions between the data points, not just those close to each other.

Combining global and local kernel functions in MKBLS is an effective strategy to leverage both strengths while mitigating their weaknesses. The fusion of the Gaussian kernel and the polynomial kernel can lead to a more comprehensive feature space that captures both global trends and local patterns. The balance between global and local information can help in scenarios where both broad trends and fine-grained details are important for decision-making. Different kernel functions correspond to implicit feature spaces with varying smoothness and scales. The fusion of multiple kernels incorporates diverse feature spaces. Therefore, the multi-kernel extracts richer features, resulting in the improvement of BLS performance.

In addition, using the proposed multi-kernel method to generate nodes reduces uncertainty in BLS and improves its feature-extraction ability. In traditional BLS, selecting the optimal number of enhancement nodes can be a challenging task that requires tuning and validation. With the multi-kernel method, the kernel matrix's dimension is inherently determined by the number of samples, simplifying this selection process. MKBLS has enough capacity to capture the complexity of the data without the need for manual tuning of enhancement nodes.

Moreover, the multi-kernel function accumulates the feature space dimensions of the single-kernel function, resulting in a higher feature dimension after multi-kernel mapping. In high dimensions, truly similar samples are distributed in clusters, with distances further reduced, making similar samples closer in the multi-kernel feature space. This proximity is beneficial for classification tasks as it can lead to more distinct and separable classes in the feature space.

The enhancement nodes $\mathbf{K}$ are mapped by the proposed multi-kernel function, so the dimension of $\mathbf{K}$ is (m, m). Subsequently, the hidden layer of MKBLS can be written as $\mathbf{H} = [\mathbf{A}\ \mathbf{K}]$. Hidden layer $\mathbf{H}$ consists of feature nodes $\mathbf{A}$ and enhancement nodes $\mathbf{K}$, so the dimension of $\mathbf{H}$ is $(m, m+m) = (m, 2m)$.

The objective function for BLS is designed as follows:

$$Loss = \|\mathbf{Y} - \hat{\mathbf{Y}}\|_2^2 + \frac{\lambda}{2}\|\mathbf{W}_h\|_2^2, \tag{6}$$

where $\hat{\mathbf{Y}} = \mathbf{H}\mathbf{W}_h$ is the prediction output and $\mathbf{Y}$ is the class label, $\mathbf{W}_h$ denotes the weight to the output and λ is the regularization term. Utilizing the objective function, the final connection weight of BLS can be calculated as follows:

$$\mathbf{W}_h = \left(\mathbf{H}^\top\mathbf{H} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}^\top\mathbf{Y}, \tag{7}$$

where $\mathbf{I}$ represents the identity matrix.

The pseudo-inverse $\mathbf{H}^\dagger$ can be calculated to facilitate the computation of the optimal weights of BLS:

$$\mathbf{H}^\dagger = \lim_{\lambda\to 0}\left(\mathbf{H}^\top\mathbf{H} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}^\top. \tag{8}$$

In conclusion, Eq. (7) can be simplified to represent the weight matrix $\mathbf{W}_h = \mathbf{H}^\dagger\mathbf{Y}$.

The pseudo-code of MKBLS is shown in Algorithm 1. MKBLS improves the performance of subsequent ensemble algorithms, as the base learner can more easily distinguish between different classes based on their feature representations. MKBLS is the fusion of multiple kernels on the feature level.

Algorithm 1 MKBLS.

Require:
Input: the dataset $\mathbf{X}$; the number of feature nodes n; the parameter of kernel functions σ; the kernel weights w_1, w_2;
Ensure:
1: **For** k in $1, \ldots, n$ do:
2: Generate feature nodes with (2);
3: **End For**
4: Concatenate all the feature nodes $\mathbf{A} = [\mathbf{A}_1\ \mathbf{A}_2\ \cdots\ \mathbf{A}_n]$;
5: Generate enhancement nodes $\mathbf{K}$ with (3);
6: Concatenate feature nodes and enhancement nodes; get the hidden layer matrix $\mathbf{H} = [\mathbf{A}\ \mathbf{K}]$;
7: Compute the weight $\mathbf{W}_h$ with (7);
Output: the weight matrix $\mathbf{W}_h$.

3.3 A-MKBLS

Adaboost based on the MKBLS (A-MKBLS) is designed to perform kernel decision-level fusion by combining the predictions of MKBLS classifiers with different kernels and parameters. Ensemble fusion takes advantage of diverse feature representations provided by various kernels. By focusing on the ensemble's overall performance rather than optimizing each kernel separately, the proposed method avoids the potential complexity of tuning multiple kernel parameters. The ensemble approach is universal and flexible, making A-MKBLS suitable for a wide range of practical applications. It can be applied to various domains where data may come from different sources and exhibit diverse characteristics.

The multi-kernel ensemble algorithm improves the statistical, expression, and computational ability of MKBLS. On the statistical level, ensemble learning integrates the prediction results of different base MKBLS to obtain the optimal ensemble result, which performs better than a single base learner. On the expression level, ensemble classifiers expand the feature space to a higher dimension, which improves the fitting ability of base MKBLS. On the calculation level, compared to single-base MKBLS, the diversity of ensemble learning makes the model more likely to jump out of local optima and achieve better global performance.

To achieve twice the result with half the effort in ensemble learning, it is important to improve the diversity between base learners. However, most of the ensemble methods have just one kind of diversity enhancement method. Bagging [38] creates diverse data subsets by random sampling to improve data diversity. The random subspace method [39] enhances feature diversity by training each base learner on a random subset of the features. Random forest [40] extends the idea of the random subspace by introducing randomness in the selection of samples and features to improve the ensemble diversity.

The proposed boosting framework with MKBLS is designed to enhance the ensemble diversity in three aspects: data diversity, feature diversity, and decision diversity, which is critical for improving ensemble performance.

The ensemble framework of A-MKBLS is as follows. Firstly, we initialize the sample weight W_1 with (9). If there are m samples in the dataset, then each sample weight is $\frac{1}{m}$, which means the first base MKBLS is trained in the initial data distribution:

$$W_1 = \left[\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}\right]. \tag{9}$$

For the jth base classifier with $j = 1, 2, \ldots, N$, we assign weight W_j to the sample set and get the base MKBLS h_j. Data diversity is achieved through the sequential sampling of the Boosting framework, leading to a diverse set of training data for each MKBLS in the ensemble method.

Adaboost [41] uses the sequential sampling method, the sample weight of the subsequent base learner is determined by the preceding training prediction. When the jth base MKBLS misclassifies some samples, Adaboost increases the weight of these hard samples W_{j+1} with W_j. Adaboost also reduces the weight of the samples that were correctly classified. In this way, the subsequent base classifiers can pay more attention to difficult samples, which continuously improves the function-fitting ability of the ensemble model. It can be seen that the kernel mapping distribution is optimized during the ensemble iterations.

MKBLS combines the global polynomial kernel and local Gaussian kernel. Parameter diversity is introduced by setting different kernel parameters for each base MKBLS learner. Different kernel combinations and parameters result in different feature expressions. When the Gaussian parameter σ in (4) is large, it can be used to classify samples with gentle changes, resulting in a better generalization ability. When σ is small, the Gaussian kernel can classify samples with drastic changes.

Therefore, A-MKBLS first uses a large-scale Gaussian kernel to fit the samples in the smooth region of the corresponding decision function. Through the boosting process, A-MKBLS gradually reduces kernel parameters and uses the small-scale kernel to fit the samples in the region where the decision function changes relatively dramatically. The change of the kernel parameter exactly corresponds to the attribute of the boosting framework, where the former base MKBLS classifies easy samples, and the latter base learners deal with the difficult ones. Therefore, the subsequent ensemble iteration utilizes the results of the previous steps to perform step-by-step optimization, ultimately obtaining better classification results. The parameter of the polynomial kernel is set to 2, aligned with the Gaussian kernel dimension for linear weighted fusion.

In this way, parameter diversity brings feature diversity, since different kernel combination ways have different feature expressions, which ensures that each MKBLS learner captures unique aspects of the data.

Then, calculate the classification error of the jth base MKBLS:

$$e_j = W_j \cdot \mathbf{I}\left(h_j(x_i) \neq Y_i\right), j = 1, 2, \ldots, N, \tag{10}$$

where $h_j(x_i)$ is the result of the jth base learner, Y_i represents the ith data label, $\mathbf{I}$ denotes the indicator function:

$$\mathbf{I}_j = \mathbf{I}\left(h_j(x_i) \neq Y_i\right) = \begin{cases} 1, & \text{if } h_j(x_i) \neq Y_i, \\ 0, & \text{if } h_j(x_i) = Y_i. \end{cases} \tag{11}$$

The error of the jth base MKBLS is e_j, so $1-e_j$ is the accuracy of this learner, then $\frac{1-e_j}{e_j}$ is the relative accuracy. Obviously, the higher the relative accuracy, the greater the weight coefficient of the classifier should be.

The sequential sampling method updates base MKBLS weights by adaptively evaluating the former base learner's performance. Therefore, the weight coefficients α_j of the jth base MKBLS in the final decision process can be defined as

$$\alpha_j = \frac{1}{2}\log\frac{1-e_j}{e_j} + \log(R-1), j = 1, 2, \ldots, N, \tag{12}$$

where R is the number of dataset categories, N is the number of classifiers. For the testing set, we can use the average training α_j as its weight.

In this way, A-MKBLS benefits from the diverse base of MKBLS and the varied influence each learner has on the final decision. Decision diversity arises from the combination of weights and predictions of base MKBLS.

The sample weight of the ith sample in the $(j+1)$th base learner is

$$W_{j+1} = \frac{W_j}{Z_j}\mathrm{e}^{\alpha_j \cdot \mathbf{I}_j}, \tag{13}$$

where Z_j denotes the normalization factor:

$$Z_j = \sum_{j=1}^{N} W_j \mathrm{e}^{\alpha_j \cdot \mathbf{I}_j}. \tag{14}$$

To ensure that the training and testing samples lie in the same subspace, the average of the training data weights can be used as the weights for the testing data, so the same linear transformation operation can be performed on the testing set. As a result, the testing samples can be mapped to the same subspace as the training samples.

Finally, compute the ensemble prediction of A-MKBLS:

$$\widehat{\mathbf{Y}} = \sum_{j=1}^{N} \alpha_j \widehat{\mathbf{Y}}_j. \tag{15}$$

The loss function of A-MKBLS is as follows:

$$\min_h \frac{1}{m}\sum_{i=1}^{m} \mathbf{I}(\widehat{Y}_i \neq Y_i), \tag{16}$$

where m is the number of samples. Base MKBLS are trained to minimize the probability of misclassification.

The error bound for training the final learner of A-MKBLS is

$$\frac{1}{m}\sum_{i=1}^{m} \mathbf{I}(h(x_i) \neq Y_i) \leqslant \frac{1}{m}\sum_{i=1}^{m} \mathrm{e}^{-Y_i \widehat{Y}_i} = \prod_j Z_j, \tag{17}$$

where $h(x)$ is the final ensemble method, and Z_j is a linear combination of the weights of each sample in the jth training round, with $Z_j \leqslant 1$. This ensures the convergence of the proposed algorithm during the iteration process.

3.4 AS-MKBLS

The diversity of A-MKBLS leads to unstable performance of the base classifiers, requiring the selective ensemble of MKBLS. It is necessary to choose a series of base learners with good kernel combinations and prune the redundant and poorly performing base learners. Good kernel combinations lead to good feature extraction ability and better performance.

Table 2 Multidimensional metric.

$f(x)$	$p_1(x)=+1$	$p_1(x)=-1$
$p_2(x)=+1$	1	0
$p_2(x)=-1$	0	-1

Generally speaking, selective ensemble methods only consider a single evaluation metric, such as accuracy or diversity. However, the same accuracy or diversity does not necessarily represent the same performance of different base learners.

Therefore, selective ensemble learning based on the multi-dimensional metric (accuracy and diversity) can measure the performance of the base classifier in a more comprehensive and precise way, thereby improving the overall performance of selective ensemble learning.

The paper proposed a multidimensional selective ensemble learning method based on MKBLS (Adaboost selective-multi-kernel broad learning system, AS-MKBLS). The design of the multidimensional selection indicator is as follows.

Firstly, count the classification performance of each base MKBLS h for each sample x_i:

$$p(x_i)=\begin{cases}+1, \text{ if } h(x_i)=Y_i,\\ -1, \text{ if } h(x_i)\neq Y_i,\end{cases} \tag{18}$$

where $\mathbf{Y}_i$ is the label of sample x_i.

Then, accumulate the performance of the jth classifier $h_j(x)$ for the ith sample and get $f(x_i)$,

$$f(x_i)=\frac{1}{j}\sum_{j=1}^{j}p_j(x_i). \tag{19}$$

Table 2 shows the condition of two base learners. It can be seen that $f(x)\in[-1,1]$. The ensemble accuracy is highest when $f(x)=1$, denoting the perfect consistency without errors. The ensemble diversity is highest when $f(x)=0$. The ensemble accuracy and diversity reach the minimum value when $f(x)=-1$. Using 0 as the dividing line, $f(x)>0$ means that more than half of the base MKBLS predict correctly, which is the situation we want. Otherwise, more than half of the base learners predict wrong, then we prune the underperforming MKBLS.

Then, calculate the multi-dimensional performance of the jth base MKBLS h_j among all data, where $j=1,2,\ldots,N$, N is the number of base learners, and m denotes the data number,

$$g(h_j)=\frac{1}{m}\sum_{i=1}^{m}f(x_i)=\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{j}\sum_{j=1}^{j}p_j(x_i)\right). \tag{20}$$

$g(h_j)$ is the proportion of samples where two or more base models agree on the same task. The proposed multi-dimensional metric can simultaneously demonstrate the diversity and accuracy of the proposed selective ensemble method. The higher the value of $g(h)$, the better the performance of the proposed method. Calculate the average $\bar{g}=\frac{g}{N}$ of all base learners in A-MKBLS, and use it as a threshold to select the base MKBLS. In this way, the following base MKBLS is added when it can improve the accuracy and the diversity of the ensemble algorithm.

AS-MKBLS selects the base MKBLS with high accuracy and diversity to form the final ensemble result, thereby contributing to the overall ensemble BLS performance. Meanwhile, the proposed measurement method has a low computational burden, which avoids affecting the overall computational efficiency of the AS-MKBLS algorithm.

Figure 3 shows the whole structure of the proposed algorithm. First, the sequential sampling method is used to determine the sample diversity. Second, construct multi-kernel spaces to fuse multi-kernel in the feature level, and then apply them in several MKBLS base classifiers. Then, select the base MKBLS with the multi-dimensional evaluation criterion. When the performance of the base learner is under the multi-dimensional threshold, as the second MKBLS shown in Figure 3, its prediction is not included in the final ensemble decision. Last, AS-MKBLS uses the soft label to fuse well-performing kernel features at the decision level. The pseudo code of the proposed algorithm is shown in Algorithm 2.

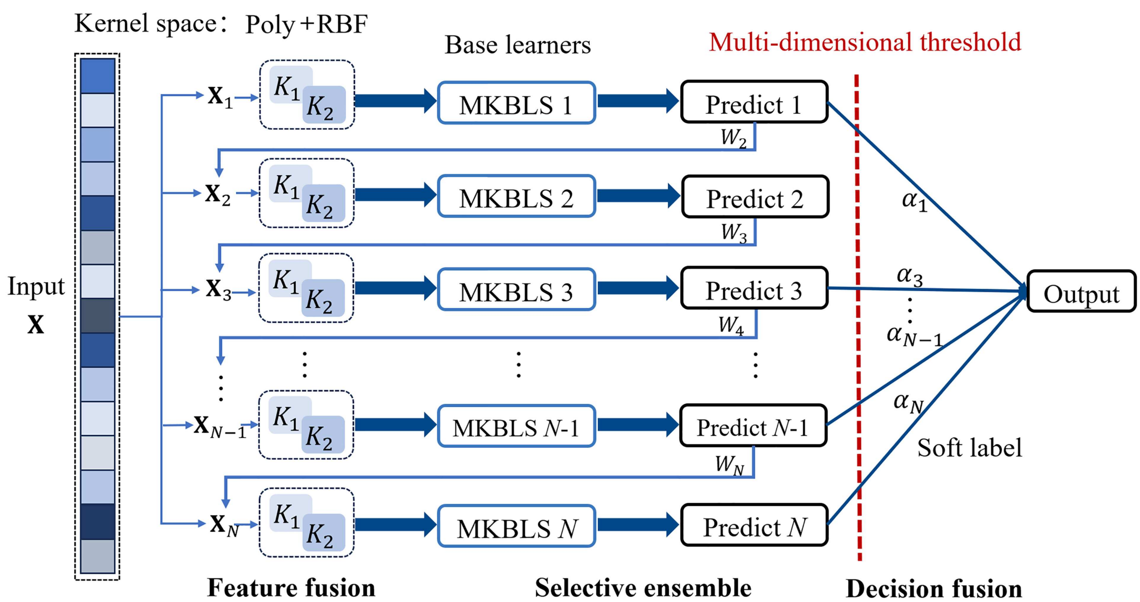

Figure 3 (Color online) Network structure of AS-MKBLS.

Algorithm 2 AS-MKBLS.

Require:

Input: the dataset $\mathbf{X}$; the number of feature nodes n_a; the regularization coefficient λ; the number of base learners N;

Ensure:

For j in $1, 2, \ldots, N$ **do**
  If $j = 1$ **do**
    Initialize the sample weights W_1 by (9); calculate $\mathbf{X}_1 = W_1\mathbf{X}$;
  Else do
    Assign the sample weight W_j to dataset $\mathbf{X}_j = W_j\mathbf{X}_{j-1}$;
  Train the base classifier MKBLS using Algorithm 1;
  Store the result $\widehat{\mathbf{Y}}_j$ of base MKBLS and the classification error e_j;
  Calculate the base BLS weight α_j with e_j by (12), normalize, and store;
  Update the sample weight W_{j+1} with α_j by (14);
End For
Choose the base MKBLS by multi-dimensional metric to participate in the final decision; calculate the ensemble result $\widehat{\mathbf{Y}}$ with $\widehat{\mathbf{Y}}_j$ and α_j by (17);
Calculate the final accuracy of AS-MKBLS;

Output: AS-MKBLS accuracy.

---

4 Experiments

This section includes ablation experiments of the proposed method and comparison experiments with classification methods.

All the experiments were carried out on a computer with a 12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz and 16.0 GB memory. The programming language is Python 3.9.

The performance evaluation indicator of models is the classification accuracy in (1).

4.1 Data preprocess

For the experiment, we utilized 20 real-world datasets in University of California, Irvine (UCI) and knowledge extraction based on evolutionary learning (KEEL), which include 8 binary classification and 12 multi-classification datasets of varying sizes across fields, such as data classification, pattern recognition, and anomaly detection, etc. The dataset information is shown in Table 3, including the dataset name, source, data volume, feature dimension, number of categories, and the corresponding task. Datasets are sorted alphabetically.

Datasets are preprocessed before training. The missing values for some datasets are filled with 0. Each dataset is randomly divided into two parts, 80% of the data is the training set, and 20% of the data belongs to the test set. All data is standardized by the scikit-learn library before training. We randomly repeat the experiment 10 times to calculate the average testing accuracy, as well as the standard deviation of the model in each dataset.

Table 3 A summary of the real-world datasets. All datasets come from the UCI benchmark datasets [42], and KEEL datasets [43].

Dataset	Data	Feature	Category	Task
Allbooks [43]	590	8265	2	Book classification
Bupa [43]	345	6	2	Disease detection
Bands [43]	539	19	2	Industrial inspection
Balance [43]	625	4	3	Balance problem
Climate [43]	540	18	2	Anomaly detection
Cancer [42]	569	30	2	Disease detection
Cleveland [43]	303	13	5	Disease detection
Contraceptive [43]	1473	9	3	Contraceptive prediction
Column_2C [43]	310	6	2	Anomaly detection
Digit [42]	1797	64	10	Digit classification
Newthyroid [43]	215	5	3	Disease detection
Pageblocks [43]	5472	10	5	Format classification
Spectfheart [42]	267	44	2	Protein localization
Segment [42]	2310	19	7	Image segmentation
Thyroid [42]	7200	21	3	Disease detection
Vehicle [43]	676	18	4	Vehicle classification
Vowel [43]	990	13	11	Disease detection
Winequality [42]	4898	11	11	Wine classification
Wine [42]	178	13	3	Wine classification
Wisconsin [42]	699	9	2	Disease detection

Table 4 Basic parameters of MKBLS.

Dataset	n	n'	σ
Allbooks	2	1	0.1
Bupa	19	6	1
Bands	18	15	0.4
Balance	6	11	0.9
Climate	2	1	0.1
Cancer	11	19	0.5
Cleveland	4	7	0.8
Contraceptive	4	7	0.8
Column_2C	2	11	0.1
Digit	16	18	0.4
Newthyroid	2	12	0.9
Pageblocks	8	8	0.1
Spectfheart	2	1	0.1
Segment	6	14	0.8
Thyroid	2	16	0.6
Vehicle	19	14	0.2
Vowel	7	16	0.5
Winequality	6	16	0.9
Wine	5	5	0.5
Wisconsin	5	1	0.5

The basic parameters are obtained by grid search, as Table 4 shows. Datasets are separated into train:validation:test sets = 6:2:2 to avoid the information leakage problem. n and n' are the number of groups and feature nodes in each group. Enhancement nodes are generated by the multi-kernel function, so there is no need to set the number of enhancement nodes. As an alternative, we need to choose the Gaussian kernel coefficient in the kernel function is σ.

4.2 Ablation experiments

The ablation experiment includes two parts: one is the analysis of the multi-kernel fusion performance, and the other is the multi-kernel selection ablation experiment. The ablation experiment is conducted on 6 datasets, including bupa, bands, cancer, Cleveland, newthyroid, and wisconsin.

Table 5 Ablation experiment of KBLS, MKBLS, A-KBLS, A-MKBLS, and AS-MKBLS (mean testing accuracy ± standard deviation), where the best values are highlighted in bold.

Algorithms	Single classifiers			Ensemble classifiers			
Datasets	KBLS (poly)	KBLS (Gauss)	MKBLS	A-KBLS (poly)	A-KBLS (Gauss)	A-MKBLS	AS-MKBLS
Bupa	65.507 ± 4.409	65.056 ± 5.197	66.103 ± 4.551	70.692 ± 2.021	70.048 ± 2.899	75.924 ± 4.643	**77.433** ± 2.049
Bands	55.144 ± 5.615	57.613 ± 2.210	59.465 ± 4.030	56.893 ± 0.817	58.951 ± 3.525	69.136 ± 2.854	**72.407** ± 2.483
Cancer	63.840 ± 3.664	90.058 ± 3.426	93.469 ± 2.198	68.324 ± 8.896	92.690 ± 1.316	95.906 ± 1.641	**98.597** ± 1.804
Cleveland	43.898 ± 12.532	55.009 ± 3.676	56.102 ± 3.043	46.794 ± 3.195	57.741 ± 2.429	59.199 ± 2.773	**66.393** ± 2.839
Newthyroid	78.139 ± 5.047	93.721 ± 2.697	95.581 ± 3.544	78.605 ± 2.859	94.419 ± 3.501	96.279 ± 2.500	**97.342** ± 2.848
Wisconsin	81.286 ± 8.321	84.611 ± 1.867	92.785 ± 2.270	89.143 ± 4.565	96.357 ± 0.855	96.428 ± 0.979	**97.571** ± 0.838
Average	64.636	74.345	77.251	68.409	78.368	82.145	**84.957**

Table 6 Runtime experiment of BLS, KBLS, MKBLS, and AS-MKBLS (average runtime±standard deviation, in seconds).

Datasets	BLS	KBLS	MKBLS	AS-MKBLS
Bupa	0.081 ± 0.022	0.014 ± 0.003	0.028 ± 0.002	0.583 ± 0.051
Bands	0.186 ± 0.013	0.036 ± 0.004	0.057 ± 0.001	1.099 ± 0.145
Cancer	0.202 ± 0.018	0.035 ± 0.003	0.054 ± 0.003	0.896 ± 0.042
Cleveland	0.045 ± 0.002	0.006 ± 0.001	0.027 ± 0.002	0.264 ± 0.091
Newthyroid	0.057 ± 0.016	0.005 ± 0.001	0.019 ± 0.001	0.141 ± 0.050
Wisconsin	0.041 ± 0.010	0.049 ± 0.002	0.063 ± 0.002	0.485 ± 0.196
Average	0.102	0.024	0.042	0.578

4.2.1 *Multi-kernel fusion experiment*

Based on KBLS, the paper proposed 3 algorithms: MKBLS, A-MKBLS, and AS-MKBLS. The proposed single classifiers and ensemble kernel methods are compared in the ablation experiment. The number of base MKBLS is set to 10 in each ensemble method. For multi-kernel methods, we used the polynomial kernel and the Gaussian kernel. Therefore, the single-kernel BLS methods with the polynomial and Gaussian kernel are also compared separately.

Table 5 shows the ablation results on 6 datasets. The table statistics the average test acc percentages and the corresponding standard deviations, and highlights the best scores of each dataset in bold.

As shown in Table 5, the proposed multi-kernel methods combine the advantages of different kinds of kernels, so MKBLS and A-MKBLS have better performance than the single-kernel method polynomial KBLS and Gaussian KBLS. The variance of AS-MKBLS is lower than that of single-kernel BLS, indicating that the proposed ensemble method improves model stability. In general, single kernel BLS with Gaussian kernels performs better than the polynomial kernel, demonstrating the universality of the Gaussian kernel. In specific datasets, such as bands, cancer, and newthyroid, the single MKBLS classifier performs even better than A-KBLS with polynomial or Gaussian kernel, which indicates that MKBLS outperforms single-kernel ensemble methods in certain cases.

Comparing the single classifiers and the ensemble method with the same kernel, the ensemble methods outperform the corresponding base kernel learners, proving the effectiveness of multi-kernel fusion on the decision level. The selective ensemble learning method can select superior kernel fusion combinations, so AS-MKBLS has higher accuracy than A-MKBLS.

Overall, the complete algorithm AS-MKBLS has the best accuracy among all datasets in the ablation experiment. Experimental evidence indicates that the proposed multi-kernel fusion and selection method can more effectively capture data feature information, thereby improving the performance of KBLS.

Table 6 shows the runtime of BLS, KBLS, MKBLS, and AS-MKBLS, where the unit is seconds. MKBLS uses 2 kinds of kernels, so the runtime of MKBLS is less than twice that of single kernel BLS. AS-MKBLS uses 10 base MKBLS, which takes about 10 times as long as MKBLS. The runtime of BLS is greater than that of MKBLS. The feature extraction capability of BLS is inferior to KBLS, so BLS needs to use more enhancement nodes to get better performance. The more hidden nodes of BLS, the larger the matrix for calculating the pseudo-inverse, the more runtime is needed.

4.2.2 *Multi-kernel selection experiment*

The selective ensemble learning method AS-MKBLS uses a multi-dimensional metric to select the base MKBLS. The ablation experiment compares A-MKBLS without the selective method, AS-MKBLS with the one-dimensional

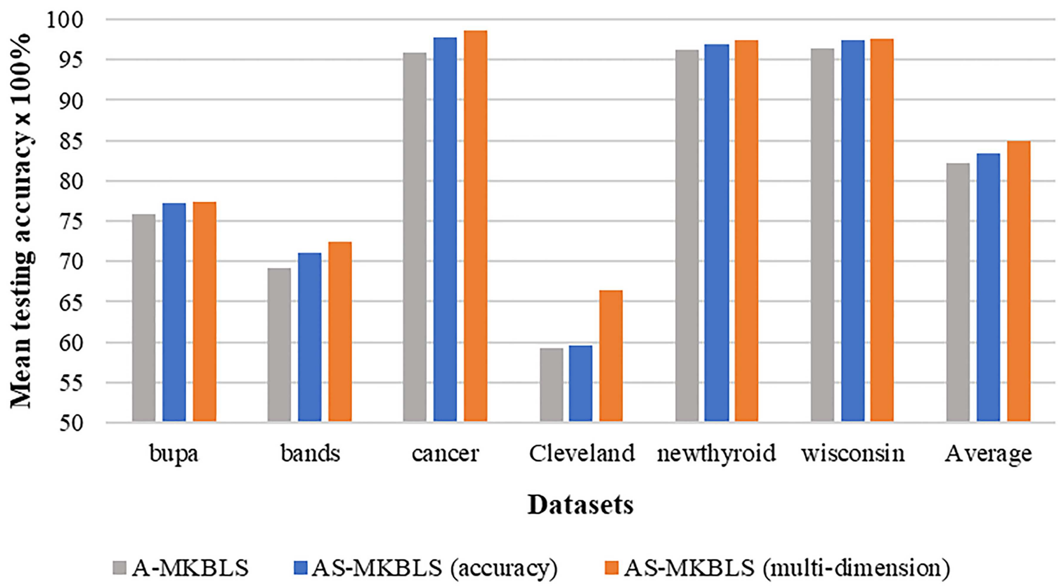

Figure 4 (Color online) Ablation experiment of multi-kernel selection.

metric, and AS-MKBLS with the proposed multi-dimensional metric. Two selective ensemble learning methods AS-MKBLS use the average training performance index as the threshold to select the base MKBLS.

Figure 4 shows the histogram of multi-kernel selection on the 6 datasets. The vertical axis represents the percentage accuracy of the ensemble result. Grey columns stand for the performance of the base ensemble MKBLS, blue columns indicate the performance of the selective ensemble MKBLS with accuracy in (1) as the selective metric, and orange columns represent the performance of the selective ensemble MKBLS with accuracy and diversity as the evaluation criterion. The 3 columns on the right are the average score of the 3 methods compared on the 6 datasets.

The 3 columns are getting taller from left to right, indicating that the algorithm performance gradually improves from A-MKBLS, AS-MKBLS (accuracy), to AS-MKBLS (multi-dimension). Both of the AS-MKBLS methods perform better than the baseline A-MKBLS, which proves that selective ensemble methods can improve the performance of ensemble learning. This is because the selective method improves the lower bound of the base MKBLS classifier performance, therefore choosing better multi-kernel combinations to generate the final ensemble decision.

Comparing the blue and orange columns in each dataset, the performance of the multi-dimensional selective metric is better than that of the accuracy-only selection on all datasets. Specifically, the performance of the proposed multi-dimensional selective method is 6% higher than the accuracy-only selection on the Cleveland dataset. The one-dimensional metric accuracy only focuses on the performance of each base MKBLS, which is calculated separately in each individual base MKBLS. The proposed two-dimensional metric $g(\cdot)$ calculates the agreement proportion of each sample between different base MKBLS, which focuses on the performance of each base learner, and the diversity between different base MKBLS.

The higher the accuracy of the base MKBLS, the greater the diversity between the base MKBLS, and the better the AS-MKBLS ensemble performance. The experiment results indicate that the proposed accuracy-diversity selective learning method outperforms the common single performance metric accuracy in ensemble MKBLS.

4.3 Comparison experiments

AS-MKBLS is compared with 3 single classifiers and 3 ensemble methods, to prove the effectiveness and superiority of the proposed algorithm.

4.3.1 *Comparison with single classifiers*

We compare AS-MKBLS with single classifiers, including broad learning system (BLS), kernel extreme learning machine (KELM), and Gaussian naive Bayes (GNB). BLS has been introduced in related work, so we will briefly introduce KELM and GNB.

KELM [44]: a single hidden layer feedforward neural network. KELM maps input to a high-dimensional feature space through the Gaussian function. The parameters of KELM include the number of hidden nodes and the

Table 7 Comparison experiment with single kernel classifiers and ensemble methods (mean testing accuracy ± standard deviation), where the best values are highlighted in bold.

Algorithms	Ensemble methods				Single classifiers		
Datasets	AS-MKBLS	PEKB	A-KSVM	A-KNN	BLS	KELM	GNB
Allbooks	**100** ± 0	99.576 ± 0.599	99.661 ± 0.415	**100** ± 0	99.407 ± 0.698	94.237 ± 2.847	99.491 ± 0.415
Bupa	**77.433** ± 1.414	69.565 ± 3.804	71.015 ± 3.995	63.810 ± 11.308	64.947 ± 3.853	70.597 ± 3.059	57.681 ± 5.136
Bands	**72.407** ± 2.483	65.370 ± 3.885	60.463 ± 0.228	54.546 ± 14.060	58.551 ± 3.789	67.500 ± 3.161	62.569 ± 3.101
Balance	**92.720** ± 0.952	87.440 ± 2.233	85.760 ± 4.811	84.039 ± 3.349	80.600 ± 0.400	87.620 ± 1.103	87.520 ± 2.610
Climate	94.074 ± 1.576	91.222 ± 3.825	91.482 ± 0.370	87.870 ± 1.770	91.204 ± 0.655	45.920 ± 3.114	**94.815** ± 0.741
Cancer	**98.597** ± 0.613	90.351 ± 3.410	92.448 ± 3.208	91.288 ± 8.498	93.801 ± 0.016	94.649 ± 1.517	93.852 ± 1.165
Cleveland	**66.393** ± 2.834	57.049 ± 6.268	53.792 ± 0.375	55.714 ± 7.468	55.410 ± 5.508	53.115 ± 6.834	57.381 ± 8.425
Contraceptive	51.729 ± 1.074	**54.745** ± 2.322	51.665 ± 1.925	50.885 ± 9.469	47.763 ± 1.397	43.966 ± 0.322	49.083 ± 2.165
Column_2C	**84.329** ± 1.354	81.936 ± 1.881	78.387 ± 2.223	81.191 ± 3.760	80.000 ± 2.281	80.645 ± 3.724	82.158 ± 2.451
Digit	93.389 ± 0.690	**97.666** ± 0.680	96.328 ± 2.008	86.232 ± 1.690	56.204 ± 1.238	49.517 ± 9.648	88.203 ± 2.560
Newthyroid	**97.342** ± 1.605	95.116 ± 2.992	81.861 ± 4.509	92.604 ± 1.079	88.372 ± 7.104	77.674 ± 3.139	93.500 ± 3.425
Pageblocks	95.096 ± 0.516	**96.593** ± 0.404	90.024 ± 0.226	94.793 ± 1.238	90.703 ± 4.406	88.448 ± 0.987	90.004 ± 0.218
Spectfheart	**86.161** ± 3.096	81.462 ± 0.994	81.191 ± 9.797	83.333 ± 5.773	62.037 ± 2.928	62.857 ± 1.813	77.667 ± 11.647
Segment	**94.416** ± 0.498	90.758 ± 1.496	85.974 ± 1.782	89.810 ± 4.883	91.889 ± 6.321	93.306 ± 0.472	80.217 ± 0.558
Thyroid	**93.972** ± 0.463	92.604 ± 1.079	93.097 ± 0.168	93.125 ± 0.208	92.069 ± 0.592	93.306 ± 0.472	92.583 ± 0.028
Vehicle	**73.171** ± 1.556	72.882 ± 4.586	50.127 ± 4.011	65.294 ± 8.504	63.706 ± 9.959	41.353 ± 1.242	47.873 ± 2.554
Vowel	**89.675** ± 2.274	89.494 ± 2.573	49.394 ± 9.215	72.632 ± 7.935	80.859 ± 14.238	52.424 ± 2.342	54.849 ± 8.122
Winequality	**47.392** ± 0.698	47.020 ± 2.359	44.345 ± 0.849	45.612 ± 5.163	46.857 ± 2.087	44.653 ± 1.214	47.100 ± 2.251
Wine	**98.333** ± 1.942	93.056 ± 4.943	69.730 ± 5.944	80.833 ± 2.429	85.000 ± 7.878	91.944 ± 4.431	96.635 ± 2.113
Wisconsin	**97.571** ± 0.838	64.429 ± 2.328	96.429 ± 4.791	97.142 ± 4.738	71.214 ± 3.451	74.571 ± 2.108	95.714 ± 4.738
Average	**85.210**	80.917	76.159	78.683	75.030	70.415	77.445

Gaussian kernel parameter.

GNB [45]: a variant of Naive Bayes based on the Bayesian theorem, which is suitable for features with a Gaussian distribution. The parameter of GNB is var_smoothing, which is the portion of the maximum variance among features, and is used to ensure computational stability.

All parameters of comparison methods are set by the grid search. Analyzing single classifiers comparison in Table 7, AS-MKBLS has the best performance among 19/20 datasets, and it also exceeds the average accuracy of other methods. Compared with BLS, AS-MKBLS effectively improves the feature extraction and classification ability. On 16/20 datasets, the variance of the proposed AS-MKBLS is smaller than that of BLS, reducing the uncertainty of randomly generated enhanced nodes by BLS.

AS-MKBLS also outperforms other kernel classifiers KELM and GNB. On 12 datasets, the accuracy of the proposed method is over 2.6% higher than the optimal comparison algorithm. In addition, AS-MKBLS performs well on disease detection tasks, since the proposed multi-kernel method can extract complex nonlinear relationships among multiple variables in disease data. From the last row of the table, it can be seen that the average score of AS-MKBLS is 7.7% higher than the runner-up method GNB. The comparison result shows the effectiveness of the proposed method.

4.3.2 *Comparison with ensemble methods*

Except for single classifiers, the paper also compares the proposed method with ensemble kernel methods. Since AS-MKBLS uses the boosting ensemble framework with kernel methods, 3 similar ensemble methods are selected in the comparison experiment, which are A-KSVM, A-KNN, and PEKB. The following is the introduction of the competitive ensemble methods.

Adaboost-kernel support vector machine (A-KSVM): the base learner is a support vector machine [46] with a Gaussian kernel, and the ensemble framework is Adaboost.

Adaboost-K nearest neighbors (A-KNN): Adaboost framework with base KNN [47] classifiers.

Progressive ensemble kernel-based broad learning system (PEKB) [36]: a progressive boosting method with Gaussian kernel BLS base learners. PEKB uses the residual of the gradient and subgradient as labels in the training process.

The number of base learners of all the ensemble algorithms is uniformly set to 10, and other parameters in each method are determined by the grid search.

From Table 7 we can see that AS-MKBLS outperforms in 17/20 datasets among the ensemble methods, so it is competitive in ensemble learning. Experimental results indicate that the proposed MKBLS outperforms other base learners in the Adaboost framework, and the proposed MKBLS ensemble method performs better than other ensemble BLS methods. The accuracy of the proposed method is over 2.2% higher than the best comparison ensemble algorithm on 11 datasets, most of which are small-scale data, which shows that the proposed method significantly improves the classification performance on small-scale datasets. This is because the proposed multi-kernel method is suitable for small-scale datasets. In addition, the average score of the proposed AS-MKBLS is the highest among other comparison algorithms, 4.2% higher than the second algorithm PEKB, further indicating the effectiveness of AS-MKBLS among state of the arts.

As shown in Table 7, the ensemble methods outperform single classifiers in most datasets. In particular, comparing BLS, PEKB, and AS-MKBLS, AS-MKBLS outperforms PEKB, and PEKB outperforms BLS. The comparison experiments further demonstrate that the combination of kernel method, ensemble learning, and BLS can enhance feature extraction capabilities, improve model stability, thereby significantly enhancing BLS classification performance.

In experiments, 7 datasets, including bupa, cancer, Cleveland, newthyroid, thyroid, vowel and wisconsin belong to disease diagnosis in the healthcare domain, which are shown in Table 3. In Table 7, AS-MKBLS performs the best among all disease diagnosis tasks, since the proposed MKBLS-based ensemble framework can better extract complex and diverse features in disease data.

5 Conclusion and future work

The paper introduces a selective ensemble approach based on an MKBLS framework. The fusion of MKBLS occurs in two stages: MKBLS integrates global and local kernels at the feature stage, then A-MKBLS extends the fusion of multiple kernels to the decision stage using an ensemble method. By leveraging the complementarity of different kernel functions, the proposed method can better capture underlying structures and patterns in data, thereby enhancing the accuracy and robustness of the model. Additionally, the paper proposes a multi-dimensional performance metric to more effectively measure the accuracy and diversity of the base MKBLS models. Ablation studies validate the effectiveness of each component of the proposed algorithm, while comparative experiments highlight the method's progressive nature, demonstrating its superiority among kernel-based learners and ensemble methodologies.

The proposed AS-MKBLS is a general classification algorithm that can be applied in other research domains, such as medical diagnosis and more complex multimodal data in the future. For more complex multimodal medical data, the proposed AS-MKBLS can use different kernel functions to extract features across different data modalities. For example, the Gaussian kernel can be used to extract medical text features. In addition, other suitable kernels can also be employed for different types of data, for example, using a convolutional kernel to extract medical image features and then fusing the multimodal feature with the proposed ensemble framework.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62572199, 92467109, 62476101, 21A20478), Major Key Project of Peng Cheng Laboratory (Grant Nos. PCL2025A11, PCL2025A13), National Key R&D Program of China (Grant No. 2023YFA1011601), and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515140137).

References

1 Chen C P, Liu Z. Broad learning system: a new learning paradigm and system without going deep. In: Proceedings of 2017 32nd Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2017. 1271–1276
2 Yang K, Wang H, Yang Q, et al. An ensemble broad learning scheme for short-term load forecasting. In: Proceedings of 2022 China Automation Congress (CAC), 2022. 4901-4905
3 Sun T, Jiao L, Liu F, et al. Selective multiple kernel learning for classification with ensemble strategy. Pattern Recognit, 2013, 46: 3081–3090
4 Pao Y H, Takefuji Y. Functional-link net computing: theory, system architecture, and functionalities. Computer, 1992, 25: 76–79
5 Yun F, Yu Z, Yang K. Broad learning system in data mining and machine learning. Data Min Mach Learn, 2025, 1: 100002
6 Yang L, Song S, Chen C P. Transductive transfer learning based on broad learning system. In: Proceedings of 2018 IEEE International Conference on Systems, Man and Cybernetics (SMC), 2018. 912–917
7 Jin J, Liu Z, Chen C L P. Discriminative graph regularized broad learning system for image recognition. Sci China Inf Sci, 2018, 61: 1–14
8 Zhou N, Lu J, Yan Z, et al. Short-term load forecasting using broad learning system. In: Proceedings of 2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2), 2019. 497–501
9 Yang L, Yang Z, Li F, et al. Single-layer feedforward neural networks with dynamic width for domain adaptation. Sci China Inf Sci, 2025, 68: 182205
10 Zhang S, Liu Z, Chen C L P. Broad learning system based on the quantized minimum error entropy criterion. Sci China Inf Sci, 2022, 65: 222203
11 Zheng Y, Chen B, Wang S, et al. Broad learning system based on maximum correntropy criterion. IEEE Trans Neural Netw Learn Syst, 2019, 32: 3083–3097
12 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

13 Jin J, Chang S, Duan J, et al. Groupwise label enhancement broad learning system for image classification. IEEE Trans Cybern, 2025, 55: 3431–3444
14 Zhang D, Li T, Chen C L P, et al. Target tracking algorithm based on a broad learning system. Sci China Inf Sci, 2022, 65: 154201
15 Li Y, Zhang T, Chen C L P. Broad siamese network for facial beauty prediction. IEEE Trans Artif Intell, 2024, 5: 5786–5800
16 Xu M, Han M, Chen C L P, et al. Recurrent broad learning systems for time series prediction. IEEE Trans Cybern, 2018, 50: 1405–1417
17 Yi J, Huang J, Zhou W, et al. Intergroup cascade broad learning system with optimized parameters for chaotic time series prediction. IEEE Trans Artif Intell, 2022, 3: 709–721
18 Zhang T, Wang X, Xu X, et al. GCB-net: graph convolutional broad network and its application in emotion recognition. IEEE Trans Affective Comput, 2019, 13: 379–388
19 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
20 Qiu T, Liu X, Zhou X, et al. An adaptive social spammer detection model with semi-supervised broad learning. IEEE Trans Knowl Data Eng, 2022, 34: 4622–4635
21 Lin J, Liu Z, Chen C L P, et al. Quaternion broad learning system: a novel multi-dimensional filter for estimation and elimination tremor in teleoperation. Neurocomputing, 2020, 380: 78–86
22 Gong X, Zhang T, Chen C L P, et al. Research review for broad learning system: algorithms, theory, and applications. IEEE Trans Cybern, 2021, 52: 8922–8950
23 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
24 Guo J, Liu Z, Chen C L P. An incremental-self-training-guided semi-supervised broad learning system. IEEE Trans Neural Netw Learn Syst, 2025, 36: 7196–7210
25 Kwak N. Nonlinear projection trick in kernel methods: an alternative to the kernel trick. IEEE Trans Neural Netw Learn Syst, 2013, 24: 2113–2119
26 Liu J, Liu X, Yang Y, et al. Contrastive multi-view kernel learning. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 9552–9566
27 Niazmardi S, Safari A, Homayouni S. Similarity-based multiple kernel learning algorithms for classification of remotely sensed images. IEEE J Sel Top Appl Earth Observations Remote Sens, 2017, 10: 2012–2021
28 Chen G, Lu J, Feng J, et al. Localized multi-kernel discriminative canonical correlation analysis for video-based person re-identification. In: Proceedings of 2017 IEEE International Conference on Image Processing (ICIP), 2017. 111–115
29 Xu Z, Jin R, King I, et al. An extended level method for efficient multiple kernel learning. In: Proceedings of Advances in Neural Information Processing Systems 21, 2008. 1825–1832
30 Xia H, Hoi S C H. MKBoost: a framework of multiple kernel boosting. IEEE Trans Knowl Data Eng, 2013, 25: 1574–1586
31 Chen W, Yang K, Shi Y, et al. Kernel-based class-specific broad learning system for software defect prediction. In: Proceedings of 2021 8th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), 2021. 109–114
32 Chen W, Yang K, Zhang W, et al. Double-kernelized weighted broad learning system for imbalanced data. Neural Comput Appl, 2022, 34: 19923–19936
33 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
34 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 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 2019. 193–197
35 Wang Q, Jin P, Xiong H, et al. Broad graph attention network with multiple kernel mechanism. IEEE Trans Emerg Top Comput Intell, 2025, 9: 2296–2307
36 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
37 Ng A. Sparse autoencoder. CS294A Lecture notes, Stanford University, 2011
38 Breiman L. Bagging predictors. Mach Learn, 1996, 24: 123–140
39 Kuncheva L I, Rodriguez J J, Plumpton C O, et al. Random subspace ensembles for fMRI classification. IEEE Trans Med Imag, 2010, 29: 531–542
40 Breiman L. Random forests. Machine Learn, 2001, 45: 5–32
41 Hastie T, Rosset S, Zhu J, et al. Multi-class AdaBoost. Stat Its Interface, 2009, 2: 349–360
42 Frank A. Uci machine learning repository. http://archive.ics.uci.edu/ml, 2010
43 Dinesh N, Joshi A, Lee I, et al. Permission to speak: a logic for access control and conformance. J Logic Algebraic Program, 2011, 80: 50–74
44 Huang G, Siew C. Extreme learning machine with randomly assigned RBF kernels. Int J Inf Technol, 2005, 11: 16–24
45 Jahromi A H, Taheri M. A non-parametric mixture of gaussian naive bayes classifiers based on local independent features. In: Proceedings of 2017 Artificial Intelligence and Signal Processing Conference (AISP), 2017. 209–212
46 Hearst M A, Dumais S T, Osuna E, et al. Support vector machines. IEEE Intell Syst Their Appl, 1998, 13: 18–28
47 Zhang M L, Zhou Z H. ML-KNN: a lazy learning approach to multi-label learning. Pattern Recognit, 2007, 40: 2038–2048