Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 8857818, 16 pages
https://doi.org/10.1155/2021/8857818

Research Article

An Optimized Neural Network Classification Method Based on Kernel Holistic Learning and Division

Hui Wen, Tongbin Li, Deli Chen, Jianlu Yang, and Yan Che

Department of Information Engineering, Putian University, Putian 351100, China

Correspondence should be addressed to Hui Wen; wen_hui81@163.com

Received 9 September 2020; Revised 17 February 2021; Accepted 20 February 2021; Published 28 February 2021

Academic Editor: David Bigaud

An optimized neural network classification method based on kernel holistic learning and division (KHLD) is presented. The proposed method is based on the learned radial basis function (RBF) kernel as the research object. The kernel proposed here can be considered a subspace region consisting of the same pattern category in the training sample space. By extending the region of the sample space of the original instances, relevant information between instances can be obtained from the subspace, and the classifier's boundary can be far from the original instances; thus, the robustness and generalization performance of the classifier are enhanced. In concrete implementation, a new pattern vector is generated within each RBF kernel according to the instance optimization and screening method to characterize KHLD. Experiments on artificial datasets and several UCI benchmark datasets show the effectiveness of our method.

1. Introduction

In the field of pattern recognition, set classification [1–3] is a common classification task. It is widely applied in text classification, speech recognition, image recognition, and multiple other fields. Taking the classification task based on image set as an example, each image set is composed of a class of image frames with a certain number of similar features. Due to the use of relevant information from adjacent frames, image changes can be effectively explored in the actual conditions. The main challenge is how to effectively integrate the information from all existing images to reach a reliable decision. A typical approach is to establish an optimized representation of different subsets of images and to achieve effective measurements between different subsets.

Different from the set classification methods mentioned above, almost all the current neural network [4–7] optimization algorithms and models are based on the training and classification of instances instead of learning and partitioning the subspace region containing those instances. Because the classification surface of the network classifier is essentially determined by the probability distribution of the training samples, if the size of the training sample set is too small or the dimension of the classified dataset is too high, the error in the final classification will be relatively large, which leads to the reduction in the generalization performance of the neural network classifier.

To improve this problem effectively, inspired by the idea of set classification, this paper attempts to introduce the idea of set classification into the neural network and presents an optimized neural network classification method based on kernel holistic learning and division (KHLD), which can improve the performance of the neural network classifier under a given sample set. Different from set classification, the method of KHLD is based on the effective coverage of a local region of the sample space, so the kernel proposed here can be considered a subspace region consisting of the same pattern category in the training sample space. Though it might not obtain the spatial distribution directly, relevant information between instances can be obtained from the subspace. The main reason is that the instances of the same pattern category are relatively close to each other in the spatial distribution and can be considered to have some similarity. Compared to single-pattern vector classification, KHLD considers the similarity information of the local region in the sample space. Due to the expansion of the

region presented by the original pattern vector in the sample space, it can be improved to a certain extent when the size of the sample set is too small or when the dimension of the sample space is too high. On the other hand, KHLD can make the classifier's boundary farther away from that of the original sample, which can further strengthen the robustness and generalization ability of the classifier.

The primary task of achieving KHLD is the establishment and representation of the kernel. In this paper, considering the local characteristics of a certain region covered by the sample space, we take the Gaussian distribution function under different parameters as the representative to establish the corresponding subspace set. Moreover, to integrate the subkernel with different parameters and the mapping effect into the original sample space, we first construct the corresponding RBF kernel by learning the original sample space to realize the local mapping of the different regions of the sample space. Then, RBF kernels with different parameters are further studied and divided. Thus, the KHLD presented in this work has two meanings: the establishment of different RBF kernel parameters and the holistic division of the coverage region.

Typical optimization algorithms for establishing RBF kernels with different parameters include K-mean clustering [8], fuzzy clustering [9, 10], orthogonal forward selection [11], evolutionary algorithm [12], particle swarm optimization [13], and other algorithms [14–16]. It is worth noting that the above methods for optimizing the RBF kernel parameters effectively combine the holistic information of the training sample space, but the number of hidden nodes in the RBF network cannot be determined automatically, which may lead to poor adaptability for different sample sets. To automatically estimate the number of RBF kernel parameters, several sequence learning RBF network kernel parameters, including minimum resource allocation network (MRAN) [17], sequential learning algorithm for growing and pruning the RBF (GAP-RBF) [18], and other incremental design of radial basis function networks [19–21], can be used. However, the holistic information of the sample space is not taken into account in these methods, and the classification performance will be affected to some extent.

To generate the optimal number and parameters of the RBF kernel, in our previous work, an incremental learning algorithm for the hybrid RBF-BP network (ILRBF-BP) [22] and a hybrid structure adaptive RBF-ELM network (HSARBF-ELM) [23] are presented. In ILRBF-BP, the method of potential density clustering is presented to generate RBF kernels automatically, which utilizes the global distribution information of sample space. However, the local adaptability of each RBF kernel parameter in ILRBF-BP is not fully considered; this disadvantage is overcome by HSARBF-ELM. By combining the potential density clustering and the center-oriented heterogeneous sample repulsive force, the density information of different regions of the sample space and the neighborhood information of the region covered by the initial hidden nodes of the RBF can be used effectively. The optimal number and parameters of the RBF kernel can be generated adaptively according to the distribution of the sample space. However, when the size of the training sample set is too small or the dimension of the sample set is too high, the distribution of the sample set will be very sparse, which leads to the failure of the optimization algorithm to some extent, and the generalization performance of the neural network classifier will be reduced. To solve this problem, an optimal neural network based on KHLD is proposed. The premise of the method of KHLD is to establish the optimized kernel parameters. The geometry of these kernels is a regular hypersphere, and the optimization of the number and parameters of RBF kernels in HSARBF-ELM is just in line with this requirement. Thus, the RBF kernel established in HSARBF-ELM is the research object of this study.

When the number and parameters of the optimized RBF kernel are established, the subsequent task is to realize KHLD. In practice, the training of the weights of network classifiers is carried out in a single instance. When all the RBF kernels are established, according to the probability density distribution of the pattern vectors in each subkernel, we consider generating new pattern vectors within each RBF kernel, which is equivalent to extending the existing pattern vector subset in the current RBF kernel, to characterize KHLD. Intuitively, when the number of samples generated in the region covered by the kernel is sufficient, the covered region can be approximated. In this way, the KHLD is transformed for training and dividing more pattern vectors. On the basis of generating a suitable sample set size, the existing network classifier is used for training and classification; thus, the final classification surface can be modified to improve the generalization performance of the network classifier.

To achieve the effective expansion of the pattern vector in the region covered by the RBF kernel, a suitable sample probability distribution model is first needed to generate new pattern vectors. For this problem, we consider that the effective region covered by the RBF subkernel contains a certain number of original pattern vectors. In the region near the center, the probability density is relatively dense, and the probability density near the boundary is relatively sparse; thus, it can be considered that these pattern vectors similarly obey the multivariate Gaussian distribution with the current RBF kernel as the parameter. Moreover, the new pattern vectors should be constrained by the region covered by the current RBF kernel, and the initial filling of the RBF kernel can be accomplished in this way. Second, we need to measure the density of the region of the original pattern vectors in each RBF kernel. In the dense region of the sample space, the number of generated pattern instances is relatively large; conversely, in the sparse region of the sample space, the number of generated pattern instances is relatively small. When the generated instances are in the mixed region covered by different pattern classes, the probability of preserving the sample is further reduced. In this way, by combining the density and location information of the region, the optimal selection of the generated pattern instances can be completed without changing the probability density distribution of the original sample space.

According to the above methods, we take the idea of KHLD as the prototype and approximate the idea of KHLD

by filling and screening the pattern vector of each kernel. On the other hand, the KHLD of each RBF kernel is converted to learning and division of more pattern vectors, which can improve the sparse sample spatial distribution caused by a sample size that is too small or sample space dimension that is too high, and the classification accuracy of the classifier can be enhanced. Note that, due to the inhomogeneity of the sample distribution inside the kernel, the approximation of the idea of KHLD by filling and screening the pattern vector of each kernel can be considered a soft partition; that is, the final classified surface can pass through the kernel to improve the overlap of different pattern subclasses effectively. Thus, it is more conducive to the adjustment of actual classification surface parameters.

In summary, the main contributions of this work are as follows:

(1) The idea of KHLD is introduced into the neural network classifier, and its characteristics are analyzed

(2) The internal sample generation and optimization screening mechanism of the RBF kernel is designed to achieve the approximation of KHLD

(3) The performance of KHLD is combined with existing classification algorithms and compared with these algorithms in two artificial datasets and several benchmark datasets, and the experimental results show the superiority of the proposed method

2. Methods

2.1. The Establishment of KHLD. Considering that the method of KHLD is based on the RBF kernel of HSARBF-ELM, here, we give the optimization of RBF kernels in HSARBF-ELM, which is ready for the optimization method of kernel holistic learning and division.

For the input sample x, when it passes through the RBF kernel function, its output can be expressed as

$$\varphi_k(\mathbf{x}) = \exp\left(-\frac{1}{2\sigma_k^2}\left\|\mathbf{x} - \mu_k\right\|^2\right), \tag{1}$$

where μ_k and σ_k are the center and width of k-th RBF kernels.

In HSARBF-ELM, by combining the methods of density clustering with a potential function and center-oriented unidirectional repulsive force, the numbers of RBF kernels and parameters can be effectively generated. The main methods are as follows.

Given a training set S, where $S = \cup_{i=1}^{h} S_i$, S_i is the i-th pattern category set, $S_i = \left\{x_1^i, \ldots, x_{L_i}^i\right\}$, h is the number of pattern categories, and L_i is the number of samples in the i-th pattern analogy, for each pattern category set S_i:

(1) Compute the potential value of $\mathbf{x}_\mathbf{v}^\mathbf{i}$ according to

$$\rho\left(\mathbf{x}_\mathbf{v}^\mathbf{i}\right) = \sum_{u=1,u\neq v}^{N_i} \frac{1}{1 + T \cdot d^2\left(\mathbf{x}_\mathbf{u}^\mathbf{i}, \mathbf{x}_\mathbf{v}^\mathbf{i}\right)}, \quad v = 1, 2, \ldots, N_i, \tag{2}$$

where T is the distance weighting factor and $d(\mathbf{x}_\mathbf{u}^\mathbf{i}, \mathbf{x}_\mathbf{v}^\mathbf{i})$ is the distance measure between $\mathbf{x}_\mathbf{u}^\mathbf{i}$ and $\mathbf{x}_\mathbf{v}^\mathbf{i}$.

(2) Determine the sample with the maximum potential as the center of the hidden nodes of the generated RBF, and set the sample with the maximum potential to be $\mathbf{x}_\mathbf{p}^\mathbf{i}$; the corresponding expression is as follows:

$$\begin{aligned} \rho\left(\mathbf{x}_\mathbf{p}^\mathbf{i}\right) &= \max\left\{\rho\left(\mathbf{x}_1^\mathbf{i}\right), \rho\left(\mathbf{x}_2^\mathbf{i}\right), \ldots, \rho\left(\mathbf{x}_{\mathbf{N_i}}^\mathbf{i}\right)\right\}, \\ \mu_k &= \mathbf{x}_\mathbf{p}^\mathbf{i}. \end{aligned} \tag{3}$$

(3) Adjust the center

$$\begin{aligned} \mu_k' &= \mu_k + \frac{1}{M}\sum_{q=1}^{M_j} \mathbf{F}_{\mathbf{x}_\mathbf{q}^\mathbf{j}} \\ s.t.\ M_i' &\geq M_i, \\ M_j' &\leq M_j, \end{aligned} \tag{4}$$

where

$$\mathbf{F}_{\mathbf{x}_\mathbf{q}^\mathbf{j}} = \exp\left(-\alpha \cdot d\left(\mathbf{x}_\mathbf{q}^\mathbf{j}, \mu_k\right)\right) \cdot \frac{\mathbf{x}_\mathbf{q}^\mathbf{j} - \mu_k}{\left\|\mathbf{x}_\mathbf{q}^\mathbf{j} - \mu_k\right\|}, \tag{5}$$

where $\mathbf{F}_{\mathbf{x}_\mathbf{q}^\mathbf{j}}$ denotes the heterogeneous repulsive force from $\mathbf{x}_\mathbf{q}^\mathbf{j}$ to μ_k, $\mathbf{x}_\mathbf{q}^\mathbf{j}$ is a heterogeneous sample covered by the current hidden nodes of the RBF:$\mathbf{x}_\mathbf{q}^\mathbf{j} \notin S_i$ and $\|\mathbf{x}_\mathbf{q}^\mathbf{j} - \mu_\mathbf{m}\| < \lambda \cdot \sigma$, where λ is the width covering factor, α is the repulsive force control factor, and M is the iteration step. M_i and M_j denote the number of samples covered by the current hidden nodes of the RBF before updating. M_i' and M_j' denote the number of samples covered by the current RBF hidden nodes after updating.

(4) The width is adjusted as follows:

$$\sigma_k = \begin{cases} \max\left\{\dfrac{\min\ d\left(\mu_k', \mathbf{x}_\mathbf{q}^\mathbf{j}\right)}{\beta}, \sigma_{\min}\right\}, & \text{if } M_j' > 0, \\ \sigma, & \text{if } M_j' = 0, \end{cases} \tag{6}$$

where β is the width constraint factor, $\sigma_{\min}$ is the constrained minimum width parameter, and σ is the initial width. This adjustment ensures the relative diversity of each generated RBF hidden node, which can achieve a balance between the coverage effect and the generalization performance.

(5) Counteract each sample potential of the region covered by the current RBF hidden node and find the sample with the maximum potential to generate the next RBF hidden node

$$\rho'\left(\mathbf{x}_\mathbf{n}^\mathbf{i}\right) = \rho\left(\mathbf{x}_\mathbf{n}^\mathbf{i}\right) - \rho\left(\mathbf{x}_\mathbf{p}^\mathbf{i}\right) \cdot \exp\left(-\frac{1}{2\sigma_k^2}\left\|\mathbf{x}_\mathbf{n}^\mathbf{i} - \mathbf{x}_\mathbf{p}^\mathbf{i}\right\|^2\right), \quad n = 1, 2, \ldots, N_i, \tag{7}$$

where $\rho'(\mathbf{x}_{\mathbf{n}}^{\mathbf{i}})$ is the updated potential value of $\mathbf{x}_{\mathbf{n}}^{\mathbf{i}}$.

(6) Set the iteration termination condition as follows:

If $\max\{\rho'(\mathbf{x}_1^{\mathbf{i}}), \rho'(\mathbf{x}_2^{\mathbf{i}}), \ldots, \rho'(\mathbf{x}_{\mathbf{N_i}}^{\mathbf{i}})\} > \delta$
Go to Steps 2-4.
Else
The process of learning the current pattern category is complete. Go to learn other pattern categories.
EndIf

According to the above steps, the number of RBF hidden nodes and the center and width, which can be denoted as $\{K, \mu_k, \sigma_k\}_{k=1}^{K}$, can be generated optimally. For HSRBF-ELM, once the optimized RBF hidden nodes are generated, the output $g(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_j(\mathbf{x}), \ldots, g_K(\mathbf{x}))$ can be the input of the subsequent ELM network. The update of the ELM network weights is based on the existing ELM [24] learning algorithm.

2.2. The Method of KHLD

2.2.1. Main Idea. To explain the characteristics and advantages of the method based on KHLD, Figure 1 gives a diagram of the direct classification and comparison of KHLD and direct pattern vector classification. The method of KHLD is transformed for training and dividing more pattern vectors.

To realize KHLD, it is necessary to establish a suitable RBF kernel to complete the effective coverage of the different regions in the original sample space. Then, to ensure the validity of the generated samples, the newly generated samples in the kernel should be approximately consistent with the original pattern vector distribution, and the number of newly generated samples should be proportional to the distribution density of the original sample region. In addition, when the kernels of different pattern categories overlap, it is necessary to further screen the generated pattern vector in overlapping regions. To this end, the following steps need to be completed:

Step 1: the optimal coverage of the original sample space is completed by the potential function density clustering and center-oriented heterogeneous sample repulsive force; the appropriate RBF kernel parameters, including the number, center, and width of the RBF kernels, can be determined adaptively according to the distribution of the sample space

Step 2: with the center and the width of each RBF kernel as constraints, a probability distribution similar to the original sample is set up to generate a new pattern vector in the effective region covered by each RBF kernel

Step 3: the newly generated pattern vector is judged to determine whether it is retained or not, and finally, a new pattern vector subset is formed

Step 4: w new set of samples is formed by combining the original sample with all the screened pattern vectors that are eventually retained to train the weights of the output classifier

The difficulty of realizing the above steps lies in Step 3, that is, to establish the appropriate standard to measure the relationship between the newly generated pattern vector and the original sample density and to determine whether the kernels with different pattern categories overlap each other to complete the optimization screening of the newly generated pattern vector.

2.2.2. The Implementation. In this section, we first give the definitions of KHLD, overlapping region samples, and nonoverlapping region samples to prepare for the description and implementation of subsequent algorithms.

(1) Definition.

KHLD. Training and partitioning labeled RBF kernels after covering the original sample space

Overlapping Region Samples. The samples in the overlapped region are covered by different pattern categories of RBF kernels

Nonoverlapping Region Samples. Samples outside each overlapped region

According to the definition, Figure 2 gives the schematic diagrams of the overlapping region samples and the nonoverlapping region samples, which represent the valid regions covered by two different RBF kernels. In Figure 2(b), $C = A \cap B$, where C is the overlapping region, sample 1 and sample 2 are the overlapping region samples, and the other samples are nonoverlapping region samples.

To realize the classification method based on the kernel holistic division and the selection of generated samples, it is necessary to establish each RBF kernel as the research object and randomly generate the pattern category samples within each kernel to further optimize the screening process. To this end, two factors need to be considered:

(1) To facilitate the optimization of subsequent generated pattern samples, the probability distribution of the initial generated pattern samples should be approximately the same as that of the original sample.

(2) In the process of sample screening, the probability of the generated sample being retained should be proportional to the density of the original sample region. It is also necessary to consider whether the sample is an overlapping region sample and, if so, further reduce the probability that the sample is retained.

For case (1), since the establishment of each RBF kernel parameter is based on the potential function density clustering, overall, the probability density of the region near the center of the original sample is relatively large, and the probability density of the region near the boundary of the original sample is relatively small, it can be considered that the probability density of pattern vectors in these kernels approximately obeys a multivariate Gaussian distribution with the current RBF kernel as the parameter, and it can be taken as the new pattern vector probability distribution model.

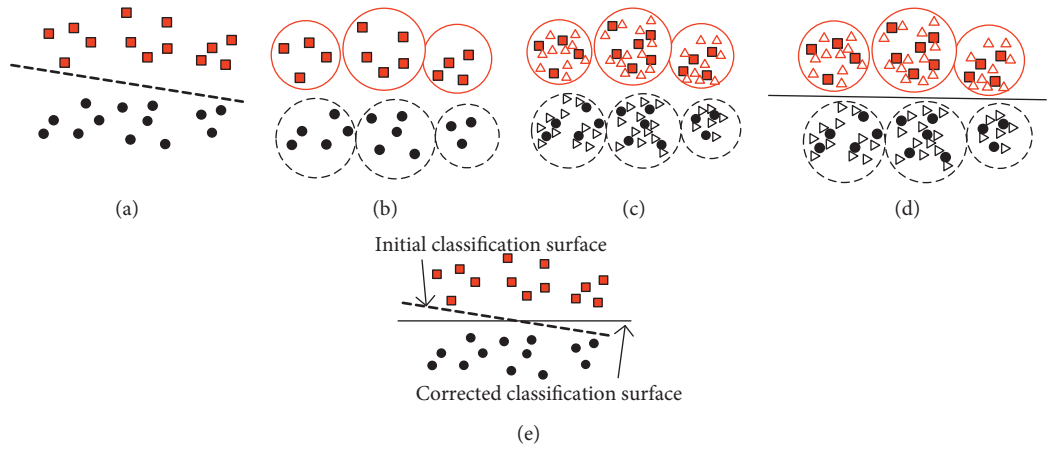


Figure 1: A schematic comparison between the kernel holistic partition and the direct pattern vector classification. (a) Directly partitioning the original sample set. (b) Density clustering of the original sample set and establishing the corresponding RBF kernels to complete the coverage of the original sample space. (c) Filling each subkernel pattern class to establish a new pattern vector to partition the whole kernel. (d) Dividing the original sample and the new filled sample into new sample sets to obtain a new classification surface. (e) Comparing the modified classification surface with the original one.

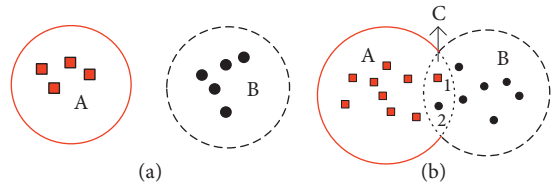


Figure 2: The schematic diagrams of distribution form of different kernels. (a) Nonoverlapping region samples. (b) Overlapping region samples.

For case (2), the key is to establish an appropriate measure to determine the density of the region where each generated sample is located and determine whether the generated sample is retained. If the generated sample is retained, it is necessary to determine whether the generated samples are in the overlapping region and further complete secondary optimization.

According to the above description, given a dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{L}$, where L is the number of training samples, y_i is the category labels$\mathbf{x}_i \in \mathbf{R}^l$, and $y_i \in \mathbf{R}^h$, let S_i be the training sample set of the i-th pattern category, $S_i = \left\{\mathbf{x}_1^{\mathbf{i}}, \mathbf{x}_2^{\mathbf{i}}, \ldots, \mathbf{x}_{L_i}^{\mathbf{i}}\right\}$; here, $S = \cup_{i=1}^{h} S_i, S_i \cap S_j = \varnothing, \ \forall i \neq j$. For each training sample category, the number and parameters of the RBF kernels are optimized by the potential function density and the repulsive force between heterogeneous samples, expressed as $\{\mu_k^i, \sigma_k^i, t_k^i\}_{k=1}^{K_i}$, where μ_k^i, σ_k^i are the center and width of the kth RBF kernel, respectively, t_k^i is the $i$thpattern category label of the RBF kernel, K_i is the number of RBF kernels generated under each pattern category, and $K = \sum_{i=1}^{h} K_i$ is the number of RBF kernels.

When all the RBF kernels are built, the effective coverage of the different regions of the original sample space is completed. To achieve sample filling for each RBF kernel, it is necessary to establish a suitable sample probability distribution model to generate new pattern vectors. For the current kth RBF kernel, the probability distribution $f(\mathbf{z})$ for generating arbitrary pattern vectors $\mathbf{z}$obeys the Gaussian distribution with μ_k^i being the mean and $\Sigma_k^i = \mathrm{diag}(\sigma_k^i, \sigma_k^i, \ldots, \sigma_k^i)$ being the variance matrix; that is, $\mathbf{z} \sim N(\mu_k^i, \Sigma_k^i)$. Moreover, the newly generated pattern vectors should be in the effective region covered by the RBF kernel, which is given by

$$f(\mathbf{z}) = \frac{1}{\sqrt{2\pi}\sigma_k^i} \exp\left(-\frac{\left\|\mathbf{z} - \mu_k^i\right\|^2}{2\sigma_k^2}\right) \quad s.t. \left\|\mathbf{z} - \mu_k^i\right\| \leq \sigma_k^i. \tag{8}$$

According to the above method, for the kth RBF kernel in the ith pattern category, let W_k^i be the generated initial vector set in the kernel; here, $W_k^i = \left\{z_1^i, z_2^i, \ldots, z_{\mathrm{N}_k}^i\right\}$, $N_k$is the number of generated samples in the k-th kernel. After the initial pattern vectors are generated, they need to be optimized and screened. During the screening process, in the dense region of the sample space, the number of generated pattern instances should be relatively large; conversely, in the sparse region of the sample space, the number of generated pattern instances should be relatively small. In this way, the probability distribution of the sample space can be combined with the density of the region where the pattern

vector is generated, and the validity of the resulting pattern vector can be enhanced.

Let C_k^i be the initial sample set of the $k$thRBF kernel in the current $i$thpattern category and P_k be the number of C_k^i. For each initial pattern vector $\mathbf{x}$, when $\mathbf{x} \in S_i$ and $\|\mathbf{x} - \mu_k^i\| \le \sigma_k^i$, then $\mathbf{x} \in C_k^i$. Thus, $C_k^i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_{\mathbf{P}_k}^i\}$. For each generated pattern vector $z_m^i (1 \le m \le N_k)$ in W_k^i, the probability density of each new pattern vector can be estimated, which is given as

$$\widehat{p}\left(z_m^i\right) = \frac{1}{P_k} \sum_{j=1}^{P_k} \frac{1}{(2\pi)^{l/2}\theta_k^l} \exp\left(-\frac{\left\|z_m^i - \mathbf{x}_\mathbf{j}^\mathbf{i}\right\|^2}{2\theta_k^2}\right), \quad (9)$$

where θ_k is the width of the corresponding Parzen window in the k-th RBF kernel.

To achieve this metric while preserving the randomness of sample generation, we consider generating a uniformly distributed random number r between 0 and 1, which is used to for comparison with the probability density of each newly generated pattern vector. If $r \le \widehat{p}(z_m^i)$, z_m^i is retained; otherwise, z_m^i is eliminated. Therefore, in the region where the original samples are relatively densely distributed, the probability that the newly generated samples will be retained is relatively high.

Due to the complexity of different sample sets, heterogeneous samples are often mixed into the generated RBF kernel. Thus, it is necessary to further improve the sample screening in the overlapping region. When the generated sample is in the overlapping region, two factors need to be considered:

(1) The probability of the sample being retained should be reduced.

(2) It is necessary to consider the sample spatial distribution density under the current pattern category and other pattern categories at the same time. According to the principle of inhibiting the probability density of heterogeneous samples, when the spatial distribution density of the sample in the current pattern category is higher than that in other pattern categories, the probability of the sample being retained is relatively large.

Combined with the above two factors, for the sample z_m^i generated in the kth RBF kernel, we can get $\|z_m^i - \mu_k^i\| \le \sigma_k^i$. Moreover, when $\|z_m^i - \mu_n^j\| \le \sigma_n^j$ is satisfied, z_m^i can be considered the sample in the overlapping region between the k-th and the n-th RBF kernel.

When the samples in the overlapping region are determined, it is necessary to further screen the samples. Let C_n^j be the initial sample set contained in the j-th pattern category by the kernel $\{\mu_n^j, \sigma_n^j\}$, and set the number of samples in C_n^j as P_n. For an arbitrary pattern vector $\mathbf{x}$, when $\mathbf{x} \in S_j$ and $\|\mathbf{x} - \mu_n^j\| \le \sigma_n^j$ are satisfied, $\mathbf{x} \in C_n^j$. Thus, $C_n^j = \{\mathbf{x}_1^\mathbf{j}, \mathbf{x}_2^\mathbf{j}, \ldots, \mathbf{x}_{\mathbf{P}_n}^\mathbf{j}\}$. For the sample $\mathbf{z}_\mathbf{m}^\mathbf{i}$ in the overlapping region intersecting the k-th and the n-th RBF kernel, the probability density estimations of the heterogeneous sample regions are expressed as

$$\widehat{q}\left(\mathbf{z}_\mathbf{m}^\mathbf{i}\right) = \frac{1}{P_n} \sum_{n=1}^{P_n} \frac{1}{(2\pi)^{l/2}\theta_k^l} \exp\left(-\frac{\left\|\mathbf{z}_\mathbf{m}^\mathbf{i} - \mathbf{x}_\mathbf{n}^\mathbf{j}\right\|^2}{2\theta_k^2}\right). \quad (10)$$

According to the above method, for a randomly generated number r between 0 and 1, when $\widehat{p}(\mathbf{z}_\mathbf{m}^\mathbf{i}) \ge \lambda r$ and $\widehat{p}(\mathbf{z}_\mathbf{m}^\mathbf{i}) \ge \gamma \widehat{q}(\mathbf{z}_\mathbf{m}^\mathbf{i})$, the sample $\mathbf{z}_\mathbf{m}^\mathbf{i}$ in the overlapping region is retained; otherwise, $\mathbf{z}_\mathbf{m}^\mathbf{i}$ is removed. Here, $\lambda > 1$ and $\gamma \ge 1$.

Combined with the above description, Algorithm 1 gives the concrete implementation of the classification method based on kernel holistic learning and kernel interior sample generation.

2.2.3. The Computational Complexity Analysis of KHLD. In this study, a method of the potential density clustering and the center-oriented heterogeneous sample repulsive force is used to generate optimized kernel parameters. Then, a method of optimized sample filling and screening can realize the effective approximation of KHLD. Assume that the number of samples in the initial training set S is L, and the initial training set contains two pattern categories; the number of samples are L_1 and L_2, respectively. Here, $L_1 + L_2 = L$. The computational complexity of the proposed method is analyzed as follows:

(1) The optimal kernel parameters are generated by the combination of potential density clustering and heterogeneous sample repulsive force. In the process of quantifying the sample potential value by potential function density clustering, the label information of each category of samples is considered. The calculation of the sample potential value needs to traverse all other samples in the current pattern category. Then, Gaussian kernels with different parameters are needed to cover the sample subspace to update the sample potential. The computational complexity is $O((L_1 - 1)^2 + (L_2 - 1)^2)$. Set the number of kernels as K; in the process of optimizing the kernel parameters, the distance between all samples and the center should be considered; the computational complexity is $O(LK)$. After merging, the computational complexity of this part is $O(L^2 - 2L_1L_2 - 2L + LK)$.

(2) The process of sample generation and screening will also take a certain amount of time. Let the number of samples generated in all kernels be P, where the number of samples generated in the k-th kernel is P_k; thus, $P = \sum_{k=1}^{K} P_k$. In the process of calculating the density measurement of the generated sample, the distance between the generated sample and the center of the current kernel should be considered; here, the computational complexity of the generated sample in the k-th kernel is $O(P_k)$. The computational complexity of all kernel generated samples is combined, which can be expressed as $O(\sum_{k=1}^{K} P_k)$; here, $O(\sum_{k=1}^{K} P_k) = O(P)$. Then, in the process of sample screening, we need to further consider whether the generated samples in the current kernel are overlapping region samples, which requires us to

```
Initialization;
for i = 1: h % h is the number of pattern categories
  for k = 1: K_i
    Count the number of initial samples P_k belonging to the ith pattern category covered by each RBF hidden node;
    Use (8) to generate a sample set W_k^i = {z_1^i, z_2^i, ..., z_{N_k}^i} and count the number of generated samples N_k;
    for m = 1: N_k % Screening of generated samples according to the density
      Use (9) to estimate the probability density p̂(z_m^i) belonging to the current ithpattern category;
      r = rand(1);
      if p̂(z_m^i) < r
        C_k^i = C_k^i − {z_m^i};
      end if
    end for
    updateN_k;
  end for
for m = 1: N_k %further screening of the overlapping region samples
  for n = 1: K − K_i
    if ‖z_m^i − μ_n^j‖ < σ_n^j
      Use (10) to estimate the probability density q̂(z_m^i) belonging to the jth pattern category;
      if p̂(z_m^i) ≤ q̂(z_m^i) ‖ p̂(z_m^i) ≤ γr
        C_k^i = C_k^i − {z_m^i};
      end if
    end if
  end for
end for
end for
```

Algorithm 1: Kernel holistic learning and kernel interior sample generation.

compare the distance between these samples and all other centers, and the computational complexity is $O((K-1)P_k)$. The computational complexity of all kernel screening samples is combined, which can be expressed as $O(\sum_{k=1}^{K}(K-1)\cdot P_k)$. Thus, the computational complexity of sample generation and screening in all established kernels is $O(P)+O(\sum_{k=1}^{K}(K-1)\cdot P_k)$, which can be simplified as $O(KP)$.

Combined with Steps 1 and 2, the computational complexity of the proposed KHLD is $O(L^2-2L_1L_2-2L+LK+KP)$. Then, the generated training samples and the original training samples are combined to complete the training of the existing algorithms.

3. Results and Discussion

In this section, the performance of KHLD is evaluated with two artificial datasets: Double Moon (DM) [25] and Concrete Circle (CC); 8 UCI benchmark datasets [26]: Blood, Climate, Heart Disease (HD), Sonar, SPECT Heart (SH), Image Segmentation (IS), Forest, and Wilt; and 1 LIBSVM benchmark dataset [27]. Figure 3 shows the graphical display of two artificial datasets. Except for the DM, CC, and IS datasets, all benchmark datasets are imbalanced datasets. In each dataset, the inputs to all the classifiers are scaled to appropriately [−1, 1]; the classification performance of each network is measured by the overall (η_o) and average (η_a) per-category classification accuracies [23]. Table 1 gives the description of the classification datasets.

The performance of KHLD is combined with existing classification algorithms and compared with these algorithms, including SVM [27], ELM [24], HSARBF-ELM, a constrained optimization method based on BP neural network (CO-BP) [28], and an optimized RBF network based on fractional order gradient descent with momentum (FOGDM-RBF) [29]. For SVM, the simulations are implemented with LIBSVM [27]. All these simulations are conducted in MATLAB R2013b running on a PC with 3.2 GHz CPU and 4G RAM. Each algorithm is conducted in 20 trials.

3.1. Artificial Datasets: DM and CC. In this section, two artificial datasets are used to verify the graphical and intuitive characteristics of KHLD. In the phase of classification performance comparison, KHLD is combined with HSARBF-ELM and compared with HSARBF-ELM. Figures 4(a)–4(d) give a comparison of the learning and classification effects based on the original training set and the KHLD under the DM dataset. It can be seen that the RBF kernel generated in HSARBF-ELM can effectively cover the sample space. The combination of KHLD and HSARBF-ELM can fill the training sample space and effectively improve the classification performance of the method of HSARBF-ELM.

Figures 5 and 6 show the optimization effect of the kernels and the samples generated in each kernel after adjusting parameters σ and θ_k, respectively, which shows that the adjustment of parameters σ and θ_k has good adaptability to the sample space on DM dataset.

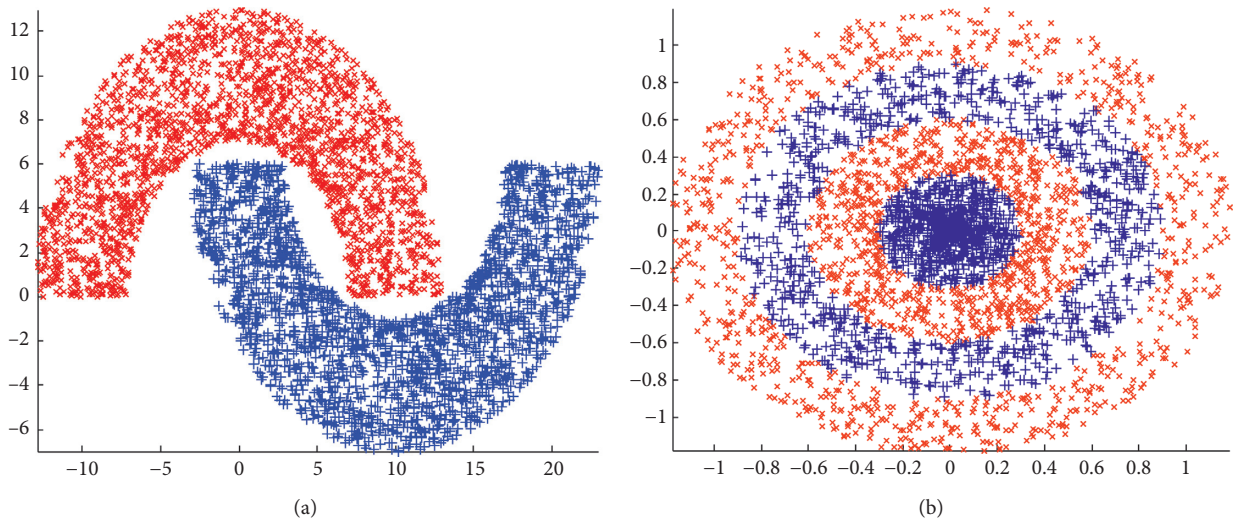


Figure 3: Artificial datasets classification problem. (a) Double Moon. (b) Concrete Circle.

Table 1: Descriptions of the classification datasets.

Datasets	No. of features	No. of classes	No. of samples		
			Training	Validation	Testing
DM	2	2	50–350	200	3000
CC	2	2	50–350	200	3000
Blood	4	2	374	187	187
Climate	18	2	270	135	135
HD	13	2	151	76	76
Sonar	60	2	104	52	52
SH	22	2	80	47	140
IS	19	7	210	1000	1100
Forest	27	4	198	100	225
Svmguide1	6	2	3089	1500	2500
Wilt	4	2	3000	1339	500

Figure 7 compares the number of samples generated and the classification accuracy under different initial training sets, where $\theta_k = \delta_k/10$. It can be seen that, under the condition of small number of training samples, when the initial kernel width is too small, the established kernel cannot effectively cover the sample space, which leads to the decline of network generalization performance; when the kernel width is large and the number of training samples is sufficient, the performance of the proposed method will also show a certain degree of decline, which shows that the method of KHLD has certain restrictions on the number of training samples and the selection of kernel width parameter.

Figure 8 shows the learning and classification comparison of the HASRBF-ELM network classifier based on the original training set and KHLD under the CC dataset. It can be seen that when the generated kernels of different categories overlap each other seriously, the proposed method can still generate new samples in different kernels and improve the classification performance of the original HASRBF-ELM network classifier, which shows the effectiveness of KHLD method for complex classification problems.

Figure 9 shows the learning effect of the proposed method on the training set as the initial kernel width varies. By changing different kernel width parameters, the method of KHLD can optimize the selection of samples in each kernel. When the kernel width increases, the generated kernels may cover the heterogeneous samples, resulting in the increase of the overlapping of the samples of different pattern categories in the kernel.

Figure 10 further shows the number of generated training samples and the performance comparison of the classification accuracy under different initial training sets, where $\theta_k = \delta_k/5$. When the width parameters of the RBF kernels are in a certain range, the method of KHLD has a good classification effect. Similar to Figure 7(b), when the initial kernel width is too small, the testing accuracy of the proposed method is greatly reduced, which means that the failure of the initial RBF kernel may invalidate the method of kernel holistic learning and further deteriorate the final classification performance. Thus, it is necessary to avoid such a situation. This condition is also a restrictive condition for KHLD in this study.

Figures 11 and 12 show that the combination of KHLD and HSARBF-ELM increases the training time. However, the proposed method improves the network classification performance of HSARBF-ELM, especially when the number of

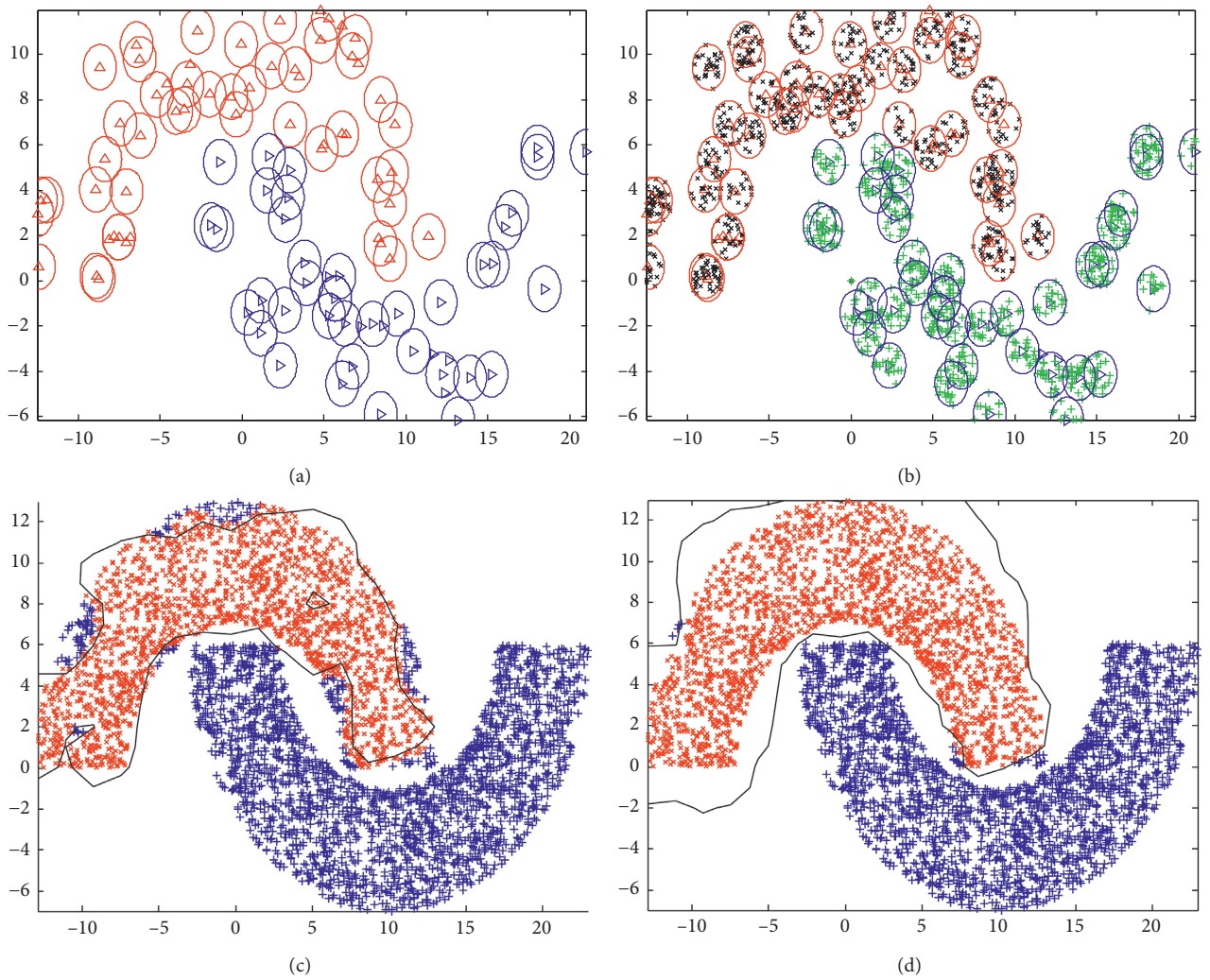


Figure 4: The learning and classification comparison of the HSARBF-ELM network classifier based on the original training set and kernel holistic learning and division in the DM dataset, where the number of original training sets is 100 and the initial kernel width is 0.1. (a) Learning the original training set to generate different RBF kernels. (b) Further learning and screening on the basis of each RBF kernel to generate new sample vectors. (c) Classification effect obtained by learning the parameters of the classifier using the original training set. (d) Classification effect obtained by merging the original sample set with the newly generated sample set and learning the classifier parameters.

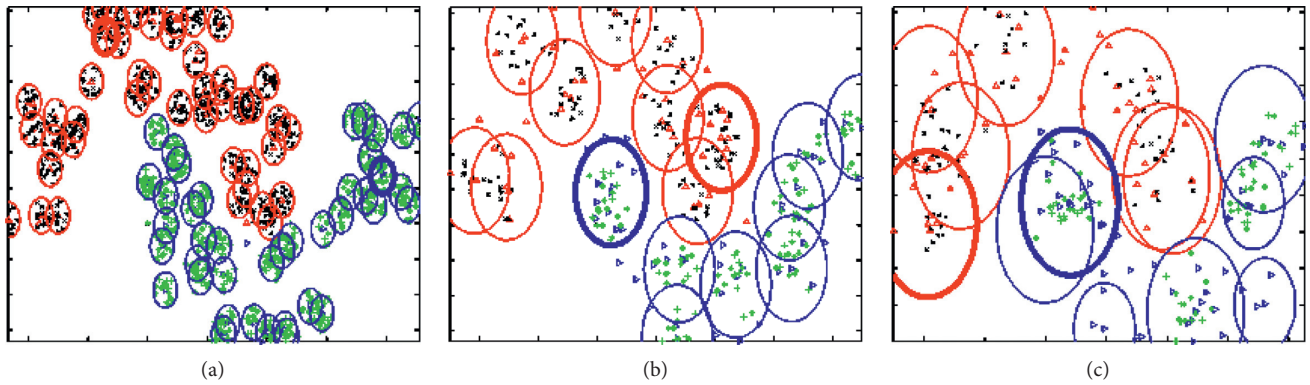
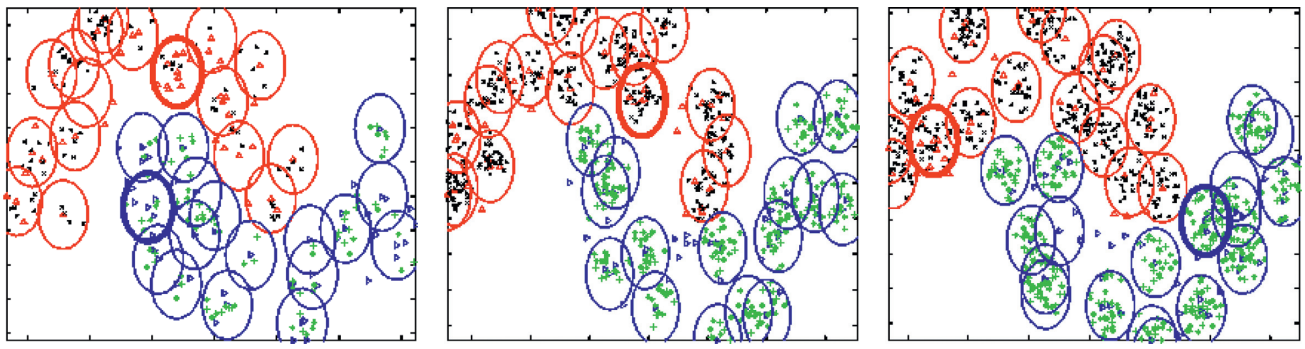
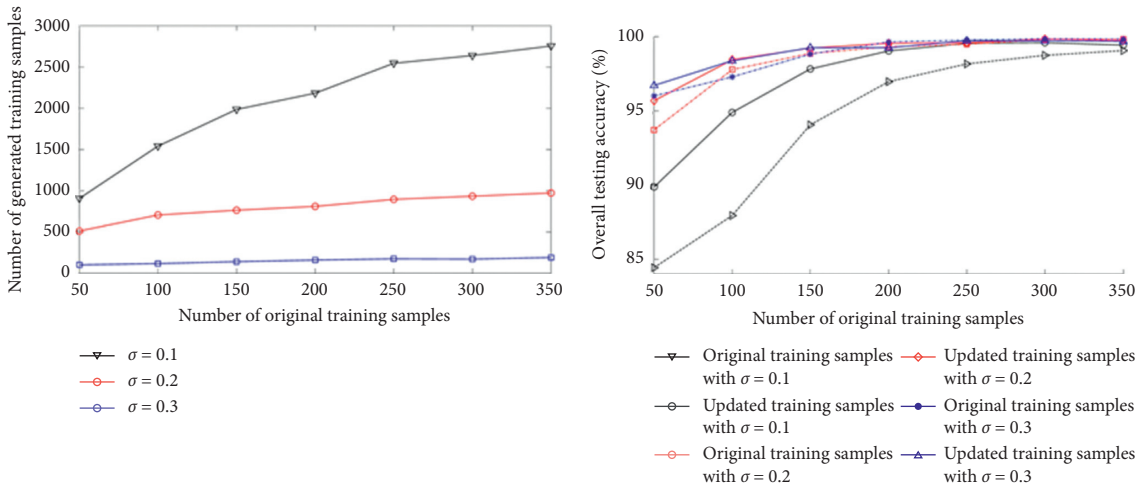


Figure 5: The effect comparison of kernel holistic learning when the number of original training samples is 100 and the initial kernel width takes different parameters. (a) $\sigma = 0.1$. (b) $\sigma = 0.3$. (c) $\sigma = 0.4$.



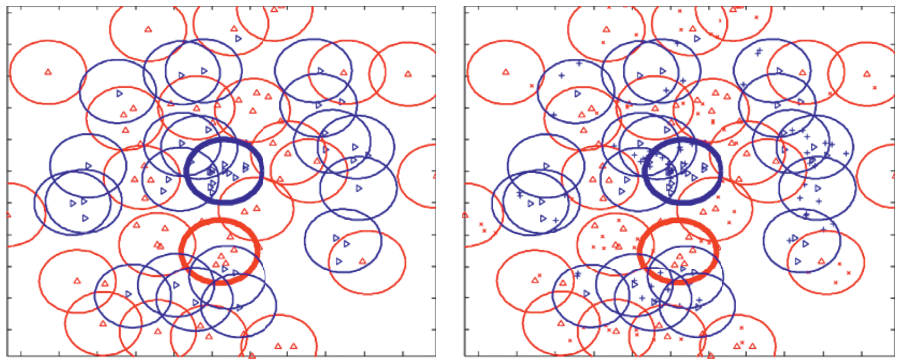
(a) (b) (c)

Figure 6: Comparison of the kernel holistic learning effect under different Parzen window width parameters. (a) $\theta_k = \sigma_k/5$. (b) $\theta_k = \sigma_k/10$. (c) $\theta_k = \sigma_k/20$.



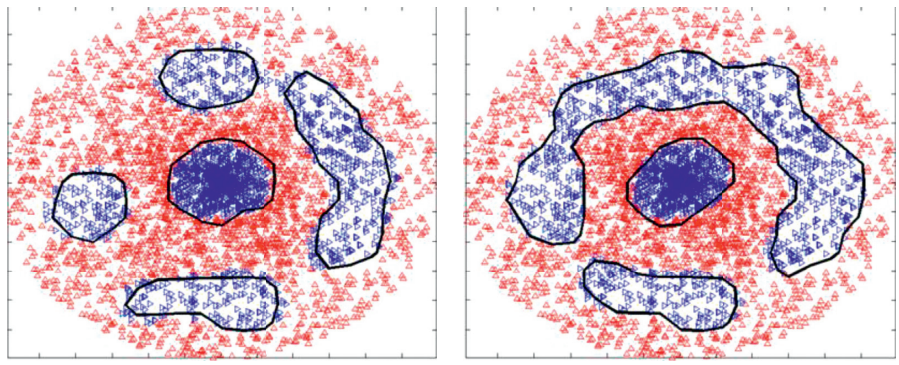
(a) (b)

Figure 7: Comparison of the number of training samples and the classification accuracy under different initial training sets in the DM dataset. (a) Number of original training samples, number of generated training samples. (b) Number of original training samples, overall testing accuracy (%).



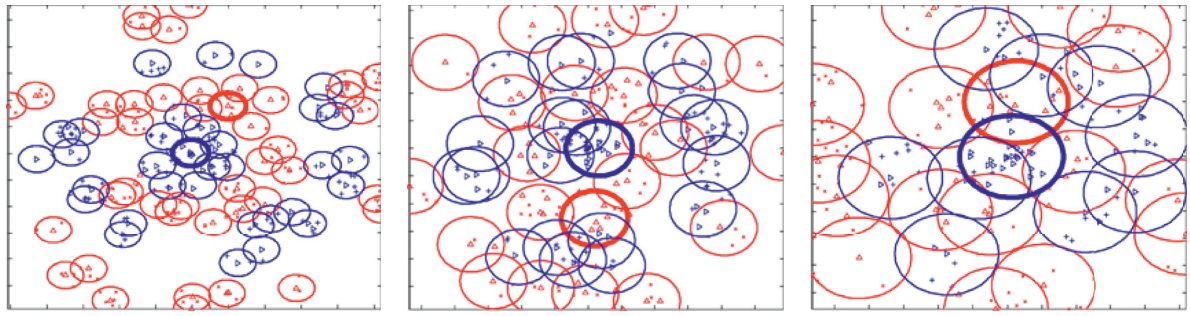
(a) (b)

Figure 8: Continued.



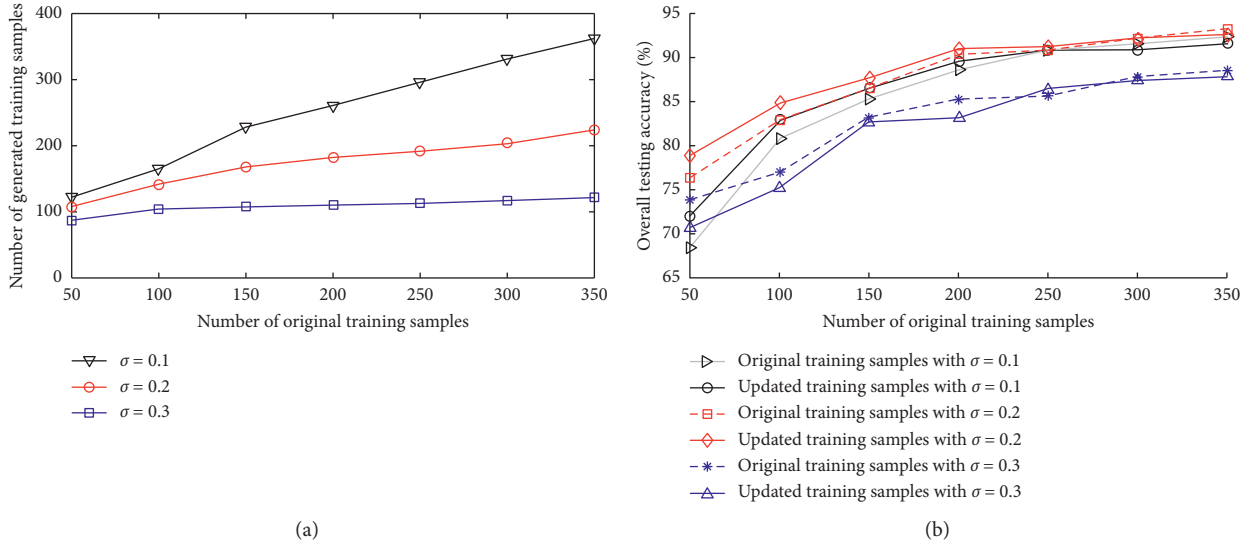
(c) (d)

Figure 8: Sample generation of the original training set under the CC dataset and the comparison of the classification results on the test set. (a) Learning the original training set to generate the subkernel. (b) Further learning and screening to generate new sample vectors. (c) Classification results obtained by using the original training set to learn the classifier parameters. (d) Classifying the original training set with the newly generated training set and then learning the classifier parameters.



(a) (b) (c)

Figure 9: Comparison of the effect of sample generation when the number of original training samples is 100 and the initial kernel width is different. (a) $\sigma = 0.1$. (b) $\sigma = 0.2$. (c) $\sigma = 0.3$.



(a) (b)

Figure 10: Comparison of the classification performance and the number of generated training samples on the CC dataset. (a) Number of original training samples, number of generated training samples. (b) Number of original training samples, overall testing accuracy (%).

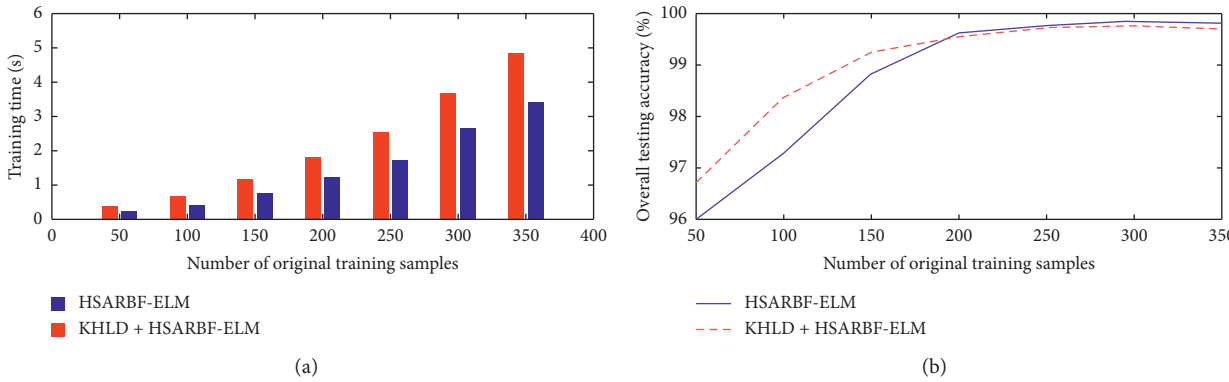


Figure 11: Performance comparison of the proposed method and HSARBF-ELM in the DM dataset. (a) Number of original training samples, training time. (b) Number of original training samples, overall testing accuracy (%).

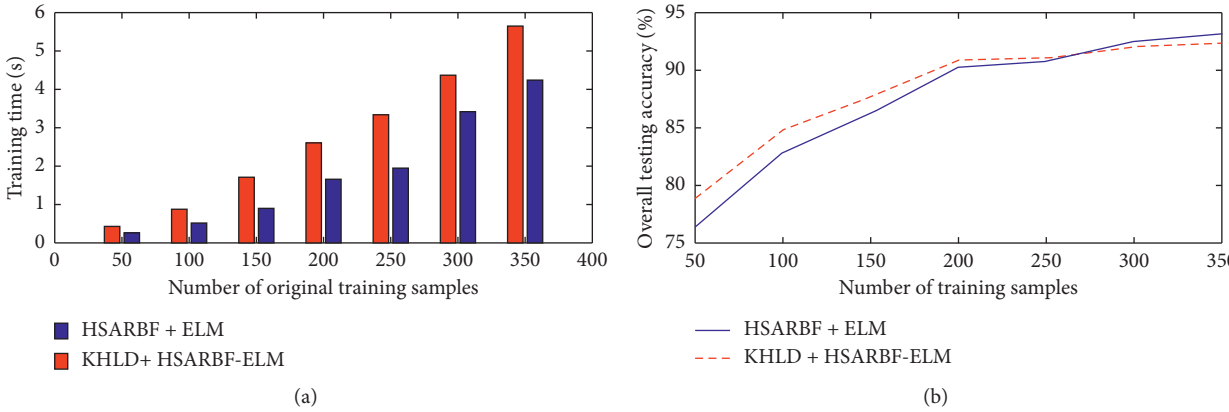


Figure 12: Performance comparison of the proposed method and HSARBF-ELM in the CC dataset. (a) Number of original training samples, training time. (b) Number of original training samples, overall testing accuracy (%).

training samples is small. When the number of training samples is sufficient, the proposed method will reduce the performance of HSARBF-ELM to a certain extent, which shows that this method of KHLD is suitable for the situation with less training samples or sparse spatial distribution of samples.

3.2. UCI Benchmark Datasets. Tables 2 and 3 give the comparisons of the classification performance of the proposed method and other learning algorithms under the benchmark sample datasets. It can be seen that, in high-dimensional small sample datasets, the combination of KHLD and other classification algorithms increases the training time. Although the testing results of different classification algorithms on different datasets are different, the combination of KHLD and these classification algorithms can improve the testing accuracy of these algorithms to varying degrees. As an auxiliary method, KHLD is an effective method when the spatial distribution of samples is sparse. The effectiveness of the proposed method can be further verified. However, for the benchmark large sample datasets, the combination of KHLD and existing algorithms reduces the test performance of these algorithms, which further shows that the method of KHLD in this study is not suitable for large sample set learning and classification.

3.3. Discussion of KHLD. In this study, under the premise of the given initial kernel width σ, according to the optimization method of RBF kernel parameters in HSARBF-ELM, the parameters $\{K, \mu_k, \sigma_k\}_{k=1}^{K}$ of KHLD are automatically generated according to the distribution of sample space, where σ_k is chosen in $\sigma - 0.2 \le \sigma_k \le \sigma$. When each kernel parameter is established, the main parameter affecting KHLD is θ_k, which determines the number of samples generated in the kernel. θ_k is chosen in the $[\sigma_k/5, \sigma_k/10, \sigma_k/20]$. Thus, we mainly discuss the influence of parameters of σ and θ_k on KHLD. Figure 13 shows the stress test when KHLD is combined with HSARBF-ELM in the Climate high-dimensional dataset. In general, when σ and θ_k are in a certain range, the combination of KHLD and HSARBF-ELM can improve the network performance of HSARBF-ELM. When σ is too low, for example, σ is set as 0.1 or 0.2, the classification performance of KHLD combined with HSARBF-ELM is poor. The main reason is that the

Table 2: Performance comparison on benchmark small sample datasets.

Datasets	Methods	Training time (s)	Testing		No. of kernels
			η_o	η_a	
Blood	SVM	5.74	77.27	75.53	265
	KHLD + SVM	13.85	77.92	75.36	113,265[a]
	ELM	0.01	76.48	75.32	80
	KHLD + ELM	2.69	76.83	75.51	113,80[a]
	HSARBF-ELM	4.21	79.12	77.69	113,60[a]
	KHLD + HSARBF-ELM	7.39	79.85	78.14	
	CO-BP	3.85	72.13	71.28	8
	KHLD + CO-BP	6.83	73.89	73.34	113,8[a]
	FOGDM-RBF	5.69	77.19	75.37	30
	KHLD + FOGDM-RBF	7.87	78.35	76.92	113,30[a]
Climate	SVM	1.86	92.32	92.64	49
	KHLD + SVM	3.74	92.86	92.82	13,49
	ELM	0.02	91.85	91.53	50
	KHLD + ELM	0.86	92.13	91.79	13,50
	HSARBF-ELM	2.78	93.47	92.41	13,50
	KHLD + HSARBF-ELM	3.82	94.21	93.13	
	CO-BP	1.73	92.80	92.38	8
	KHLD + CO-BP	2.53	93.68	92.76	13,8
	FOGDM-RBF	2.11	92.26	92.07	8
	KHLD + FOGDM-RBF	3.20	93.39	92.82	13,8
HD	SVM	0.15	81.70	85.85	42
	KHLD + SVM	0.28	82.58	85.76	12,42
	ELM	0	79.56	78.62	30
	KHLD + ELM	0.05	80.41	79.83	12,30
	HSARBF-ELM	0.35	83.13	83.64	12,20
	KHLD + HSARBF-ELM	0.67	84.32	84.39	
	CO-BP	0.08	80.58	80.37	5
	KHLD + CO-BP	0.23	81.52	81.26	12,5
	FOGDM-RBF	0.13	82.74	82.18	10
	KHLD + FOGDM-RBF	0.27	83.56	82.90	12,10
Sonar	SVM	0.12	80.85	84.97	46
	KHLD + SVM	0.26	82.56	85.34	49,46
	ELM	0	70.37	70.06	50
	KHLD + ELM	0.13	72.92	72.73	49,50
	HSARBF-ELM	0.64	76.74	76.09	49,40
	KHLD + HSARBF-ELM	0.95	79.59	80.31	
	CO-BP	0.14	68.32	66.86	5
	KHLD + CO-BP	0.36	70.31	69.25	49,5
	FOGDM-RBF	0.27	73.63	72.58	16
	KHLD + FOGDM-RBF	0.62	74.82	74.29	49,16
SH	SVM	0.18	70.05	72.41	78
	KHLD + SVM	0.76	72.59	73.61	14,78
	ELM	0	66.24	65.56	50
	KHLD + ELM	0.23	67.92	66.72	14,50
	HSARBF-ELM	0.41	67.58	65.84	14,40
	KHLD + HSARBF-ELM	0.71	71.24	70.38	
	CO-BP	0.12	71.26	70.68	7
	KHLD + CO-BP	0.38	73.18	72.80	14,7
	FOGDM-RBF	0.45	68.61	67.37	12
	KHLD + FOGDM-RBF	0.82	70.65	70.13	14,12

Table 2: Continued.

Datasets	Methods	Training time (s)	Testing		No. of kernels
			η_o	η_a	
IS	SVM	7.46	90.56	—	96
	KHLD + SVM	20.53	91.84	—	32,96
	ELM	0.03	90.31	—	100
	KHLD + ELM	1.68	90.82	—	32,100
	HSARBF-ELM	5.17	92.23	—	32,80
	KHLD + HSARBF-ELM	7.23	93.17	—	
	CO-BP	1.52	90.74	—	8
	KHLD + CO-BP	2.67	91.85	—	32,8
	FOGDM-RBF	4.63	89.58	—	36
	KHLD + FOGDM-RBF	6.39	90.54	—	32,36
Forests	SVM	1.86	72.19	74.51	125
	KHLD + SVM	2.94	73.60	74.78	51,125
	ELM	0.01	68.56	67.63	60
	KHLD + ELM	1.49	69.32	68.52	51,60
	HSARBF-ELM	1.83	69.93	69.37	51,50
	KHLD + HSARBF-ELM	3.35	71.65	71.32	
	CO-BP	1.27	63.79	62.23	9
	KHLD + CO-BP	1.85	64.16	62.58	51,9
	FOGDM-RBF	2.32	69.24	68.87	30
	KHLD + FOGDM-RBF	3.94	70.62	70.42	51,30

[a]The number of generated RBF kernels and the number of kernels/support vectors in each classifier.

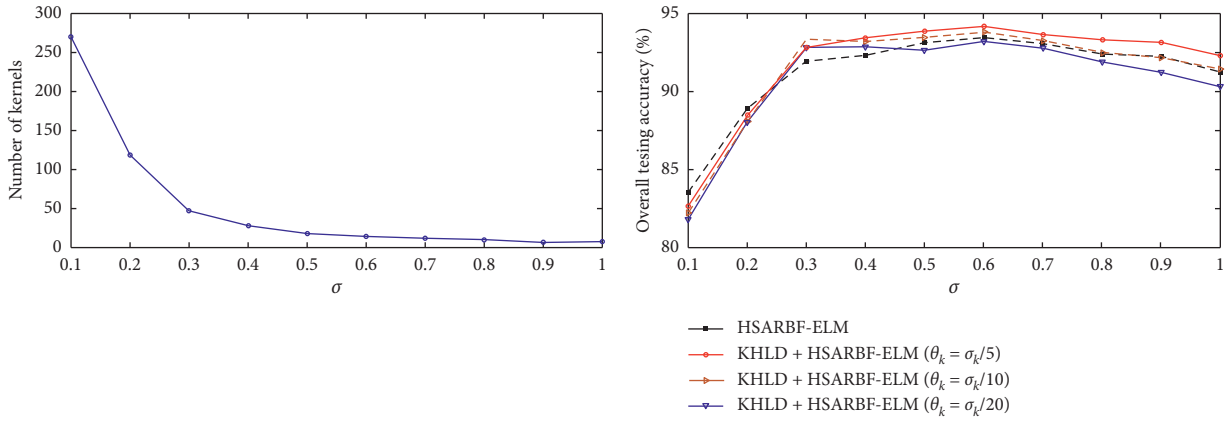
Table 3: Performance comparison on benchmark large sample datasets.

Datasets	Methods	Training time (s)	Testing		No. of kernels
			η_o	η_a	
Svmguide1	ELM	0.15	90.52	90.36	200
	KHLD + ELM	32.53	89.81	89.73	442,200
	HSARBF-ELM	387.52	91.76	91.34	442,160
	KHLD + HSARBF-ELM	423.64	91.24	90.81	
	CO-BP	18.54	90.41	90.13	30
	KHLD + CO-BP	55.37	89.38	89.22	442,30
	FOGDM-RBF	28.53	92.34	92.18	70
	KHLD + FOGDM-RBF	65.70	91.82	91.56	442,70
Wilt	ELM	0.06	62.63	60.40	100
	KHLD + ELM	37.91	61.82	59.84	173,100
	HSARBF-ELM	358.61	64.73	63.93	173,80
	KHLD + HSARBF-ELM	387.42	64.17	63.49	
	CO-BP	38.52	60.28	59.52	30
	KHLD + CO-BP	78.65	59.56	58.72	173,30
	FOGDM-RBF	24.73	62.85	62.37	60
	KHLD + FOGDM-RBF	63.85	62.39	62.13	173,60

generated kernel cannot effectively cover the sample space, so the effectiveness of the kernel cannot be guaranteed, which leads to the performance degradation of the proposed method. When σ is too large or θ_k is too small, the probability of overlapping samples in the generated kernel will increase, which leads to the performance degradation of the proposed method.

From experiments on multiple datasets, the method of KHLD improves the problem of network generalization performance degradation when the sample size is too small or the sample space distribution is too sparse.

However, when the number of training samples is sufficient or the spatial distribution of training samples is dense, the network performance of the proposed method shows a certain degree of decline compared with the direct training of the classifier. This situation shows that when the constructed kernel can be effectively represented by the existing training samples, the generated samples in the kernels are equivalent to increasing the noise samples, which leads to the redundancy of network training and is not conducive to the improvement of the boundary partition surface. Thus, the proposed method is not suitable for classification problems with sufficient number of training samples or dense spatial distribution of samples. In the selection of parameters, the kernel width should be chosen so that it is not too small or too large. If the kernel width is too small, the validity of the



(a) (b)

Figure 13: Stress testing of the proposed method on climate dataset. (a) σ, number of kernels. (b) σ, overall testing accuracy.

established kernel may not be guaranteed, which makes the method of KHLD ineffective to a certain extent. If the kernel width is too large, the overlapping degree between the samples generated in the kernel and the heterogeneous samples increases, which also leads to the performance degradation of the proposed method.

4. Conclusion

An optimized neural network classifier based on KHLD is presented. The established kernels in KHLD are based on the generated RBF kernel parameters in the HSARBF-ELM algorithm. An optimized sample filling and screening method can realize the effective approximation of KHLD in different classification problems. Combining KHLD with other algorithms can effectively improve the network performance of these algorithms, especially when the sample space distribution is sparse. Experiments on artificial datasets and benchmark datasets further verify the effectiveness of our method.

One of the main shortcomings of this work is the representation of kernels. In this study, for the convenience of problem description, the representation of the kernel is a regular hypersphere. The proposed method is mainly suitable for the case of sparse spatial distribution of samples but is not suitable for large sample set learning and classification. The establishment and representation of kernel are worthy of further study. Exploring more optimized kernel representation and combining it with KHLD are our future work.

Data Availability

The artificial dataset Double Moon comes from "S Hayin, Neural networks and learning machines (Third Edition). Beijing: China Machine Press, China, 2009, pp. 61-63." The artificial dataset Concrete Circle is generated by the authors. The data are available upon request by email wen_hui81@163.com. The benchmark datasets Blood, Climate, Heart Disease (HD), Sonar, SPECT Heart (SH), Image Segmentation (IS), Forest, and Wilt come from UCI repository of machine learning databases, available at http://archive.ics.uci.edu/ml. The Svmguide1 dataset comes from LIBSVM databases, available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China (No. 61741111); Natural Science Foundation of Fujian (Nos. 2019J01815 and 2019J01816); Natural Science Foundation of Jiangxi (No. 20181BAB202011); Department of Education of Fujian Province (Nos. JT180486 and FJJKCG20-101); Putian Science and Technology Bureau Project (Nos. 2018RP4004 and 2018ZP10); and Introduction of Talents to Start Scientific Research Projects in Putian University (No. 2018088).

References

[1] W. Yan, H. Sun, Q. Sun et al., "Multiple kernel dimensionality reduction based on collaborative representation for set oriented image classification," *Expert Systems with Applications*, vol. 137, pp. 380–391, 2019.

[2] W. Yan, Q. Sun, H. Sun, Y. Li, and Z. Ren, "Multiple kernel dimensionality reduction based on linear regression virtual reconstruction for image set classification," *Neurocomputing*, vol. 361, pp. 256–269, 2019.

[3] K. Selvakumar, M. Karuppiah, L. SaiRamesh et al., "Intelligent temporal classification and fuzzy rough set-based feature selection algorithm for intrusion detection system in WSNs," *Information Sciences*, vol. 497, pp. 77–90, 2019.

[4] X. Liang, L. Zhu, and D.-S. Huang, "Multi-task ranking SVM for image cosegmentation," *Neurocomputing*, vol. 247, pp. 126–136, 2017.

[5] H. Wang, P. Shi, H. Li, and Q. Zhou, "Adaptive neural tracking control for a class of nonlinear systems with dynamic uncertainties," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3075–3087, 2017.

[6] H. Wang, W. Sun, and P. X. Liu, "Adaptive intelligent control of nonaffine nonlinear time-delay systems with dynamic

uncertainties," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1474–1485, 2017.

[7] X. Zhao, P. Shi, X. Zheng, and J. Zhang, "Intelligent tracking control for a class of uncertain high-order nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 9, pp. 1976–1982, 2016.

[8] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.

[9] W. Pedrycz, "Conditional fuzzy clustering in the design of radial basis function neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 601–612, 1998.

[10] A. D. Niros and G. E. Tsekouras, "A novel training algorithm for RBF neural network using a hybrid fuzzy clustering approach," *Fuzzy Sets and Systems*, vol. 193, pp. 62–84, 2012.

[11] S. Chen, X. Hong, C. J. Harris, and L. Hanzo, "Fully complex-valued radial basis function networks: orthogonal least squares regression and classification," *Neurocomputing*, vol. 71, pp. 3421–3433, 2008.

[12] R. Mohammadi, S. M. T. Fatemi Ghomi, and F. Zeinali, "A new hybrid evolutionary based RBF networks method for forecasting time series: a case study of forecasting emergency supply demand time series," *Engineering Applications of Artificial Intelligence*, vol. 36, pp. 204–214, 2014.

[13] H. M. Feng, "Self-generation RBFNs using evolutional PSO learning," *Neurocomputing*, vol. 70, pp. 41–251, 2006.

[14] H. Wang, R. Feng, Z. F. Han, and C. S. Leung, "ADMM-based algorithm for training fault tolerant RBF networks and selecting centers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3870–3878, 2018.

[15] J. Raitoharju, S. Kiranyaz, and M. Gabbouj, "Training radial basis function neural networks for classification via class-specific clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2458–2471, 2016.

[16] H. Yin and N. M. Allinson, "Self-organizing mixture networks for probability density estimation," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 405–411, 2001.

[17] L. Yingwei, N. Sundararajan, and P. Saratchandran, "A sequential learning scheme for function approximation using minimal radial basis function neural networks," *Neural Computation*, vol. 9, no. 2, pp. 461–478, 1997.

[18] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 6, pp. 2284–2292, 2004.

[19] M. Bortman and M. Aladjem, "Growing and pruning method for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 1030–1045, 2009.

[20] H. Yu, P. D. Reiner, T. Xie, T. Bartczak, and B. M. Wilamowski, "An incremental design of radial basis function networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 10, pp. 1793–1803, 2014.

[21] G. Vachkov, V. Stoyanov, and N. Christova, "Incremental RBF network models for nonlinear approximation and classification," in *Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–8, Istanbul, Turkey, August 2015.

[22] H. Wen, W.-X. Xie, J.-H. Pei, and L.-X. Guan, "An incremental learning algorithm for the hybrid RBF-BP network classifier," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 57, 2016.

[23] H. Wen, H. Fan, W. Xie, and J. Pei, "Hybrid structure-adaptive RBF-ELM network classifier," *IEEE Access*, vol. 5, pp. 16539–16554, 2017.

[24] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "A new learning scheme of feedforward neural networks," in *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks*, pp. 985–999, Budapest, Hungary, July 2004.

[25] S. Hayin, *Neural Networks and Learning Machines*, China Machine Press, Beijing, China, 3rd edition, 2009.

[26] C. Blake and C. Merz, *UCI Repository of Machine Learning Databases [Online]*, University of California, Irvine, Department of Information and Computer Sciences, Irvine, CA, USA, 1998, http://archive.ics.uci.edu/ml.

[27] C.-C. Chang and C.-J. Lin, *LIBSVM: A Library for Support Vector Machines [Online]*, National Taiwan University, Taiwan, Department of Computer Science and Information Engineering, Taipei, Taiwan, 2016, http://www.csie.ntu.edu.tw/%7Ecjlin/libsvm.

[28] Z. Li, F. Wang, T. Sun, and B. Xu, "A constrained optimization method based on BP neural network," *Neural Computing and Applications*, vol. 29, no. 2, pp. 413–421, 2018.

[29] H. Xue, Z.-P. Shao, and H.-B. Sun, "Data classification based on fractional order gradient descent with momentum for RBF neural network," *Network-Computation in Neural Systems*, vol. 31, no. 1–4, pp. 166–185, 2020.