

## Research Article

# A Novel Kernel for RBF Based Neural Networks

Wasim Aftab,<sup>1,2</sup> Muhammad Moinuddin,<sup>1,3</sup> and Muhammad Shafique Shaikh<sup>1,3</sup>

<sup>1</sup> Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, P.O. Box 80204, Jeddah 21589, Saudi Arabia

<sup>2</sup> Centre of Nanotechnology, King Abdulaziz University, Jeddah, Saudi Arabia

<sup>3</sup> Center of Excellence in Intelligent Engineering Systems (CEIES), King Abdulaziz University, Jeddah, Saudi Arabia

Correspondence should be addressed to Muhammad Moinuddin; [mmsansari@kau.edu.sa](mailto:mmsansari@kau.edu.sa)

Received 10 April 2014; Revised 25 May 2014; Accepted 25 May 2014; Published 19 June 2014

Academic Editor: Dumitru Baleanu

Copyright © 2014 Wasim Aftab et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Radial basis function (RBF) is well known to provide excellent performance in function approximation and pattern classification. The conventional RBF uses basis functions which rely on distance measures such as Gaussian kernel of Euclidean distance (ED) between feature vector and neuron's center, and so forth. In this work, we introduce a novel RBF artificial neural network (ANN) where the basis function utilizes a linear combination of ED based Gaussian kernel and a cosine kernel where the cosine kernel computes the angle between feature and center vectors. Novelty of the proposed work relies on the fact that we have shown that there may be scenarios where the two feature vectors (FV) are more prominently distinguishable via the proposed cosine measure as compared to the conventional ED measure. We discuss adaptive symbol detection for multiple phase shift keying (MPSK) signals as a practical example to show where the angle information can be pivotal which in turn justifies our proposed RBF kernel. To corroborate our theoretical developments, we investigate the performance of the proposed RBF for the problems pertaining to three different domains. Our results show that the proposed RBF outperforms the conventional RBF by a remarkable margin.

## 1. Introduction

Computational model for neural networks was first proposed by McCulloch and Pitts [1]. Since then, artificial neural networks (ANN) have been recognized as a decision making tool by many researches [2–4]. ANN is particularly very useful in solving problems which are difficult to solve with the conventional rule-based programming [5].

Simple but yet powerful generalization capability of ANN had drawn the attention of numerous past and present researchers [2, 3, 6, 7]. It all started with Rosenblatt when he created the perceptron [8], a pattern recognition algorithm for supervised classification.

However, Rosenblatt's idea could not be translated into a computer program until the development of backpropagation algorithm which has so far been the most popularly used algorithm in ANN paradigm [9]. Thereafter, immense research was done in this field, and in last 50 years or so there has been extraordinary growth in this domain and the result is invention of several sophisticated algorithms [6, 10, 11].

Radial basis function (RBF) network [12] is an ANN and its activation functions are radial basis functions. It was first introduced by Broomhead and Lowe [10] and since then it has become a very popular methodology to solve problems that suit ANN paradigm [11–14]. The main advantage of RBF when compared with other algorithms based on ANN paradigm is the simplicity of the computation of network parameters [12]. Another very important feature of radial basis function neural networks is to be able to perform complex nonlinear mappings that allow a fast linear and robust learning mechanism [5]. Originally, RBF networks were developed for data interpolation in high dimensional space [12]. Nonetheless, RBF networks have been used in diverse domains, including pattern classification [7], time series prediction [13], system control [14], and function approximation [15].

Some of the most commonly used basis functions are Gaussian functions [12], multiquadric functions [12], thin plate spline function [12], inverse multiquadric functions [12], and so forth. There is no general rule, but the choice of a radial basis function is highly problem specific. Also, most

applications using RBF make use of a free shape parameter that plays pivotal role in the accuracy of the method and is commonly chosen with the help of cross-validation [16] technique. It is a standard practice [12] to learn three sets of parameters for RBF network: locations, widths, and weight factors of RBF kernel. Enormous amount of work [17–19] has already been done to select those parameters optimally.

In the conventional RBF kernel, mostly Gaussian of the Euclidean distance between feature vector and neuron's center is used [17]. However, there can be scenarios where Euclidean distance is not the dominant measure to find separation among the features, for example, if two feature vectors are separated by equal distance from a center but separated from the center via unequal angles. In that case, the cosine of the angle can play a vital role in differentiating the feature vectors. We have discussed in detail such scenarios in Section 3.

Motivated by this observation, we propose a novel RBF kernel which consists of a linear combination of Gaussian and cosine RBF kernels. The cosine RBF kernel computes the cosine of the angle between supplied feature vector and the center vector associated with that neuron.

There are some existing works in the literature that had discussed usage of cosine measure with RBF kernels [20–26]. Karayiannis and Randolph-Gips [27] have proposed a novel RBF which is a normalized version of the multiquadratic radial basis function, where the cosine represents the angle between the transformed vectors rather than the original vectors. Liu et al. [28] have used cosine similarity measure to achieve high performance of classification by selecting meaningful features. They compute the cosine similarity among the kernels rather than the original vectors. By doing so, they transformed all the vectors to the same length, whereas we do not perturb the feature space. Moreover, these cosine kernels are developed for Support vector machine (SVM). Cho and Saul [29] have used arc cosine of the angles between inputs in their kernel.

To the best of our knowledge, the existing works in the literature related to the idea of incorporating cosine measure inside kernels have been used either in SVM kernel [28, 29] without using ANN paradigm or in a transformed domain [30] in ANN paradigm. Our work is different from all of them in many aspects. Firstly, we proposed a cosine kernel in the original vector space rather than in any transformed space.

Secondly, we incorporated the effect of both ED and cosine measures with a linear combination. Finally, unlike the existing works [27–29], we used the proposed kernel with RBF ANN classifier.

In order to validate our theoretical developments, we have investigated two different research problems: pattern recognition and nonlinear plant identification.

We composed the rest of this paper in the following manner. Section 2 provides an overview of conventional RBF. In Section 3, we describe our proposed algorithm. In Sections 4 and 5, we provide the proof-of-principle of our method with application examples. Then, in Section 6, we discuss our results in detail.

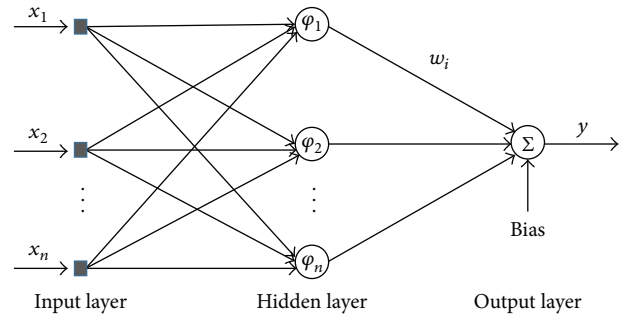


FIGURE 1: Architecture of the radial basis function neural network.

## 2. Overview of the Conventional RBF

RBF networks in their general form consist of three layers: an input layer, a hidden layer where nonlinear activation functions operate, and a linear output layer as shown in Figure 1. Generally, the input is a real vector,  $\mathbf{x} \in \mathbf{R}^n$ . The network output maps the input vector to a scalar,  $y: \mathbf{R}^n \rightarrow \mathbf{R}$ , which is achieved by employing the following equation:

$$y_j = \sum_{i=1}^N w_i \varphi_i(\|\mathbf{x} - \mathbf{c}_i\|) + b_j \quad \forall j = 1, 2, \dots, N_0, \quad (1)$$

where  $N$  and  $N_0$  are the number of hidden and output layer neurons, respectively,  $\mathbf{c}_i \in \mathbf{R}^n$  is the center for  $i$ th neuron,  $w_i$  is output layer weight for  $i$ th neuron,  $b_j$  is the bias term for the  $j$ th output neuron, and  $\varphi_i$  is the basis function associated with  $i$ th hidden neuron.

RBF solves a problem by mapping it into a high dimensional space in a nonlinear manner and then applies linear decision boundary. The concept of transformation to high dimensional space is justified by Cover's theorem, according to which classification via linear separation becomes easier by translating the features from low dimension to high dimension [31].

The significance of adding bias to the output is to improve the approximation quality by shifting the decision boundary. The weights of the network govern the position of the decision boundary in the feature space. However, during the adaptive weight update, if bias is not used, then the hyperplane is forced to pass through the origin of the feature space defined by the inputs or feature vectors. Although it is valid for some problems, in many others this separation boundary is desired to be located somewhere else.

As a general rule, all inputs are connected to each hidden neuron. The domain of activation function is a norm which is typically taken to be the Euclidean distance between input and the centers of every neuron. Most commonly used RBF kernels are as follows [5].

Multiquadratics:

$$\varphi_i(\|\mathbf{x} - \mathbf{c}_i\|) = (\|\mathbf{x} - \mathbf{c}_i\|^2 + \tau^2)^{1/2}. \quad (2)$$

Inverse multiquadrics:

$$\varphi_i(\|\mathbf{x} - \mathbf{c}_i\|) = \frac{1}{(\|\mathbf{x} - \mathbf{c}_i\|^2 + \tau^2)^{1/2}}. \quad (3)$$

Gaussian:

$$\varphi_i(\|\mathbf{x} - \mathbf{c}_i\|) = e^{-\|\mathbf{x} - \mathbf{c}_i\|^2/\beta^2}, \quad (4)$$

where  $\tau > 0$  is a constant and  $\beta$  is spread parameter. The sensitivity of a hidden neuron towards a data point varies in proportion with the distance of the data point from its center. For example, in case of a conventional ED based RBF network that uses Gaussian in its kernel, this sensitivity can be fine-tuned by adjusting  $\beta$ ; if  $\beta$  is large, it implies less sensitivity and vice versa. The weights and biases are usually updated adaptively by implying the following steepest decent approach [32]:

$$\begin{aligned} w_i(n+1) &= w_i(n) + \eta e(n) \varphi_i(\|\mathbf{x} - \mathbf{c}_i\|) \\ b_i(n+1) &= b_i(n) + \eta e(n), \end{aligned} \quad (5)$$

where  $\eta$  is the learning rate for the network and  $e(n)$  is the error between the desired and the actual output of the RBF for the  $n$ th iteration.

### 3. Proposed RBF

Intuition suggests that ED is not the only measure to contrast the FVs. For example, in the case when FVs are equally separated in distance, then the ED will be no more effective. To deal with this issue, we proposed a generalized RBF kernel by linearly combining the conventional ED based RBF kernel and our proposed cosine based RBF kernel which can be formulated as follows:

$$\varphi_i(\mathbf{x}, \mathbf{c}_i) = \alpha_1 \varphi_{i1}(\mathbf{x} \cdot \mathbf{c}_i) + \alpha_2 \varphi_{i2}(\|\mathbf{x} - \mathbf{c}_i\|), \quad (6)$$

where  $\alpha_1, \alpha_2$  are weightage parameters for cosine and Euclidean kernels, respectively, which can acquire values in this range:  $0 \leq \alpha_1, \alpha_2 \leq 1$ . In (6),  $\varphi_{i1}(\mathbf{x} \cdot \mathbf{c}_i)$  and  $\varphi_{i2}(\|\mathbf{x} - \mathbf{c}_i\|)$  are the cosine and the Euclidean kernels, respectively, for  $i$ th neuron. These are defined as follows:

$$\varphi_{i1}(\mathbf{x} \cdot \mathbf{c}_i) \triangleq \cos(\theta_i) = \frac{\mathbf{x} \cdot \mathbf{c}_i}{\|\mathbf{x}\| \cdot \|\mathbf{c}_i\|}, \quad (7)$$

$$\varphi_{i2}(\|\mathbf{x} - \mathbf{c}_i\|) = e^{-\|\mathbf{x} - \mathbf{c}_i\|^2/\beta^2}, \quad (8)$$

where  $\mathbf{x} \cdot \mathbf{c}_i$  represents the dot product between the two vectors. Consequently, (7) can be rewritten as follows:

$$\varphi_{i1}(\mathbf{x} \cdot \mathbf{c}_i) = \frac{\sum_{j=1}^L \sum_{j=1}^L X_j \cdot C_{ij}}{\sqrt{\sum_{j=1}^L X_j^2} \cdot \sqrt{\sum_{j=1}^L C_{ij}^2}}, \quad (9)$$

where  $L$  is the length of each incoming feature vector  $\mathbf{x}$ . By observing (7), we can notice that the kernel  $\varphi_{i1}(\mathbf{x} \cdot \mathbf{c}_i)$  computes the cosine of angle between  $\mathbf{x}$  and  $\mathbf{c}_i$ . Hence,  $\varphi_{i1}(\mathbf{x} \cdot \mathbf{c}_i)$  may

attain the values in the range  $[-1, +1]$ . If it returns to 1, it implies that the  $\mathbf{x}$  is aligned with  $\mathbf{c}_i$ , whereas its 0 return value corresponds to the scenario when  $\mathbf{x}$  is perfectly orthogonal to  $\mathbf{c}_i$ ; and the return value of  $-1$  indicates that  $\mathbf{x}$  and  $\mathbf{c}_i$  are aligned in opposite directions.

**3.1. Some Special Scenarios Related to the Proposed RBF Kernel.** In order to get more insight into the advantage of our proposed RBF kernel, we consider 2 feature vectors  $\mathbf{f}_1$  and  $\mathbf{f}_2$  separated by distances  $d_1$  and  $d_2$  and angles  $\theta$  and  $\delta$ , respectively, from the center vector  $\mathbf{c}$  (see Figure 2). We explore some special scenarios of the aforementioned situation in the ensuing subsections.

**3.1.1. Scenario 1.** Consider

$$d_1 = d_2, \quad \theta \neq \delta. \quad (10)$$

In this case, we clearly see (see Figure 2(a)) that a RBF kernel based on Euclidean distance will not be able to differentiate between  $\mathbf{f}_1$  and  $\mathbf{f}_2$ . Using the cosine of such angle as activation, we propose a new kernel for RBF which can work in the described scenario. Therefore, we get rid of the ED component from (6) by setting  $\alpha_2 = 0$  and  $\alpha_1 = 1$  and hence obtain one cosine kernel which is defined by (7) and (9). A similar scenario is observed in our pattern classification application dataset discussed in Section 5.2. To support our argument, we performed a statistical analysis using the *Silhouette widths* [33].

**3.1.2. Scenario 2.** Consider

$$d_1 \neq d_2, \quad \theta = \delta. \quad (11)$$

We clearly see (see Figure 2(b)) that, for this scenario, the Euclidean kernel is more suitable than ours, so we set  $\alpha_2 = 0$  and  $\alpha_1 = 1$  in (6) and the resulting kernel is defined in (6).

**3.1.3. Scenario 3.** Consider

$$d_1 \neq d_2, \quad \theta \neq \delta. \quad (12)$$

Form (6), it is clear that we fused the two classifiers with certain weightage [34]. It is also evident that the weightage parameters in (6) can be tuned according to various different mentioned scenarios to produce good classification results. For example, we saw that, in scenario 1, we have  $\alpha_1 = 1$  and  $\alpha_2 = 0$  and in scenario 2 they will assume value is complementary to the scenario 1. However, in scenarios like 3 (see Figure 2(c)), weightage parameters for the kernel can assume any values in the range (0, 1) but with one condition: summation of  $\alpha_1$  and  $\alpha_2$  is always unity.

We have another scenario where  $d_1 = d_2$  and also  $\theta = \delta$ . In this case although individually both conventional and our RBF will fail to produce good results, we anticipate that proper choice weightage parameters may improve the results.

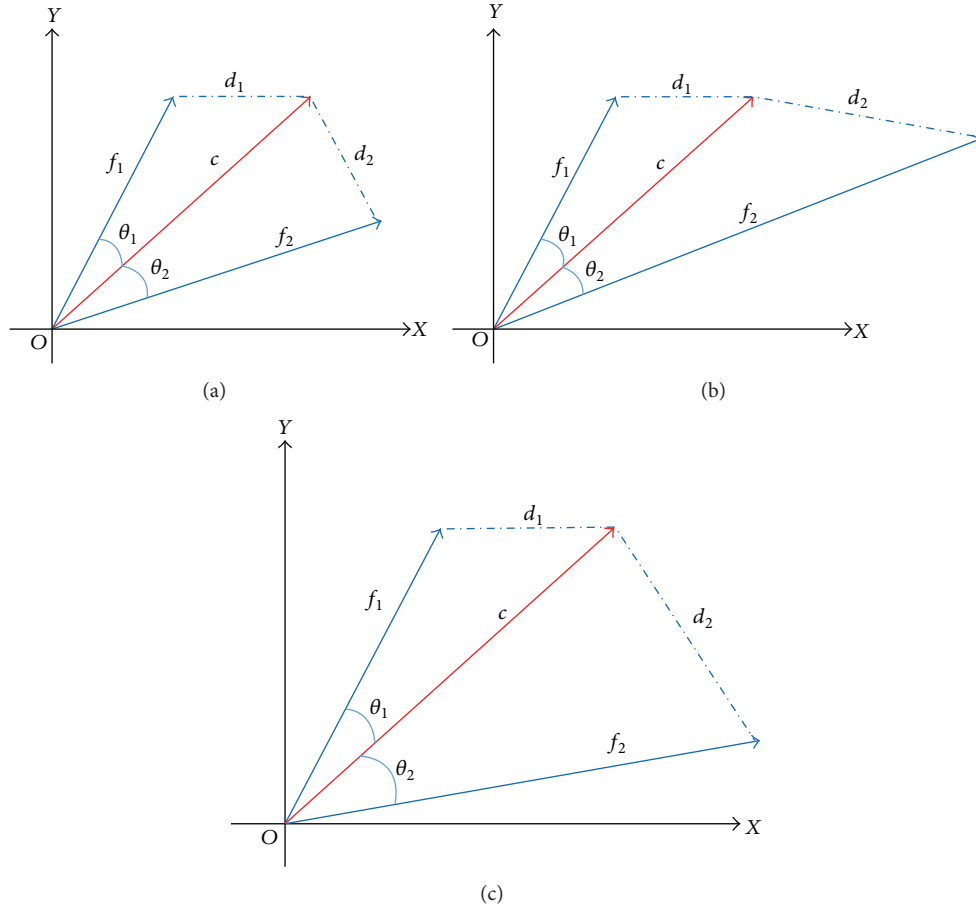


FIGURE 2: (a) Geometric representation of scenario 1:  $d_1 = d_2$  but  $\theta_1 \neq \theta_2$ . (b) Geometric representation of scenario 2:  $d_1 \neq d_2$  but  $\theta_1 = \theta_2$ . (c) Geometric representation of scenario 3:  $d_1 \neq d_2$  but  $\theta_1 \neq \theta_2$ .

3.2. *Properties of the Proposed RBF Kernel.* Consider

- (i)  $\varphi_i(\mathbf{x}, \mathbf{c}_i) = \varphi_i(\mathbf{c}_i, \mathbf{x})$
- (ii)  $\varphi_i(\mathbf{x}, \mathbf{c}_i) = \varphi_i(\mathbf{c}_i, \mathbf{x}) = 1 \quad \forall \mathbf{x} = \mathbf{c}_i$
- (iii)  $|\varphi_i(\mathbf{x}, \mathbf{c}_i)| \leq 1$ .

The above mentioned properties can be justified with the help of (6)–(8). For example, in (6), if we swap the orders of  $\mathbf{x}$  and  $\mathbf{c}_i$ , then the results of the equation remain unchanged; hence, proposed kernel obeys commutative property.

Similarly, if we plug  $\mathbf{x} = \mathbf{c}_i$  in (6), then with the help of (7) and (8) the right-hand side of (6) boils down to the summation of the weightage parameters  $\alpha_1$  and  $\alpha_2$ . We have already discussed earlier that summation of the weightage parameters is 1 always, which justifies the second property of our kernel.

Finally, from (7) and (8), we see that upper bounds of both the kernels are 1; deploying this concept in (6), we can write  $|\varphi_i(\mathbf{x}, \mathbf{c}_i)| \leq 1$ , which is the third property.

#### 4. Intuition from a Practical Example: Adaptive Symbol Detection of MPSK Modulated Symbols

In digital communication systems, MPSK modulation is a common and well-known practice. When such signals are transmitted through noisy channel, the symbols are dispersed around its original locations (see Figure 3). As a result, the system performance degrades severely. Therefore, it is crucial to design efficient receiver capable of recovering the original symbols without errors. For this purpose, there are many well-known methods in the literature. However, our rationale to discuss this problem here is to show a practical example where the angle measure is more discriminative than the ED measure. To get more insights, observe the scatter plot of MPSK modulated signals (For  $M = 16$  symbols) shown in Figure 4. It can be clearly seen from Figure 4 that these modulated signals have equal amplitudes but differ in their phases. Thus, intuitively, one can say that in this scenario a more suitable receiver will be the one which can deal with the phases only. Therefore, our proposed kernel will be more preferred candidate in this case. Our simulation study for this problem presented in Section 5.1 supports our intuition

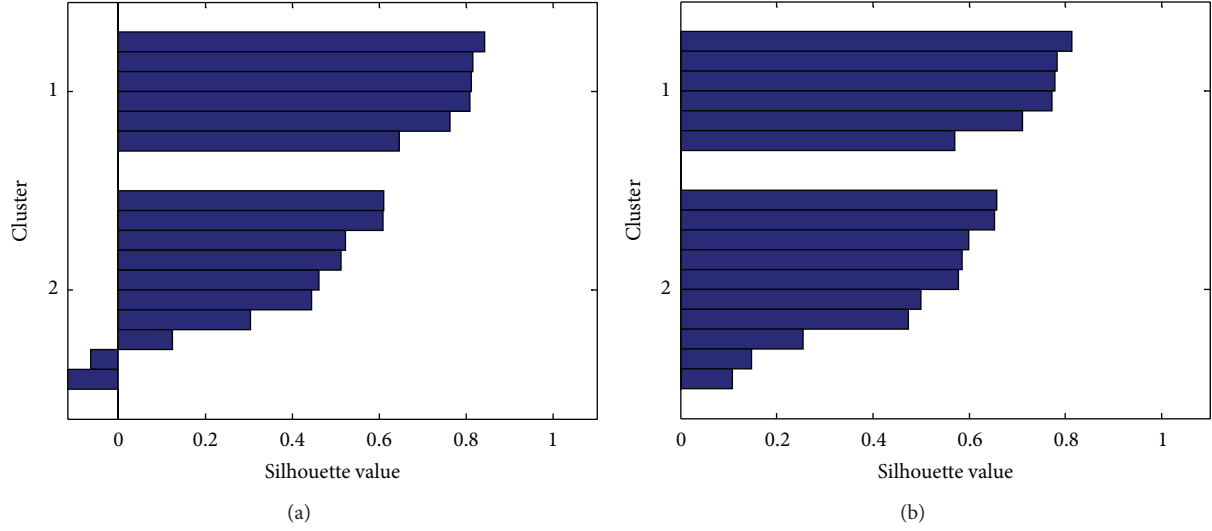


FIGURE 3: (a) Silhouette plot for Euclidean RBF kernel: negative Silhouette indicates misclassification. (b) Silhouette plot for proposed RBF kernel: large Silhouette widths with no negatives.

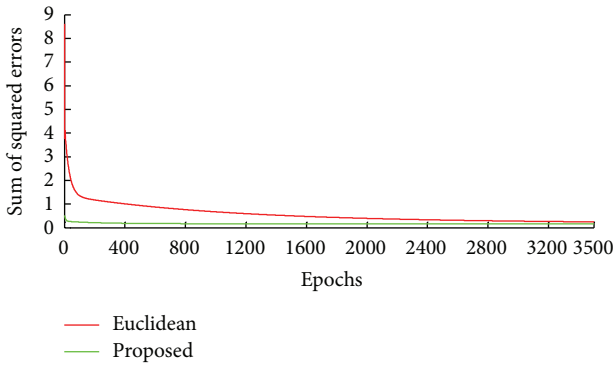


FIGURE 4: Comparative objective function plot showing difference in convergence for the two RBF kernels: Euclidean and proposed; proposed kernel converges faster than its conventional counterpart.

and our proposed RBF kernel is found to be better than the conventional ED based RBF kernel.

## 5. Comparative Study of the Conventional and the Proposed RBF

In the present work, we aim to show a proof-of-principle application for our proposed algorithm. We conducted a comparative study by applying both the conventional RBF and our proposed RBF to the following domains, where very often RBF network has been used for classification.

**5.1. Recovery of MPSK Modulated Signals.** The problem of adaptive recovery of MPSK modulated signals is already described in Section 4. In this section, we present simulations to validate the performance of the proposed kernel. Specifically, we present the simulations for adaptive recovery of

QPSK signals (i.e.,  $M = 4$ ) in adaptive white Gaussian noise (AWGN). In Figures 11 and 12, the scatter plots of transmitted QPSK signals are shown for  $M = 16$  and  $M = 4$  respectively. These signals, when transmitted through AWGN channel, are dispersed as shown in Figure 10. In order to recover the original transmitted signals, we employ our proposed RBF kernel and compare its performance with the conventional ED based RBF. For this purpose, we used four neurons in the hidden layer and one in the output layer with  $\eta = 0.05$  and  $\beta = 1$ .

**5.2. Pattern Recognition: Classification of Leaves.** Our dataset is extracted from [35]. The dataset contains three features: shape, margin, and texture for one hundred plant species leaves and, for each feature, a 64-element vector is given per sample of leaf. We have concatenated these features into a single feature vector for every sample. As a result, the length of one feature vector is 192. In order to perform a binary classification, we had chosen samples for the species *Acer campestre* and *Zelkova serrata* which consisted of 16 instances per class originally.

Therefore, the problem to be solved is as follows. Given these features, classify the leaves of the abovementioned species with minimum possible classification error. We made a comparative study of convergence time in terms of number of epochs and accuracy using both approaches: conventional ED based RBF and the proposed RBF whose results are shown in Figures 4, 5 and 6 and are discussed in Section 5.2.

We have used adaptive kernels for both the conventional and the proposed RBF. We tried different combinations of  $\eta$ ,  $\beta$ , and  $N$  and after exhaustive investigation we got 100% accuracy at  $\eta = 0.05$ ,  $\beta = 1$ , and  $N = 6$ .

**5.3. Control Theory: Nonlinear Plant Identification.** Plants are dynamic systems with high complexity and nonlinearity.



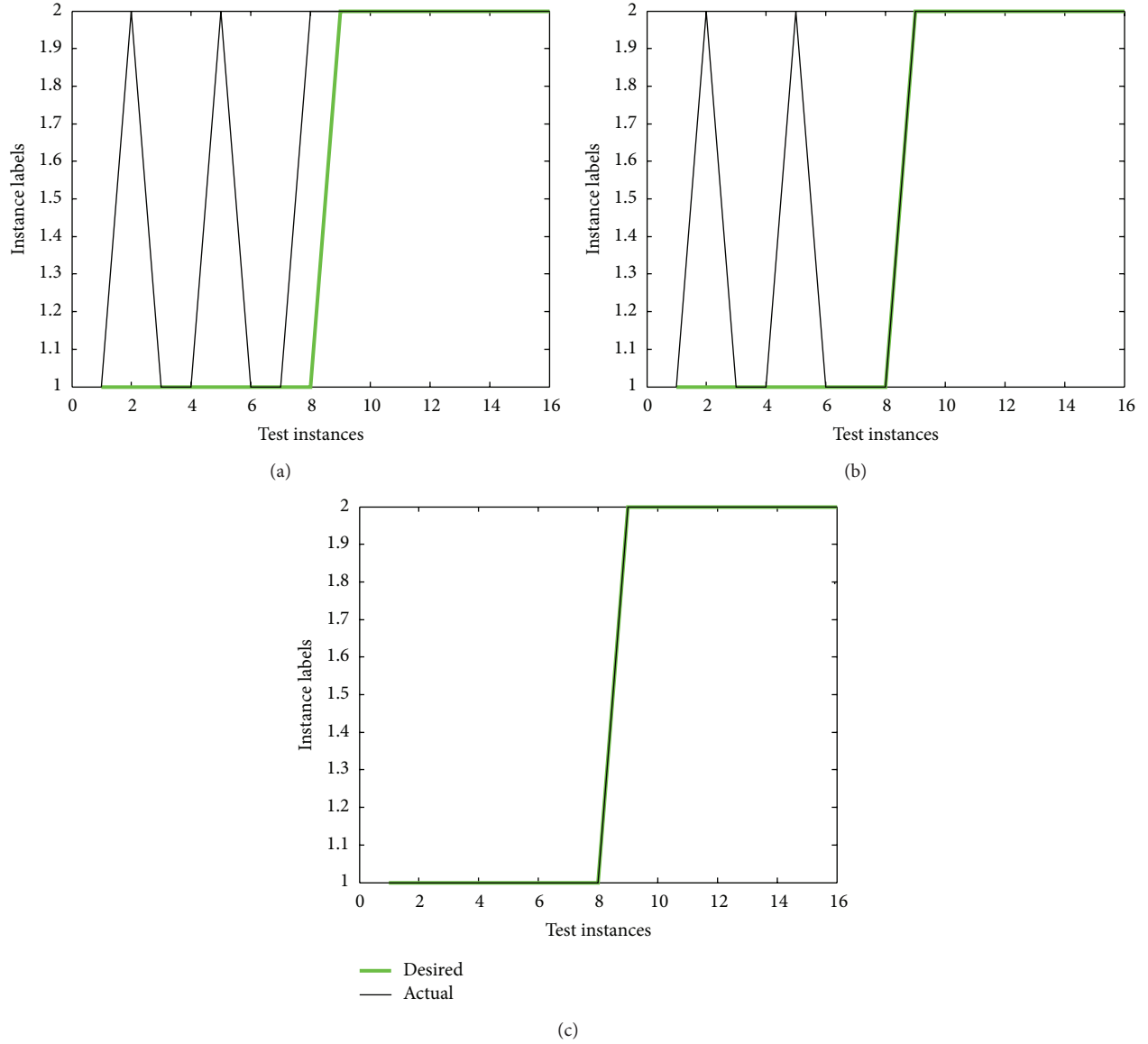


FIGURE 5: (a) Testing results after 100 epochs using Euclidean RBF kernel; misclassifications exist. (b) Testing results after 150 epochs using Euclidean RBF kernel; misclassifications still exist. (c) Testing results after 3500 epochs using Euclidean RBF kernel; 100% accuracy is achieved.

A vital step towards nonlinear plant identification is the development of a nonlinear model [36]. Therefore, it is very important to develop as accurate models as possible for plants which have highly nonlinear behavior. ANN in general and RBF to be specific have been used very often in this regard [37–39]. The system model is shown in Figure 9. In our study, we have considered a highly nonlinear plant whose output and input can be mapped with the following relation:

$$y(t) = a_1 r(t) + a_2 r(t-1) + a_3 r(t-2) + a_4 (\cos(br(t)) + e^{-|r(t)|}) + n(t), \quad (13)$$

where  $r(t)$  is plant's input,  $n(t)$  is the plant's disturbance which we have modeled as zero mean normal distributed random variable,  $a_i$ 's are polynomial coefficients defining systems

zeros, and  $b > 0$  is a constant. In our experiment, we have chosen the polynomial coefficients as  $a_1 = 2$ ,  $a_2 = -0.5$ ,  $a_3 = -0.1$ ,  $a_4 = -0.7$ , and  $b = 3$ .

In Figure 9,  $p(t)$  is the plant's impulse response and  $y(t)$  is the final output of the plant.

$\hat{p}(t)$ ,  $\hat{y}(t)$  are estimations of  $p(t)$ ,  $y(t)$ , respectively, and  $e(t)$  is error in the estimation. In this study, we have generated plant's disturbance with a variance of 0.0025.

Since our method relies on the fact that norms of the vectors will be greater than zero and in the plant's input we may have zero values in some occasion, hence we need to modify (7) in order to be meaningful here as follows:

$$\varphi_{il}(\mathbf{x} \cdot \mathbf{c}_i) \triangleq \cos(\theta_i) = \frac{\mathbf{x} \cdot \mathbf{c}_i}{\epsilon + \|\mathbf{x}\| \cdot \|\mathbf{c}_i\|}, \quad (14)$$

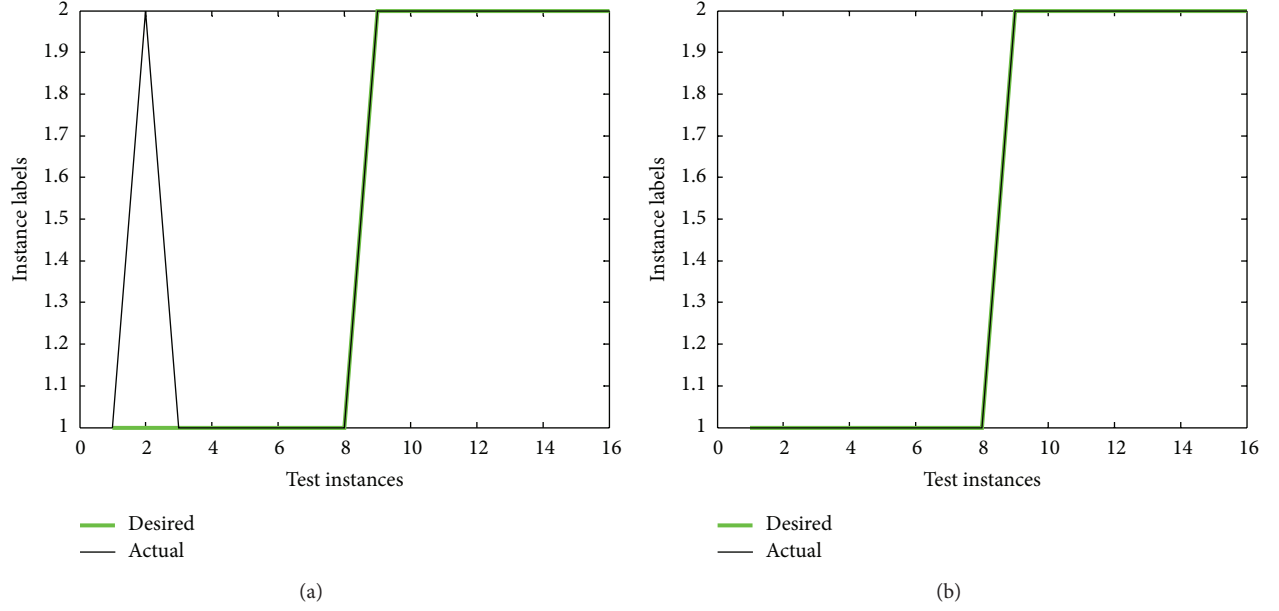


FIGURE 6: (a) Output after 100 epochs using proposed kernel: misclassification exists. (b) Output after 150 epochs using proposed kernel: 100% accuracy is achieved.

where the term  $\epsilon > 0$ ; a very small number is added to the denominator in order to avoid divide by zero scenario.

As before, we have used adaptive kernels for both the conventional and the proposed RBF. We tried different combinations of  $\eta$ ,  $\beta$ , and  $N$  but ultimately got 100% accuracy at  $\eta = 0.05$ ,  $\beta = 1$ , and  $N = 41$ . The results of this application are shown in Figures 10-11 which are discussed in Section 5.2.

## 6. Results and Discussions

**6.1. Recovery of MPSK Modulated Signals in Noisy Environment.** In this application, we compare the total mean square error (MSE) of our algorithm with that of conventional ED based RBF. The MSE of the two algorithms is computed as the sum of average value of the squared error and mathematically it can be defined as follows:

$$\text{MSE} = \sum_{k=1}^T E(\mathbf{o}'_i - \mathbf{o}_i)^2, \quad (15)$$

where  $T$  is the total number of epochs,  $\mathbf{o}$  is vector of true symbols,  $\mathbf{o}'$  is the vector of predicted symbols, and  $E$  is the expectation operator showing ensemble average. For comprehensive comparison, we investigated the performance of the two algorithms for three values of signal-to-noise ratio (SNR) which are 10 dB, 20 dB, and 30 dB. These results are reported in Table 1, which show that the proposed RBF kernel has superior performance over the conventional one.

Moreover, the scatter plot of the recovered signals at 10 dB SNR is shown in Figure 13, which clearly shows that the effect of noise is eliminated efficiently by the proposed RBF kernel.

TABLE 1: Comparison of the total mean square error.

SNR	Total MSE of the proposed kernel	Total MSE of The RBF kernel
10	0.0958	0.1098
20	0.0097	0.0110
30	$9.6461 \times 10^{-4}$	0.0011

**6.2. Classification of Leaves.** Through statistical data analysis, we have evaluated our claim that, for our data, cosine RBF is more powerful than its Euclidean counterpart. *Silhouette widths*, which were first described by Rousseeuw [33], provide a succinct graphical representation of how well each object lies within its cluster. In other words, we can visualize how well the algorithm associates each data point with its center.

For validation with standard approach, we used k-means and Silhouette functions from MATLAB statistics toolbox. The result is shown in Figure 3. Ideally, observations with large and positive Silhouette value ( $\sim 1$ ) are well clustered, those with Silhouette value around 0 lie between clusters, and those with negative Silhouette value are placed in “wrong” cluster. It is worth noticing that in Figure 3(a) where we used Euclidean kernel some sample values are clustered as negatives indicating misclassification. We found that it is fixed in Figure 3(b) where we used cosine kernel.

Our observation is directly backed by testing accuracies of both the approaches. We started our experiments with 100 epochs for both the conventional RBF kernel and our proposed kernel and repeated the experiments for many runs. In every subsequent run, we increased the number of epochs by 50. However, we achieved 100% accuracy with our kernel once it reached to 150 epochs as shown in Figure 8. In

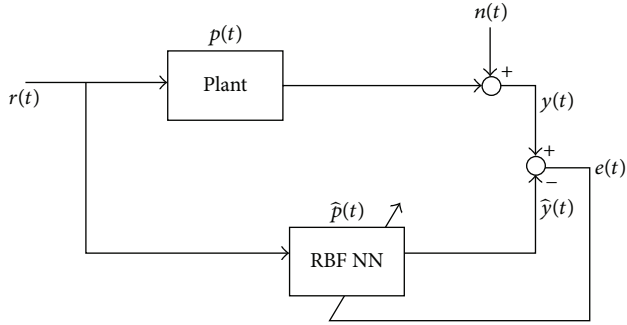


FIGURE 7: RBF NN based modeling of a nonlinear plant.

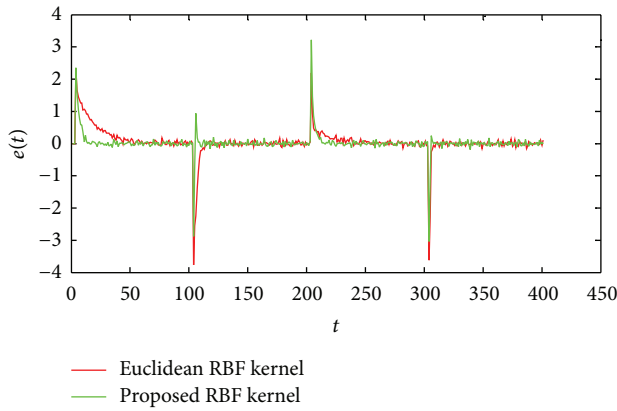


FIGURE 8: Comparing plant identification error for the two RBF kernels: Euclidean and proposed.

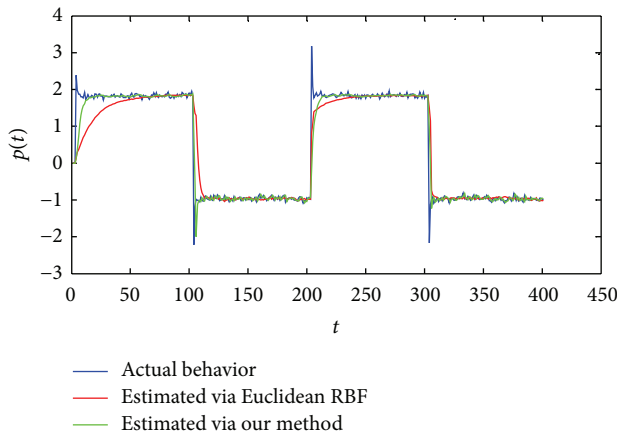


FIGURE 9: Comparing the actual plant's output with the two RBF kernels: Euclidean and proposed.

Figure 7, we noticed that Euclidean kernel takes 3500 epochs to acquire 100% classification accuracy, which is much higher than the one obtained by the proposed kernel. This fact is also reflected in Figure 6, where we show the comparative objective functions for the three approaches used and it is evident from there that both the proposed kernel and only cosine part of it work very well with this dataset, whereas the conventional kernel takes much more epochs to minimize the objective function.

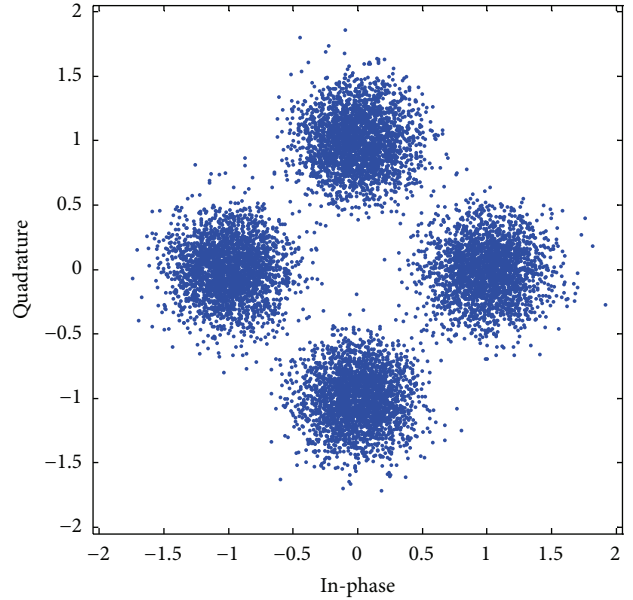
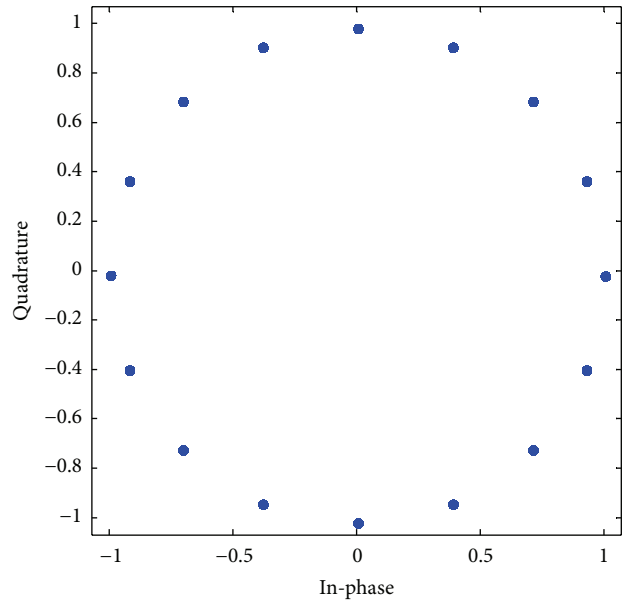
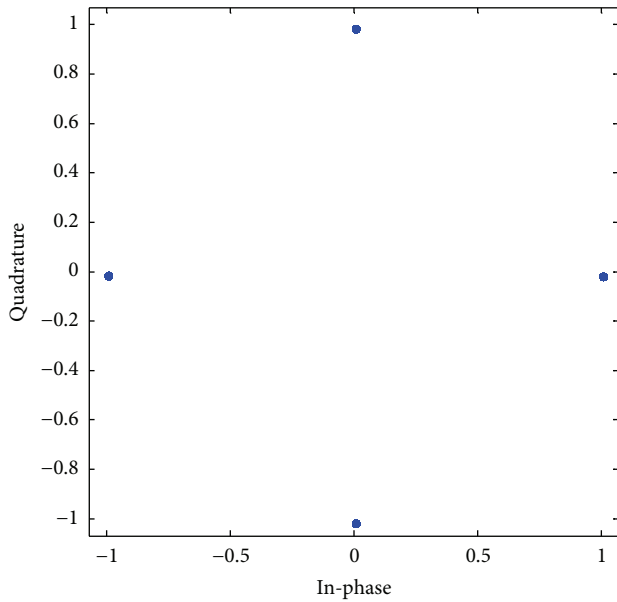
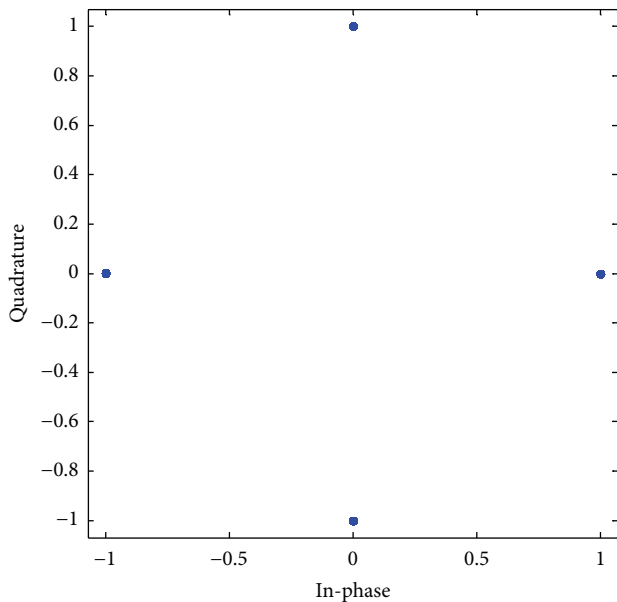


FIGURE 10: Noisy PSK scatter plot.

FIGURE 11: Transmitted symbols scatter plot for  $M = 16$ .

**6.3. Nonlinear Plant Identification.** The plant's identification error for both the RBFs is shown in Figure 10 which indicates that our method converges faster than ED based RBF. Moreover, it can be seen from Figure 11 that the proposed kernel emulating the plants behavior very well except in transition phases where it produces spikes, which results due to the state transition of the input square wave. Furthermore, the proposed RBF has faster convergence speed as compared to its counterpart; that is, it takes smaller number of iterations to get accustomed again as compared to the conventional RBF as evident from Figure 11.



FIGURE 12: Transmitted symbols scatter plot for  $M = 4$ .FIGURE 13: Recovered symbols scatter plot for  $M = 4$ .

## 7. Conclusion

In this work, we have introduced a new generalized RBF kernel by fusing the conventional ED kernel and the proposed cosine kernel. The proposed RBF kernel promises good performance in the scenarios where the angle between the feature and center vectors is distinguishable. This fact is also observed in our application studies where we have shown via statistical analysis of Silhouette widths why the proposed RBF kernel is more suitable in such scenarios. For validating the performance of the proposed RBF kernel, we have investigated three diverse nature of applications:

adaptive symbol detection of MPSK modulated symbols, classification of leaves, and nonlinear plant identification. Our algorithm has outperformed the conventional RBF kernel in terms of recognition accuracy and run time in epochs. We have achieved 100% accuracy in case of pattern recognition with faster convergence, for the adaptive symbol detection example the effect of noise is eliminated more efficiently by the proposed RBF kernel and in case of nonlinear plant identification we observed that our kernel converges faster and traces the nonlinear plant output function better than its conventional counterpart. Our work promises to find more interesting applications in other research areas too.

## Abbreviations

ANN: Artificial neural networks  
 RBF: Radial basis function  
 ED: Euclidean distance  
 SVM: Support vector machine  
 MPSK: M-ary phase shift keying  
 MSE: Mean square error.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This paper was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah. Therefore, the authors acknowledge, with thanks, the DSR financial support.

## References

- [1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [2] J. Khan, J. S. Wei, M. Ringnér et al., "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, vol. 7, no. 6, pp. 673–679, 2001.
- [3] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: the state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.
- [4] D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-dynamic programming: an overview," in *Proceedings of the 34th IEEE Conference on Decision and Control*, vol. 1, pp. 560–564, December 1995.
- [5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1994.
- [6] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.
- [7] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.

- [8] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [10] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, no. 3, pp. 321–355, 1988.
- [11] D. H. Ackley and G. E. Hinton, "A learning algorithm for boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, 1985.
- [12] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.
- [13] H. Leung, T. Lo, and S. Wang, "Prediction of noisy chaotic time series using an optimal radial basis function neural network," *IEEE Transactions on Neural Networks*, vol. 12, no. 5, pp. 1163–1172, 2001.
- [14] S. Seshagiri and H. K. Khalil, "Output feedback control of nonlinear systems using RBF neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 1, pp. 69–79, 2000.
- [15] J. Platt, "A resource-allocating network for function interpolation," *Neural Computation*, vol. 3, no. 2, pp. 213–225, 1991.
- [16] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Advances in Neural Information Processing Systems*, vol. 7, pp. 231–238, MIT Press, Cambridge, Mass, USA, 1995.
- [17] D. Wettschereck and T. Dietterich, "Improving the performance of radial basis function networks by learning center locations," in *Advances in Neural Information Processing Systems*, vol. 4, Morgan Kaufmann, San Mateo, Calif, USA, 1992.
- [18] S. S. A. Ali, M. Moinuddin, K. Raza, and S. H. Adil, "An adaptive learning rate for RBFNN using time-domain feedback analysis," *The Scientific World Journal*, vol. 2014, Article ID 850189, 9 pages, 2014.
- [19] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris, and D. M. Hummels, "On the training of radial basis function classifiers," *Neural Networks*, vol. 5, no. 4, pp. 595–603, 1992.
- [20] R. J. Schilling, J. J. Carroll Jr., and A. F. Al-Ajlouni, "Approximation of nonlinear systems with radial basis function neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 1, pp. 1–15, 2001.
- [21] M. J. Er, W. Chen, and S. Wu, "High-speed face recognition based on discrete cosine transform and RBF neural networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 679–691, 2005.
- [22] F. Rudzicz, "Articulatory knowledge in the recognition of dysarthric speech," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 947–960, 2011.
- [23] Y. J. Zeng and J. D. Zhuang, "Construction cosine radial basic function neural networks based on artificial immune networks," in *Advanced Data Mining and Applications*, vol. 6441 of *Lecture Notes in Computer Science*, pp. 134–141, Springer, Berlin, Germany, 2010.
- [24] K. Zhang, Y. Li, P. Scarf, and A. Ball, "Feature selection for high-dimensional machinery fault diagnosis data using multiple models and radial basis function networks," *Neurocomputing*, vol. 74, no. 17, pp. 2941–2952, 2011.
- [25] U. R. Acharya, F. Molinari, S. V. Sree, S. Chattopadhyay, K. Ng, and J. S. Suri, "Automated diagnosis of epileptic EEG using entropies," *Biomedical Signal Processing and Control*, vol. 7, no. 4, pp. 401–408, 2012.
- [26] D. Sridhar and M. Krishna, "Brain tumor classification using discrete cosine transform and probabilistic neural network," in *Proceedings of the International Conference on Signal Processing, Image Processing and Pattern Recognition (ICSIPR '13)*, pp. 92–96, Coimbatore, India, February 2013.
- [27] N. B. Karayiannis and M. M. Randolph-Gips, "On the construction and training of reformulated radial basis function neural networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 4, pp. 835–846, 2003.
- [28] Z. Liu, X. Zhao, M. J. Zuo, and H. Xu, "Feature selection for fault level diagnosis of planetary gearboxes," *Advances in Data Analysis and Classification*, 2014.
- [29] Y. Cho and L. K. Saul, "Kernel methods for deep learning," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS '09)*, pp. 342–350, December 2009.
- [30] N. B. Karayiannis and Y. Xiong, "Training reformulated radial basis function neural networks capable of identifying uncertainty in data classification," *IEEE Transactions on Neural Networks*, vol. 17, no. 5, pp. 1222–1234, 2006.
- [31] T. M. Cover, "Geometrical and Statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Transactions on Electronic Computers*, vol. 14, no. 3, pp. 326–334, 1965.
- [32] D. Psaltis, A. Sideris, and A. A. Yamamura, "A multilayered neural network controller," *IEEE Control Systems Magazine*, vol. 8, no. 2, pp. 17–21, 1988.
- [33] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.
- [34] L. I. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 281–286, 2002.
- [35] C. Mallah, J. Cope, and J. Orwell, "Plant leaf classification using probabilistic integration of shape, texture and margin features," in *Proceedings of the International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA '13)*, pp. 279–286, February 2013.
- [36] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.
- [37] V. T. S. Elanayar and Y. C. Shin, "Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 594–603, 1994.
- [38] R. P. Lippmann, "Pattern classification using neural networks," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 47–50, 1989.
- [39] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 697–710, 2002.