

Learning Enhancement of Radial Basis Function Neural Network with Harmony Search Algorithm

Mohamed Hassan Ahmed^{1,2}, Shafaatunnur Hasan^{1,2}, and Aida Ali^{1,2}

¹UTM Big Data Centre, Ibnu Sina Institute for Scientific and Industrial Research
Universiti Teknologi Malaysia, 81310 UTM Skudai,
Johor, Malaysia

e-mail: mxaxmed114@simad.edu.so, shafaatunnur@utm.my, aida@utm.my

²Faculty of Computing, Universiti Teknologi Malaysia
81310 UTM Skudai,
Johor, Malaysia

e-mail: shafaatunnur@utm.my, aida@utm.my

Abstract

Training Radial Basis Function (RBF) neural network with Particle Swarm Optimization (PSO) was considered as a major breakthrough that overcome the stuck to the local minimum of Back Propagation, time consuming and computation expensive problems of Genetic Algorithm. However, PSO converged too fast, and hence stuck to the local optimum. Furthermore, particles may move to an invisible region. Therefore, to realize the enhancement of the learning process of RBF and overcome these PSO problems, Harmony Search Algorithm (HSA), a new meta-heuristic algorithm was employed to optimize the RBF network and to attain the desired objectives. The study has conducted comparative experiments between the integrated HSA-RBF network and the PSO-RBF network. The results proved that HSA increased the learning capability of RBF neural network in terms of accuracy and correct classification percentage, error convergence rate, and less time consumption with less mean squared error. The new HSA-RBF model provided higher performance in most cases and give promising results with better classification proficiency compared with that of PSO-RBF network.

Keywords: *Radial Basis Function, Meta-heuristic, Harmony Search Algorithm, Particle Swarm Optimization, Stochastic Global Optimization.*

1 Introduction

Artificial Neural Network's mimicking ability to human talent and their similarity to the structure of the neurons of the human brains attracted the eyes of many researchers due to its unparalleled properties, such as adaptability, learning and generalization capability [43]. Basically the principles of the Artificial Neural Networks (ANNs) were first formulated by McCulloch and Pitts in 1943 [32]. According to Chan *et al.* [14], Neural Network not only have the competence to learn a complex nonlinear dataset from massive body of given attributes, but can tolerate to fault and noisy condition in resemblance to human brain as well.

One of the outstanding examples of Neural Networks is Radial Basis Function (RBF). According to Gan *et al.* [22], RBF Neural Network which was originally conceived by Broomhead and Lowe in 1988 has characterized with fast training speed, strong learning capability and simple topological architecture. Idri *et al.* [35] and Gan *et al.* [23], described that it consists of only three different layers which are the input layer which accepts source dataset; the hidden layer that uses radial basis function to compute its output, and the output layer which represents the result of the network.

The emergence of radial bases function as an alternative of ANN was first perceived in late 80's, although their related counterpart, pattern recognition technique has existed long ago [9]. RBF Neural Network was originally perceived and added to the ANN by Broomhead and Lowe (1988), who were inspired by the local response observation in the biologic neurons. RBF Networks have been implemented in a wide area of engineering and science fields, because of their advantages over other well known networks such as: their optimized ability, simple topological architecture, accuracy in dynamically nonlinear approximation and fast and easy learning algorithms [22].

According to Kurban *et al.* [40], in various literatures, different algorithms were proposed for training the RBF Network. Therefore, it is necessary to find the appropriate training algorithms for the RBF Neural Network. One of the most popular training algorithms in the domain of RBF Neural Networks is the Back-Propagation technique, which introduces gradient-descent method to minimize the mean squared error between the desired outputs and the actual outputs for the particular inputs to the networks. However, according to Pan *et al.* [43], Kattan *et al.* [37] and Hamed *et al.* [1] training RBF with BP faced some problems such as poor convergence and trapping at the local minima.

To deal with these convergence problems, some researchers proposed two derivative based algorithms for training RBF networks, such as the gradient descent (GD) algorithm and Kalman filtering (KF). Kurban and Beşdok [40] proved that both algorithms need a lengthy time and have convergence weaknesses to the local minima and procedure of discovering the optimal gradient. Genetic Algorithm performed robust training without suffering from local

minimum problem. However, its output production is time consuming and computation expensive [34, 54].

Particle Swarm Optimization attracted the attention of many researchers after several experiments proved its performance is better than GA. Although the experiments conducted by many researches showed a plausible achievement, Rini *et al.* [47] mentioned that PSO easily suffers from the partial optimism, which is related to the regulation of its speed and direction. Moreover Grosan and Abraham [33] summarized the pitfalls of PSO in their book of “Intelligent Systems: A Modern Approach”, with the following three problems: firstly, particles tend to cluster, converge too fast and get stuck at local optimum. Secondly, the movement of particle carried it into infeasible region and finally, inappropriate mapping of particle space into solution space.

Pan *et al.* [43] proved that HS which is a new SGO meta-heuristic algorithm is a good candidate and the most promising variant for training feed forward NNs. More over Soltani *et al.* [50] confirmed that HS is not only faster than PSO but also has a significant convergence rate to reach the optimal solution.

This paper employs Harmony Search Algorithm (HSA) to improve the learning capability of RBF neural network and to achieve higher convergence rate and classification performance compared with PSO based RBF neural network.

2 Radial Basis Function Neural Network

As in Alejo *et al.* [5] RBF Neural Network is an extremely powerful type of feed-forward Neural Network first perceived by Broomhead and Lowe. RBF strongly differ from MLP in terms of usage and activation functions.

In its basic form, Radial Basis function can be defined as:

$$f(x) = \sum_{i=1}^k w_i h_i(x) + b = \sum_{i=1}^k w_i \exp\left(-\frac{\|x-c_i\|^2}{\sigma_i^2}\right) + b \quad (1)$$

According to Qasem *et al.* [45] and Kurban and Beşdok [40], RBF network gained their popularity due to their simpler architecture, faster learning process, and better approximation capabilities compared with other types of ANN. Moreover, RBF have attracted a considerable attention and have been applied in many science and engineering fields.

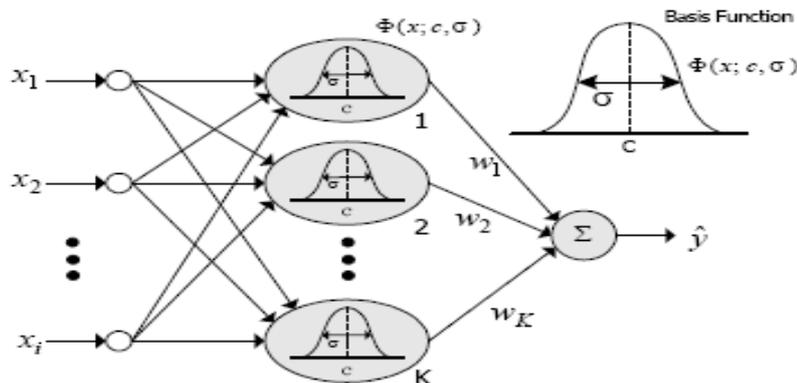


Fig. 1: The structure of RBF Neural Network [42]

2.1 RBF Neural Network architecture

Tsimboukakis and Tambouratzis [51], highlighted that RBF structure strictly comprises three layers as shown in Fig. 1. The first layer is the input layer that deals with the transmission of the inputs to the network. The second layer, known also as the hidden layer usually contains the activation function of the radial basis function. This activation function depends on the Euclidian distance from the input pattern vector to the center of hidden neuron. Finally, the third or the output layer usually contains a linear activation function. This output neuron is similar to a simple perceptron that obtain its activation as a weighted sum of the hidden neurons' outputs and supplies the results to the network.

RBF Neural Network configuration which implements the output-input relation is a nonlinear mapping with linear mapping composition realized by the hidden layer and the output layer respectively [35]. The next section will describe several RBFN functions in accordance with their activation functions.

2.2 RBF activation functions

Wang [53], ruled out that the value of RBF depends on the distance from the origin, i.e., $\phi(\mathbf{x}) = \phi(\|\mathbf{x}\|)$; or alternatively on the distance from some other point \mathbf{c} , called a center, so that $\phi(\mathbf{x}, \mathbf{c}) = \phi(\|\mathbf{x} - \mathbf{c}\|)$. Any function ϕ with the satisfying property of $\phi(\mathbf{x}) = \phi(\|\mathbf{x}\|)$ is radial basis function. The norm is usually Euclidian distance.

The most common types of RBFs include $r = \phi(\|\mathbf{x}\|)$:

Gaussian:	$\phi(r) = \exp(-\beta r^2)$	for some $\beta > 0$
Multiquadric:	$\phi(r) = \sqrt{r^2 + \beta^2}$	for some $\beta > 0$
Polyharmonic spline:	$\phi(r) = r^k, k = 1,3,5, \dots$ and $\phi(r) = r^k \ln(r), k = 2,4,6, \dots$	
Thin plate spline:	$\phi(r) = r^2 \ln(r)$	

Gaussian function is the most commonly used radial basis function.

2.3 Training RBF Neural Networks

According to Dhubkarya *et al.* [17], RBF networks are used mainly in supervised applications. In a supervised application, the training set of the data samples are usually provided and from there the corresponding network outputs are known.

Yu *et al.* [55] and Dhubkarya *et al.* [17] expressed the simplicity of the design of RBF Networks by organizing it into three quite straightforward steps:

- i. Find the number of hidden units
- ii. Determine the parameters of each RBF unit

iii. Train the RBF Networks to adjust parameters.

To determine the weights, the centers, and the width, a training process is used. Optimization algorithm for minimizing a suitable error function (E) are applied to compute the weights [34].

2.4 Construction of the hidden layer of RBF networks

The construction of hidden layer of the RBF networks can be achieved using clustering techniques. The role of clustering in the design of RBFN is to set up an initial distribution of receptive fields (hidden neurons) across the input space of the input variables [35].

According Xie *et al.* [54] and Idri *et al.* [35], the basic computations in the Radial Basis Function network above include:

i. Input layer computation

At the input of hidden unit l , the input vector x is weighted by input weights w^h :

$$s_l = [x_1 w_{1,l}^h, x_2 w_{2,l}^h \dots \dots x_n w_{n,l}^h \dots \dots x_N w_{N,l}^h] \tag{2}$$

where: n is the index of input; l is the index of hidden units; x_n is the n^{th} input; $w_{n,l}^h$ is the input weight between input n and hidden unit l .

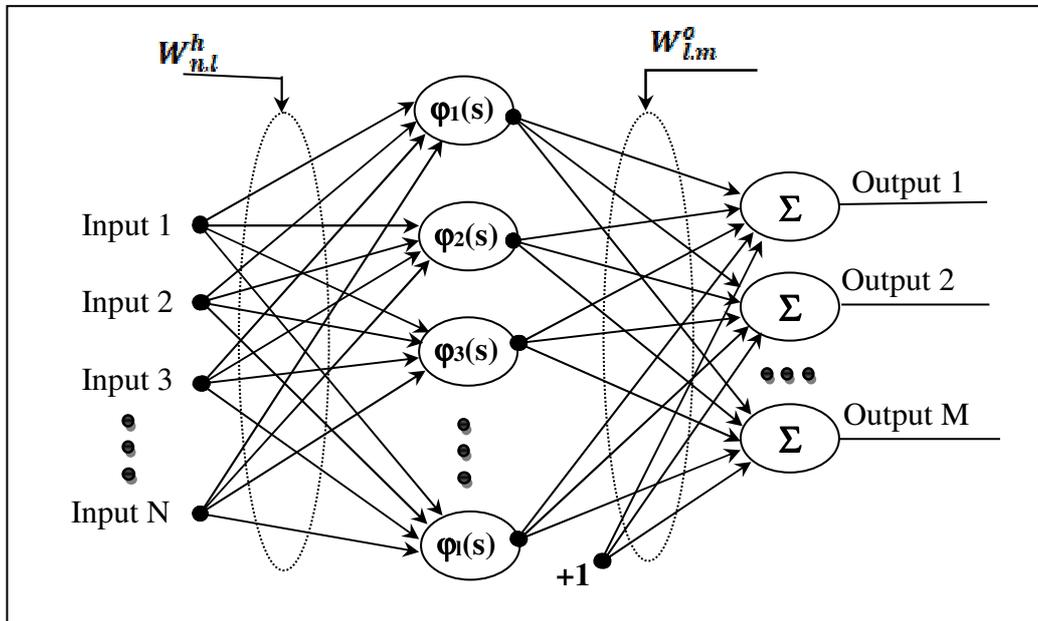


Fig. 2: RBF network with N inputs, L hidden units and M outputs [54]

ii. Hidden layer computation

The output of the hidden unit l is calculated by:

$$\varphi_l(s_l) = \exp\left(-\frac{\|s_l - c_l\|^2}{\sigma_l}\right) \quad (3)$$

Where: the activation function $\varphi_l(\cdot)$ for hidden unit l is normally chosen as Gaussian function; c_l is the center of hidden unit l and σ_l is the width of hidden unit l and $\|\cdot\|$ denotes the Euclidian distance. This implies that $\varphi_l(s_l)$ has an applicable value only when the distance $\|s_l - c_l\|$ is smaller than the width σ_l [17].

iii. Output layer computation

The network output m is calculated by:

$$o_m = \sum_{l=1}^L \varphi_l(s_l) w_{lm}^o + w_{0,m}^o \quad (4)$$

Where: m is the index of output; w_{lm}^o is the output weight between hidden unit l and output unit m ; $w_{0,m}^o$ is the bias weight of output unit m .

You may notice from this basic computation the existence of four types of parameters, the input weight matrix w^h , the output weight matrix w^o , the center matrix c and the width vector σ . Normally, the input weights are all set as '1' [54]. In the subsequent section, the other three parameters will be discussed briefly.

iv. Computation of the centers

As in Bors [9], Gan et al. [21] and Noman et al. [42], K-means clustering algorithms, learning vector quantization, decision trees, or self-organizing feature maps can be used to find the hidden unit parameters. The centers of the radial basis functions are initialized randomly. Upon averaging each cluster the centroid vector is updated with the following formula [34, 53]:

$$c_k = \frac{1}{R} \sum_{k=1}^R x_k \quad (5)$$

Where: c_k is the center of the cluster k which has R elements

v. Computation of the widths

Determination of the widths of unit function and its inherent complexity is considered to be one of the most important tackled aspects. To calculate the widths for each hidden neuron of RBFN with the variance σ^2 , the following formula is used as in Hamadneh et al. [34]:

$$\sigma_k^2 = \frac{1}{R_k} \sum_{x \in z_k} |x - c_k| \quad (6)$$

Where: σ_k is the width of the neuron k. R_k is the number of elements in the cluster Z_k

vi. Computation of weights

Idri *et al.* [35] illustrated when the center and the width of RBFN have been evaluated, the weights of the output layer can be determined and adjusted. As in Wang [53] the learning rules of the output layer's weights can be written as:

$$b_i(\mathbf{x}) = \frac{\exp\left(-\frac{\|\mathbf{x}_p - \mathbf{c}_i\|^2}{2\delta_i^2}\right)}{\sum_{l=1}^m \exp\left(-\frac{\|\mathbf{x}_p - \mathbf{c}_l\|^2}{2\delta_l^2}\right)} \quad (7)$$

$$\Delta w_{ij} = -\frac{\eta}{m} \sum_{p=1}^m [y_j'(\mathbf{x}_p) - y_j(\mathbf{x}_p)] * b_i(\mathbf{x}_p) \quad (8)$$

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (9)$$

Here $y_j'(\mathbf{x}_p)$ is the actual output, $y_j(\mathbf{x}_p)$ is the target output and η is the learning rate.

It is customary to train artificial neural networks before they can achieve their desired task. Some learning algorithms that will be used to train RBF network will be discussed in the subsequent sections.

3 Harmony Search Algorithm

According to Geem and Williams [26] and Geem *et al.* [29], solutions based on traditional mathematics such as linear programming, nonlinear programming and dynamic programming suffered from losses on the development of nonlinear real world problems, missing the optimality in non-differentiable computing functions and dependence of the increase in number of variables and number of evolutions of the recursive functions respectively. To overcome these deficiencies, researchers introduced heuristic optimization techniques based on simulations.

Although the aforementioned heuristic algorithms solved several shortcomings of the traditional mathematical methods, however, researchers asked themselves whether it is possible to develop a heuristic algorithm that can perform better with less iteration in compared with the existing ones. Geem *et al.* [29] developed a new meta heuristic stochastic global optimization (SGO) method and named it a harmony search algorithm (HSA) [13 and 37].

Harmony search (HS) is a recent stochastic search mechanism inspired not by biological nor physical processes but by the natural improvisation process of music players in which the notes played by each musician is tested in order to find the best harmony of all [36, 56]. When applied to optimization problems, the

musicians represent the decision variables of the cost function, and HS acts as the meta heuristic algorithm which attempts to find a solution vector that optimizes this function. During the process, each decision variable (musician) generates a value (note) for finding global optimum (best harmony).

According to Pan *et al.* [43], HS imposes a few mathematical obligations compared with the other existing meta heuristics algorithms. According to Alia and Mandava [6], HS has attracted many researchers to use it as an easy and successful benchmark in solving many engineering optimization problem such as: music composition [24], Sudoku puzzle solving [31], tour planning [30], web page clustering [20], ground water modeling [8], soil stability analysis [15], ecological conservation [26], energy system dispatch [52], heat exchanger design [19], transportation energy modeling [12], satellite heat pipe design [25], medical physics [44], RNA structure prediction [41], Medical image [6,7], time tabling [2,3,4], image segmentation [8], etc.

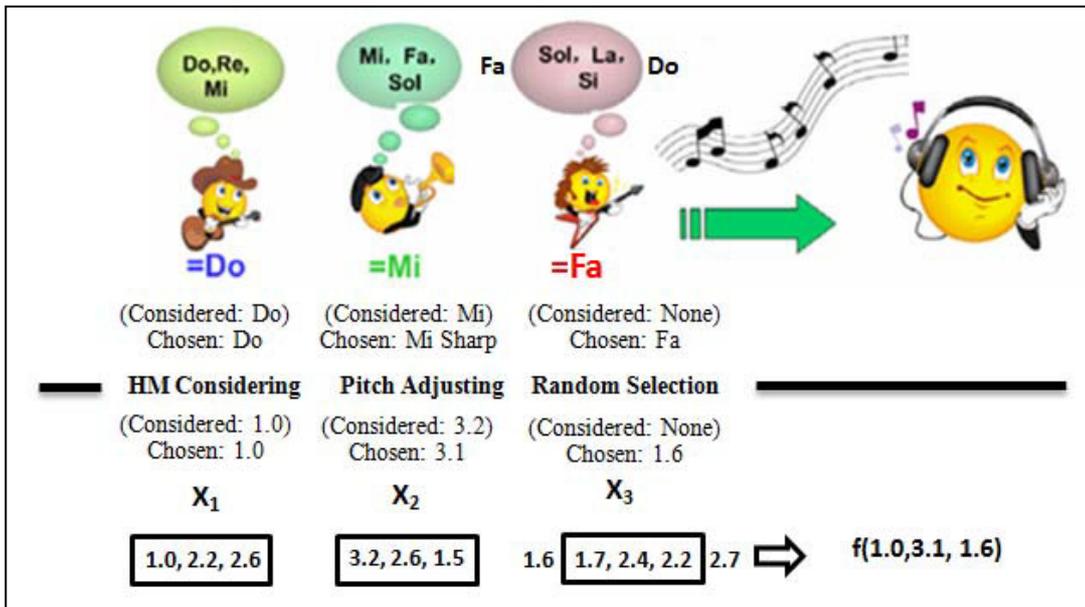


Fig. 3: Analogy between improvisation and optimization [38]

Table 1: The optimization terms in the musical context [3]

Musical Terms	Optimization Terms
Improvisation	Generation or construction
Harmony	Solution vector
Musician	Decision variable
Pitch	Value

Pitch range	Value range
Audio-aesthetic standard	Objective function
Practice	Iteration
Pleasant harmony	(Near-) optimal solution

HS isn't like gradient in differential calculus that deals with continuous variable, instead it is a stochastic derivative based on the population density of musicians experience that deals with discrete variables [29].

Similar to Table 1, Alia and Mandava [6] summarized the analogy between the improvisation and the optimization process expressed in [28] in four easy steps listed as follows:

1. Each musician corresponds to each decision variable.
2. Musical instruments' pitch range corresponds to the decision variable's value range.
3. Musical harmony at a certain time corresponds to the solution vector at a certain iteration.
4. Audiences'-aesthetics corresponds to the objective function.

The five steps of the optimization process of the basic harmony search are as follows: [3 and 48].

- Step 1:** Initialize the problem and HSA parameters.
- Step 2:** Initialize a Harmony Memory (HM).
- Step 3:** Improvise a new harmony from HM.
- Step 4:** If the new harmony is better than minimum harmony in HM, include the new harmony in HM, and exclude the minimum harmony from HM.
- Step 5:** If stopping criteria are not satisfied, go to Step 3.

4 Data Representation

The process of training RBF neural network needs determining the weights between the hidden layer and the output layer of the network which minimizes the error and training and testing datasets that should be fully implanted. The harmony vector in HM represents the weights of the RBFNN and the XOR, Iris, Cancer and Heart Disease Classification datasets are used to ensure reliable results.

4.1 HS parameter setting

In the process of training RBFN with HS algorithm, there are five parameter settings have been employed in all experiments as shown below:

Parameter Name	Value
Harmony memory size (HMS)	15
Harmony memory consideration rate (HMCR)	0.9
Pitch adjusting rate (PAR)	0.3
Number of iteration (NI)	100
Number of decision variables (n)	2

4.2 Datasets

XOR Dataset: A connective in logic known as the "exclusive or" or exclusive disjunction is a logical operation on two operands that results in a logical value of true if and only if one of the operands but not both has a value of true. XOR is a basic dataset that widely use to train and test NN. In this study, 4 data patterns are used in both algorithms.

Iris Dataset: The dataset is small consisting of 150 records. The target variable is categorical specifying the species of iris. The predictor variables are measurements of plant dimensions. The dataset is used for classifying all the information into three classes which are iris setosa, iris versicolor, and iris virginica. The classification is based on its four input pattern which are sepal length, sepal width, petal length and petal width. Each class refers to type of iris plant contain 50 instances. In NN learning, the network has four input patterns and 3 output patterns.

Breast cancer dataset: The dataset was originally generated at hospitals at the University of Wisconsin Madison, by Dr. William H. Wolberg to correctly diagnose breast lumps as either benign or malignant based on data from automated microscopic examination of cells collected by needle aspiration. The dataset includes nine inputs and one output. The exemplars are split with 599 for training, and 100 for testing, totaling 699 exemplars. All inputs are continuous variables and 458 (65.5%) of the examples are benign and the remaining 241 (34.5%) are malignant.

Heart disease dataset: This database is taken from the Cleveland Clinic Foundation and was supplied by Robert Detrano, M.D., Ph.D. of the V.A. Medical Center, Long Beach, CA. It is part of the collection of databases at the University of California, Irvine (UCI) collected by David Aha. The aim of the dataset is to classify the presence or absence of heart disease given the results of various medical tests carried out on a patient. The original dataset contains 13 numeric attributes and a fourteenth attribute indicating whether the patient has a heart condition. This dataset is interesting because it represents real patient data and has been used extensively for testing various data mining techniques.

4.3 Normalization of datasets

Normalizing data means fitting the data within unity so that all attributes will take a value between 0 and 1. Since some attributes may have inapplicable value that can be impossible to apply to some models, (i.e., non-numeric value or numbers that are too large or too small to represent, etc.). Gaussian RBF employs $\exp()$ function which cannot accept any number greater than 709.782, so it is necessary to normalize some datasets. The following formula is used to normalize datasets.

$$X_{i,0 \text{ to } 1} = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad (10)$$

Where X_i denotes each data point i , X_{\min} is the minimum amount in all data points in the dataset, X_{\max} is the maximum amount in all data points of the dataset and $X_{i,0 \text{ to } 1}$ is the data point at i that is normalized to between 0 and 1.

4.4 Partition of datasets

K-fold cross-validation which is the base for all other existing cross-validations is applied to these experiments. In k-fold cross validation, the datasets are partitioned into an equally or semi-equally k sized segments. Trainings and validations are performed on iteration of sub sequent k-iterations, holding out one segment of different fold of the data for validation and the remaining k-1 folds for training. In this experiment, a k=10 fold of cross validation is employed so that each data exemplars is subdivided into 10 folds.

5 Training RBF Neural Network Using HS Algorithm

Schwenker *et. al.* [49] distinguished three learning schemes for RBF which are, one phase learning in which only the weights of the output layer of the RBF is adjusted by some kind of optimization algorithm. The second scheme is two phase learning in which both layers of the RBF are trained by determining the centers and widths of the hidden layer and subsequently adjusting the weights of the output layer and finally, the third scheme is three phase learning in which after utilizing the two phase learning, a further optimization process is applied to adjust the whole architecture of the RBF.

This study implemented one-phase learning, whereby the weights of the output layer of the RBF network have been initialized by the HS optimization algorithm. The acquired difference between the desired target value and the actual output value is considered as an error rate and is minimized by updating and adjusting the output layer weights with fresh values obtained from HSA. New output has been regenerated repeatedly until set of conditions are satisfied or the end of loop is reached.

HSA-RBF networks are trained based on the following eight steps. Meanwhile, Fig. 4 illustrates the network architecture of HSA-RBF network.

- Step 1:** Obtain the dataset and normalize if necessary
Step 2: Decide on how many hidden neurons there should be using the formula: $\text{number of hidden nodes} = \sqrt{(\text{input} * \text{output})}$
Step 3: Decide on their centers and the sharpness of their Gaussians.
Step 4: Generate the best harmony from HSA and initialize the weights of the RBF output layer with this value.
Step 5: Start training the output layer.
Step 6: Calculate error rate by finding the difference between the desired output and the actual output values.
Step 7: If error rate beyond the threshold, go to step 4 to adjust the weights.
Step 8: If the condition is satisfied or end of iteration is reached, stop training process.

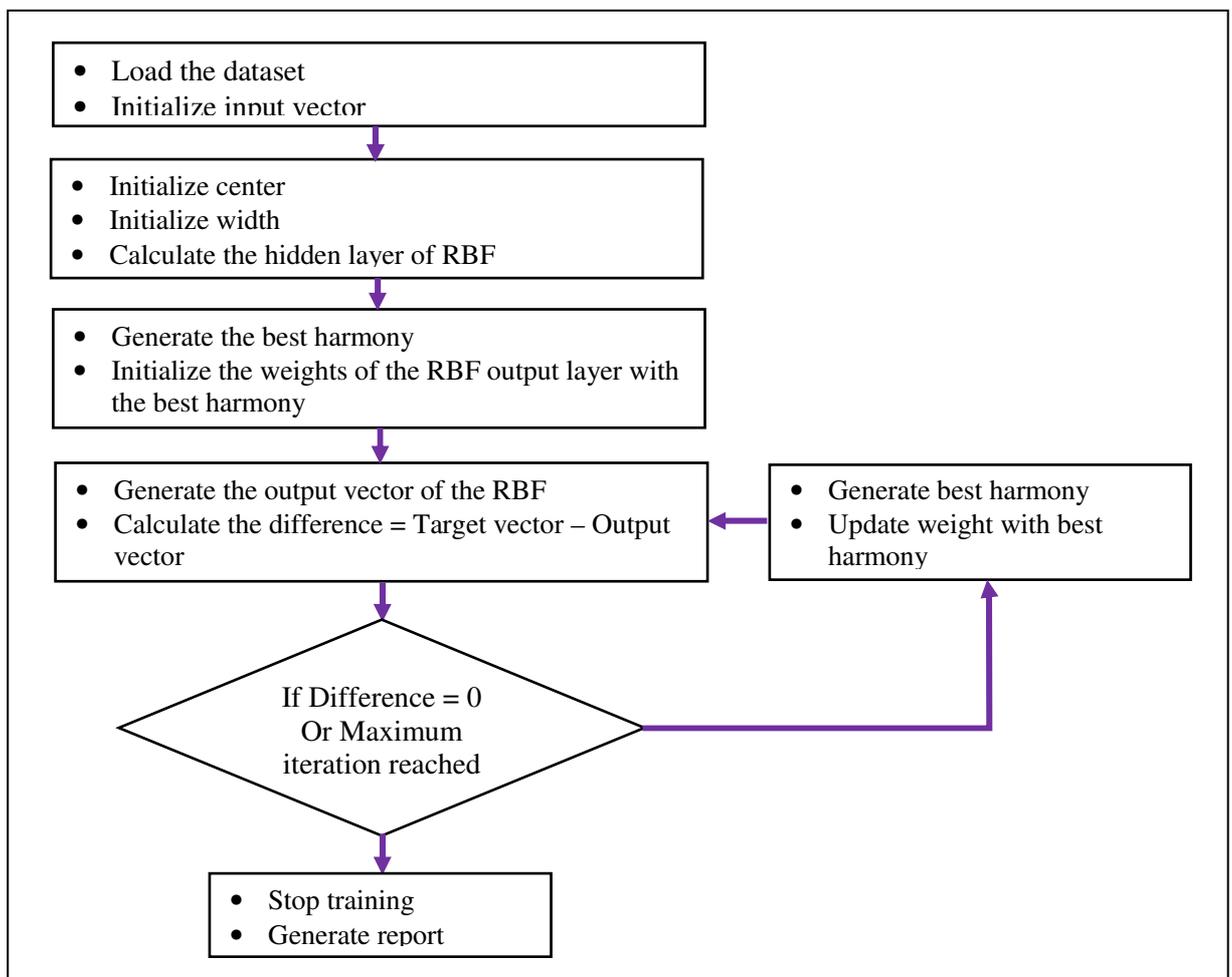


Fig. 4: The network architecture of HSA-RBF network

6 Results and Discussion

In this section, several substantial experiments are conducted. The normalized UCI machine learning repository datasets, such as, XOR dataset, iris dataset and breast cancer dataset are used for the experiments. The analysis from the experiments are given in the next sub-section.

6.1 Analysis on Iris dataset

In this problem, the iris dataset consisting of three classes; Setosa, Versicolour, and Virginica with 150 samples were divided into two parts, 120 input patterns are applied to train the network, and 30 input patterns are employed to test the network. The network contains four input nodes, three hidden nodes and three output nodes.

In iris learning, HSA-RBF completed the training with its best performance epoch at iteration 172 in 71.738 seconds. The correct classification percentage is 98.89% and MSE of 0.00833. Similarly the testing process of HSA-RBF with iris dataset reached the best performance epoch at iteration 3 with execution time of 16.442 seconds. This provides a significant improvement to the overall performance of RBF neural network in terms of time, error convergence and correct classification.

Table 2: Result of HSA-RBF on Iris Dataset

	Training	Testing
Number of iterations	500	500
Number of samples	120*3	30*3
No. of correct classifications	356	90
Execution time in seconds	71.74	16.44
Mean square error (MSE)	0.0083	0.000
Correct classification %	98.89%	100%

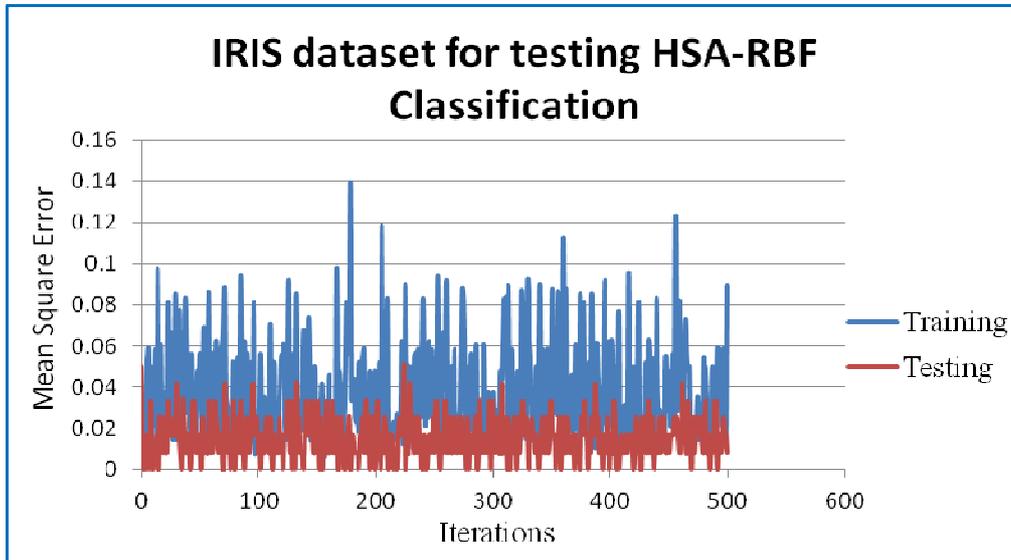


Fig. 5: Convergence of IRIS trained with HSA-RBF neural network

6.2 Analysis on cancer dataset

There are two classes in the breast cancer dataset; benign, and malignant. About 699 samples have been used for the experiment and the data are divided into two parts, 599 input patterns are applied to train the network, and 100 input patterns are employed to test the network. The network contains nine input nodes, three hidden nodes and one output node.

Table 3: Result of HSA-RBF on Cancer Dataset

	Training	Testing
Number of iterations	500	500
Number of samples	599	100
No. of correct classifications	552	96
Execution time in seconds	340.75	64.11
Mean square error (MSE)	0.0196	0.01
Correct classification %	92.15%	96%

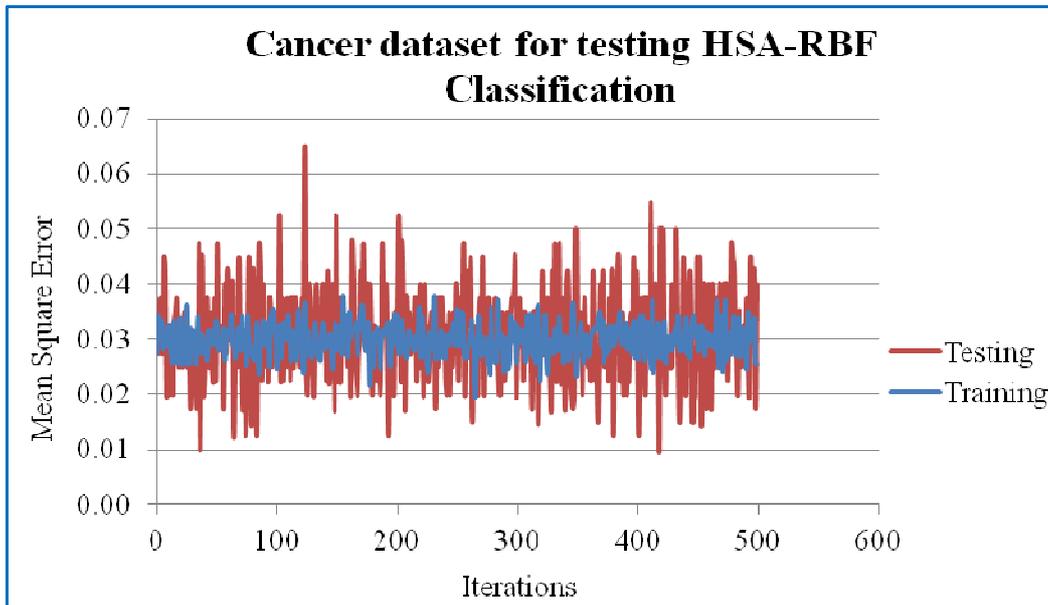


Fig. 6: Convergence of cancer dataset trained with HSA-RBF neural network

In breast cancer dataset, HSA-RBF completed training with the best performance epoch at iteration 266 in 340.75 seconds. The correct classification percentage is 92.15% and MSE value is 0.0196. Likewise, the testing process of HSA-RBF with cancer dataset found best performance epoch at iteration 37 with the execution time of 64.11 seconds and the error rate of 0.01.

This experiment has lowest performance compared with the other two experiments conducted in terms of time, error convergence and correct classification.

6.3 Analysis on heart disease dataset

The heart disease dataset used in this problem is starlog heart dataset. It consists of two classes; absence and presence of heart disease. There are 13 attributes extracted from a larger set of 75 attributes. The 270 samples contained by this dataset are divided into two parts, 216 input patterns are applied to train the network, and 54 input patterns are employed to test the network. The network contains 13 input nodes, three hidden nodes and one output node.

Table 4: Result of HSA-RBF on Heart Disease Dataset

	Training	Testing
Number of iterations	500	500
Number of samples	216	54
No. of correct classifications	206	54
Execution time in seconds	182.83	34.03
Mean square error (MSE)	0.011574	0.000
Correct classification %	95.37%	100.00%

In the heart disease learning, HSA-RBF completed training at iteration 212 in 182.83 seconds. The correct classification percentage is 95.37% and MSE value is 0.01157. Meanwhile, the testing process of HSA-RBF with heart disease dataset found best performance epoch at iteration 165 with execution time of 34.03 seconds. This provides a significant improvement to the overall performance of RBF neural network in terms of time, error convergence and correct classification.

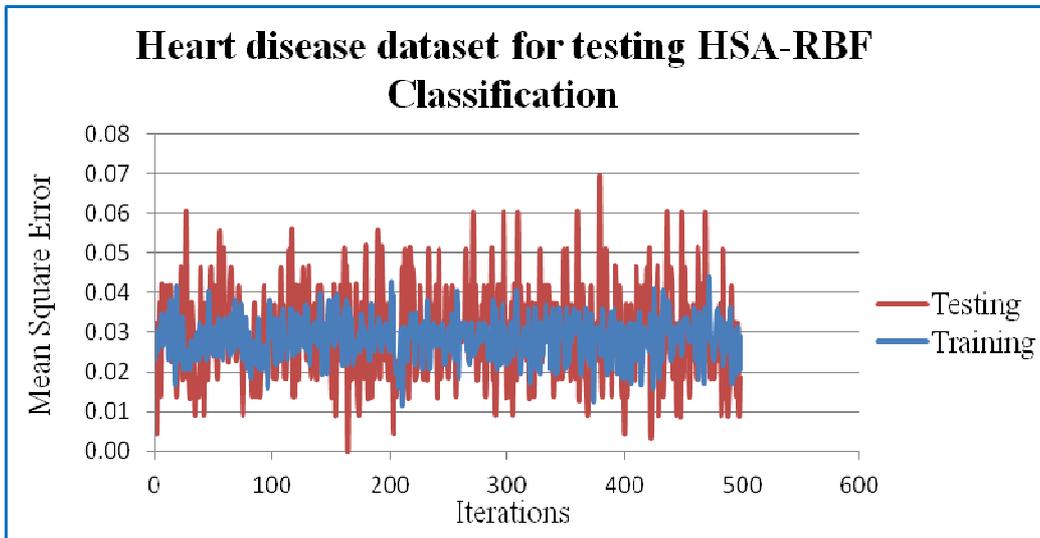


Fig. 7: Convergence of heart disease dataset trained with HSA-RBF

6.4 Performance comparisons

In this section, the classification results of the HSA-RBF are compared with the classification accuracy of PSO-RBF in several experiments conducted by Qasem [46] and Das [16] in their respective studies and RBF-Alone. The analysis uses four datasets, XOR, IRIS, Cancer and Heart disease datasets in order to compare

their correctly classified percentages. Table 6 and the graph in Fig. 8 depict their comparative results.

Table 5: Comparative summary of Standard RBF, HSA-RBF and PSO-RBF

Dataset	Standard RBF	HSA-RBF	PSO-RBF
	Correct Classification %	Correct Classification %	Correct Classification %
IRIS	95.28%	98.89%	95.48%
Breast Cancer	85.64%	92.15%	97.65%
Heart Disease	87.96%	95.37%	93.23%

According to the above summary, the XOR, Iris and Heart disease datasets demonstrated that HSA-RBF has better results with higher convergence rate in a short period of time than both RBF-Alone and PSO-RBF. On the other hand, breast cancer dataset depicted that for this time PSO-RBF has better convergence rate and higher correctly classified result than that of HSA-RBF. Although both algorithms converge to the solution with minimum error, however, HSA-RBF has significantly reduced the error rate with minimum iteration since HSA-RBF converged in 500 iterations while PSO-RBF converged at 93 iterations for EXOR, 3774 iterations for iris and 1000 iterations for cancer [46], and HSA-RBF classifications are better that that of RBF-Alone and PSO-RBF in most cases.

Generally, HSA-RBF shows an improved overall performance and considerable enhancement in terms of time, error convergence and classification process of RBF neural network compared with RBF-Alone and PSO-RBF neural network.

7 Validation Results

In this section datasets are validated in crossed-over successive rounds so that each data segment receives a chance to be validated against the network. Data cross-validations are used for evaluating or comparing learning algorithms. K-fold cross-validation which is the base for all other existing cross-validations is applied to these experiments.

Table 6: 10 Fold Cross-Validation of three datasets on HSA-RBF

No-CV	Iris		Breast Cancer		Heart disease	
	MSE	Classification	MSE	Classification	MSE	Classification
1	0.0074	99.01	0.0286	88.55	0.01235	95.06
2	0.0074	99.01	0.0223	91.10	0.01749	93.00

3	0.0074	98.51	0.0302	87.92	0.01337	94.65
4	0.0056	98.76	0.0254	89.83	0.01543	93.83
5	0.0125	97.78	0.0278	88.87	0.01337	94.65
6	0.0074	99.01	0.0250	89.98	0.02160	91.36
7	0.0074	99.01	0.02385	90.46	0.018520	92.59
8	0.0000	100.00	0.03021	87.91	0.015430	93.83
9	0.0074	98.51	0.02504	89.98	0.018520	92.59
10	0.0111	98.51	0.02901	88.39	0.014400	94.24
Average	0.00736	98.811	0.02675	89.29	0.016048	93.58

Table 7 illustrates the validation results of iris, cancer and heart disease datasets consisting of 150, 699 and 270 instances respectively. The generalized accuracy of the datasets are obtained through dividing the three datasets into 10 folds (10-k cross-validations) in which the first round ($k = 1$) of 15, 70 and 27 patterns respectively are used for validation and the remaining exemplars for training. In the next round ($k = 2$) the second patterns of the same size are hold back for validation and the rest for training. This will continue until last round ($k = 10$) of same size patterns are used for validation and the rest for training.

8 Statistical Test of HSA-RBF Classifier

In this section, summary of statistical tests are conducted on the training and testing process of HSA-RBF to increase the strength of the study. There are 500 separate runs of the network have been done on each of iris, breast cancer, and heart disease datasets. The mean square error and the correctly classified percentage of each run is recorded. A statistical hypothesis test is applied to the data obtained to make sure that the results are statistically significant.

Table 7: Statistical test report of Iris, Cancer and Heart

		Best Classification	Min. MSE	Mean	Std Dev	Std Err	Median	Mode	
Iris dataset	Training	98.89%	0.0063	0.0378	0.0234	0.0011	0.0271	0.0208	
	Testing	100.00%	0.0000	0.0150	0.0101	0.0005	0.0167	0.0167	
	T-Test of the MSEs in the training and testing processes of HSA-RBF							7.16E-66	
	T-Test of the classification % in training and testing of HSA-RBF							1.15E-107	

Cancer dataset	Training	92.15%	0.0196	0.0299	0.0029	0.0002	0.0296	0.0288	
	Testing	96.00%	0.0100	0.0309	0.0084	0.0004	0.0300	0.0275	
	T-Test of the MSEs in the training and testing processes of HSA-RBF							0.0192134	
	T-Test of the classification % in training and testing of HSA-RBF							0.0192129	
Heart dataset	Training	95.37%	0.0116	0.0284	0.0053	0.0002	0.0287	0.0283	
	Testing	100.00%	0.0000	0.0298	0.0114	0.0005	0.0303	0.0292	
	T-Test of the MSEs in the training and testing processes of HSA-RBF							0.0153567	
	T-Test of the classification % in training and testing of HSA-RBF							0.0153533	

The statistical student's t-test conducted separately on the MSEs of each of the testing and training results of those 500 runs produced results surprisingly similar to t-test result obtained from the relative outcome of the correct classification percentages of these training and testing processes in HSA-RBF network.

8.1 Sensitivity and specificity

Finally, in this section the specificity and the sensitivity concepts were employed on the breast cancer and heart disease medical dataset. Both sensitivity and specificity address how often is the medical test right? And they do their test on two different populations, those with the disease and those who are well. Sensitivity is about those people having the disease while specificity deals with the people who have no disease. Sensitivity should answer the question of, among the people with disease how often is the test accurate? On the other hand, specificity should answer the question of, among the people who are well, how often is the test accurate?

Therefore the sensitivity and specificity in the Table 9 illustrates the diagnostic test performance of the HSA-RBF classifier and the amount of true positive or negative hits compared with those of false misclassified positive or negative hits.

Table 8: Sensitivity and Specificity

Test Result	Disease Status			
	Breast Cancer		Heart Disease	
	Malign	Benign	Present	Absent
Positive Test	219	96	94	2

Negative Test	1	283	19	101
Sensitivity	99.55%		83.19%	
Specificity	74.67%		98.06%	

The experiments show that the classification accuracy of the HSA-RBF network is high and reliable. The sensitivity obtained from cancer and heart disease datasets is 99.55% and 83.19% respectively, while the specificity result of the two datasets are 74.67% and 98.06% respectively.

8.2 Receiver operating characteristics

A receiver operating characteristics (ROC) graph is a technique for visualizing, organizing and selecting classifiers based on their performance. ROC graphs have long been used in signal detection theory to depict the tradeoff between hit rates and false alarm rates of classifiers; it is a good way of visualizing a classifier's performance in order to select a suitable operating point, or decision threshold [11, 18]. A ROC curve is a graph where the x-axis represents the number of true negatives and the y-axis the number of true positives [10].

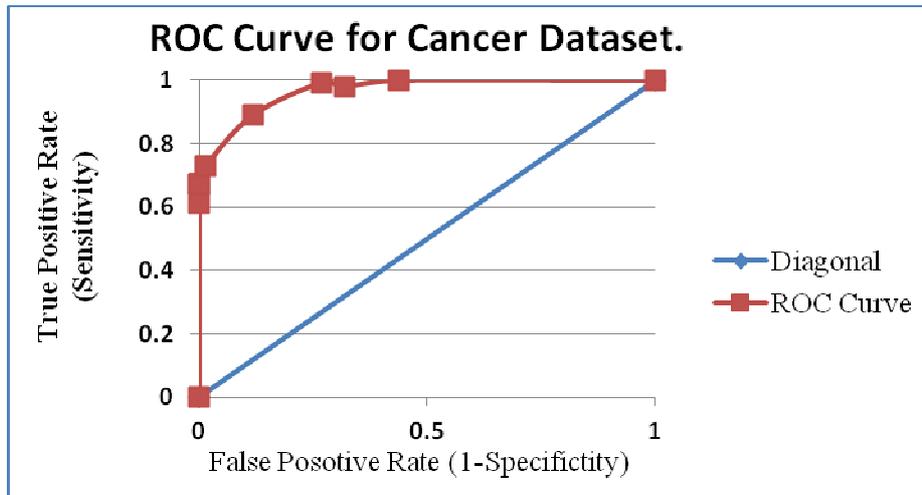


Fig. 8: ROC curve for HSA-RBF on the breast cancer dataset

As a case study, the performances of the classification schemes on two real world medical data sets are employed. These data sets are chosen to characterize those typically found in medical diagnostics; they have primarily continuous input attributes and have overlapping output classes. For each dataset, 10 sets of results (one for each of the 10-fold cross-validation partitions) were generated. The raw data were stored in the form of a confusion matrix. The experiment results represented in the form of ROC curves are illustrated in Fig. 8 and Fig. 9.

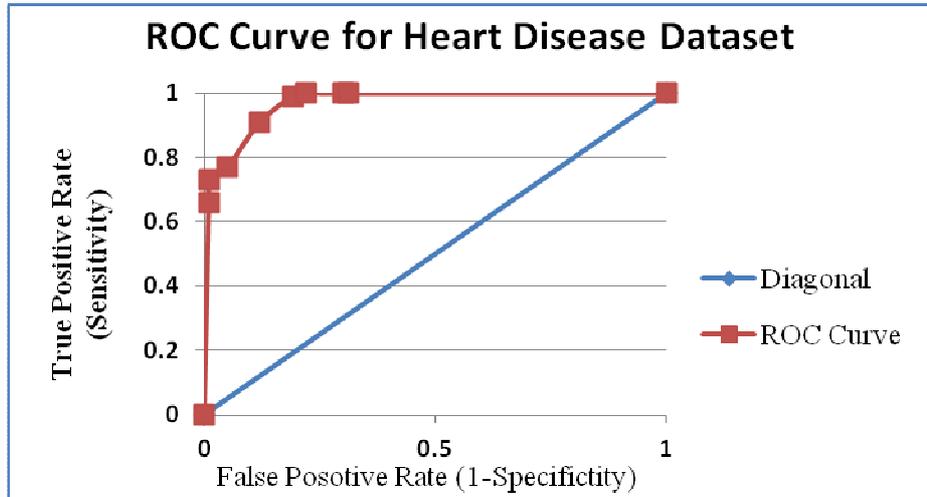


Fig. 9: ROC curve for HSA-RBF on the heart disease dataset

Bouckaert [10] determined that the area under the ROC Curve (AUC) is a summary statistic that indicates higher overall performance with higher AUC. AUC is typically calculated by summing over all trapezoids with base x_{i+1} to x_i , that is:

$$AUC = \sum_{i=0}^n \frac{(y_{i+1} + y_i)}{2} * (x_{i+1} - x_i) \quad (11)$$

The AUC calculations conducted on the two datasets (Breast cancer and Heart disease) produced two close results which are 0.9653 and 0.9684 respectively, that is slightly over 96% of the overall performance of the HSA-RBF classifier.

9 Conclusion

In this study, the hybrid model of RBF Neural Network and the Harmony Search Algorithm is proposed to enhance the learning process of the RBF network. Several experiments were carried out using some UCI machine learning datasets, such as XOR, Iris, Cancer and Heart disease datasets in order to evaluate the abilities of the new model. The proposed model proved a high performance in terms of accuracy and correct classification percentage, the convergence rate, the least time consumption and mean squared error. However, the model showed a slight decrease in performance when the size of the training datasets increases. The statistical tests conducted have confirmed the accuracy and reliability of the HSA-RBF network.

References

- [1] Abdull Hamed, H. N., Shamsuddin, S. M. and Salim, N. "Particle Swarm Optimization For Neural Network Learning Enhancement". *Jurnal Teknologi*, Vol 49, (2012), pp. 13–26.

- [2] Al-Betar, M. and Khader, A. "A Hybrid Harmony Search for University Course Timetabling". *Proceedings of the 4TH Multidisciplinary Conference on Scheduling: Theory and Applications (MISTA 2009)*, (2009), pp.10-12.
- [3] Al-Betar, M. A. and Khader, A. "A Harmony Search Algorithm for University Course Timetabling". *Springer Science + Business Media, LLC*, (2012), Vol. 194, pp.3-31.
- [4] Al-Betar, M. A., Khader, A. T. and Liao, I. Y. "A Harmony Search with Multi-pitch Adjusting Rate for the University Course Timetabling". *Recent Advances In Harmony Search Algorithm*. Springer. Vol. 270 (2010), pp. 147-161.
- [5] Alejo, R., Garcia, V., Sotoca, J., Mollineda, R. and Sanchez, J. "Improving the Performance of the RBF Neural Networks Trained with Imbalanced Samples". *Computational and Ambient Intelligence*, Vol. 4507, Springer, (2007), pp. 162-169.
- [6] Alia, O. M. D. and Mandava, R. "The variants of the harmony search algorithm: An Overview". *Artificial Intelligence Review*, Springer, Vol. 26, Issue 1, (2011), pp. 1-20.
- [7] Alia, O. M., Mandava, R., Ramachandram D. and Aziz, M. E. "A Novel Image Segmentation Algorithm Based on Harmony Fuzzy Search Algorithm". *Proceedings of the International Conference of Soft Computing and Pattern Recognition, SOCPAR'09*, IEEE, (2009), pp. 335-340.
- [8] Ayvaz, M. T. "Simultaneous Determination of Aquifer Parameters and Zone Structures with Fuzzy C-Means Clustering and Meta-Heuristic Harmony Search Algorithm". *Advances in Water Resources*, Vol. 30, (2007), pp.2326-2338.
- [9] Bors, A. G. "Introduction of the Radial Basis Function Networks". Online Symposium for Electronics Engineers, (2001), pp.1-7.
- [10] Bouckaert, R. R. "Efficient AUC Learning Curve Calculation". *AI 2006: Advances in Artificial Intelligence*, Springer, (2006).
- [11] Bradley, A. P. "The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms". *Pattern Recognition*, Vol. 30, (1997), pp. 1145-1159.
- [12] Ceylan, H., Haldenbilen, S. and Baskan, O. "Transport energy modeling with meta-heuristic harmony search algorithm, an application to Turkey". *Energy Policy*, Vol 36, (2008), pp. 2527-2535.
- [13] Chakraborty, P., Roy, G. G., Das, S., Jain, D. and Abraham, A. "An Improved Harmony Search Algorithm with Differential Mutation Operator". *Fundamenta Informaticae* , Vol 95, (2009), pp1-26.
- [14] Chan, W., Chow, Y. and Liu, L. 1995. "Neural Network: An Alternative to Pile Driving Formulas". *Computers and Geotechnics*, Vol. 17, (1995), pp. 135-156.
- [15] Cheng, Y., Li, L., Lansivaara, T., Chi, S. and Sun, Y. "An Improved Harmony Search Minimization Algorithm Using Different Slip Surface Generation

- Methods for Slope Stability Analysis”. *Engineering Optimization*, Vol. 40, (2008), pp. 95-115.
- [16] Das, S. R., Panigrahi, P. K., Das, K. and Mishra, D. 2012. “Improving RBF Kernel Function of Support Vector Machine using Particle Swarm Optimization”. *International Journal of Advanced Computer Research*, Vol. 2, (2012), pp.2277-7970.
- [17] Dhubakarya, D.C., Nagariya, D. and Kapoor, R. “Implementation of a Radial Basis Function Using VHDL”. *Global Journal of Computer Science and Technology*, Vol. 10, No.10, (2010), pp.16-19.
- [18] Fawcett, T. “An Introduction to ROC Analysis”, *Pattern Recognition Letters*, Vol. 27, (2006), pp. 861-874.
- [19] Fesanghary, M., Mahdavi, M., Minary-Jolandan, M. and Alizadeh, Y. “Hybridizing Harmony Search Algorithm with Sequential Quadratic Programming for Engineering optimization problems”. *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, Issues 32-40, (2008), pp.3080-3091.
- [20] Forsati, R., Mahdavi, M., Kangavari, M. and Safarkhani, B. “Web Page Clustering Using Harmony Search Optimization”. *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, CCECE 2008, (2008). IEEE, 001601-001604.
- [21] Gallegos-Funes, F. J., Gomez-Mayorga, M. E., Lopez-Bonilla, J. L. and Cruz-Santiago, R. “Rank M-type Radial Basis Function (RMRBF) Neural Network for Pap Smear Microscopic Image Classification”. *Apeiron*, Vol.16, (2009), pp. 542.
- [22] Gan, M., Peng, H. and Chen, L. “Global–local Optimization Approach to Parameter Estimation of RBF-type Models”. *Information Sciences*. Vol.197, (2012), pp.144-160.
- [23] Gan, M., Peng, H. and Dong, X. “A Hybrid Algorithm to Optimize RBF Network Architecture and Parameters for Nonlinear Time Series Modeling”. *Applied Mathematical Modeling*. Vol. 36, No. 7, (2011), pp.2911-2919.
- [24] Geem, Z. and Choi, J. Y. “Music Composition Using Harmony Search Algorithm”. *Applications of Evolutionary Computing, Lecture Notes in Computer Sciences*, Vol. 4448, (2007), pp.593-600.
- [25] Geem, Z. W. and Hwangbo, H. “Application of Harmony Search to Multi-Objective Optimization for Satellite Heat Pipe Design”. Online at: <http://citeseerx.ist.psu.edu> . (2006), pp. 1-3.
- [26] Geem, Z. W. and Williams, J. C. “Ecological Optimization Using Harmony Search”. *Proceedings of the American Conference on Applied Mathematics, World Scientific and Engineering Academy and Society (WSEAS)*, (2008), pp.148-152.
- [27] Geem, Z. W. “Particle-swarm Harmony Search for Water Network Design”. *Engineering Optimization*, Vol. 41, (2009), pp. 297–311.
- [28] Geem, Z. W. “State-of-the-Art in the Structure of Harmony Search Algorithm”. *Recent Advances in Harmony Search Algorithms. Studies in*

- Computational Intelligence*, Springer-Verlag Berlin Heidelberg, Vol. 270, (2010), pp. 1-10.
- [29] Geem, Z. W., Kim, J. H. and Loganathan, G. V. "A New Heuristic Optimization Algorithm: Harmony Search". *SIMULATION*, Vol. 76, No.2, (2001), pp60-68.
- [30] Geem, Z. W., Tseng, C.-L. and Park, Y. "Harmony Search for Generalized Orienteering Problem: Best Touring in China". *Advances in Natural Computation. Lecture Notes in Computer Sciences*, Springer. Vol. 3612, (2005), pp. 741-750.
- [31] Geem, Z. "Harmony Search Algorithm for Solving Sudoku". *Knowledge-Based Intelligent Information and Engineering Systems*, Springer, (2007). pp. 371-378.
- [32] Graupe, D. "*Principles of Artificial Neural Networks. Advanced Series in Circuits and Systems*". Vol. 6, World Scientific Publishing Co. Pte. Ltd., (2007)
- [33] Grosan, C. and Abraham, A. "*Intelligent Systems: A Modern Approach*", Springer. (2011)
- [34] Hamadneh, N., Sathasivam, S., Tilahun, S. L. and Choon, O. H., "Learning Logic Programming in Radial Basis Function Network via Genetic Algorithm". *Journal of Applied Sciences*, Vol. 12, (2012), pp. 840-847.
- [35] Idri, A., Zakrani, A. and Zahi, A., "Design of Radial Basis Function Neural Networks for Software Effort Estimation". *International Journal of Computer Science Issues*, Vol. 7, Issue 4, No. 3.(2010), pp. 11-17.
- [36] Kattan, A. and Abdullah, R. "An Enhanced Parallel and Distributed Implementation of the Harmony Search Based Supervised Training of Artificial Neural Networks". *Proceedings of the Third International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*, (2011), pp. 275-280.
- [37] Kattan, A., Abdullah, R. and Salam, R. A., "Harmony Search Based Supervised Training of Artificial Neural Networks". *Proceedings of the 2010 International Conference on Intelligent Systems, Modeling and Simulation (ISMS '10)*, (2010), IEEE Computer Society, pp.105-110.
- [38] Khadwilard, A., Luangpaiboon, P. and Pongcharoen, P. "Full Factorial Experimental Design for Parameters Selection of Harmony Search Algorithm". *The Journal of Industrial Technology*, Vol. 8, (2012)
- [39] Kulluk, S., Ozbakir, L. and Baykasoglu, A. "Training Neural Networks With Harmony Search Algorithms for Classification Problems". *Engineering Applications of Artificial Intelligence*, Vol. 25, (2012), pp. 11-19.
- [40] Kurban, T. and Besdok, E. 2009. "A Comparison of RBF Neural Network Training Algorithms for Inertial Sensor Based Terrain Classification". *Sensors*, Vol. 9, No. 8, (2009), pp. 6312-6329.
- [41] Mohsen, A., Khader, A. and Ramachandram, D. "An Optimization Algorithm Based on Harmony Search for RNA Secondary Structure Prediction". *Recent*

- Advances in Harmony Search Algorithm. Studies in Computational Intelligences*, Springer, Vol. 270, (2010), pp. 163-174.
- [42] Noman, S., Shamsuddin, S. and Hassanien, A. "Hybrid Learning Enhancement of RBF Network with Particle Swarm Optimization". *Foundations of Computational, Intelligence*, Vol. 1, (2009), pp. 381-397.
- [43] Pan, Q. K., Suganthan, P., Tasgetiren, M. F. and Liang, J. "A Self-Adaptive Global Best Harmony Search Algorithm for Continuous Optimization Problems". *Applied Mathematics and Computation*, Vol. 216, (2010), pp. 830-848.
- [44] Panchal, A. "Harmony Search in Therapeutic Medical Physics". *Music-inspired harmony search algorithm. Studies in Computational Intelligences*, Springer, Vol. 191, (2009), pp. 189-203.
- [45] Qasem, S. N. and Shamsuddin, S. M. "Radial Basis Function Network Based on Time Variant Multi-Objective Particle Swarm Optimization for Medical Diseases Diagnosis". *Applied Soft Computing*, Elsevier, Vol. 11, Issue 1, (2011), pp. 1427-1438.
- [46] Qasem, S. N., "Learning Enhancement of Radial Basis Function Network with Particle Swarm Optimization". *Masters Thesis*, Universiti Teknologi Malaysia, Faculty of Computer Science and Information System. (2008)
- [47] Rini, D. P., Shamsuddin, S. M. and Yuhaniz, S. S. "Particle Swarm Optimization: Technique, System and Challenges". *International Journal of Computer Applications*, Foundation of Computer Science, Vol 14, No. 1, (2011), pp.19-27.
- [48] Santos Coelho, L. and Andrade Bernert, D. L. "An Improved Harmony Search Algorithm for Synchronization of Discrete-Time Chaotic Systems". *Chaos, Solitons & Fractals*, Elsevier, Vol. 41, Issue 5, (2009), pp. 2526-2532.
- [49] Schwenker, F., Kestler, H. A. and Palm, G. "Three Learning Phases for Radial-Basis-Function Networks". *Neural Networks*, Elsevier, Vol. 14, (2001), pp. 439-458.
- [50] Soltani, M. A. Z., Haghghat, A. T., Rashidi, H. and Chegini, A. T. G. "A Couple of Algorithms for K-Coverage Problem in Visual Sensor Networks". *Proceedings of the International Conference on Communication Engineering and Networks, IPCSIT*. Vol. 19, (2011), pp. 96-101.
- [51] Tsimboukakis, N. and Tambouratzis, G., "Neural Networks for Author Attribution". *Proceedings of the Fuzzy Systems Conference, FUZZ-IEEE 2007*.(2007). IEEE, pp. 1-6.
- [52] Vasebi, A., Fesanghary, M. and Bathaee, S. "Combined Heat and Power Economic Dispatch by Harmony Search Algorithm". *International Journal of Electrical Power & Energy Systems*, Vol. 29, Issue 10, Elsevier, (2007), pp. 713-719.
- [53] Wang, Weihua. "Face Recognition Based On Radial Basis Function Neural Networks". *International Seminar on Future Information Technology and Management Engineering*,(2008).

- [54] Xie, T., Yu, H. and Wilamowski, B., “Comparison Between Traditional Neural Networks and Radial Basis Function Networks”. *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE) 2011*, IEEE, (2011), pp. 1194-1199.
- [55] Yu, H., Xie, T., Hamilton, M. and Wilamowski, B. “Comparison of Different Neural Network Architectures for Digit Image Recognition”. *Proceedings of the 4th International Conference on Human System Interactions (HSI) 2011*, IEEE, (2011), pp. 98-103.
- [56] Diao, R. and Shen, Q. “Two New Approaches to Feature Selection with Harmony Search”. *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ)*, IEEE, (2010), pp. 1-7.