

# Optimizing the Topology and Learning Parameters of Hierarchical RBF Neural Networks Using Genetic Algorithms

Mohammed Awad

*Department of Computer Systems Engineering, Faculty of Engineering and Information Technology,  
Arab American University, Jenin, West-Bank Palestine*

## Abstract

This paper proposed a hierarchical topology of Radial Basis Function Neural Networks (HTRBFNs) combined with the using of evolutionary process by applying Genetic Algorithms (GAs). The GAs process used to optimize the topology of the HTRBFNs and its learning parameters. The number of sub-RBFNs in the hierarchical topology will be predetermined, and the GAs in its whole process will determine which of the input data variables group that directed to each sub-RBFN and it will optimize the sub-RBFN learning parameters (centers  $c$ , radii  $r$  and weights  $w$ ). The proposed model implies that we can train exactly the system to optimize the topology and the learning parameters of each sub-RBFN depending on the entire model approximation error. GA-HTRBFNs produce automatically the most suitable topology which can apply to complex problems of function approximation. Therefore, the model goal is to find the most suitable topology of the proposed GA-HTRBFNs, the best optimization parameters and the number of RBF in each sub-RBFN in order to approximate a problem of input/output (I/O). The results show that the proposed GA-HTRBFNs model has the ability to produce the best topology of parallel RBFN structure using GAs, with the better value of approximation means square error.

**Keywords:** Hierarchical Topology, RBF Neural Networks, Genetic Algorithms, Parallel RBFN.

## INTRODUCTION

For the application of supervised learning techniques, the process starts with a set of initial data (inputs) for which a series of results (outputs) is observed in a series of determining events, which will allow for mapping the results that will be obtained for a new event [1,2]. In the process of achieving the smallest error between real output and target output, the selection of the learning algorithm and the topology of the Neural Networks (NNs) is important [24]. Because of any conventional or standard function approximation systems, the number of parameters increases exponentially with the increase in the number of input data variables; suppose there are  $n$  input variables and  $m$  parameters for each variable, then we will need  $m*n$  computational parameters [3,25]. In general, the complexity of a problem increases exponentially due to the number of variables involved, this phenomenon called the curse of dimensionality [4, 25]. Evolutionary algorithms (EAs)

in their different variants have been very successful in the solution of a wide range of real-world problems [26]. EAs can be used to optimize the parameters and the topology of the NNs [5, 26]. Genetic Algorithms (GAs) can be used to optimize the parameters of NNs, it can obtain the best combination of the learning parameters and the NNs topology too [6, 7, 8]. The combination of NNs and GAs are soft-computing technologies that can be very effective when it used together [7].

In this work, hierarchical RBFN model [9, 10] will be applied to approximate created complex functions [3]. The problem of estimating an unknown function  $F$  from samples of input/output data type  $\{X, F(X)\}$ ; has been used in a great variety of scientific disciplines [3]. Excessively a high computational complexity can occur when developing multivariate real-world models for industrial or medical applications, where the best set of input data variables is unknown [3]. The most real-world applications nowadays depend on a vast number of input data variables; this obviously will produce computational complexity problems. In general, training NNs is more difficult with a big number of input variables.

In general, EAs learning is formulated as an optimization problem. Given a fixed topology, training of NNs can be seen as an optimization process whose objective is to find a set of parameters that minimizes the error that the NNs produce on the training dataset [5]. This makes EAs an alternative to traditional training methods, since they are not able to be trapped in local minimums and, as they move towards the solution through evolutionary operators, they can increase the speed of convergence to the solution. EAs as a training method can replace the traditional training algorithms [5].

The proposed hierarchical topology of parallel processing Radial Basis Function Neural Networks using Genetic Algorithms (GA-HTRBFNs) is a model which used to train and create the hierarchical RBFN depends on; distribution the input data variables forwards to a determined number of sub-RBFNs, the GAs process select the best hierarchical topology and the best optimization values of the learning parameters. The proposed model consists of a number of sub-RBFNs distributed hierarchically, which have the great advantage that the total number of executed parameters decreases depending on the proposed methodology.

The RBFNs as single hidden layer neural networks will generally be unsuitable for dealing with complex problems [25]. To obtain maximum flexibility to deal with complex problems, a hierarchical structure of RBFN is proposed. The HRBFNs is characterized by a linear combination of the output, which means all sub-RBFN produce one output vector and so one evaluation error. In order to give strength to the hierarchical structure model, we use a predetermined number of sub-RBFN in the proposed model. In a more detailed way, each sub-RBFN receives a number of input data variable/s which determine by the GAs process. The parameters of each sub-RBFN determine by GAs too, which used to optimize the centers  $c$ , radii  $r$  and weight  $w$ . The output of all determined sub-RBFN will be calculated depends on a linear equation, where the entire HT-RBFN grouped the sub-RBFN as shown in Figure 1, and Figure 2. For example, a two-input data variables will produce one or two sub-RBFN.

During the process of optimization of the input data variables distribution and the sub-RBFN learning, parameters will be determined automatically using GAs. This Process will produce many topologies; the best topology which produced minimum approximation error will be the winner. The proposed topology combines the advantages of the global search algorithm performed by GAs. The research paper is organized as follows. Section 2 describes the related works. Section 3; presents the basic model and the hierarchical parallel processing structures of RBFN. Section 4; presents a created simulation example. The conclusion will present finally.

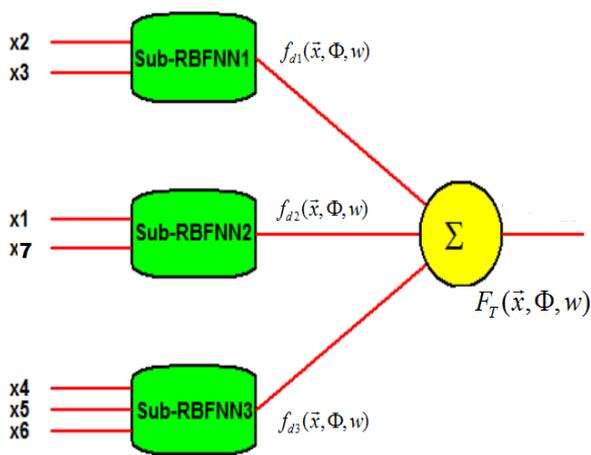


Figure. 1: The topology of HT-RBFNs produced by GAs

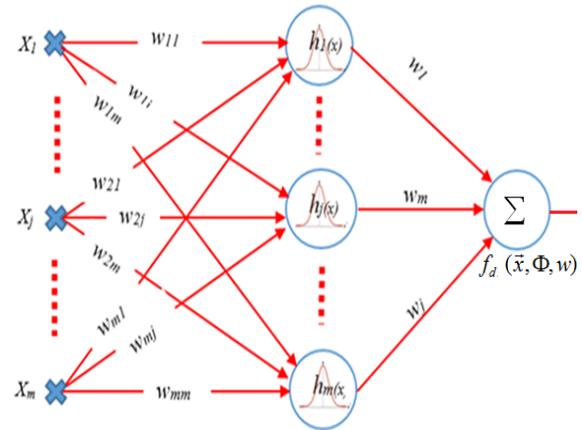


Figure. 2: The sub-RBFN which present traditional RBFN.

### RELATED WORKS

In general, the neural network structure consists of input layer; hidden layer/s and output layer, the mapping of complex problems produces often huge neural networks, with an excessively large number of hidden neurons. One effective solution is to incorporate proper hierarchical processing structure into the network. Hierarchical processing structures have a very rich variety of applications in computing since they provide representations that can be composed, modified, and manipulated in a very flexible way [3]. Some researchers introduce hierarchical NNs in general. In [11] the authors used a hierarchical neural network is proposed to build distributed representation of news articles. This proposed model can capture specific n-grams and model the sequenced relations among sentences. The experimental result shows that the model outperforms traditional methods and is especially good at identifying underlying popular news. The authors in [12] proposed deep hierarchical neural network applied for traffic flow prediction which considers and extracts the time-variant trend. It's used two stacked layers: extraction layer and prediction layer. This model needs pre-trained layer-by-layer to make it more effective. The result of the model can noticeably boost the prediction performance compared with some traditional prediction models and LSTM with other based methods. This paper [13] present a novel hierarchical deep neural network (HDNN) used for general multivariate regression problem. They used multiple subnets to build the HDNN using a divide and conquer technique, the proposed HDNN applied on Chinese handwriting recognition and speech enhancement task. The result of the experiments shows that HDNN outperforms DNN in the two applied fields. In [14] the authors describe a hierarchical evolutionary technique developed to design and train feedforward NN with different activation functions on their hidden layer neurons. A genetic algorithm is used to determine the number of neurons in the hidden layer and the type of the activation function of those neurons. The result presents as a created NN by adjusting both the weights of the hidden layer neurons and the parameters for their activation functions.

Other research's used hierarchical RBFNs as in [15]. The researchers present a Hierarchical Markovian RBF Neural Network model with the opportunity to enable recursive operations. They used nested RBFNs with random levels of hierarchy. Truly RBFNs with 2 weight matrices composed of the model, centers, and output weights are summed linearly using one neuron. The hidden layer of RBFN is recursively response. They used k-means clustering, the classical tree-based recursion function, and the standard regularized least squares solver to optimize RBFN learning parameters. The simulation result compared with two standard model meta-learning architectures shows that the model produces similar results. Hierarchical RBFNs which consist of multiple RBFNs assembled in different architecture level is presented on [16]. This model is employed to detect the breast cancer. They use Genetic Programming, a tree-structure based evolutionary algorithm, and the Differential Evolution to find an optimal detection model. The comparison of the result with other models like Flexible Neural Tree, Neural Network, and RBFN by using the same breast cancer data appeared that the model outperforms these models in having a fewer number of variables with the reduced number of input features and with the high detection accuracy. In [17] the authors present a two-level learning method for the design of the Beta Basis Function Neural Network BBFNN. They used a genetic algorithm to construct BBFNN, the learning parameters: width, centers, and the Beta are optimized using the gradient algorithm. They applied the model to non-linear approximation function in order to demonstrate that the model is effectiveness. Each sub-RBFN. The purpose model in [18] depends on hierarchical RBFNs and select important input features for each sub-RBF neural network automatically. They used a tree-structure based evolutionary algorithm to improve the performance of a hierarchical RBFN. Particle swarm optimization algorithm is used to optimize RBFN parameters. The result appeared that the model produces an efficient result. This paper [19] proposes a hierarchical multi-dimensional differential evolution (HMDDE) approach, which is a computational framework for the optimization of beta basis function neural network (BBFNN) in automatic form. The weights and learning method parameters are adapted according to the problem. The population of the model forms multiple beta networks with different structures at the higher level and each individual of the previous population is optimized at a lower hierarchical level to improve the performance of each individual. Empirical results illustrate that the HMDDE produces a better generalization performance.

In this paper, a hybrid model that combines the RBFNs and GAs is used to approximate complex functions. In order to give strength to the hierarchical structure model, we use a predetermined number of sub-RBFN in the proposed model. These sub-RBFNs receive a subset of the input data variables  $\{x_1, \dots, x_n\}$  depends on the GAs process, which means the assigned numbers of input variables to each sub-RBFN determined using GAs. The advantage of the GA-HTRBFNs structure consists of the fact that the problem will be divided into many problems that are connected in parallel. In general, 3 basic steps used to solve complex problems of function

approximation; determine the number of sub-RBFN in the proposed system GA-HTRBFNs, then we determine using GAs the suitable group of input data variables as input of each sub-RBFN, the next step is the optimization of the parameters of each sub-RBFN (centres, radii, and weights) and the number of RBF (Neurons) in each sub-RBFN, which depends on the calculation of the total output  $F(x)$  of the GA-HTRBFNs using GAs.

### THE PROPOSED MODEL

The RBFNs considered as universal function approximation, the general design process of RBFN start by determining an appropriate architecture, a subset of input data is used to train the RBFN by means of a suitable learning algorithm, and finally, the performance of the trained RBFN is tested with data that has not been used initially generalization. The proposed approach basically follows the same step for each sub-RBFN in the system where the process of determining the suitable inputs and directed it to the each sub-RBFN, the process of determining suitable number of RBF in each sub-RBFN, and the determining the RBFN learning parameter (centers, radii, and weights) are determined using GAs as a powerful optimization tool in the optimization phase. The advantage of the proposed model structure consists of that the main problem will be divided into subproblems that are connected in parallel; with the advantage that each sub-RBFN receives input or inputs of the data, variables depends to the optimization process performed by GAs. The entire proposed model has a total output that is the sum of all output of each sub-RBFN.

In general, the construction of the proposed model depends on determining the number of sub-RBFNs manually. Use GAs to select the suitable single or group of input variables directed to each sub-RBFN, optimization of the sub-RBFNs learning parameters (centers, radii, and weight), and optimization of the number of RBF in each sub-RBFN. With produced structure and optimized parameter the model calculates the output  $F(x)$  of GA-HTRBFN using in the following expression:

$$F(\vec{x}, \Phi, w) = \sum_{s=1}^S \sum_{i=1}^{m_s} \phi_i^s(\vec{x}) \cdot w_i^s \quad (1)$$

where  $\phi_i^s$  are the  $i$ -th basis functions of the  $s$ -th RBFN, and  $w_i^s$  is its weight. The basis function  $\phi$  can be calculated as a Gaussian function using the following expression:

$$\phi(\vec{x}, \vec{c}, r) = \exp\left(-\frac{\|\vec{x} - \vec{c}\|^2}{r}\right) \quad (2)$$

where  $\vec{c}$  is the central point of the function  $\phi$  and  $r$  is its radii.

The real output of each of sub-RBFN is calculated using the following expression:

$$f(\vec{x}, \Phi, w) = \sum_{i=1}^m \phi_i(\vec{x}) \cdot w_i \quad (3)$$

The GAs is selected to carry out the optimization process of the topology and learning parameters because it does not depend on the gradient, which helps to avoid stuck in a local minimum, and GAs are known as a powerful heuristic search method for the optimization of non-linear and complex function approximation problems [20]. A GA has several parameters whose values must be determined before starting the optimization process.

The main objective of using GAs is to determine the best topology of the proposed model HTRBFN, it will find the dynamic topology of HTRBFN and optimize the learning parameters that minimize the approximation error. GAs start with a population of chromosomes, which are the elements of a search space of higher dimensionality, the chromosomes present a string of float numbers that represent the structure of the proposed HTRBFN. Each chromosome in the population we are divided into sub-chromosomes, where the all sub-chromosomes will represent one entire chromosome in the GAs population. This main chromosome will have 3 sub chromosomes; input variables sub-chromosome which have the possibility of creating subsets of input variables that must be more than one variable that can be assigned to sub-RBFN and less than the total number of input variables with the condition that no sub-RBFN have empty input variable. The number of sub-RBFN that determined manually will be the final number of this sub-chromosome. The number of RBF in each sub-RBFN will be the second sub-chromosome, and the learning parameters for each RBF (centers, radii, and weights) will present the thread sub-chromosome. The manipulation of these sub-chromosomes will allow the model to find the best combinations to find new HRBFN topology. The fittest function which presents the error between real output and the target of the GA-HTRBFNs, it is calculated for each chromosome in the population. The selection of the best chromosomes depends on the normalized geometric ranking method. To avoid the stocking in local minima we will use crossover and mutation steps. This process is repeated until it is accomplished with a stop condition, so at the end, you have the best individual found by the GA, which represents the best GA-HTRBFNs topology and best leaning parameters with use on the testing phase. The general algorithm of the proposed model is illustrated as a following:

---

**Algorithm:** The general procedure that was used in optimizing the GA-HTRBFN model

---

**Input:** Input data variables

**Output:** Approximated output

**Data Preprocessing:** Normalization between [0 1] the input data variables and target data.

**Determine** the number of sub-RBFNs in the GA-HTRBFN model.

**For** each iteration in the GAs process

**Generate** the divided **Initial Population**  $P$  with each individual  $S$  which represents:

**Part1:** # of input data variables [its final element is # of *sub-RBFN*]

**Part2:** # of RBF (Neurons) in each sub-RBFN.

**Part3:** Learning parameters of each sub-RBFN (centers  $c$ , radii  $r$ , and weights  $w$ )

**Evaluate** the fitness function for each individual  $S$  using MSE of the Total output  $F(\vec{x}, \Phi, w)$

**If termination** criteria (MSE)  $\leq$  threshold value ( $\alpha$ )  
 or # of Generation  $\leq$  threshold value ( $\beta$ )

**Select** using the normalized geometric ranking method the best two individuals

**Apply Arithmetical Crossover** on the two selected individuals

**Apply Uniform Mutation** to create two offspring

**Insert the best Individual** in the newly generated population

**Else**

**Stop** the optimization process.

**Return** the training approximation Error.

**Return** the GA-HTRBFN structure with the best-optimized parameters (centers  $c$ , radii  $r$ , and weights  $w$ )

**Test**  $\leftarrow$  dataset

**Return** the testing approximation error.

---

### Initializing HTRBFN Population

Usually, the initial population is chosen by generating random strings, each chromosome containing one of the possible values that used to evaluate the model. The non-random initialization of the initial population can accelerate the convergence of the GAs. The operation of determining the initial element of the chromosome and sub-chromosomes depends on each chromosome in the population present; input data variables, number of sub-RBFN, number of neurons (RBF) in each sub-RBFN, and the learning parameters of all sub-RBFNs. The initial population will be generated randomly using the sub-

chromosome restrictions. The learning parameters of the chromosome initialize using; k-Means Clustering [21] to initialize centers, k-nearest neighbor k-nn [22] to initialize radii, and singular value decomposition SVD [23] to initialize weights. The main chromosome is represented as follows:

$$\left\{ \{\vec{x}_1, \dots, \vec{x}_n\}, \{\#sub_d\}, \{\#rbf_1, \dots, \#rbf_d\}, \{c_1 r_1 w_1, \dots, c_d r_d w_d\} \right\}$$

Where  $\vec{x}$  are the input data vectors, where the GAs processes select signal or group of these variables to direct it to the determined sub-RBFN, with condition that each sub-RBFN at least receive on the input vector. #sub is the number of determined sub-RBFN, this element inserted to the chromosome to connect the process of assign input variables  $\vec{x}$  and #rbf which present the number of RBF (neurons) in each sub-RBFN. In #rbf depends on the length of the final part of this chromosome that represents the learning parameters of the model which are centers  $c$ , radii  $r$ , and weights  $w$ .

### Evaluating the Chromosomes

The evaluation function is a crucial issue in GAs. Building a good evaluation function means it should reflect the value of the individual in a real way, but in this model we have a combinatorial optimization, where there is a restriction, as in selecting the input variables to sub-RBFNs. in this work the process depends on the ability of GAs in selecting the best structures and the best learning parameters. GAs uses the objective functions of GA-HTRBFNs model as information to guide the search, the evaluation mechanism is required to calculate the value of the fitness in each chromosome. The proposed method uses Fitness = Mean Square Error (MSE) which can be presented by the following expression:

$$MSE = \frac{1}{n} \sum_{i=1}^n (T_i - F(\vec{x}, \phi, w))^2 \leq \theta \quad (4)$$

Where  $n$  is the number of the input data variables, and  $\Theta$  is the threshold value of the approximation process.  $T_i$  is the target output of the GA-HTRBFNs model,  $F(\vec{x}, \phi, w)$  is the real output after the optimization process of the topology and the learning parameters of the proposed model.

### Select the Best Chromosome

The process of selecting the most common parents in which each chromosome has a probability of being selected as a parent that is proportional to the value of its target function. One of the ways to overcome the problem related to the rapid convergence coming from the super individuals, which arises when applying the selection function, is to make the selection proportional to the individual's rank, which results in a more even distribution of the probability of selection.

This method assigns a value of probability of selection  $p_j$  to each individual  $j$  depends on its value of fitness (MSE). A series of numbers  $N$  are generated and compared against the

accumulated probability  $C_i$  which equal the  $\sum P_i$  start  $j = 1$  to a population  $i$ . Then copy the selected individual  $j$  in the new population if  $C_{i-1} < U(0,1) \leq C_i$ . The normalized geometric ranking method [7] uses a partially ordered set; it assigns  $P_i$  based on the line of solution  $I$  when all the solutions are classified. The normalized geometric ranking method of definite classification the probability  $P_i$  for every individual is defined as:

$$P_i = q^+ (1 - q)^{r-1} \quad (5)$$

Where  $q$  is the probability of selecting the best individual,  $r$  is the line of the individual, where 1 is the best.

$$q = \frac{q}{1 - (1 - q)^P} \quad (6)$$

Where  $P$  is the population size. The normalized geometric ranking method selects only the best HTRBFNs for the next generation, which increases the speed of convergence.

### GAs Operators on Selected Chromosomes

The idea of operators depends on creating new population by crossover and mutation with the opportunity to release the best chromosome. This step in the process of GAs depends on crossover and mutation. This step causes an increase in the efficiency of the GAs. Arithmetical crossover is used to produce two linear combinations of parents. The crossover step applied on the best two chromosomes, in this case, this operation applied to each sub-chromosome taking into account each sub-chromosome the cross operate only in change the variables which have the same characteristics. (Input variables with input variables, center with the center, and ..., etc). If parent1 and parent2 are the parents with best fitness value in the population, the function returns the child's for each sub-chromosome as in the following equations:

$$cross(X, Y) = \alpha \tilde{X} + (1 - \alpha) \tilde{Y} \quad (7)$$

$$cross(Y, X) = (1 - \alpha) \tilde{X} + \alpha \tilde{Y} \quad (8)$$

Where  $\tilde{X}$  and  $\tilde{Y}$  are the random value dimensional of each sub-chromosome in each HTRBFN chromosome parts,  $\alpha$  is probability random value between [0, 1]. Applying these equations on the selected chromosomes of HTRBFN parameters will produce a new HTRBFN topology and learning parameters presented on the best chromosome produced from the parents. The best one of these chromosomes will pass to mutation step. A mutation is an unusual event; it contributes to the genetic process diversity of the chromosomes. The mutation process contributes in that the GAs cannot is stack in a local solution but will allow jumping to other points of the function, thus allowing a global search. Uniform mutation is used to in this model to mutate the sub-chromosome parameters; this step performed using the following expression:

$$x_i = \begin{cases} \lfloor U(a_i, b_i) \rfloor & \text{if } i = j \\ x_i & \text{otherwise} \end{cases} \quad (9)$$

Where  $a_i$  and  $b_i$  are down and top level, for every variable  $i$ .

After applying GAs operators (crossover and mutation) a new HTRBFN chromosome that has the topology and learning parameters will appear. This best HTRBFN will be chosen to be part of the new population.

### Termination Criteria

The termination criteria in the process of GAs are very important to stop the evolutionary process of GAs. In the proposed model, two types of criteria are used as stop conditions; the maximum number of generation's  $\varepsilon$ , and the specified threshold value of the approximation error  $\theta$ . The termination criteria are expressed as in the flowing equation:

$$Stop_c = |B_{Fc} - B_{Fp}| \leq \varepsilon \text{ or } Stop_c \leq \theta \quad (10)$$

where  $B_{Fc}$  is the best fitness value in the current population,  $B_{Fp}$  is the best fitness value in the previous population,  $\varepsilon$  is the maximum number of generations, and  $\theta$  is the minimum fitness threshold.

### SIMULATION RESULTS

This section presents the experimental results obtained by training and testing the GA-HTRBFN model. The designed experiments carried out were with the objective of establishing. Experiments have been performed to test the performance of the proposed model. The system is simulated in MATLAB 2017. Under Windows 10 with processor i7 running with 8 GRAM. In this section different examples are given to verify the procedure in the proposed model. Two 3 types of results are presented:

- Determine the optimal topology of the proposed model.
- Determine the optimal learning parameters of the training algorithm.

Parameters and initial value have been used as initial population generated using 50 chromosomes, each chromosome divide into 3 sub-chromosomes. The length of the chromosome depends on the number of input data variables related to the number of sub-RBFNs and the number of learning parameters which depends on the number RBF in each sub-RBFN. For each particular configuration of a training model the results obtained in 5 executions; these result will appear;  $num\_SubRBF$ ; the number of sub-RBFN used, input data variable or variables presented as  $numInput\_RBFN$ ,  $numRBF\_networks$ ; present the number of neurones in each

sub-RBFN;  $C$ ,  $W$ , and  $R$  present the learning parameters centers, weights and radii, and  $MSE$  is the mean squared error.

### First Simulation Experiment

Suppose we take an example with 4 possible input variables to choose from. Let us consider a set of 5000 I/O data pairs randomly taken from the proposed function.

$$f(x) = 3\sin(\pi x_1 \cdot x_2) + 9(x_3 - 0.25)^2 + 5x_4 \quad x_1, \dots, x_4 \in [0, 1]$$

where each input variable is defined in the interval [0,1]. The proposed algorithm selects the best structure and learning parameters compatible with the model parameters and threshold as shown in Figure.3 and 4.

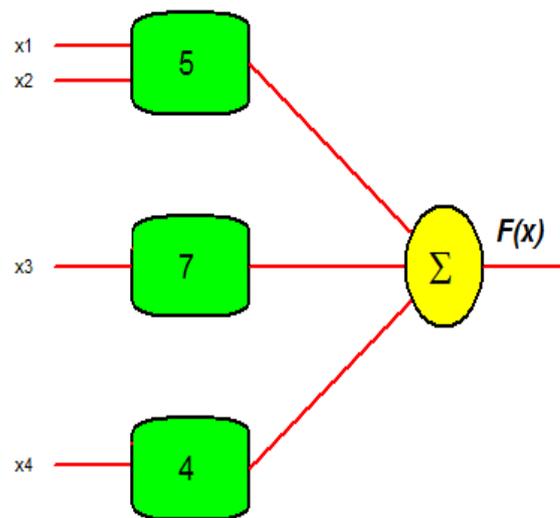


Figure 3: Produced structure GA-HTRBFN model for  $f_i(x)$

```

Editor - C:\Program Files\MATLAB71\work\work3\anetGA\Untitled.m*
File Edit Text Cell Tools Debug Desktop Window Help
1 - 'GA-HTRBFN'
2
3 - num_SubRBFN: 3
4 - maxnumInputs: 4
5 - numInput_RBFN: {[2] [1] [1]}
6 - numRBF_network: [5 7 4]
7 - data: [5000x4 double]
8 - C: {[5x1 double] [7x1 double] [4x1 double]}
9 - W: {[5x1 double] [7x1 double] [4x1 double]}
10 - R: {[5x1 double] [7x1 double] [4x1 double]}
11 - Target: [5000x1 double]
12 - ActivFun: 'gaussian : the Activation Function'
13 - OutPFun: 'linear : The Output Calculation Function'
14 - MSE: 0.00328
15
script Ln 13 Col 45 OVR
    
```

Figure 4: Produced topology parameters, learning parameters and MSE using GA-HTRBFN model for  $f_i(x)$

From the result, it is observable that the proposed model GA-HTRBFN can select the topology and optimize the learning parameters, which is clear in the small value of the approximation error MSE.

From the result, it is observable that the proposed model GA-HTRBFN can select the topology and optimize the learning parameters, which is clear in the small value of the approximation error MSE.

**Second Simulation Experiment**

This example presents two variable input data; let us consider a set of 866 I/O data pair randomly taken from the proposed function. This function presented in the following expression:

$$f(x) = 42.651(0.1 x_1(0.05x_1^4 - 10x_1^2 x_2^2 + 5x_2^4)), \dots x_1, x_2 \in [0,1]$$

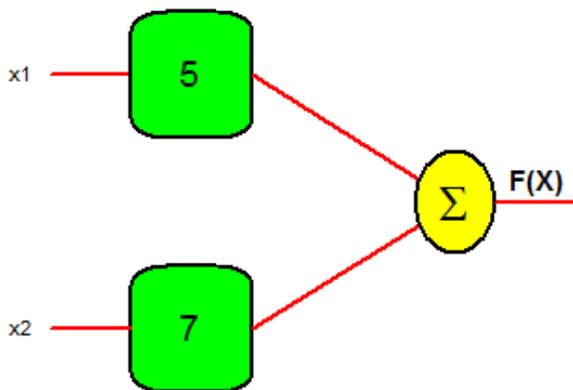
The results obtained by the model present in figures 5 and 6.

**CONCLUSIONS**

This paper based on combining GAs with RBFN applied on a proposed hierarchical topology of sub-RBFNs with the aim of determining the optimal topology of the proposed model and the optimal learning parameters of the training algorithm. The results show that the proposed model GA-HRBFNs obtained better performance in terms of approximation error, with the best topology selected by the model. The obtained results in this work show that the proposed model is an interesting method for complex function approximation problems. The proposed model is capable of the train the RBFN by producing the best structure of the GA-HTRBFN and optimizes the learning parameters of the proposed system. The approximation MSE produced by the selected topology is small, but we can say that the execution time required to construct the best topology and to optimize the learning parameters is not long. So we can use the parallel GAs in the future to improve the model.

**REFERENCES**

- [1] Foqaha, Monir, and Mohammed Awad. "Hybrid Approach to Optimize the Centers of Radial Basis Function Neural Network Using Particle Swarm Optimization." JCP 12.5 (2017): 396-407.
- [2] Conneau, Alexis, et al. "Supervised learning of universal sentence representations from natural language inference data." arXiv preprint arXiv:1705.02364 (2017).
- [3] Awad, Mohammed. "Input Variable Selection Using Parallel Processing of RBF Neural Networks." Int. Arab J. Inf. Technol. 7.1 (2010): 6-13
- [4] Bach, Francis. "Breaking the curse of dimensionality with convex neural networks." Journal of Machine Learning Research 18.19 (2017): 1-53.
- [5] Saleh, Abdulrazak Yahya, Siti Mariyam Shamsuddin, and Haza Nuzly Abdull Hamed. "A hybrid differential evolution algorithm for parameter tuning of evolving spiking neural network." International Journal of Computational Vision and Robotics 7.1-2 (2017): 20-34
- [6] Aljarah, Ibrahim, Hossam Faris, and Seyedali Mirjalili. "Optimizing connection weights in neural networks using the whale optimization algorithm." Soft Computing 22.1 (2018): 1-15.
- [7] Awad, Mohammed. "Optimization RBFNNs parameters using genetic algorithms: applied on function approximation." International Journal of Computer Science and Security (IJCSS) 4.3 (2010): 295.



**Figure 5:** Produced topology GA-HTRBFN model for  $f_2(x)$

```

Editor - C:\Program Files\MATLAB71\work\work3\anetGA\Untitled.m*
File Edit Text Cell Tools Debug Desktop Window Help
[Icons]
1 - 'GA-HTRBFN'
2
3 -     num_SubRBFN: 2
4 -     maxnumInputs: 2
5 -     numInput_RBFN: {[1] [1]}
6 -     numRBF_network: [5 7]
7 -         data: [866x2 double]
8 -         C: {[5x1 double] [7x1 double] }
9 -         W: {[5x1 double] [7x1 double] }
10 -        R: {[5x1 double] [7x1 double] }
11 -        Target: [866x1 double]
12 -        ActivFun: 'gaussian : the Activation Function'
13 -        OutPFunc: 'linear : The Output Calculation Function'
14 -        MSE: 0.0092
15
getdata.m x Untitled* x Untitled2* x Untitled.m* x
script Ln 15 Col 3 OVR
    
```

**Figure 6:** Produced topology parameters, learning parameters and MSE using GA-HTRBFN model for  $f_2(x)$

- [8] Zhang, Tengfei, Dan Liu, and Dong Yue. "Rough Neuron Based RBF Neural Networks for Short-Term Load Forecasting." *Energy Internet (ICEI)*, IEEE International Conference on. IEEE, 2017.
- [9] Chen, Yuehui, Yan Wang, and Bo Yang. "Evolving hierarchical RBF neural networks for breast cancer detection." *International Conference on Neural Information Processing*. Springer, Berlin, Heidelberg, 2006.
- [10] Peng, Lizhi, Bo Yang, and Yuehui Chen. "Hierarchical RBF neural network using for early stage internet traffic identification." *Computational Science and Engineering (CSE)*, 2014 IEEE 17th International Conference on. IEEE, 2014.
- [11] Guan, Xinyu, et al. "Hierarchical neural network for online news popularity prediction." *Chinese Automation Congress (CAC)*, 2017. IEEE, 2017.
- [12] Dai, Xingyuan, et al. "DeepTrend: A Deep Hierarchical Neural Network for Traffic Flow Prediction." *arXiv preprint arXiv:1707.03213* (2017).
- [13] Hierarchical deep neural network for multivariate regression." *Pattern Recognition* 63 (2017): 149-157.
- [14] Weingaertner, Daniel, et al. "Hierarchical evolution of heterogeneous neural networks." *Evolutionary Computation*, 2002. CEC'02. Proceedings of the 2002 Congress on. Vol. 2. IEEE, 2002
- [15] Kokkinos, Yiannis, and Konstantinos G. Margaritis. "Topology and simulations of a hierarchical Markovian radial basis function neural network classifier." *Information Sciences* 294 (2015): 612-627
- [16] Chen, Yuehui, Yan Wang, and Bo Yang. "Evolving hierarchical RBF neural networks for breast cancer detection." *International Conference on Neural Information Processing*. Springer, Berlin, Heidelberg, 2006.
- [17] Dhahri, Habib, and Mohamed Adel Alimi. "Hierarchical learning algorithm for the beta basis function neural network." *arXiv preprint arXiv:1210.8124* (2012).
- [18] Chen, Yuehui, Lizhi Peng, and Ajith Abraham. "Hierarchical radial basis function neural networks for classification problems." *International Symposium on Neural Networks*. Springer, Berlin, Heidelberg, 2006.
- [19] Dhahri, Habib, Adel M. Alimi, and Ajith Abraham. "Hierarchical multi-dimensional differential evolution for the design of beta basis function neural network." *Neurocomputing* 97 (2012): 131-140.
- [20] Afiatdoust, F., and M. Esmailbeigi. "Optimal variable shape parameters using genetic algorithm for radial basis function approximation." *Ain Shams Engineering Journal* 6.2 (2015): 639-647.
- [21] Hartigan J. and Wong M., "Algorithm AS 136: a K-means Clustering Algorithm," *Applied statistics*, vol. 28, no. 1, pp. 100-108. 1979.
- [22] Altman, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46 (3), (1992). 175-185
- [23] Golub, G. H., & Van Loan, C. F. *Matrix Computations*. Hopkins University Press. (2013).
- [24] Qasrawi, Ibrahim, and Mohammad Awad. "Prediction of the power output of solar cells using neural networks: solar cells energy sector in Palestine." *Int J Comput Sci Secur IJCSS* 9.6 (2015): 280.
- [25] Awad, Mohammed. "Enhanced hybrid method of divide-and-conquer and RBF neural networks for function approximation of complex problems." *Turkish Journal of Electrical Engineering & Computer Sciences* 25.2 (2017): 1095-1105.
- [26] Mohammed Awad, Monir Foqaha. "Email Spam Classification Using Hybrid Approach of RBF Neural Network and Particle Swarm Optimization, *International Journal of Network Security & Its Applications* 8(4):17-28,.