# Artificial RBF Neural Network for Big Data Analysis and New Cases Prediction

Faiez Musa Lahmood Alrufaye

Middle Technical University, Baghdad, Iraq

Faiez.Alrufaye@mtu.edu.iq

## Abstract

AI-based treatments have shown promise in a variety of fields, particularly those directly connected to human health. Some AI processors are used to categorize and distinguish groupings and patterns, while others are used to forecast future values based on data from previous study and the environment in which that data was employed. An artificial neural network that employs radial basis functions as activation functions is known as a radial basis function network. The radial basis functions input and the neural parameters are combined linearly to produce the network output. There are several applications for radial-based functional networks, such as function approximation, classification, time series prediction, and system control. In this paper, the RBF network will be used in two phases: the data training phase, where the data is trained with the inputs and outputs to obtain new values for the outputs and compare them with the original outputs, and the testing phase, where only the inputs are entered without the outputs and the outputs are evaluated using the RMSE calculation, where it reached a performance of RMSE of 0.018. In the training phase of utilizing the system, the mistake rate was 0.04 and the success rate was 96%; in the testing phase, the error rate was 0.05 and the success rate was 95%.

**Keywords: RBF; Big data; Artificial intelligence; Neural networks.**

## 1. Introduction

Treatments using artificial intelligence have been successful in many aspects of life, particularly those directly connected to human health.

Artificial intelligence processors are employed in a variety of ways; some of them are used to categorize and separate classes and patterns, while others are used to forecast new values using data from earlier studies and the environment in which that data functions.

A radial basis function network is an artificial neural network that employs radial basis functions as activation functions in the area of mathematical modeling. A linear combination of the inputs' radial basis functions and the neuronal parameters makes up the network's output. Numerous applications of radial basis function networks exist, such as function approximation, classification, time series prediction, and system control. Broomhead and Lowe, two researchers of the Royal Signals and Radar Establishment, initially proposed them in a 1988 study [1,2,3].

For issues involving function approximation, radial basis function (RBF) networks are a popular variety of artificial neural network. The universal approximation and quicker learning pace of radial basis function networks set them apart from other neural networks. Three layers, the input layer, the hidden layer, and the output layer, make up an RBF network, a form of feed-forward neural network [4]. These layers each do distinct responsibilities. Once the computed error has achieved the desired values (i.e. 0.01) or the required number of training iterations (i.e. 500), the RBF model's training is finished. It is decided to use an RBF network with 10 nodes or more in the hidden layer. In computing units, the transfer function is a Gaussian function [5].

In this study, RBF networks will be used in two stages: the data training stage, where the RBF network trains the data with inputs and outputs to obtain new values for the outputs, compares them to the original outputs, and the test stage, where only the inputs are entered without the outputs, and the outputs are assessed using the RMSE calculation.

During training and testing, large amounts of data that were gathered by specialized equipment were employed. These data include meteorological variables that may be used to forecast temperature and humidity in the upcoming years using a variety of atmospheric features.

## 2. Related Works

By accurately forecasting future outcomes that would be similar to the current results, RBF neural networks contributed significantly to numerous studies by data science experts. We will explore the most significant researchers and their work in this subject in this section of the essay.

In their study, Zhongqi Wang, Bo Yang, Yonggang Kang, and Yuan Yang suggested the RBF neural network prediction model as a tool for designing and improving the placement arrangement of sheet metal assemblies. From the training dataset chosen by systematic sampling and finite element simulation analysis, the RBF neural network model was created. The proposed strategy was then tested using a case study [6].

The authors of the research by Baoliang Ma, Yuzhu Zhang, and Lixing Ma first examined calcium ferrite (CFA), a crucial precursor substance for the production of complex calcium copper (SFCA), before examining a number of XRD phase detection procedures for the production of complex calcium scatterin, likewise pre-processing the data. To determine the impact of MgO and Al2O3 on the production of the composite calcium ferrite, data correlation and data fit analysis were coupled with energy spectrum analysis of the phase diagram of the composite calcium ferrite. Then, to forecast the direction of formation of complicated calcium scatterers, a modified RBF neural network model utilizing the resource allocation network (RAN) method was applied. As opposed to the conventional RBF neural network, the Dragonfly method optimizes the neural network parameters. With a forecast result of 97.6%, it was discovered that the revised algorithm had a big prediction accuracy. Finally, the outcomes of comparative metallurgical tests and data analysis served as the foundation for the anticipated findings. This article has proven the results' reliability and correctness [7].

A method based on the radial basis function (RBF) optimized using the quantitative genetic algorithm (QGA) to construct the photosynthesis rate prediction model is suggested in the study by Liuru Pu et al. that examines the impact of light quality on photosynthetic rate. They used "Golden Embryo Formula 2 98-1F1" cucumber seedlings as their experimental subjects, and they utilized the LI-6800 to record photosynthetic rates at various temperatures, light intensities, and light quality. The suggested model was tested and trained using the experimental data. The straight slope for linear fitting was 1.000, the straight intercept for linear fitting was 0.061, and the specific coefficient of the model between predicted and measured values was 0.996. Moreover, six artificial intelligence algorithms were contrasted with the suggested strategy. The comparison findings also showed that, when compared to other algorithms, the suggested model had the best accuracy [8].

In their paper, Eva Chondrodima et al. describe a novel data-driven method for PT-ETA prediction based on RBF neural networks and a modified PSO-NSFM training algorithm. To clean and reconstruct GTFS data, a new preparation pipeline (CR-GTFS) is also created. PSO-NSFM and CR-GTFS together established a comprehensive framework for precisely forecasting PT-ETA using real-world data sources. The suggested technique was validated through experiments using GTFS data, outperforming cutting-edge prediction accuracy and computation speed [9].

In our current research, a system based on RBF neural networks has been built that predicts new values based on big data.

## 3. Radial Basis Function Neural Network Technique

Human intellect assisted in society's growth. The human mind's mode of functioning was purposely replicated by neural networks in order to profit from their style of thinking. The neuronal network performs comparable duties to a human neuron [10]. An RBFNN's artificial neurons are a collection of connected nodes that flexibly mimic the neurons found in a biological brain. Any connection, such as the synapses in a real brain, can transmit a signal to other neurons. A synthetic neuron receives a signal, processes it, and then transmits a signal to the neurons it is connected to [11].

There are two processes in redial basis function neural networks: training and testing. The initial phase is training. The interactions that neurons must have with a set of weights in order to create those outputs are initially taught, as well as the inputs and outputs of neurons [12]. Next, the outputs are matched. During testing, we merely provide the inputs and weights. The system generates outputs in accordance with the knowledge base created throughout the testing phase's training [13]. The training and testing stages are depicted in Figures 1 and 2, respectively.

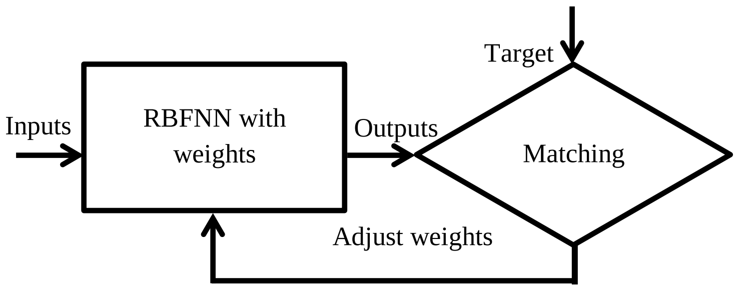


Figure 1. Training stage structure

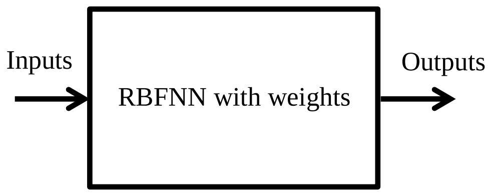


Figure 2. Testing stage structure

Redial basis function neural networks include three layers: Each node in the input layer serves as a representation of a system input. The activation function and neural network function are both contained in the second layer, or hidden layer. Results from data processing in the hidden layer will be provided by the output layer. In figure 3, RBF neural network layers are shown [14].

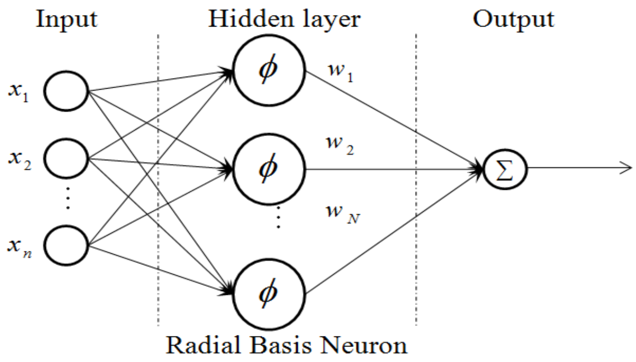


Figure 3. RBFNNs layers

The number of nodes in the input layer corresponds to the number of inputs for the redial basis neural networks, which serve as a proxy for the program's parameters, according to the picture above [15]. The number of nodes in the hidden layer depends on the weights assigned to each parameter. When the weights assigned to a particular parameter are higher, the results are more comparable to those obtained during the training phase [16].

## 4. RBFNN and Gaussian Processes

The supervised learning techniques RBF Neural Networks (RBFNNs) and Gaussian Processes (GPs) are both extremely expressive [16]. The quantity and types of parameters utilized in each framework may be used as one criterion for comparing the similarities and differences between these two models. The training dataset used to create Gaussian Processes (GPs) impacts how well they forecast in major part, and they learn considerably fewer parameters than other non-parametric machine learning algorithms [17]. Their selection of values conveys all about the factors:

- Kernel/covariance function (k(x,x'))
- Mean function (m(x))
- Likelihood covariance noise ($\sigma^2$).

These choices or variables are commonly referred to as "hyper parameters". Gaussian Processes (GPs) do not employ any parameters, in striking contrast to many contemporary Deep Neural Networks (DNNs), which attempt to address machine learning challenges by utilizing as many parameters, or weights, as feasible. Using a lot of parameters has always been discouraged in the literature on statistical learning since it is predicted to result in considerable over-fitting and poor generalization to out-of-distribution data. This fundamental statistical learning theory did not account for the practical success of over-parameterized neural networks, which gave rise to a new theory of the "over-parameterized" or "interpolation" regime [11, 12].

In mathematics, a function of this kind is commonly referred to as a Gaussian function [12]:

$$f(x) = a \exp\left(-\frac{(x-b)^2}{2c^2}\right) \quad (1)$$

Where random real variables a, b, and c that are non-zero It bears the name of the mathematician Carl Friedrich Gauss. The graph of a Gaussian has a recognizable symmetric "bell curve" shape. Parameter a controls the height of the curve's peak, parameter b determines where the peak's center is located, and parameter c determines how wide the "bell" is (the standard deviation, also known as the Gaussian RMS width).

We may achieve results with the Gaussian function that are very close to those attained with the RBF neural networks function during the training stage [18]:

$$y = \sum_{1}^{n} ( w_k x_k ) + b \quad (2)$$

Using the perfect weights ($w_k$) discovered, back propagation (gradient) techniques are employed to lower ($y=||y-y_{desired}||^2$ ).

## 5. RBFNNs System Proposed

### 5.1 Dataset

Throughout the training and testing phase, sophisticated equipment collected a significant amount of data. Using a variety of atmospheric indicators, this data gives meteorological elements that may be used to forecast temperature and humidity for the upcoming years.

The dataset contains 24 values, of which 12 parameters represent the neural networks' inputs and the remaining 12 represent their outputs. In addition to the quantity of carbon dioxide ($CO_2$), which has a big influence on temperature and humidity, the data often covers atmospheric parameters such air temperatures and humidity, all recorded at various times.

### 5.2 RBFNNs phases

The two main stages of our system are the training stage for neurons, which involves entering the parameters of the system and generating random weights ranging from 0 to 1 for each of these parameters, and the application stage, which involves using the RBF neural network function on it and comparing the results to those obtained using our data. The second stage is testing, when we only input the parameters without considering the outcomes and gauge how well the recommended system works.

In our current system and according to the data, we have 24 parameters, 12 of which are inputs, and 12 other parameters represent weather characteristics.

#### *5.2.1 The training phase*

For our currently suggested system, this stage is essential since the training of the neurons during this stage heavily influences the outcomes of the subsequent stage. An Excel file with the settings for the system's input and the outputs for comparison is used to develop the system using the Matlab application. As the user enters the necessary number of nodes to approach the provided results, the system creates random weights between 0 and 1 for each parameter in each node, allowing them to increase the number of nodes in the hidden layer. It is important to note that the findings are more accurate when there are more neurons in the hidden layer. The RBF neural network function creates the inputs and their weights, and the following function sums the results:

$$y = \sum_{1}^{n} \left( w_k \, x_k \right) + b$$

$w_k$ is a weight, b is a threshold, and $x_k$ is an input parameter. With the help of Matlab's random function, the weight $w_k$ is calculated as a random value between (0,1). The ultimate outcome is obtained using the Gaussian function. RBF neural networks require the conversion of an input signal into an output signal by the Gaussian function in order to capture complex non-linear patterns that are ignored by more straightforward models.

### *5.2.2 The testing phase*

The RBF network was trained on the data during the training phase, and after its output was compared to the initial output received by measuring devices, it was feasible to go on to the testing phase of the proposed system. In the testing phase, just the input parameters are entered; the system is then run to produce outputs, which must be reasonably similar to the readings from measurement instruments. The testing phase completely depends on how the network was trained in the training phase, and it is associated with a positive connection, meaning that whenever the training phase's findings are correct, so are the test results. The chart below shows the schematic of the RBFNNs algorithm, and how it works:

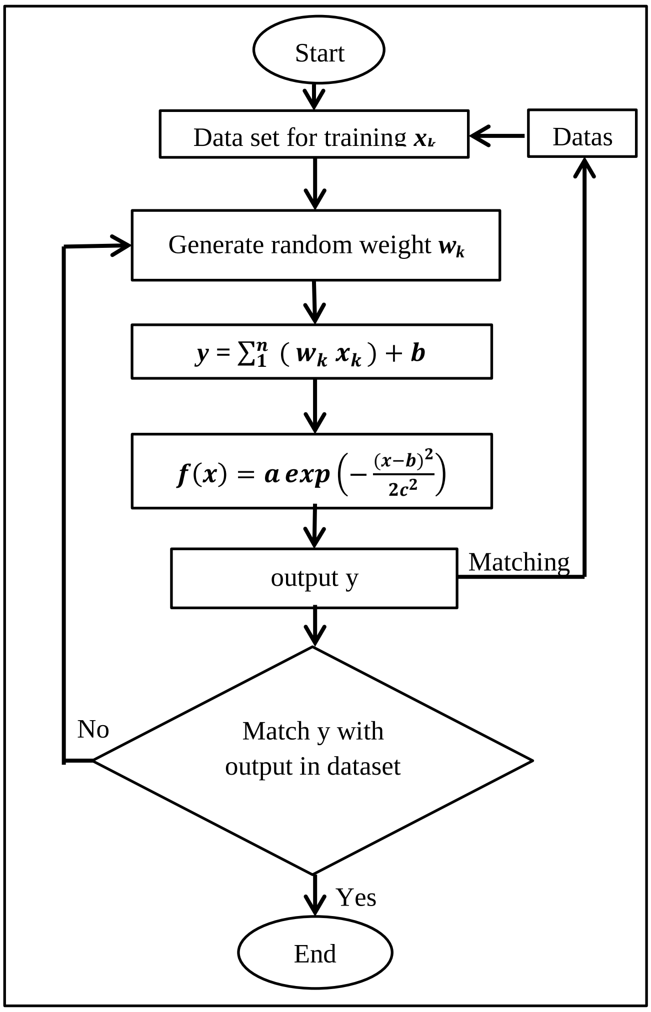


Figure 4. RBFNNs algorithm flowchart

## 5.3 The results and discussion

The Matlab-based RBF neural network program is created and implemented, and an Excel file is used to supply the system's input parameters. RBF Neural Networks function in two steps: first, the neurons are trained using inputs and outputs; second, the hidden layer is constructed using those inputs and outputs and uses the RBF Neural Networks equation to provide results that are reasonably near to the outputs. In this portion of the study, the results of the recommended system will be discussed. Results from the training period and those from the testing phase were divided as follows:

### *5.3.1 The training phase results*

The equation is applied to the inputs in the RBF neural network technique to produce new values, which are then compared to our target values. With one additional node from the base term, the input layer contains more nodes than the input layer alone. User input determines the number of nodes in the hidden layer; the more nodes, the more accurate the output. For instance, 50 examples were input as the hidden layer's number of nodes during the learning stage, and while the results were good but not quite what we needed, we upped the number to 100 and obtained excellent results. There are 12 nodes for results. After implementing the program and the first stage, we obtain results that are rather near to the intended outcomes; Table 1 lists the first layer's inputs and outputs during the training phase.

As shown in the table below, there are 12 inputs that reflect temperatures in various environments, such as outside the home or in the income, in addition to humidity levels, CO2 levels, and other inputs that have an impact on the weather.

Twelve numbers that represented the weather's temperature and humidity over time were the outputs. With 12 outcomes and the use of the **CLUSTERING** approach to determine which inputs are the best for the outputs and results, the current system poses a significant difficulty.

Table 1. The training phase results

| Init. T | Init. H | Initi. Co2 | Max speed | Min speed | Intensity | outdoor T | outdoor H | outdoor Co2 | T setpoint | H setpoint | Co2 setpoint | t1 T | t2 T | t3 T | Y1 T | Y2 T | Y3 T | t1 H | t2 H | t3 H | Y1 H | Y2 H | Y3 H |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 25 | 0.7 | 1500 | 412 | 868 | 973 | 33 | -8 | -8 | 36 | 99 | 148 | 0.9 | 0.1 | -0.3 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 23 | 0.7 | 1500 | 500 | 935 | 1170 | 33 | -9 | -8 | 37 | 80 | 130 | 1 | 0 | 0 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 21 | 0.7 | 1500 | 394 | 996 | 1144 | 33 | -8 | -10 | 35 | 67 | 122 | 1 | 0 | 0 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 20 | 0.7 | 1500 | 396 | 976 | 845 | 33 | -10 | -8 | 20 | 76 | 125 | 1 | 0 | 0 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 18 | 0.7 | 1500 | 381 | 900 | 998 | 31 | -10 | -8 | 55 | 158 | 273 | 0.9 | -0.1 | -0.1 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 25 | 0.6 | 1500 | 375 | 998 | 1165 | 30 | -9 | -9 | 32 | 72 | 126 | 0.9 | -0.1 | -0.1 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 23 | 0.6 | 1500 | 352 | 819 | 888 | 28 | -9 | -9 | 36 | 86 | 140 | 1 | -0.1 | -0.3 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 21 | 0.6 | 1500 | 200 | 746 | 805 | 30 | -10 | -10 | 28 | 55 | 117 | 1 | 0 | 0 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 20 | 0.6 | 1500 | 377 | 1000 | 1000 | 30 | -10 | -8 | 20 | 74 | 119 | 1 | 0 | 0 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 18 | 0.6 | 1500 | 491 | 943 | 953 | 30 | -8 | -9 | 50 | 79 | 138 | 1 | -0.1 | -0.1 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 25 | 0.5 | 1500 | 392 | 999 | 999 | 29 | -9 | -9 | 26 | 56 | 139 | 1 | -0.1 | -0.3 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 23 | 0.5 | 1500 | 329 | 804 | 811 | 30 | -9 | -8 | 32 | 64 | 126 | 0.9 | -0.1 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 21 | 0.5 | 1500 | 362 | 998 | 878 | 29 | -10 | -8 | 20 | 73 | 125 | 0.9 | -0.1 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 20 | 0.5 | 1500 | 375 | 998 | 1165 | 30 | -9 | -9 | 32 | 72 | 126 | 0.9 | -0.1 | -0.1 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 18 | 0.5 | 1500 | 240 | 804 | 912 | 30 | -10 | -10 | 27 | 57 | 113 | 0.9 | -0.1 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 25 | 0.4 | 1500 | 298 | 800 | 831 | 29 | -9 | -9 | 29 | 77 | 118 | 0.8 | -0.2 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 23 | 0.4 | 1500 | 428 | 937 | 1018 | 30 | -9 | -8 | 37 | 85 | 140 | 0.8 | -0.2 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 21 | 0.4 | 1500 | 467 | 811 | 895 | 30 | -10 | -8 | 31 | 89 | 137 | 0.8 | -0.2 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 20 | 0.4 | 1500 | 361 | 997 | 871 | 29 | -10 | -8 | 20 | 83 | 133 | 0.8 | -0.2 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 18 | 0.4 | 1500 | 207 | 804 | 911 | 30 | -10 | -10 | 27 | 66 | 118 | 0.8 | -0.2 | -0.2 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 25 | 0.7 | 1500 | 419 | 687 | 866 | 33 | -8 | -8 | 34 | 82 | 147 | 1 | 0 | 0 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 23 | 0.7 | 1500 | 386 | 946 | 956 | 33 | -8 | -8 | 28 | 70 | 113 | 1 | 0 | 0 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 21 | 0.7 | 1500 | 367 | 956 | 993 | 30 | -8 | -8 | 32 | 74 | 125 | 0.9 | -0.1 | -0.2 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 20 | 0.7 | 1500 | 305 | 822 | 840 | 30 | -8 | -9 | 28 | 63 | 110 | 1 | 0 | 0 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 18 | 0.7 | 1500 | 353 | 788 | 704 | 30 | -9 | -9 | 75 | 256 | 234 | 0.9 | -0.1 | -0.1 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 25 | 0.6 | 1500 | 385 | 950 | 821 | 29 | -8 | -8 | 32 | 80 | 128 | 0.9 | -0.1 | -0.2 |

### *5.3.2 The testing phase results*

In the testing phase, new outputs are only produced using inputs that have no outputs. The same inputs as in the training stage are imported into the system, and then the RBF neural network function is implemented within the hidden layer. The user then enters the same number of nodes for the hidden layer as in the training stage, and we receive the results based on the training for neurons in the training stage. Table 2 lists the system inputs and results for the testing phase.

The first and second tables display the results of the RBF neural networks' two phases as well as the degree of their convergence. The application has been designed with an interface to calculate the Root Mean Square Error (RMSE) supplied by Equations (3).

$$\text{RMSE} = \frac{1}{\text{N}} \sum (R_O - R_P) \tag{3}$$

Where $R_o$ and $R_p$ are the observed and estimated concentrations at the time steps, respectively, and N is the total number of observations in the data set. Although it is used much more frequently and receives more attention than minor mistakes, the RMSE is a metric for evaluating errors [14].

The RMSE number for the training phase was 0.04, which is a very excellent percentage as it shows the error rate of the trained data and its opposite is 96%, which represents the correctness of the proposed system's findings during the training phase.

Although the testing phase's RMSE score is 0.05, the suggested system's testing phase findings are 95% accurate.

Table 2. The testing phase results

| Init. T | Init. H | Initi. Co2 | Max speed | Min speed | Intensity | outdoor T | outdoor H | outdoor Co2 | T setpoint | H setpoint | Co2 setpoint | t1 T | t2 T | t3 T | Y1 T | Y2 T | Y3 T | t1 H | t2 H | t3 H | Y1 H | Y2 H | Y3 H |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 25 | 0.7 | 1500 | 413 | 869 | 973 | 33 | -8 | -8 | 36 | 99 | 148 | 0.9 | 0.1 | -0.31 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 23 | 0.7 | 1500 | 499 | 935 | 1170 | 33 | -9 | -8 | 37 | 80 | 130 | 1 | 0 | 0.1 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 21 | 0.7 | 1500 | 394 | 996 | 1144 | 33 | -8 | -10 | 35 | 67 | 122 | 1 | 0 | 0.1 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 20 | 0.7 | 1500 | 396 | 976 | 845 | 33 | -10 | -8 | 20 | 76 | 125 | 1 | 0 | 0.1 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 18 | 0.7 | 1500 | 381 | 900 | 998 | 31 | -10 | -8 | 55 | 158 | 273 | 0.9 | -0.1 | -0.13 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 25 | 0.6 | 1500 | 375 | 998 | 1164 | 30 | -9 | -9 | 32 | 72 | 126 | 0.9 | -0.1 | -0.1 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 23 | 0.6 | 1500 | 352 | 819 | 888 | 28 | -9 | -9 | 36 | 86 | 140 | 1 | -0.1 | -0.3 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 21 | 0.6 | 1500 | 200 | 746 | 805 | 30 | -11 | -11 | 28 | 55 | 118 | 1 | 0 | 0 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 20 | 0.6 | 1500 | 377 | 1000 | 1003 | 30 | -11 | -8 | 20 | 75 | 119 | 1 | 0 | 0 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 18 | 0.6 | 1500 | 491 | 943 | 953 | 30 | -8 | -9 | 51 | 79 | 138 | 1 | -0.1 | -0.1 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 25 | 0.5 | 1500 | 392 | 999 | 1000 | 29 | -9 | -9 | 26 | 56 | 139 | 1 | -0.1 | -0.3 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 23 | 0.5 | 1500 | 329 | 804 | 811 | 30 | -9 | -8 | 32 | 64 | 126 | 0.9 | -0.1 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 21 | 0.5 | 1500 | 362 | 998 | 878 | 29 | -10 | -8 | 20 | 73 | 125 | 0.9 | -0.1 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 20 | 0.5 | 1500 | 375 | 998 | 1165 | 30 | -9 | -9 | 32 | 72 | 126 | 0.9 | -0.1 | -0.1 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 18 | 0.5 | 1500 | 240 | 804 | 912 | 30 | -10 | -10 | 27 | 57 | 113 | 0.9 | -0.1 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 25 | 0.4 | 1500 | 298 | 800 | 831 | 29 | -9 | -9 | 29 | 77 | 118 | 0.8 | -0.2 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 23 | 0.4 | 1500 | 428 | 937 | 1018 | 30 | -9 | -8 | 37 | 85 | 140 | 0.8 | -0.2 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 21 | 0.4 | 1500 | 467 | 811 | 895 | 30 | -10 | -8 | 31 | 89 | 137 | 0.8 | -0.2 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 20 | 0.4 | 1500 | 361 | 997 | 871 | 29 | -10 | -8 | 20 | 83 | 133 | 0.8 | -0.2 | -0.2 |
| **-5** | 0.5 | 200 | 10 | 1 | 400 | -5 | 0.5 | 200 | 18 | 0.4 | 1500 | 207 | 804 | 911 | 30 | -10 | -10 | 27 | 66 | 118 | 0.8 | -0.2 | -0.2 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 25 | 0.7 | 1500 | 419 | 687 | 866 | 33 | -8 | -8 | 34 | 82 | 147 | 1 | 0 | 0 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 23 | 0.7 | 1500 | 386 | 946 | 956 | 33 | -8 | -8 | 28 | 70 | 113 | 1 | 0 | 0 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 21 | 0.7 | 1500 | 367 | 956 | 993 | 30 | -8 | -8 | 32 | 74 | 125 | 0.9 | -0.1 | -0.2 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 20 | 0.7 | 1500 | 305 | 822 | 840 | 30 | -8 | -9 | 28 | 63 | 110 | 1 | 0 | 0 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 18 | 0.7 | 1500 | 353 | 788 | 704 | 30 | -9 | -9 | 75 | 256 | 234 | 0.9 | -0.1 | -0.1 |
| **0** | 0.5 | 200 | 10 | 1 | 400 | 0 | 0.5 | 200 | 25 | 0.6 | 1500 | 385 | 950 | 821 | 29 | -8 | -8 | 32 | 80 | 128 | 0.9 | -0.1 | -0.2 |

## 6. Conclusion

The most important of them is the need for the system to collect a lot of data since it depends much on how many weather factors are fed into the test algorithm to provide findings that are comparable to the desired outcomes. We developed and implemented a program for analyzing big data and forecasting new cases using the RBF neural network. The information entered into the system has a big impact on it. The accuracy and robustness of the findings are significantly influenced by the hidden layer's number of neurons since more neurons in the hidden layer result in heavier weights, which improve system output. Root mean square error (RMSE) was used to test the system, and the outcomes were excellent. We may attain better outcomes by utilizing extra system components or by combining multiple artificial intelligence algorithms.

**References**

[1] Broomhead, D. S.; Lowe, David, "Radial basis functions, multi-variable functional interpolation and adaptive networks (Technical report)". RSRE. 4148, 1988.

[2] Broomhead, D. S.; Lowe, David, "Multivariable functional interpolation and adaptive networks" (PDF). Complex Systems. 2: 321–355, 1988.

[3] Jump up to:a b Schwenker, Friedhelm; Kestler, Hans A.; Palm, Günther, "Three learning phases for radial-basis-function networks". Neural Networks. 14 (4–5): 439-458, 2011.

[4] M. M. I. Al-Sagheer and F. M. L. Alrufaye, "Data Mining and RBF Neural Networks to Analyze Data from COVID-19 Patients and Predict New Cases Based on Symptoms," 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2022, pp. 1-6, doi: 10.1109/HORA55278.2022.9799979.

[5] V.N. Delgrange, N. Cabassud, M. Cabassud, L. Durand Bourlier, J.M. Laine, "Neural networks for prediction of ultrafiltration transmembrane pressure: application to drinking water production ", Journal of Membrane Science 150, Pages 111–123, 1998.

[6] Zhongqi Wang, Bo Yang, Yonggang Kang, Yuan Yang, "Development of a Prediction Model Based on RBF Neural Network for Sheet Metal Fixture Locating Layout Design and Optimization", Computational Intelligence and Neuroscience, vol. 2016, Article ID 7620438, 6 pages, 2016. https://doi.org/10.1155/2016/7620438.

[7] Baoliang Ma, Yuzhu Zhang, Lixing Ma, "Research on the Formation Mechanism of MgO and Al2O3 on Composite Calcium Ferrite Based on DA-RBF Neural Network", Computational Intelligence and Neuroscience, vol. 2022, Article ID 4327969, 12 pages, 2022. https://doi.org/10.1155/2022/4327969.

[8] Pu, L., Li, Y., Gao, P. et al. A photosynthetic rate prediction model using improved RBF neural network. Sci Rep 12, 9563 (2022). https://doi.org/10.1038/s41598-022-12932-9.

[9] Eva Chondrodima, Harris Georgiou, Nikos Pelekis, Yannis Theodoridis, "Particle swarm optimization and RBF neural networks for public transport arrival time prediction using GTFS data", International Journal of Information Management Data Insights, Volume 2, Issue 2, 2022.

[10] R. A. Naji, E. J. Khadum, " Efficiency of Using Artificial Neural Networks to Predict Child Mortality in Najaf Province and Neuroshell Predictor Program", AL GHAREE for Economics and Administration Sciences, Vol.14, No.38 Pages 311–320, 2016.

[11] Faiez Musa L. Alrufaye, Hakeem I. Mhaibes, Ali F. Neamah, " Neural Networks Algorithm for Arabic Language Features-Based Text Mining", IOP Conference Series: Materials Science and Engineering, Volume 1045, https://doi.org/10.1088/1757-899X/1045/1/012003.

[12] Belkin, Mikhail. "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation." arXiv preprint arXiv:2105.14368, 2021.

[13] Jacot, Arthur, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks." arXiv preprint arXiv:1806.07572, 2018.

[14] Damianou, Andreas, and Neil D. Lawrence. "Deep gaussian processes." Artificial intelligence and statistics. PMLR, 2013.

[15] Blomqvist, Kenneth, Samuel Kaski, and Markus Heinonen. "Deep convolutional Gaussian processes." arXiv preprint arXiv:1810.03052, 2018.

[16] Mikut Ralf, Reischl Markus, "Data Mining Tools"، Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery، 1 (5): 431–445، 2011.

[17] Hastie Trevor, Tibshirani, Robert, Friedman Jerome, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", 2011.

[18] Faiez Musa L. Alrufaye, Firas H. Neamah, Hakeem I. Mhaibes, Marwah Th. Ali, "Use of Artificial Neural Network method to Predict the Amount of Oxygen in the Tigris River ", IOP Conference Series: Materials Science and Engineering, Volume 1076, 2021.

Article submitted 10 November, 2022. Published as resubmitted by the authors 16 Dec. 2022.