



PREDICTION OF MICROWAVE ATTENUATION DUE TO RAIN USING ARTIFICIAL NEURAL NETWORK

Marcel, F.T.¹ and Ibrahim, S.B.²

Department of Electrical Engineering, Bayero University Kano, Nigeria

ARTICLE INFO

Key words:

Attenuation,
Prediction,
Microwave,
Scattering,
Absorption

Correspondence author:

e-mail:Email:

marcelfote@yahoo.fr and

SabKibr@yahoo.com

Phone:

ABSTRACT

Electromagnetic (EM) signal attenuation due to rain has long been recognized as a major limitation to reliable communication system operation at frequencies above 10GHz. Rain attenuation is one of the most fundamental limitations to the performance of satellite communication links. In this study, Artificial Neural Network (ANN) based model for predicting rain attenuation has been developed. Standard multilayer, feed-forward, back-propagation neural networks have been designed using MATLAB's Neural Network Tool. Eight (8) parameters namely: frequency, rain rate, elevation angle, polarization angles and station altitudes; others are latitude, the longitude and rain height were taken as input to the network, and attenuation as the target. The data used in developing the ANN model were calculated using International Telecommunication Union-Recommendation model (ITU-R). A total number of 2800 samples were used to train the network and the best network structure selected. The overall Mean Square Errors (MSE) during training is 0.0134699, regression value of 0.999396, small percentage of error (0.02175% to 0.16284%) in response to unknown input vectors indicate that the developed model can be used reliably for predicting rain attenuation. The proposed model compares favourably with the ITU-R model, thus it can be used by microwave link designers. Moreover, the model has advantages associated with ANN based systems.

1.0 INTRODUCTION

Rain affects the transmission of an electromagnetic (EM) signal in three ways: It attenuates the signal; it increases the system noise temperature; and it

changes the polarization. All three of these mechanisms cause degradation in the received signal quality and become increasingly significant as the carrier frequency increases.

The first, and most well known, effect of rain is that it attenuates the signal. The attenuation is caused by the scattering and absorption of electromagnetic waves by drops of liquid water. The scattering diffuses the signal, while absorption involves the resonance of the waves with individual molecules of water. Absorption increases the molecular energy, corresponding to a slight increase in temperature, and results in an equivalent loss of signal energy [1]. The actual amount of fading is dependent on the frequency of the signal and the size of the raindrop [2]. The two main causes of rain fading are scattering and absorption. In both satellite and terrestrial communication the prediction of rainfall attenuation is prerequisite in any communication design [2].

Various researchers have made contributions on the problem of rain attenuation prediction: The rain attenuation prediction model of millimeter wave based on artificial neural network was proposed first time, and the predicted results were compared with CCIR recommendation model showed good accuracy [3]. It was also concluded that it is very important to accurately predict rain attenuation for reliable incorporation into design process based on the climate and geographical of the particular region [4].

The artificial neural network (ANN) will be designed using Matlab 7.12.0 environment. The ITU-R rain attenuation model is the most widely accepted model by the international propagation community. The model was first accepted globally in 1982 and is continuously updated as rain attenuation modelling is better understood. The current model has been deliberately kept simple for ease of use [5]. In this research, a multilayer Perceptron with a back propagation learning algorithm will be used to develop

artificial neural network for evaluating the attenuation. This work will consider frequencies ranging from 1 to 100GHz and rain rate from 10 to 100 mm/h.

2.0 DESCRIPTION OF ANN MODEL

Artificial Neural Networks became an interesting tool of modeling of linear and non-linear systems without the need to make implicit assumptions as in most of classical statistical modeling approaches. Nowadays, ANNs are widely used in many applications in various aspects of science and engineering [6], [7]. ANNs can be grouped into two major categories: feed-forward and feedback (recurrent) networks. In the former network, no loops are formed by the network connections, while one or more loops may exist in the latter. The most commonly used family of feed-forward networks is a layered network in which neurons are organized into layers with connections strictly in one direction from one layer to another [8].

In most problems a feed-forward neural network (FFNN) with the Back-Propagation (BP) training algorithm is being used, consisting of input, hidden and output layer which is called the multi-layer perceptron (MLP). In this network the data flow forward to the output continuously without any feedback. Feed-forward neural networks without recurrent connections are still preferred to be used in scientific and industrial application because the way humans use recurrence in learning language or logical decisions is not completely recognised. On the other hand, training of recurrent neural networks is difficult to manage due to complexity of chaotic behaviour [9]. Each neuron computes a weighted sum of the inputs it receives and adds it with a bias (b) to form the net input (x). The bias is included in the neurons to allow the activation function to be offset from zero [10],

$$x = w_{1,1} \cdot P_1 + w_{1,2} \cdot P_2 + \dots + w_{1,j} \cdot P_j \quad (2.1)$$

Where:

$w_{1,j}$ Is the weight between the hidden layer and output layer.

x Is the net input

P_j Is the input

The net input (x) is then passed to the subsequent layer through a non-linear Logsig or Tansig function to form its own output (y_i).

$$y_i = \frac{1}{1 + e^{-x}} \quad (2.2)$$

Afterward, the output was compared with the target output using an error function of the form

$$\delta_k = (t_j - y_j) y_j (1 - y_j) \quad (2.3)$$

For the neuron in the hidden layer, the error term is given by the following equation [10]:

$$\delta_j = y_j (1 - y_j) \sum_k \delta_k w_k \quad (2.4)$$

Where:

δ_j is the error term of the output layer,

The error was then propagated backward from the output layer to the input layer to update the weight of each connection as follows [11]:

$$w_{ji}(t+1) = w_{ji}(t) + \eta \delta_j y_j + \alpha w_{ji}(t) - w_{ji}(t-1) \quad (2.5)$$

Where:

η Is learning rate

α Is the momentum factor

And α determines the effect of past weight changes on the current direction of

movement. Both of these constant terms are specified at the start of the training cycle and determine the speed and stability of the network.

Implementing Artificial Neural Network Model implies three steps:

- a) Choice of network model. (Feed-forward neural network was chosen)
- b) Correct dimensioning of the network;
 - The network must be able to learn the input data;
 - The network must be able to generalize for similar input data that were not in training set.
- c) The training of the network using existing data (synaptic weights synthesis).

2.2 Organization of Data

The data set contained 2800 input/output samples. One input sample consists of 8 inputs as follows: frequency f (GHz), rain rate R (mm/h), elevation angle θ (degree), polarization angles Ψ (degree), station altitudes (km), the latitude $lat.$, the longitude log , rain height and the corresponding output attenuation (dB). The 2800 data are obtained using 4 stations [12].

2.3 Optimal Neural Network Structure

The first step in training a feed-forward network is to create the network object. The function `newff` creates a feed-forward network. It requires three arguments and returns the network object. The first argument is the input vectors (frequency, rain rate...).

The second argument is a target vectors (attenuation). The inputs and outputs are used to set up network input and target dimensions and parameters. The third argument is an array containing the sizes

of each hidden layer. (The output layer size is determined from the targets.) . Fig 2.1 show schematic diagram of multilayer neural network

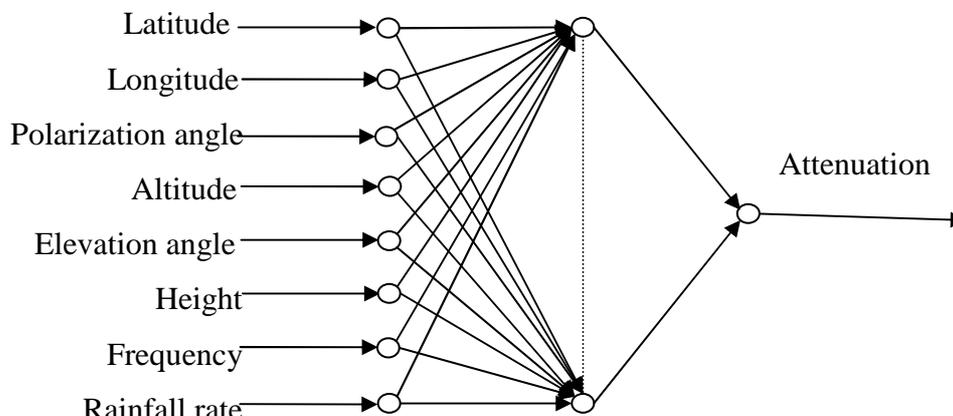


Fig 2.1 Schematic Diagram of Multilayer Neural Network

2.4 Design approaches for feed-forward neural networks

Step 1: Importing data

The input and output vectors were imported into matlab workspace environment from Microsoft excel 2003. And those data are partitioned into training, validation and testing sets. The partition is as follows (60%, 20%, 20% for training /test / validation). The training samples (1680 samples) which are presented to the network during training and the network are adjusted according to its error by random division, validation samples (560 samples) which are used to measure network generalization, and to halt training when generalization stops improving. Finally, we have the testing samples (560 samples) which have no effect on training and so provide an independent measure of network performance during and after training.

Step 2: Network structure

A feed-forward neural network was employed with Herperbolic tangent sigmoid transfer function (tansigmoid) and linear output neurons (purelin) function were used at the hidden and output layer respectively. A practical way to find the optimum structure is to begin a minimal structure and training the network by training data set. The mean square error (MSE) and correlation coefficient between actual and predicted values (R^2) are determined for performance of network on testing data set different from training data set. The structure was enlarged by adding new hidden layers and increasing the nodes in each hidden layer. Various configurations are tried using the trial and error method and the configuration with the minimum MSE and the maximum R^2 is selected. The network structure is shown in Fig 2.2.

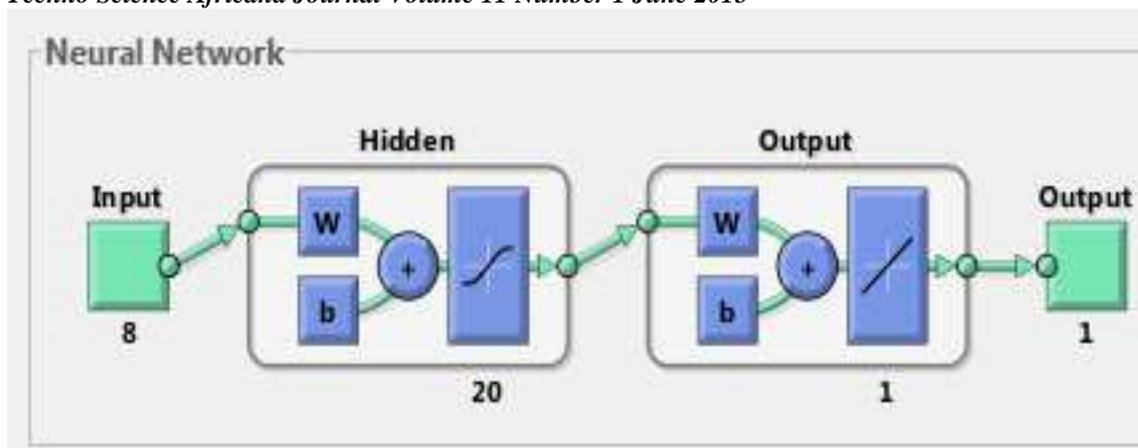


Fig 2.2 Network structure

Step 3 network training

Levenberg-Marquardt optimization. trainlm is often the fastest backpropagation algorithm in the toolbox, and is highly recommended as a

first-choice supervised algorithm, although it does require more memory than other algorithms. The training output is shown in Fig 2.3.

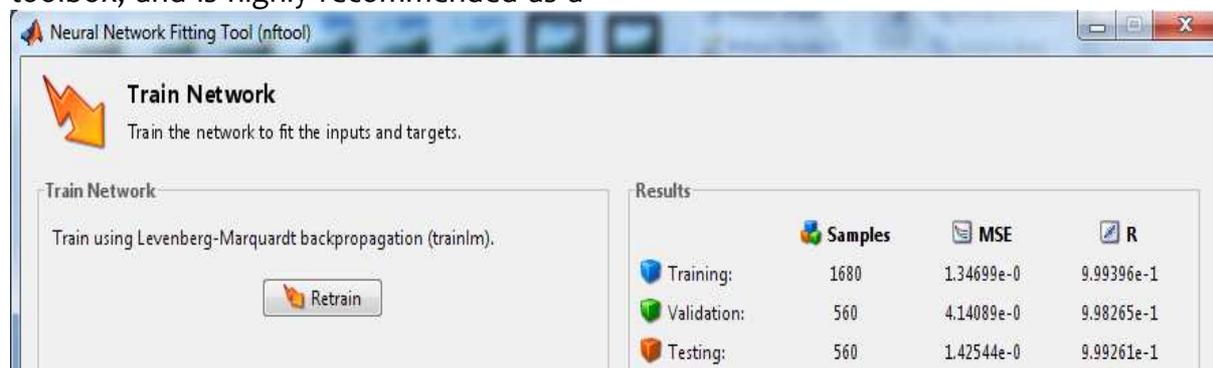


Fig 2.3 training outputs

The network is trained to fit the input and targets. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples. Mean Square Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error. Regression R values measure the correlation between outputs and targets. An R value of 1 means a close relationship. 0 means a random relationship.

3.0 RESULTS AND VALIDATION OF RESULT

3.1 Result

The ANN structure found suitable is 8-20-1 (eight inputs neuron, one hidden layer

with twenty neurons and one output neuron). With the help of trial and error method the smallest MSE for training, validation and testing of 1.34699, 4.14089 and 1.42544 respectively have been found for 20 neurons in the hidden layer shown in Fig 3.1; it shows that the network is learning. The regressions as shown in Fig 3.2 are equal to 0.999396, 0.998265, 0.999261 for training, validation and testing respectively. These network structures are optimum and give minimum mean square error (MSE) and maximum correlation coefficient R^2 values, which means the ANN model prediction is close to the ITU-R model.

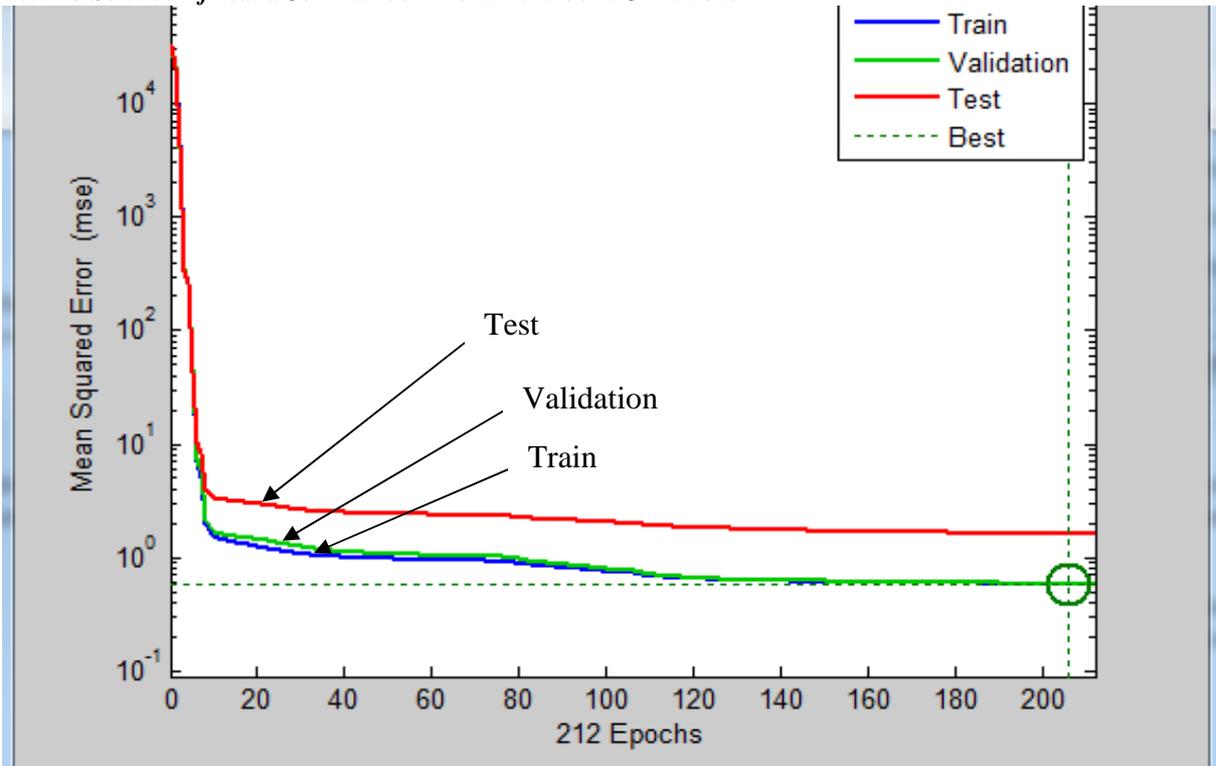


Fig 3.1 Performance Plot

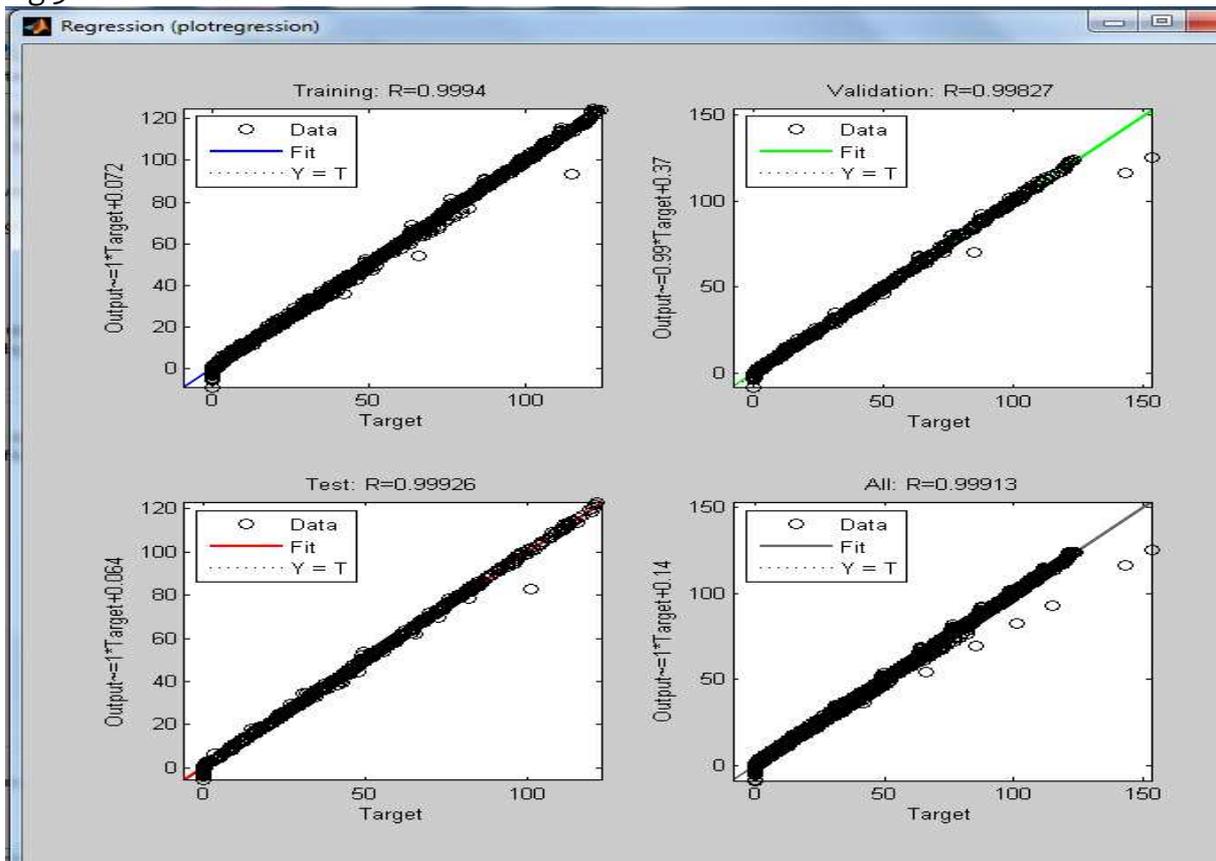


Fig 3.2 Regression Plot

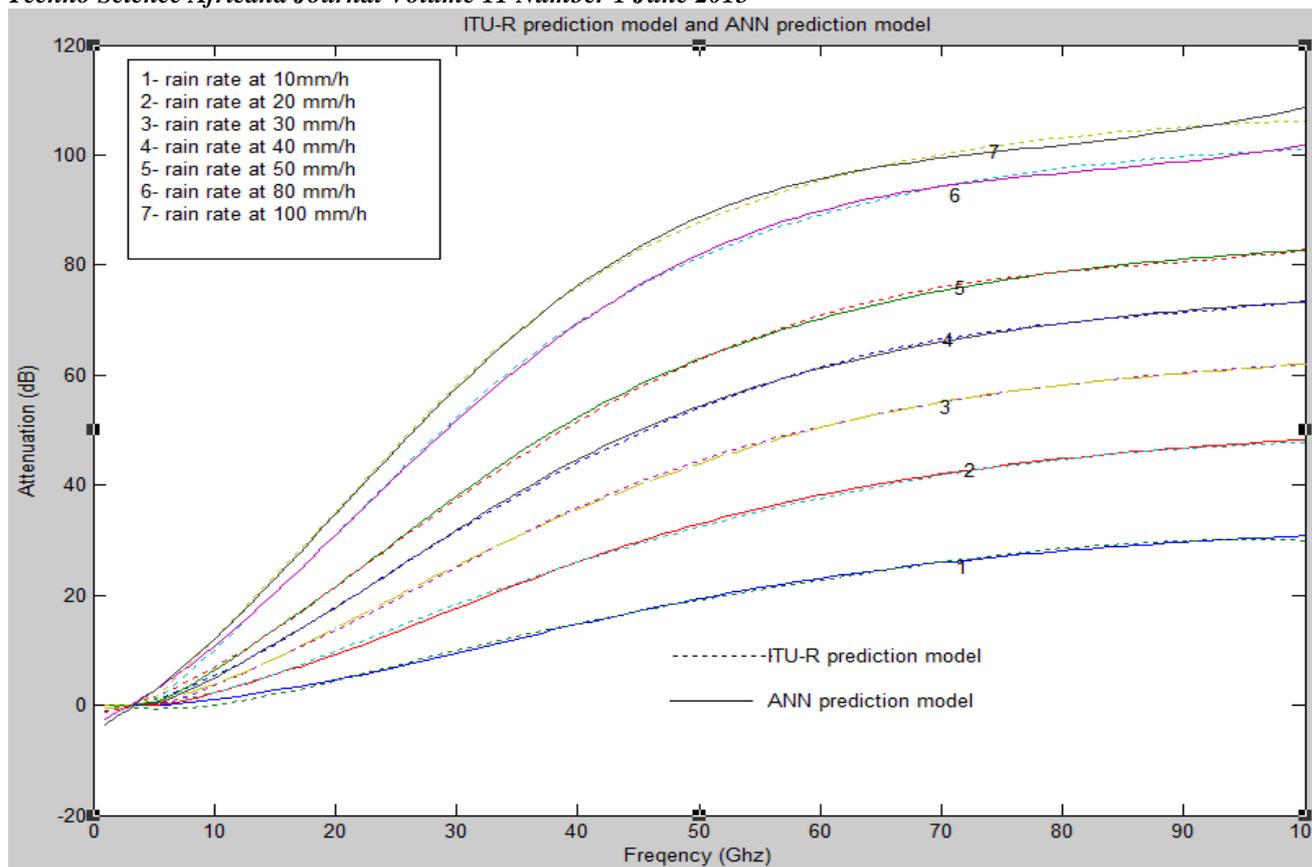


Fig3.3 Rain Attenuation Prediction of ANN Model and ITU-R Model

Fig 3.3 show curves of ITU-R prediction model and ANN prediction model based on ITU-R model. These curves were obtained at different rain rate of 10, 20, 30, 40, 50, 80, 100mm/h respectively. From the Fig 3.3, a very close relationship is observed. This close relationship is marking the closeness of the two models.

3.1 Validation of Result

After the network has been trained satisfactorily, a simulink model as shown in Fig 3.4 of that network has been created which simulates the output for

any given input vector. Here we have tested the network with four input vectors (samples) to get the response of our proposed network for any possible input patterns. Table 3.1 is showing the inputs vectors, network outputs, actual and the error in prediction of rain attenuation. The errors are as follows 0.02175%, 0.11583%, 0.15624% and 0.16284% respectively. Therefore, it is evident that our proposed network is capable of predicting rain attenuation for any given location with negligible difference.

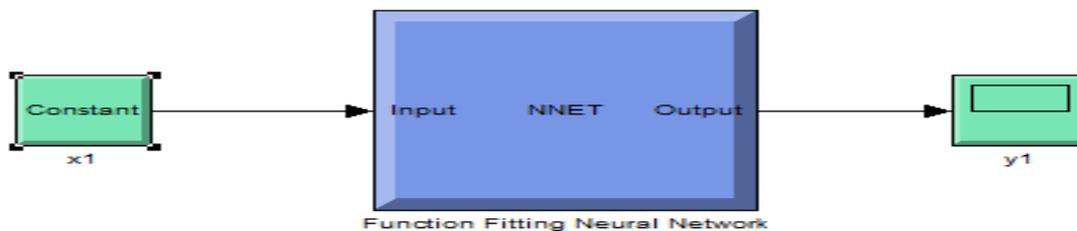


Fig3.4 Simulink Block

Table 3.1 Summary of the Responses of the Simulink Model to Unknown Samples

Samples	1	2	3	4
Latitude	-31.87	46.22	25.77	47.6
Longitude	151.2	6.12	279.8	237.67
Polarization angle(degree)	45	45	45	45
Altitude	0.017	0.814	0	0.504
Elevation angle(degree)	3	33	53	73
Height	3.43	3.18	4.57	2.47
Frequency(GHz)	61	35	100	30
Rainfall rate (mm/h)	50	80	40	40
Calculated attenuation	252.35	52.78	81.22	14.51
Predicted attenuation	252.41	52.84	81.35	14.53
Error (%)	0.02175	0.11583	0.15624	0.16284

4.0 DISCUSSION

The performance function for this network is Mean Squared Error (MSE) which is the average squared difference between outputs and targets. Lower values are better. Zero means no error. During training period of network the numbers of neurons in the hidden layer have been varied to find the smallest MSE values for training, validation and testing. With the help of trial and error method the smallest MSE for training, validation and testing of 0.0134699, 0.04140899 And 0.0142544 respectively has been found for 20 neurons in the hidden layer. It has also been observed that the training performance of the network deteriorates when the numbers of hidden layer neurons are more than 30 or less than 15 and there is no considerable difference when they are varied around 20. These small MSE values are indicating that the network performance is good to predict the rain attenuation accurately.

5.0 CONCLUSION

5.1 Conclusion

Different ANN structures had been investigated to find the optimum ANN

model. The optimum neuron number in each hidden layer was also investigated. In the ANN model architecture the input layer includes 8 inputs: the hidden layers have 20 neurons; and the output layer consist of one neuron. The input layer receives the inputs from the training data, while the hidden layer and the output layer receive it from the interconnections. The neurons use hyperbolic tangent sigmoid transfer functions to produce their outputs. Feed-forward back-propagation algorithm has been used to train the ANN models.

In this model, a supervised learning algorithm was used. The selection of the number of hidden layers and the number of neurons in each layer was an important factor in the development of the neural network. In general, there was no specific rule to estimate the number of neurons for the hidden layers and the optimum topology was obtained by experimental selection. In order to find the best structure for the ANN model many networks with different structures have been developed. These models had one to five hidden layers.

The number of neurons in each hidden layer is also very important in the structure of ANN. The lower the number of neurons in each layer decreases the learning performance. In contrast, a higher number of neurons increase the training time. Therefore, the optimum number of neurons in each layer is essential to build the ANN structure. In the present research, different number of neurons (2-30 neurons) has been tested to find the best ANN structure. By comparing performance of developed ANN models, in this research, the optimum number of neurons in the

hidden layer was 20 for this application. By examining various network structures it was found that the ANN models with more than one hidden layers are not superior to ANN models with a single layer. Therefore, an ANN models chosen to assess the model performance, has one hidden layer. The best value of 0.001 was chosen in this research for the learning rate, training parameter that controls the size of weight and bias changes during learning, to avoid over-fitting and to improve the generalisation, early stopping method was used in the training of the model.

REFERENCES

- Robert, A. N., "Satellite Communication Systems Engineering," 2nd edition. (Prentice Hall, 1993)
- Crane, R. K., "Prediction of attenuation by rain," *IEEE Transactions on Communication*, Vol. 28, No. 9, September 1980.
- Ongwei. Y.C. "Prediction of Slant Path Rain Attenuation Based On Artificial Neural Network" May 2000
- Sahadah, B. A., "Frequency Scaling of Rain Attenuation," MSc Thesis, University Science Malaysia, May 2007
- Louis J. Ippolito (1999) "Propagation Effects Handbook for Satellite Systems Design"
- S. Haykin, *Neural Networks and Learning Machines*, 2009, 3rd Edition, Pearson Education, Inc., New Jersey.
- I.M. Galvan and P. Isasi. Multi-Step Learning Rule for Recurrent Neural Models: An Application to Time Series Forecasting. *Neural Processing Letters*, (13):115-133, 2001.
- Jain, A., Mao, J. & Mohiuddin, K. (March 1996). Artificial Neural Networks: A Tutorial, *IEEE Computer*, Vol. 29, No.3, pp.31-44
- Haykin S., (1999) *Neural Networks a comprehensive foundation* (second ed), prentice- Hall, Englewood Cliff, NJ
- A. Ouammi, D. Zejli, H. Dagdougui, and R. Benchrifa, "Artificial neural network analysis of Moroccan solar potential," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 7, pp. 4876-4889, Sep. 2012.
- H. K. Elminir, Y. A. Azzam, and F. I. Younes, "Prediction of hourly and daily diffuse fraction using neural network, as compared to linear regression models," *Energy*, Vol. 32, No. 8, pp. 1513-1523, 2007.
- ITU-R P.618 RECOMMENDATION validation example for prediction methods