International Journal of Electronics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tetn20

Macrocell path loss prediction using artificial intelligence techniques

Abraham U. Usman [a] , Okpo U. Okereke [b] & Elijah E. Omizegba [b]

[a] Department of Electrical and Electronics Engineering , Federal University of Technology , P.M.B. 65, Minna , Nigeria

[b] Electrical and Electronics Engineering Programme , Abubakar Tafawa Balewa University , Bauchi , Nigeria

Published online: 14 Jun 2013.

International Journal of Electronics, 2013
http://dx.doi.org/10.1080/00207217.2013.792040

Macrocell path loss prediction using artificial intelligence techniques

Abraham U. Usman[a]*, Okpo U. Okereke[b] and Elijah E. Omizegba[b]

[a]*Department of Electrical and Electronics Engineering, Federal University of Technology, P.M.B. 65, Minna, Nigeria;* [b]*Electrical and Electronics Engineering Programme, Abubakar Tafawa Balewa University, Bauchi, Nigeria*

(*Received 31 May 2012; final version received 16 December 2012*)

The prediction of propagation loss is a practical non-linear function approximation problem which linear regression or auto-regression models are limited in their ability to handle. However, some computational Intelligence techniques such as artificial neural networks (ANNs) and adaptive neuro-fuzzy inference systems (ANFISs) have been shown to have great ability to handle non-linear function approximation and prediction problems. In this study, the multiple layer perceptron neural network (MLP-NN), radial basis function neural network (RBF-NN) and an ANFIS network were trained using actual signal strength measurement taken at certain suburban areas of Bauchi metropolis, Nigeria. The trained networks were then used to predict propagation losses at the stated areas under differing conditions. The predictions were compared with the prediction accuracy of the popular Hata model. It was observed that ANFIS model gave a better fit in all cases having higher R^2 values in each case and on average is more robust than MLP and RBF models as it generalises better to a different data.

Keywords: propagation loss; function approximation; signal strength; empirical models; neuro-fuzzy

1. Introduction

Radiowave propagation mechanisms are very complex and diverse owing first to the attenuation that occurs in the media between the receiver and the transmitter and the additional components that often stem from diffraction, scattering, reflection and refraction phenomena. Hence, the prediction of the propagation path loss is of importance in the design and planning of wireless telephone network either mobile or fixed wireless-access systems. Unfortunately, existing empirical models though easier to implement are less sensitive to the environment's physical and geometrical structures and not so accurate while the deterministic models which though are more accurate are computationally inefficient and requires more detailed site-specific information which are often difficult to come by (Abhayawardhana, Wassell, Crosby, Sellars, & Brown, 2005).

Computational intelligence techniques have been suggested as an alternative to linear regression or auto-regression models which are limited in their ability to deal with natural phenomenon whose trend is generally non-linear. According to Alotaibi, Abdennour, and Ali (2008), artificial neural networks (ANNs) and adaptive neuro-fuzzy inference systems (ANFISs) are among such newest techniques. Other authors affirm that the neural network (NN) can be used to predict these losses from measured or theoretically produced data with such parameters of the environment as the mean height and mean dimensions of the

*Corresponding author. Email: abrausman@yahoo.com

buildings and the mean width of the roads (Neskovic, Neskovic, & Paunovic, 2002; Popescu, Kanatas, Angelou, Nafornita, & Constantinou, 2002; Ruben, Lorenzo, & Narcis, 2000).

In this study, ANN and ANFIS techniques were used to predict radio propagation path loss based on signal strength measurement taken in several locations in the suburban areas of Bauchi metropolis in Nigeria. These data were used to train these networks which were later used to predict the loss in some specified locations both with sparse and dense vegetations. The results of the predictions were compared for the different techniques and also compared with that of the popular Hata (1980) model.

2. Artificial neural networks

ANNs are non-linear statistical data modelling tools inspired by biological NNs and have found applications for function approximations, pattern recognition, prediction/forecasting, optimisation, system identification, classification, data processing, robotics and control (Anil, Jianchang, & Mohiuddin, 1996). ANNs can be viewed as weighted directed graphs in which artificial neurons are nodes and directed edges are connections between neuron inputs and outputs. Figure 1 shows a simple neuron model. A neuron is the processing element that takes a number of inputs, weigh them, sums them up together with an additional scalar bias parameter and uses the result as the argument for a single-valued function (svf) called the activation function, $f(u)$.

The neuron is trained to minimise the error based on some optimisation criteria, where $[x_1, x_2, \ldots, x_n]$ is the vector of the input, $[w_{i1} \ldots w_{in}]$ is the weight, b is an additional scalar bias parameter and the output error, e, is given by

$$e = t - y \tag{1}$$

ANNs are composed of several interconnected neurons and the connection patterns can be of *Feed-Forward* network architecture or *Recurrent (or Feedback)* network architecture. However, the different network architectures must be trained with a set of typical input/output data sets using appropriate learning algorithms. The final weight vector of a properly trained NN represents its knowledge about the problem.

Application of ANN to the prediction of field strength and propagation loss in different environments by Bargallo (1998), Ruben et al. (2000), Neskovic et al. (2002), Popescu et al. (2002), Cerri, Cinalli, Michetti, and Russo (2004) and Östlin, Zepernick, and Suzuki (2004) shows a very good performance. The main discrepancy between each

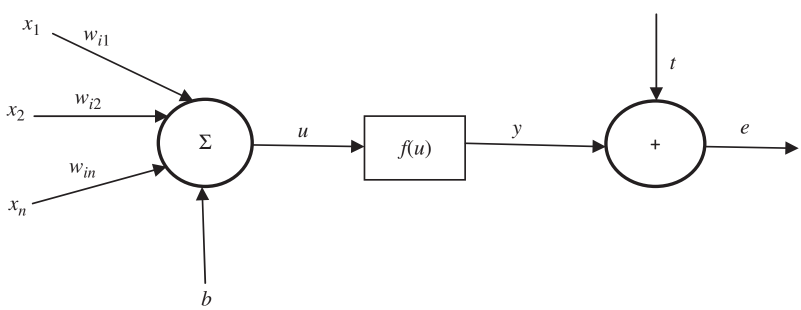

Figure 1. A simple neuron model.

work is given by the classification of the input and the network architecture. Two most common types of NNs used for prediction purposes are the multilayer perceptron neural network (MLP-NN) and the radial basis function neural network (RBF-NN).

2.1. *Multilayer perceptron neural network*

Feed-forward networks are extensions of the neuron model and have a layered structure with each layer consisting of units receiving their input from units in a layer directly below and sending their output to units in a layer directly above themselves. Figure 2 shows the configuration of a multilayer perceptron feed-forward network with one hidden layer and one output layer (Popescu et al., 2002).

The overall output of the MLP network is described by Equation (2) as

$$y = F_{\mathrm{o}}\left(\sum_{j=0}^{M} w_{0j}\left(F_{\mathrm{h}}\left(\sum_{i=0}^{N} w_{ji}x_i\right)\right)\right) \tag{2}$$

where w_{oj} represents the synaptic weight from neuron j in the hidden layer to the single output neuron, x_i represents the ith element of the input vector, F_{h} and F_{o} are the activation function of the neurons from the hidden layer and output layer, respectively, and w_{ji} are the connecting weights between the neurons of the hidden layer and the inputs.

During the training phase, the network adaptively adjusts the free parameters of the system, such as the weight and the bias based on the mean squared error, *mse*, defined by Equation (3). The task of the training process is to minimise the *mse* by optimising the weights using a set of training data.

$$mse = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3}$$

where y_i is the desired output value and $\hat{y}_i$ is the output predicted by the network.

Although multilayered networks with any number of layers may be built, Östlin et al. (2004) have shown that NN with only one hidden layer can approximate any function with finitely many discontinuities to an arbitrary precision, provided the activation functions of the hidden units are non-linear. However, for their particular application, Ruben et al. (2000) have shown that RBF-NNs when trained with supervised learning algorithms can outperform MLP-NNs in terms of generalisation ability.

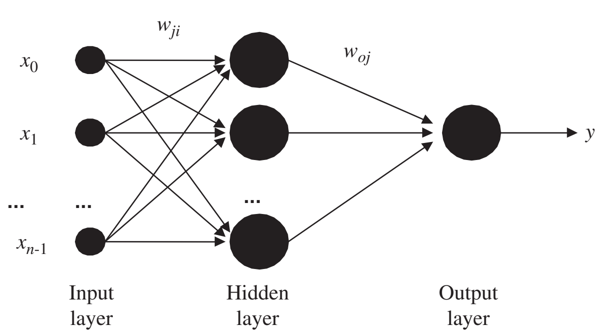

Figure 2. Configuration of the MLP.

2.2. *Radial basis function neural network*

The RBF network has a feed-forward structure with a single hidden layer of K locally tuned nodes as shown in Figure 3. Unlike the MLP, the output of hidden nodes are not calculated using the weighted-sum activation function; rather the output of each hidden-node, φ_k is obtained by the closeness of input X to an M-dimensional parameter vector μ_k associated with the kth hidden node (Popescu et al., 2002).

The response characteristics of the kth hidden node is taken as

$$\varphi_k = P\left(\frac{\|X - \mu_k\|}{\sigma_k^2}\right) \tag{4}$$

where $k = 1, 2, \ldots, K$ and P is a strictly positive radial symmetric function, μ_k is the centre of the function and σ_k is the 'width'. Given an input vector X, the output of the RBF network is the L-dimensional activity vector Y, whose lth component ($l = 1, 2, \ldots, L$) is given by

$$Y_l(X) = \sum_{k=1}^{K} w_{lk}\varphi_k(X) \tag{5}$$

The most popular choice for the function φ is a multivariate Gaussian function with an appropriate mean μ_k and autocovariance σ_k (Popescu et al., 2002). In this study, a Gaussian basis function is assumed for the hidden nodes given as φ_k.

$$\varphi_k = \exp\left(-\frac{\|X - \mu_k\|^2}{2\sigma_k^2}\right) \tag{6}$$

2.3. *Neuro-Fuzzy techniques*

Fusion of ANN and fuzzy inference systems (FIS) has attracted the growing interest of researchers in various fields of science and engineering. An analysis reveals that the

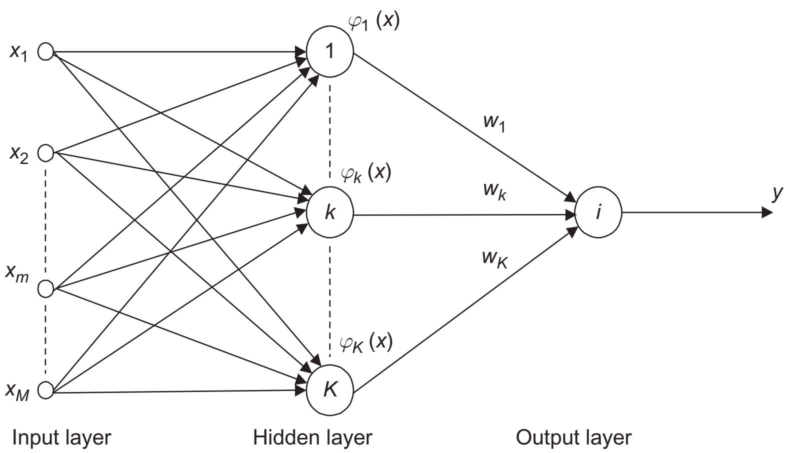

Figure 3. RBF-NN architecture.

drawbacks pertaining to ANN and FIS approaches seem complementary and therefore it is natural to consider building an integrated system combining the concepts. This so-called neuro-fuzzy (NF) networks have been used to solve complex problems.

2.3.1. *Adaptive neuro-fuzzy inference systems*

ANFIS was first proposed by Jang (1993) to combine the learning ability of NNs with the ability of fuzzy systems to interpret imprecise information. ANFIS model is one of the implementations of a first-order Takagi–Sugeno–Kang (TSK) fuzzy inference system. An example of such fuzzy inference system with two inputs, x and y and one output which is a function of the inputs is shown in Figure 4. For TSK inference system, the rule is constructed as

If x is A_i and y is B_i, then $f_i = p_i x + q_i y + r_i$

where A_i and B_i are the linguistic labels in the input spaces x and y, respectively, and f_i is a local function which depends on x and y.

Layer 1 is the fuzzification layer which generates membership grades for each linguistic label for any input value; these values are defined by membership functions (MFs). The common MFs are the bell and triangular functions depicted in Figure 5. The bell function used in this study is described by the three premise parameters, a, b and c in Equation (7).

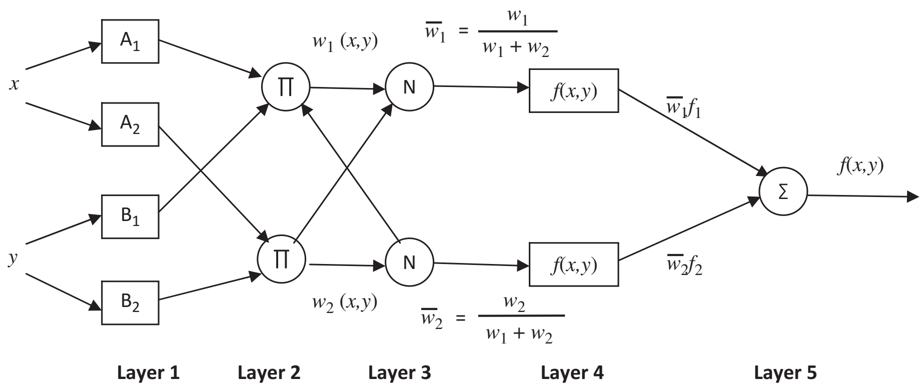

Figure 4. The structure of adaptive neuro-fuzzy inference system.

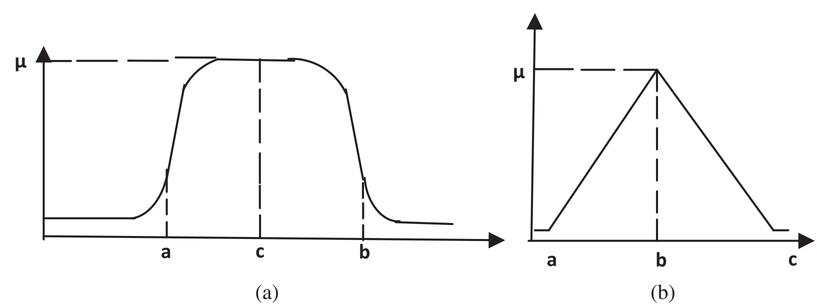

Figure 5. (a) Bell and (b) triangular MFs.

$$\mu_{Ai}(x) = \frac{1}{1 + \left[\frac{x-c_i}{a_i}\right]^{2b_i}} \tag{7}$$

Layer 2 is the application of the fuzzy operator and the output of every node in this layer is the product of all the incoming signals into the node as given by Equation (8).

$$w_i = \mu_{Ai}(x_i) X \mu_{Bi}(y_i) \tag{8}$$

Layer 3 produces an output which is the so-called normalised firing strength of each rule according to Equation (9):

$$\overline{w_i} = \frac{w_i}{\sum_{j=1}^{2} w_j} \tag{9}$$

Layer 4 is a layer of adaptive nodes each with a node function:

$$\overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \tag{10}$$

where, p_i, q_i and r_i are called consequent parameters.

Layer 5 is the defuzzification layer that generates a crisp output given by Equation (11).

$$f(x, y) = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{11}$$

where $\overline{w_i} f_i$ is the output of node i in layer 4 denoting the consequent part of rule i.

ANFIS uses a hybrid learning algorithm which is a combination of gradient descent and the least-squares approximation method in order to train the network. Gradient descent back-propagation algorithm is used for training the premise parameters and least-squares approximation is used for training the consequent parameters.

Several applications of ANFIS are reported. However, the first open literature that used it as wireless signal predictor is Alotaibi et al. (2008) which was based on a private mobile network- terrestrial trunked radio (TETRA) network. The result of their comparative work shows that the ANFIS prediction model outperforms some other empirically based prediction models and is also marginally better than RBF-NN predictor. Also, ANFIS was used by Turkan, Berna, and Apaydin (2010) to predict path loss based on data obtained in the 900 MHz band in Harbiye region of Istanbul, Turkey. ANFIS prediction error was shown to be less than that obtained using Bertoni–Walfish model.

3. Data collection

The signal strength along a straight path was measured at intervals of 200 m of varying radial distances from the test BSs (Figure 6). The longitude and latitude of each point A_i, (i = 1, 2, …, n), as well as the elevation above sea level were taken using eTrex Garmin GPS receiver while a NOKIA3310 handset equipped with net-monitor software was used to measure the received signal strength level. Four different BSs in Bauchi belonging to

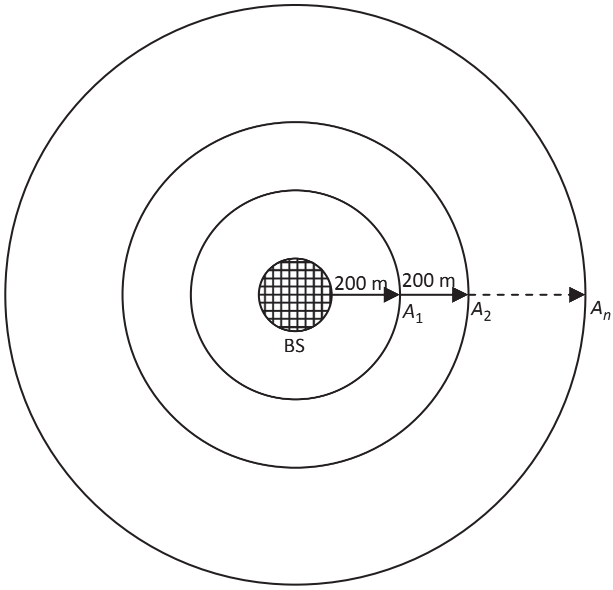

Figure 6. Measurement set-up.

the same GSM service provider were selected and labelled as BS1, BS2, BS3 and BS4 for investigation.

Measurements were taken several times at each point between June and May 2011, for each site on the Broadcast Control Channel (BCCH) control channel to eliminate the effect of frequency hopping and downlink power control algorithms. The coverage distances were 2 km, 2.4 km, 3.8 km and 4.4 km from BS1, BS2, BS3 and BS4, respectively, beginning at 200 m away from the base station (BS).

Further measurement was conducted in the Federal Polytechnic, Bauchi, in an environment well characterised with dense vegetation as shown in Figure 7. The typical neem trees in the area have an average height of 6 m and the trees are nearly equally spaced with a separation of 3 m. The leaves of neem trees form a dense canopy and the environment and weather condition were wet. The measurement was taken up to 2.4 km from another BS (BS5) within the mainlobe of the sector.

Figure 7. Test environment from BS5.

3.1. *Model input selection*

In developing a satisfactory prediction model, the selection of the input variables is crucial since the input is useful only if it is closely associated with the measured path loss values. Analysis of result in Usman, Okereke, and Omizegba (2011) shows that the four measured path loss data sets exhibit significant positive linear correlation with distance (ranging from 0.63 to 0.94).

Also, the targeted area in this study is considered as a flat area because its terrain undulation standard deviation is less than 17 m (Hernando & Pérez-Fontán, 1999), and the percentages of area covered by buildings for the suburban BSs are fixed for each BS. Thus, their effects on the signal strength are negligible, and as such, only one input to the network is used which is the distance between the mobile station (MS) and BS, and the single output is the path loss at each location. However, the algorithm was also tested on data from an area with dense vegetation.

3.2. *MLP networks design procedure*

Our first approach to fit the N data points of distance versus path loss pairs for each location is using the MLP-NN. A 1-3-1 feed-forward network is created having a one-hidden layer with three Tansig neurons in the hidden layer and a linear neuron in the output layer to approximate the path loss using the *newff* function of MATLAB (Neural Network Toolbox™). The learning phase of the network proceeds by adaptively adjusting the free parameters of the system (weights and biases) based on the mean square error (*mse*) between the predicted and measured path loss. Normally, the NN starts with random initial weights, and as such the results will differ every time it is run. Therefore, we set the random seed using rand ('seed', 62734347) to avoid this randomness.

The MLP network was trained using the Levenberg–Marquardt algorithm (LMA) which has been shown to conver considerably faster for function approximation problems than the Back-Propagation Algorithm (BPA) with adaptive learning rates and momentum. The LMA update rule is given by

$$\Delta\mathbf{W} = (\mathbf{J}^{\mathrm{T}}\mathbf{J} + \mu\mathrm{I})^{-1}\mathbf{J}^{\mathrm{T}}\boldsymbol{e} \tag{12}$$

where $\boldsymbol{e}$ is an error vector, μ is a scalar parameter, $\mathbf{W}$ is a matrix of network weights and $\mathbf{J}$ is the Jacobian matrix of partial derivatives of the error components with respect to the weights. For large values of μ, the above expression approximates the gradient descent method. On the other hand, for small values of μ, Equation (12) becomes the Gauss–Newton Method which is faster and more accurate near a minimum of the error surface. The goal is then to adjust μ, to approach the Gauss–Newton method as quickly as possible.

3.2.1. *Improving generalisation*

One of the problems that occur during NN training is called overfitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorised the training examples, but it has not learned to generalise to new situations. Reducing the size of a network can prevent overfitting, unfortunately, it is difficult to know beforehand how large a network should be for a specific application. Two other methods for improving generalisation that are implemented in Neural Network Toolbox™ software are regularisation and early stopping. These apply to those situations in which it is intended to make the most of a limited supply of data as our case in this research is and are thus utilised.

3.3. *RBF networks design procedure*

Another approach we used to find a function which fits the *N* data points of distance and Path loss pairs is with a radial basis network which is a network with two layers – a hidden layer of radial basis neurons and an output layer of linear neurons. The *newrb* function of MATLAB (Neural Network Toolbox™) is used to create a radial basis network that approximates a function defined by the set of data points. In addition to the input training set and targets, *newrb* takes two arguments, the sum-squared error goal and the spread constant, and returns the desired network.

The RBF training parameter spread controls the smoothness and generalisation of the approximation. The larger the spread is the smoother the function approximation will be. Too large a spread means a lot of neurons will be required to fit a fast changing function. Too small a spread means many neurons will be required to fit a smooth function, and the network may not generalise well. The RBF was trained with different values of spread constant and the optimal value of 0.2873 was obtained.

3.4. *ANFIS design procedure*

Apart from specifying the number of network inputs, designing an ANFIS network involves the need to specify the number of fuzzy membership function (MF) per each input, the type of fuzzy MF, and the number of epochs (Alotaibi et al., 2008). In this study, the type of the MF is chosen to be the bell-shaped function, and the number of fuzzy MF per each input and the number of epochs are 3 and 100, respectively.

3.5. *Choice of optimal parameters*

The various network parameter values were optimised using trial and error until the minimal *mse* is achieved and presented in Table 1.

3.6. *Data division*

The goal of the prediction is not only to produce small errors for the set of training examples, but to be able to perform well with examples, i.e. receiver locations, not used in the training process. A model that performs well in previously unseen situations is said to have good 'generalisation properties'. It is clear that the latter is of utmost importance in practical propagation prediction situations, where the intention is to use the propagation

Table 1. The value of MLP, RBF and ANFIS prediction model parameters.

Parameter	Value
No. of MLP hidden neurons	3
No. of epochs	100
No. of RBF neurons	Default length of data
RBF's spread	0.2873
Error goal	0.01
No. of ANFIS MF	3

loss prediction model to determine the coverage area of potential transmitter locations for which none or limited measured data is available (Alotaibi et al., 2008; Bargallo, 1998; Popescu et al., 2002).

Thus, after specifying the network structure and parameters, we created two data sets: a training set and a testing test. The training set was used for the NN's training process, and to generate the fuzzy rules as well as tune the MFs. The trained networks were then tested with the testing samples to give us a sense of how well the network will do when applied to data different from those used during training. Also, for better generalisation the inputs and targets of both the training and testing data were normalised so that they have zero mean and unity variance. The output was reversed before the graphical plots.

3.7. *Statistical basis of performance evaluation*

The performances of the models developed in this study were evaluated based on; absolute mean error (μ), standard deviation (σ) and root mean square error (RMSE) as described in Usman et al. (2011) and R^2 values.

R-square (R^2) measures how successful the fit is in explaining the variation of the data- the goodness of fit. It is also called the square of the multiple correlation coefficients and the coefficient of multiple determinations and given by:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y}_i)^2} \tag{13}$$

where $y_i$is the measured path loss, $\hat{y}_i$ is the predicted path loss and $\bar{y}_i$ is the mean of the measured path loss.

R^2 can take on any value between 0 and 1, but can be negative for models without a constant, which indicates that the model is not appropriate for the data. A value closer to 1 indicates that a greater proportion of variance is accounted for by the model.

4. Results and discussion

In the first comparative approach, the overall data were divided into 60% training set and 40% testing set and the performance of the models were compared using the afore mentioned statistical measures – μ, σ, RMSE and R^2 – and the results are presented in Tables 2–5, respectively. From these tables, it is found that performance of the proposed

Table 2. μ Comparison between AI models and the Hata empirical model.

	Model			
Base station	MLP	RBF	ANFIS	Hata
BS1	3.235	3.548	2.671	9.419
BS2	3.166	3.073	2.997	6.341
BS3	4.189	3.713	3.470	9.992
BS4	7.701	4.740	4.230	11.011
Average	4.572	3.769	3.342	9.191

Table 3. σ Comparison between AI models and the Hata empirical model.

	Model			
Base station	MLP	RBF	ANFIS	Hata
BS1	2.798	2.835	2.189	5.010
BS2	2.314	2.256	2.284	5.582
BS3	4.098	3.452	3.596	5.634
BS4	6.071	3.286	3.329	7.547
Average	3.820	2.957	2.849	5.943

Table 4. RMSE comparison between AI models and the Hata empirical model.

	Model			
Base station	MLP	RBF	ANFIS	Hata
BS1	4.264	4.529	3.444	10.663
BS2	3.915	3.807	3.762	8.442
BS3	5.828	5.043	4.967	11.458
BS4	9.789	5.759	5.373	13.341
Average	5.949	4.785	4.387	10.976

Table 5. R^2 comparison between AI models and the Hata empirical model.

	Model			
Base station	MLP	RBF	ANFIS	Hata
BS1	0.870	0.853	0.915	0.221
BS2	0.858	0.866	0.869	0.419
BS3	0.391	0.544	0.557	–1.661
BS4	0.487	0.822	0.845	0.040
Average	0.652	0.771	0.797	–0.245

ANFIS predictor model on average is the best having the lowest μ, σ, RMSE and the highest R^2 when compared to all other models. The average RMSE for the ANFIS model is 4.387 dB, whereas for the MLP, RBF and Hata models, it is 5.949, 4.785 and 10.976 dB, respectively. The RMSE values of AI predictors for all BSs are almost half of the corresponding values of the Hata empirical model. Also, we observed that in all the statistical measures, the performance of the ANFIS model is followed closely with the RBF predictions.

4.1. *Training with one BS and testing with another*

A second comparison approach taken is based on training the prediction models with data from one base station and testing with data from another BS. This approach is employed to test the robustness or the generalisation properties of the models as

Table 6. Summary of training with one BS and testing with another.

Model	Perf. indices	Training/testing BSs				Average
		BS2/BS4	BS4/BS1	BS1/BS2	BS1/BS3	
MLP	μ (dB)	7.354	5.558	4.640	4.951	5.625
	σ (dB)	4.468	4.192	2.992	2.998	3.662
	RMSE (dB)	8.543	6.855	5.484	5.709	6.648
	R^2	0.470	0.672	0.718	0.204	0.516
RBF	μ (dB)	7.426	4.272	3.461	4.708	4.967
	σ (dB)	4.454	4.622	2.998	1.649	3.431
	RMSE (dB)	8.599	6.151	4.535	4.961	6.061
	R^2	0.463	0.736	0.807	0.399	0.601
ANFIS	μ (dB)	7.493	4.303	3.551	4.226	4.893
	σ (dB)	4.106	2.285	2.287	1.750	2.607
	RMSE (dB)	8.492	4.828	4.195	4.540	5.514
	R^2	0.476	0.837	0.835	0.496	0.661
Hata	μ (dB)	9.027	8.450	5.457	9.301	8.059
	σ (dB)	6.441	5.129	4.887	6.077	5.634
	RMSE (dB)	10.991	9.773	7.251	10.943	9.739
	R^2	0.123	0.333	0.506	−1.925	−0.241

summarised in Table 6. Similarly, Figures 8–11 show the graphical presentation of the various model predictions in comparison with the measured data from the test BSs. Hata model predictions presented here are based on the network configurations of the test BSs for appropriate comparison. Again, we observed that ANFIS model gave a better fit in all cases having higher R^2 values in each case and on average is more robust as it generalises better to a different data. Hata model on average gave R^2, RMSE, σ and μ values of −0.241, 9.739, 5.634 and 8.059 dB, respectively, and thus is the worst model for this environment.

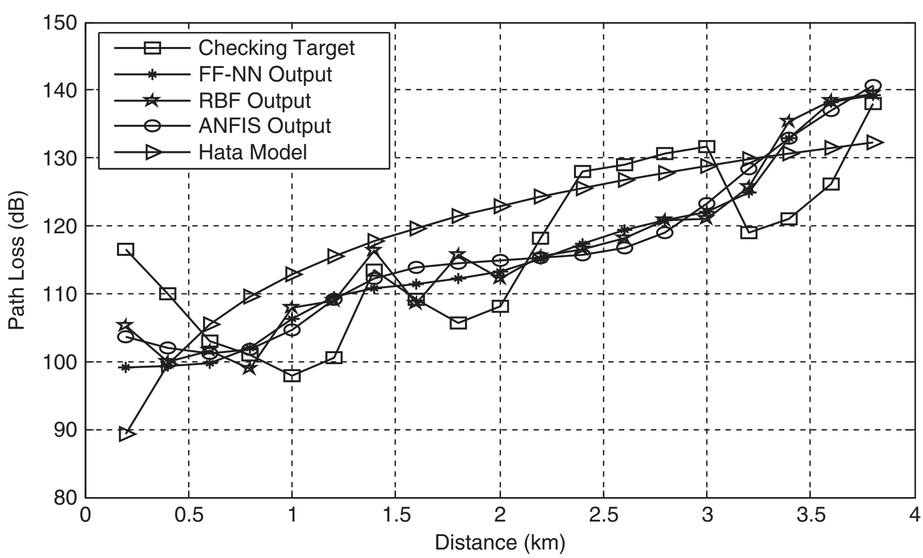

Figure 8. Model predictions for training with BS2 and testing with BS4.

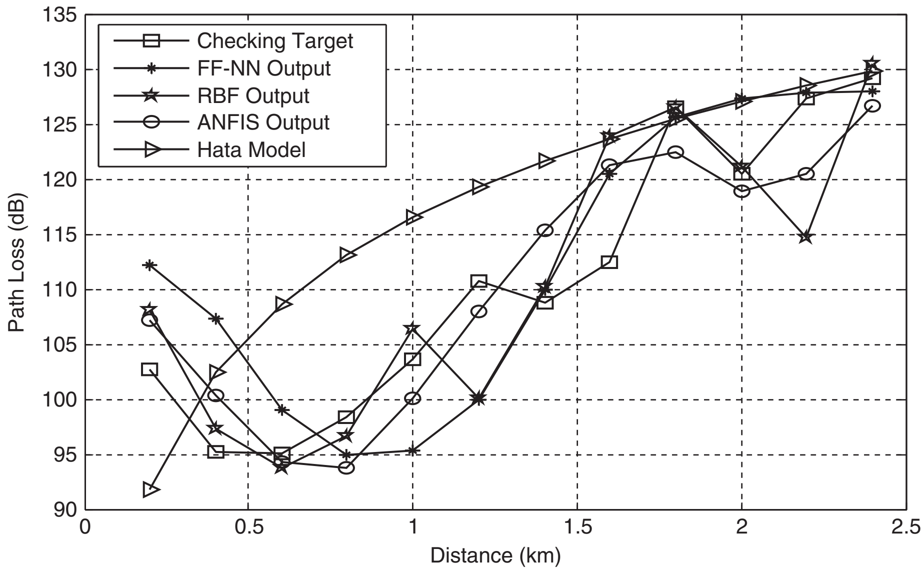

Figure 9. Model predictions for training with BS4 and testing with BS1.

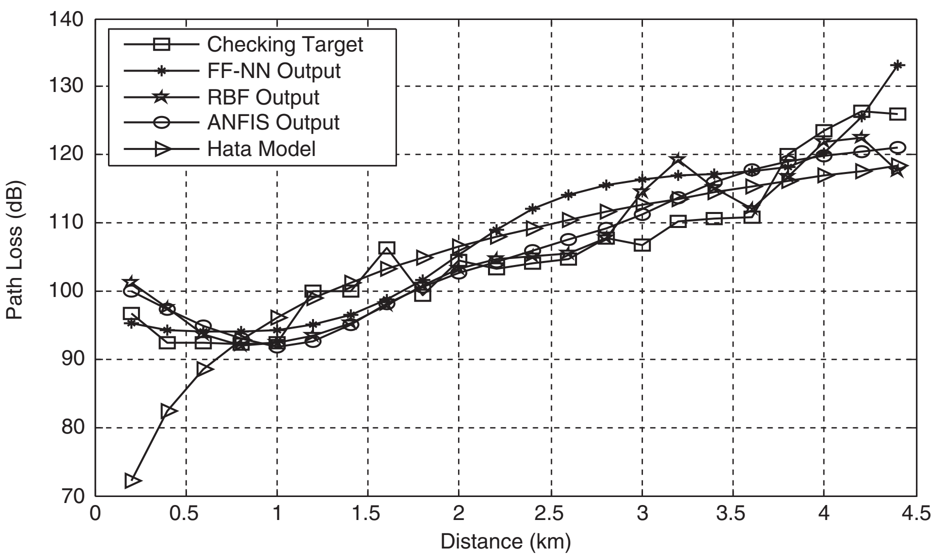

Figure 10. Model predictions for training with BS1 and testing with BS2.

4.2. *Testing on data from tree-dominated area*

The result of a third approach which tested the algorithm on data from a tree-dominated area is presented in Table 7 and plotted in Figures 12 and 13. ANFIS on average gave the best prediction with the highest R^2 value of 0.732 against the Hata model with the lowest R^2 value of 0.135.

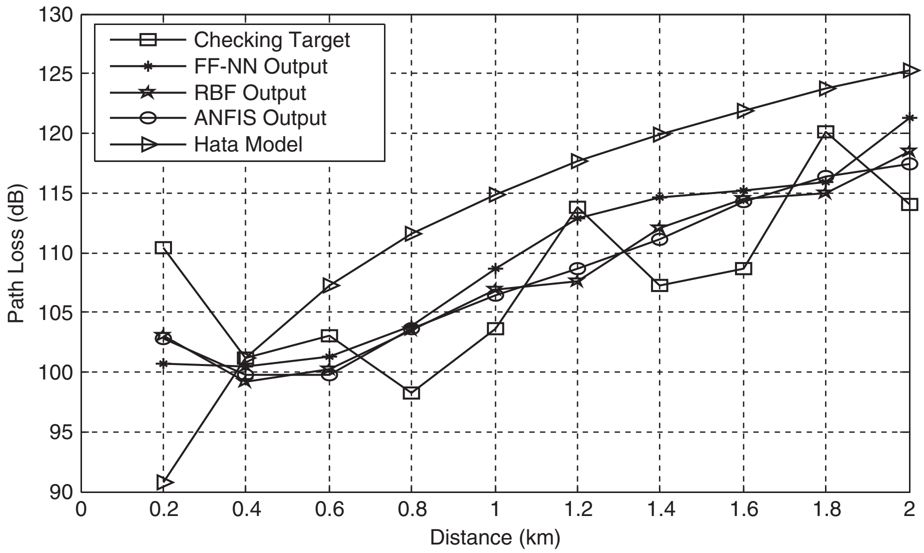

Figure 11. Model predictions for training with BS1 and testing with BS3.

Table 7. Summary of training with one BS and testing on tree-dominated area.

Model	Perf. indices	Training/testing BSs BS4/BS5	BS1/BS5	Average
MLP	μ (dB)	5.094	8.187	6.641
	σ (dB)	4.340	5.967	5.153
	RMSE (dB)	6.574	9.983	8.279
	R^2	0.791	0.517	0.654
RBF	μ (dB)	7.617	6.732	7.175
	σ (dB)	4.026	4.476	4.251
	RMSE (dB)	8.537	7.980	8.258
	R^2	0.647	0.692	0.669
ANFIS	μ (dB)	6.004	5.687	5.845
	σ (dB)	5.308	4.239	4.773
	RMSE (dB)	7.866	6.986	7.426
	R^2	0.700	0.764	0.732
Hata	μ (dB)	10.060	10.060	10.060
	σ (dB)	9.182	9.182	9.182
	RMSE (dB)	13.360	13.360	13.360
	R^2	0.135	0.135	0.135

5. Conclusion

Recent developments using Artificial Intelligent techniques for prediction purposes have been explored to model radio propagation path loss based on data taken from an existing GSM network in the surburban areas of Bauchi, Nigeria. The use of MLP, RBF and ANFIS networks for the prediction of the propagation loss was shown to provide a significant improvement over conventional empirical models like Hata model even in an environment with dense vegetation.

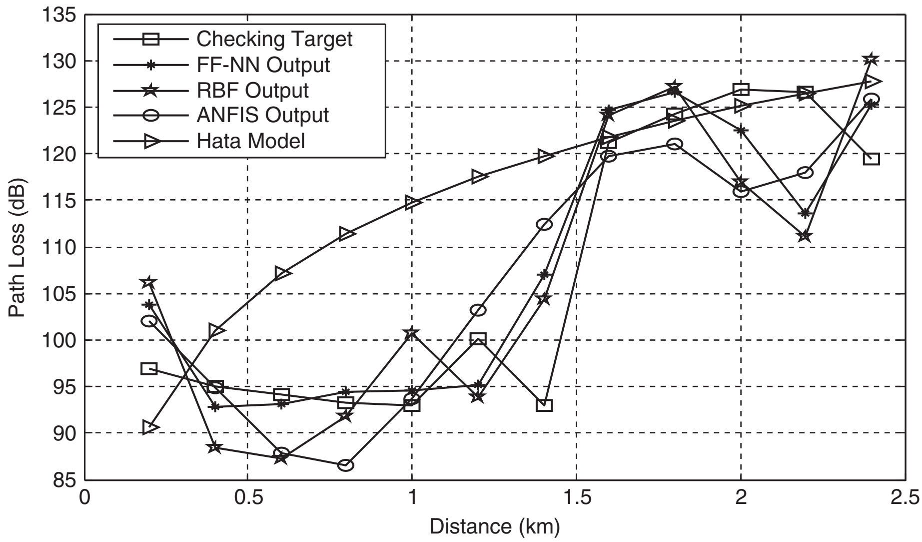

Figure 12. Model predictions for training with BS4 and testing with BS5.

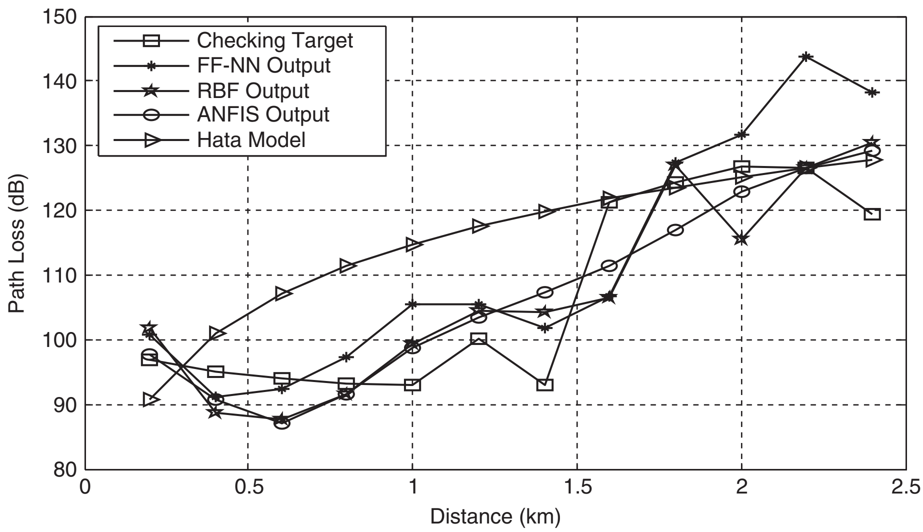

Figure 13. Model predictions for training with BS1 and testing with BS5.

It was also discovered that ANFIS model is more robust when used on data different from the training set. ANFIS was shown on average to provide standard error improvement on the order of 1 dB over other AI models and 4 dB over Hata model. Hata model on average gave R^2, RMSE, σ and μ values of −0.241, 9.739, 5.634 and 8.059 dB, respectively, and performs worse than even the worst AI model, MLP-NN, which gave R^2, RMSE, σ and μ values of 0.516, 6.648, 3.662 and 5.625 dB, respectively. ANFIS on average gave the best prediction with the highest R^2 value of 0.732 against the Hata model with the lowest R^2 value of 0.135 even when tested on data from dense vegetation.

References

Abhayawardhana, V. S., Wassell, I. J., Crosby, D., Sellars, M. P., & Brown, M. G. (2005). Comparison of empirical propagation path loss models for fixed wireless access systems. In *Proceedings of IEEE Conference on Vehicular Technology*, *1*, 73–77. doi:10.1109/VETECS.2005.1543252

Alotaibi, F. D., Abdennour, A., & Ali, A. A. (2008). A robust prediction model using ANFIS based on recent TETRA outdoor RF measurements conducted in Riyadh city – Saudi Arabia. *AEU – International Journal of Electronics and Communications*, *62*(9), 674–682. doi:10.1016/j.aeue.2007.09.001

Anil, K. J., Jianchang, M., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *IEEE Computer Society*, *9*(3), 31–44. Retrieved from http://web.iitd.ac.in/~sumeet/Jain.pdf

Bargallo, J. M. (1998). A comparison of neural network models for prediction of RF propagation loss. *Proceedings of the IEEE 48th Vehicular Technology Conference*, *48*(1), 445–449. doi:10.1109/VETEC.1998.686613

Cerri, G., Cinalli, M., Michetti, F., & Russo, P. (2004). Feed forward neural networks for path loss prediction in urban environment. *IEEE Transactions on Antennas and Propagation*, *52*(11), 3137–3139. doi:10.1109/TAP.2004.835252

Hata, M. (1980). Empirical formula for propagation loss in land mobile radio services. *IEEE Transactions on Vehicular Technology*, *29*(3), 317–325. doi:10.1109/T-VT.1980.23859

Hernando, J. M., & Pérez-Fontán, E. (1999). *Introduction to mobile communications engineering*. Norwood, MA: Artech House.

Neskovic, A., Neskovic, N., & Paunovic, D. (2002). Macrocell electric field strength prediction model based upon artificial neural networks. *IEEE Journal on Selected Areas in Communications*, *20*(6), 1170–1177. doi:10.1109/JSAC.2002.801217

Östlin, E., Zepernick, H.-J., & Suzuki, H. (2004). Macrocell radio wave propagation prediction using an artificial neural network. *IEEE Semiannual Vehicular Technology Conference*, *1*, 57–61. doi:10.1109/VETECF.2004.1399921

Popescu, I., Kanatas, A., Angelou, E., Nafornita, I., & Constantinou, P. (2002). Applications of generalized RBF-NN for path loss prediction. *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* (PIMRC 2002), *1*, 484–488. doi:10.1109/PIMRC.2002.1046748

Ruben, F., Lorenzo, R., & Narcis, C. (2000). Application of RBF neural networks to the prediction of propagation loss over irregular terrain. *IEEE Semiannual Vehicular Technology Conference*, *2*, 878–884. doi:10.1109/VETECF.2000.887127

Turkan, E. D., Berna, Y. H., & Apaydin, A. (2010). Fuzzy adaptive neural network approach to path loss prediction in urban areas at GSM-900 band. *Turkish Journal of Electrical Engineering and Computer Sciences*, *18*(6), 1077–1094. doi:10.3906/elk-0904-18

Usman, A. U., Okereke, O. U., & Omizegba, E. E. (2011). Comparative analysis of path loss model predictions with measurement taken for GSM 900/1800MHz networks in Bauchi suburbs of Nigeria. *The IUP Journal of Telecommunications, III*(4), 7–17.