Effect of Architectural Composition of MLP ANN in Neural Network Learning for Signal Power Loss Prediction

Virginia C. Ebhota and Viranjay M. Srivastava
Department of Electronic Engineering, Howard College, University of KwaZulu-Natal, Durban - 4041, South Africa
Email: ebhota@ieee.org, viranjay@ieee.org

Abstract**—**This work analyzes the architectural complexity of a Multi-Layer Perceptron (MLP) Artificial Neural Network (ANN) model suitable for modeling and predicting signal power loss in micro-cellular environments. The MLP neural network model with *one*, *two*, and *three* hidden layers respectively were trained using measurement datasets used as the target values collected from a micro-cell environment that is suitable to describe different propagation paths and conditions. The neural network training has been performed by applying different training techniques to ensure a well-trained network for good generalization and avoid over-fitting during network training. Bayesian regularization algorithm (that updates weights and biases during network training) following the Levenberg-Marquardt optimization training algorithm was used as the training algorithm. A comparative analysis of training results from *one*, *two*, and *three* hidden layers MLP neural networks show the best prediction result of the signal power loss using a neural network with one hidden layer. A complex architectural composition of the MLP neural network involved very high training time and higher prediction errors.

Index Terms**—**Architecture of MLP ANN, Micro-cellular, Neuron variation, Signal power loss, Bayesian Regularization, ANN.

I. Introduction

A basic contributing factor in planning an efficient and workable wireless radio communication networks as well as improving existing communication networks depends on the ability to precisely predict the coverage and strength of the radio signal between transmitters and receivers in the communication networks [1], [2]. The mathematical algorithms required for prediction of the signal coverage and strength is known as propagation models [3]. Propagation path loss, also known as signal power loss as electromagnetic signals transmitted from the transmitters to the receivers, is the unwanted loss in transmitted power density enroute the transmitter to the receiver in cellular radio channels. The loss arises as a result of many factors, such as environmental blockades and multi-path propagation effects [4], [5]. This causes the received signal power to fluctuate and attenuate around the User Equipment (UE).

Over the years, different traditional models such as the deterministic model, the empirical model, and the semi-empirical models have been applied to solve the effect of signal power loss in diverse environments for proper field strength prediction [6]. These traditional models have been developed to address propagation behavior in specific areas making it not fitting or too cumbersome for broader application. Artificial Neural Network (ANN) modes have been used considerably as a better alternative model for field strength prediction with wider adaptability in different propagation environments [7], [8]. A reliable and accurate signal strength prediction model, such as ANNs, ensures coverage area and power optimization and gets rid of interference problems of the radio transmitters. This helps network engineers and planners to properly optimizes the coverage area and ensure adequate use of transmitting powers

Artificial neural networks are computational models that are based on biological neural network structures. Its functions that the crude electronic models simply learn from in solving different problems such as predictions and finding trends in large quantities of data. Varieties of ANN models have been developed over the years for signal processing, system control, system optimization, pattern recognition, etc. [8]-[10]. These include the Multi-Layer Perceptron (MLP), the Radial Basis Function (RBF), the recurrent networks, the wavelet networks, etc. [11], [12]. The MLP-ANN is a common neural network structure because of its unambiguous structure and simplicity in usage. It consists of an input layer, one or more hidden layers, and an output layer. They belong to the neural network structure known as feedforward ANNs which are basically capable of function approximation, including integral and continuous functions [12]. Multi-layer perceptron ANNs have also been applied for microwave modeling & designs and have shown a smooth & measurable function approximation between the input/output vectors [13], [14].

However, a major drawback MLP-ANN exhibits is difficulty in the determination of an adequate number of layers and neurons the hidden layers required for efficient network training [15]. Too many neurons in the hidden layers of MLP-ANN result in poor network generalization during network training, this leads to non-convergence of the network [16]. The ability of the neuron weights to adequately converge at a point of satisfactory operation (during network training) is known as good network generalization. The MLP-ANN architecture is expected to be in synchronization with the underlying physical complexity of the problem to impact adequately on the training procedure [15], [16]. Also,

Manuscript received July 24, 2020; revised December 16, 2020.
Corresponding author email: ebhota@ieee.org
doi:10.12720/jcm.16.1.20-29

aside from ensuring the appropriate selection of the hidden layer neurons during network training, poor ANNs generalization can be reduced using the Bayesian Regularization (BR) training approach and application of early stopping method during network training.

The activation function for the output neurons of the MLP-ANN may be logistic functions such as sigmoid or a simple linear function that computes the weighted sum of the stimuli. Each of the neurons is a processing element with the switching activation function, with many of the neurons combining to produce the overall actual result. It is (these neurons) switching state combination which respond to various external stimuli values that permits the ANN to characterize a nonlinear mapping. Therefore, the architectural composition and the selection of an appropriate number of hidden layer neurons of the MLP-ANN is essential for its performance. The authors in ref. [17], [18] proved the universal approximation theorem for MLP-ANN, however, there is no illustration about the number of neurons required, specified layers of MLP-ANN required to approximate a given function. Thus, failure to design accurate model architecture can point to an inadequate number of the hidden neurons, inadequate training/learning, presence of stochastic relation instead of deterministic between the input and the out layers [18]. The input neurons of MLP-ANN basically relay the external stimuli to hidden layer neurons, making the input activation function work as a relay function.

The selection of appropriate neuron numbers is one of the critical factors that determine the ANN performance during the network training. This has been an open problem as there is no basic rule to guide in the neuron number selection. Most time, it is mainly based on trial and error [19], [20].

In this research work, authors have adopted two different methods in determining an appropriate MLP-ANN size for effective training/learning of electromagnetic signal power loss using a dataset collected from a micro-cell environment. An adaptive process that adds/deletes neurons (during network training and application of constructive training algorithm) such as the BR algorithm to match the complexity of the neural network model with the problem complexity has been adopted. Fig. 1 represents the flowchart of procedure of MLP-ANN signal power loss training and prediction. The MLP-ANN training at the application of different numbers of hidden layers and neurons has been performed using the BR algorithm, which is a back-propagation training algorithm that minimizes error function in accordance with Levenberg-Marquardt (LM) algorithm [21]. The BR algorithm update weight and bias in agreement with LM optimization [22], [23] and adjusts the linear combination to guarantee an improved generalized network.

The early stopping technique using the ratio of 70%:15%:15% for training, testing, and validation during network training was adopted to avoid network over-fitting during training. The 1st order statistical indices, the standard deviation, and the coefficient of correlation were used for prediction error analysis. The neural network training was done with neuron variation in the hidden layers from 5 to 100 neurons to ensure that a considerable number of neurons have been employed for network training for better assessment. The result shows that MLP neural network training with *one* hidden layer with neuron variation from 5 to 100 neurons gave the best prediction result with training using 50 neurons in the hidden layer. Training with 50 neurons gave the highest coefficient of correlation of 0.96870 and a standard deviation of 1.39040 in comparison to training with 5 and 100 neurons, which gave a coefficient of correlation of 0.90550 and 0.84470, respectively.

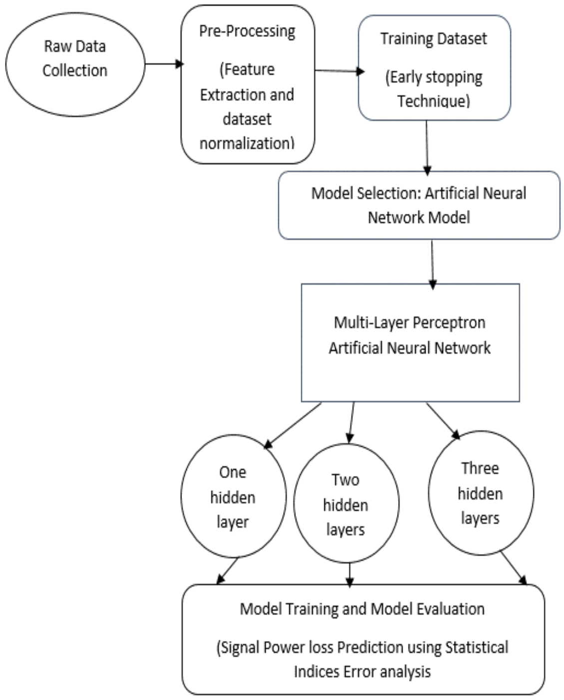

Fig. 1. Flowchart of procedure of MLP-ANN signal power loss training and prediction.

The multi-layer perceptron network training with *two* hidden layers shows the best prediction with a coefficient of correlation of 0.96790 and a standard deviation of 1.82720 with 50 neurons in the hidden layer *one* and 45 neurons in the hidden layer two, respectively. In comparison with training with the lowest number of neurons of 10 neurons in the hidden layer, *one* and 5 neurons in hidden layer *two*, and the highest number of neurons of 100 neurons in the hidden layer *one* and 95 neurons in the hidden layer *two*, the coefficient of correlation is 0.91850 and 0.75890. Lastly, the MLP network with three hidden layers was trained with the best prediction result seen with 30 neurons in hidden layer *one*, 25 neurons in hidden layer *two*, and 20 neurons in hidden layer *three* with the coefficient of correlation of 0.91830.

This work has been organized as follows: Section II describes the architectural composition of MLP-ANN, back-propagation, and MLP-ANN training. Section III described the training algorithm, method of training set selection, neuron variation, data measurement procedures with predictions. The section IV analyses the results of this work, and finally, Section V concludes the work and recommends the future aspects.

II. Basic of Multi-layer Perceptron Artificial Neural Network

A. Architectural Composition of Multi-layer Perceptron Artificial Neural Network

The universal approximation theorem states that three-layered MLP-ANN approximates practically any nonlinear function [18], [24]. However, there is no specification of the size (number) of the hidden layer of neurons for specified problem complexity. The actual required number of the hidden layers and neurons has remained an open problem for effective neural network training/learning. Various hidden layer neurons may result in over-learning during network training, while few neurons may result in inadequate training/learning of the data trend [16]. The neural network can be assessed in terms of their generalization and mapping capability. Fig. 2 shows the basic multi-layer perceptron artificial neural network with multiple hidden layers.

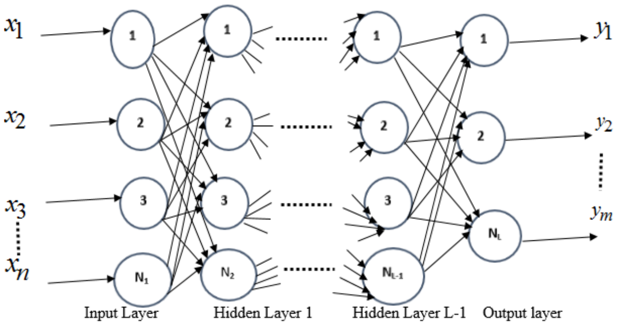

Fig. 2. Multi-layer perceptron artificial neural network with multiple hidden layers.

Practically, MLP-ANN with one or two hidden layers is usually employed for Radio Frequency (RF), and microwave applications as at least one hidden layer are necessary for nonlinear function approximation. However, a four-layered perceptron may be better in the modeling of nonlinear problems where there is the existence of certain localized behavioral components repeatedly in various regions of problem space [16]. A three-layered MLP-ANN, though can model such problems, it may require many hidden layer neurons.

Neurons are switches with information input and output, which will be activated with enough stimuli of other neurons stroking the information input and sending a pulse at the information output [25]. The neurons receive information from the synapses, which are special connections. Mathematically, a single hidden layer of MLP-ANN can arbitrary approximate functions, however, this is with only finite discontinuities and first derivations [16]. The proof thus is not constructive as there is a gap to the correct number of neurons and weights required. An n-layer has exactly n-variable weight layers and $n+1$ neuron layer.

If the total number of layers is L, the input layer is the first layer, the second layer $L-1$ is the first hidden layer, while L^{th} the layer is the output layer. If the neuron numbers in the L^{th} layer are N_L then $L = 1, 2, 3, \ldots$. If w^1_{ij} represents the weight of the j^{th} neuron of $(L-1)^{th}$ the layer and the i^{th} neuron of the L^{th} $1 \leq jN_{L-1}, 1 \leq i \leq N_L$. If x_i represents i^{th} external input, the MLP-ANN and z_i^L is the output of the i^{th} neuron of the L^{th} layer. Introducing extra weight parameter for each neuron $w_{i0}{}^L$ represents the bias for the i^{th} neuron of L^{th} layer. Therefore, the weight w of the MLP-ANN includes $w^L{}_{ij}$ s, where $j = 0, 1, 2, 3 \ldots N_{L-1}$, and $i = 1, 2, 3, \ldots N_L$, $L = 1, 2, 3, \ldots, L$, thus:

$$w = \left(w_{10}^2 w_{11}^2 w_{12}^2 \ldots\ldots\ldots\ldots w^L{}_{N_L N_{L-1}} \right)^T \qquad (1)$$

Practically, the w optimal weight values are obtained during the process of MLP-ANN training, where there is an adjustment of the weight such that error between the ANN model output and the initial problem output is minimized.

B. Back Propagation

A major objective in developing a neural network model is in finding the optimal set of weight parameters such that the actual output closely represents or approximates the initial problem behavior [5], [26]. This is actualized through a training process where a set of training data is presented to the MLP-ANN. The training data are pairs of the desired output and a total number of training samples, while the MLP-ANN performance during training is evaluated by computation of the difference between the actual and the desired output [27]. If x represents the initial problem or set of input data, then the actual output $y = (x, w)$, where w is the weight parameter. The neural network tries to find the optimal set of weight parameters w such that $y = (x, w)$, which should closely approximate the initial problem x. Let the set of the training data presented as:

$$(x_n, d_n), n = 1, 2, 3, \ldots\ldots.\mathrm{S} \qquad (2)$$

where d_n is the desired output of the ANN for input data $x_n S$ is the total number of the training samples. The performance of the ANN is assessed during the network training by computation of the ANN actual output and the desired output with the difference between the two known as error. This is expressed as [16], [28]:

$$\mathrm{E} = \frac{1}{2} \sum_{n \epsilon S_r} \sum_{j=1}^{m} (y_j(x_n, w) - d_{jn})^2 \qquad (3)$$

where d_{jn} is the j^{th} element of d_n, $y_j(x_n, w)$ is j^{th} ANN output for the x_n input and the S_r is the index of the training data. During the network training, the weight parameters w is adjusted, such that there is error minimization.

C. Multi-Layer Perceptron ANN Training

During the training process of the MLP-ANN, the weight w is initialized using small random values. The weight parameter is updated along the negative direction of the gradient of the error until the error becomes small enough [29], [30].

$$w = e - \eta \frac{\partial E}{\partial w} \tag{4}$$

where η is the learning rate parameter and e is the error. Using one training sample per time for weight update, the per-sample error function E_S is expressed as:

$$E_s = \frac{1}{2}\sum_{j=1}^{m}(y_j(x_n, w) - d_{jn})^2 \tag{5}$$

Multi-layer perceptron with sigmoid transfer function and linear transfer function in the hidden and output layer respectively, are universal approximators [31]. The weight and biases are adjusted during the training process in accordance with the training algorithm to give minimum Mean Squared Error (MSE). The network performance i.e. the MSE is expressed as:

$$MSE = \frac{1}{S}\sum_{j=1}^{m}[d(j) - y(j)]^2 \tag{6}$$

where S denotes the total number of training samples j in one epoch. Multi-layer perceptron-ANN is trained to employ back-propagation algorithms [14], [32]. The basic algorithm calculates an adjustable network parameter update for every calculation step m:

$$\chi(m+1) = \chi(m) - \eta.\frac{\partial MSE}{\partial \chi(m)} \tag{7}$$

where χ denotes a vector of present weights and biases, the partial derivative of MSE is the present gradient, η which is the learning rate.

For proper training of MLP-ANN, the process of training requires a set of examples of network behavior i.e. the network input and the target outputs. Therefore, the values of signal power measured at different points of the considered micro-cell environment over a distance of *800 m* were employed. During the process of training, MLP-ANN learns the relationship between the location of measurement points i.e. from the measured data, the link between the input vector and the target vector for the given environment. The number of the layers and neurons for the proposed MLP-ANN model are experimentally determined. The early stopping method and Bayesian Regularization were applied during network training to improve network generalization [14].

III. Training Algorithm, Method of Training Set Selection, Data Measurement Procedures

A. Training of ANN Using BR Algorithm

Bayesian Regularization (BR) algorithm is applied in weight update during network training in agreement with LM algorithm and has demonstrated near better training by linear permutation of squared error and weight variables [33]. The algorithm uses BR and modifies all variables in accordance with LM function approximation method [34], [35].

$$\left[J^t J + \lambda I\right]\delta = J^t E \tag{8}$$

where J and δ are the Jacobian matrix and the weight update vector (unknown), respectively, and E and λ are error and the damping factor, respectively. The damping factor for the optimization process is modified at all iteration. Usually, the Hessian is approximated using the Jacobian matrix [36]:

$$H = J^t J \tag{9}$$

The LM algorithm is based on the network preliminary weight value, convergence may occur at local minima, or there will be no convergence at all. The data outliers and initial weights are not considered, resulting in poor generalization [36]. Therefore, the BR algorithm is used to avoid poor network generalization by permitting adequate weights that are vital for solving the specific problem [37]. It increases the cost function to detect the smallest error in applying the smallest weight. Two hyper-parameters, alpha, and beta are introduced to advise the direction of the learning process. The cost function is expressed as [38]:

$$F = \beta E_d + \alpha E_s \tag{10}$$

where E_d is the sum squared error while E_s is sum squared weight. The addition of BR to LM adds up a small overhead to the process of network training.

B. Selection of Training Set and Neuron Variation

There is a need for appropriate training set selection from the real propagation path from which the MLP-ANN will learn to calculate received power, which is the most crucial factor in the training phase. For training optimization, the training set involves measurement data from different routes with different characteristics of propagation such as reflection, diffraction, reflection, direct rays, etc. The selected routes also include received positions that show various ranges of the input parameter. Hence the network can learn to behave in different situations and thus make an appropriate generalization on application to new cases.

The first important step in the training process is appropriate measurement points characterization in the training route according to their type of dominant path. The choice of training routes was a planned process and enough balanced number of measured data points that represent different propagation conditions supplied. A total 1970 data of measurements were recorded, each with different received signal power. The neural network was trained using different numbers of neurons in the hidden layers that vary from 5 to 100 neurons to extract a better assessment. The neuron variations are shown in Table I, Table II, and Table III for training with *one* hidden layer, *two* hidden layers, and *three* hidden layers, MLP-ANN, respectively.

C. Data Measurement, Tools, and Procedure

Typically, a single transmitter was assumed, while different received locations were part of the route or mesh grid. The received power was measured along different routes using drive test equipment [5], [39]. There were repeated measurements to average out the signal cancellations and enhancements as a result of the multi-path phenomenon. For each measurement point, the coordinates and received power level information in dBm were recorded. The signal power transmits at the *1900 MHz* band using *4 dBi* antenna gain and *35 dBm* transmit power. The receiver is mobile phones that were connected to a Personal Computer (PC). The receivers termed MI and M2 (mobile station 1 and mobile station 2) were set at idle and dedicated modes. M1 was set at idle mode while M2 was set to dedicated mode. The Base station is the transmitter.

The measurements started from a point very close to the transmitter, ending at a location *800 m* away from the transmitter. The location information using the driving test equipment was automatically recorded simultaneously. The measurement set-up is shown in Fig. 3.

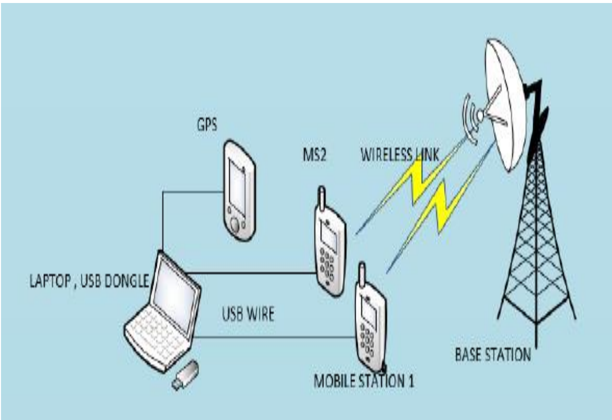

Fig. 3. A set-up of field test measurement [40].

D. Neural Network Training and Prediction

The neural network training was carried out using BR algorithm with early stopping data division technique. The dataset training was done via neural network toolbox (nntraintool) in MATLAB 2015a using *1970* input data, which is also the target data. Dataset was normalized in an excel spreadsheet before the neural network training to avoid bias in the order of presentation of the data pattern during the network training. The early sopping technique at a ratio of *70:15:15* for training, testing, and validation of the dataset to avoid overfitting was adopted. The dataset was trained for an average of ten runs to ensure adequate learning of the dataset by the network, while the training values with the least error recorded. The 1st order statistical indicators, the Standard Deviation (SD), and the coefficient of correlation (r) were employed for analysis of the performance of the trained dataset by measuring the difference between the actual value (target value) and the prediction value. The training time and the number of epochs required for training each number of neurons were also recorded. The measured dataset normalized before the neural network training to improve the training phase of the network using the vector normalization technique is expressed as [16], [41]:

$$d_n = \frac{d}{\sqrt{\sum_{l=1}^{n}(d_o)}} \tag{11}$$

where d_n and d_o are the normalized data value and the original data value, respectively. The *1st* order statistical indices use for result analysis are defined as [42], [43]:

$$SD = \sqrt{\left(\frac{1}{S}\sum_{o=1}^{S}|d_o - a_o| - MAE\right)^2} \tag{12}$$

$$r = \frac{\sum_{o=1}^{N}\left(d_o - \overline{d_o}\right)\left(d_o - \overline{a_o}\right)}{\sqrt{\left[\sum_{o=1}^{S}\left[do - \overline{do}\right]^2\right]\left[\sum_{o=1}^{S}\left[do - \overline{a_o}\right]^2\right]}} \tag{13}$$

where d_o and a_o are the desired and actual network output, respectively. $\overline{a_o}$ is the mean of the actual output and o = *1,2, 3, ……… s*, which are values for samples of signal power.

IV. Analysis of Results and Discussions

The field measurement result of the signal power at different points from the transmitting antenna was collected using mobile phones installed with TEMS software via a drive test. The collected data were transferred to a laptop installed with TEMS software, where the data had been extracted and normalized using an excel spreadsheet. The normalized data were used as input in training the artificial neural network model.

Table I shows the training result using MLP-ANN with *one* hidden layer. The neuron numbers in the hidden layer were varied between 5 to 100 neurons during network training using the BR training algorithm. The MLP-ANN was trained to vary the neuron numbers in the hidden layer, with 50 neurons giving the highest coefficient of correlation of 0.96870 and the least standard deviation 1.39040. This shows that training the network with 50 neurons gives the best prediction of the target values. However, the training time required was averagely high in comparison to the time required for training the network with fewer neurons, however, with fewer neurons, the network did not appropriately learn the training dataset. As the neuron numbers increases, the training time decreases with a strong deviation of the prediction values from the target values. At 100 neurons in the hidden layer, the coefficient of correlation dropped to 0.84470 and a standard deviation of 4.22880 was recorded except for 5 to 20 neurons in the hidden layer that required 199, 207, 364, and 564 epochs to get the network trained, other values from 25 to 100 neurons in the hidden layer required 1000 epoch to train the network.

In Table II, the MLP-ANN was trained using *two* hidden layers while varying the neurons in hidden layer *one* and hidden layer *two*. The neurons were varied between 10 to 100 in the hidden layer *one* and between 5 to 95 in the hidden layer *two*. The variation of neurons with 50 neurons in the first hidden layer and 45 neurons in the second hidden layer gave the highest coefficient of correlation of 0.96790 and the least standard deviation of 1.82720 i.e. best prediction of the target values. However, a very high training time of 3:36:10 was required, which was high in comparison to the time required for training the network with fewer and more neurons in the hidden layer *one* and hidden layer *two*, respectively. From 60 neurons in hidden layer *one* to 55 neurons in hidden layer 2, the training epoch required dropped from 1000 to 116, and at 100 neurons in hidden layer *one* and 95 neurons in hidden layer two, the required training epoch becomes only 56. However, at 100 and 95 neurons in the first and second hidden layers, the coefficient of correlation has dropped to 0.75890, and the standard deviation was 3.05660 showing a strong deviation of the prediction values from the target values.

In Table III, training of the MLP-ANN was performed using three hidden layers while varying the neurons in the three hidden layers. The variation of the neurons ranges from 15 to 100 in hidden layer *one*, from 10 to 95 in the hidden layer two and from 5 to 90 in the hidden layer three, respectively. The best prediction result is seen with 30, 25, 20 neurons in the first, second, and third hidden layers, respectively, with the highest coefficient of correlation of 0.91830 and a standard deviation of 1.80390 and training time of 00:32:12 recorded. Further training of the network required very high training time, and at 100, 95, 90, neurons in the first, second, and third hidden layers, the network training time was over seven hours. The values with the highest prediction accuracy are highlighted with astrics (*) in the Tables.

TABLE I: ANALYSIS OF NEURON VARIATION IN MLP-ANN WITH ONE HIDDEN LAYER

Neuron number for *one* hidden layer MLP	Training time	Epoch (1000)	Coefficient of Correlation (r)	Standard Deviation (SD)
5	00:00:02	199	0.90550	1.90550
10	00:00:04	207	0.91290	1.87660
15	00:00:13	364	0.92440	1.81460
20	00:00:25	564	0.94580	1.60600
25	00:00:26	1000	0.94770	1.58570
30	00:00:28	1000	0.95530	1.55170
35	00:00:31	1000	0.95650	1.54460
40	00:00:38	1000	0.96050	1.47900
45	00:00:39	1000	0.96170	1.47280
50*	00:00:39	1000	0.96870	1.39040
55	00:00:40	1000	0.96350	1.39710
60	00:00:44	1000	0.96250	1.43570
65	00:00:45	1000	0.95690	1.97650
70	00:00:49	1000	0.94540	2.24330
75	00:00:53	1000	0.92830	2.55390
80	00:01:00	1000	0.91150	2.98340
85	00:01:08	1000	0.90950	3.34270
90	00:01:14	1000	0.87430	3.74050
95	00:01:16	1000	0.86450	4.19010
100	00:01:22	1000	0.84470	4.22880

*Highest prediction accuracy

TABLE II: ANALYSIS OF NEURON VARIATION IN MLP-ANN WITH TWO HIDDEN LAYERS

Neuron number for two hidden layers MLP	Training time	Epoch (1000)	Coefficient of Correlation (r)	Standard Deviation (SD)
[10 5]	00:00:29	1000	0.91850	1.83180
[15 10]	00:00:47	1000	0.95590	1.56310
[20 15]	00:01:50	1000	0.96520	1.58540
[25 20]	00:04:40	1000	0.93390	2.45180
[30 25]	00:10:34	1000	0.90830	2.64050
[35 30]	00:30:37	1000	0.90490	2.00360
[40 35]	00:38:43	1000	0.90790	1.90320
[45 40]	03:30:40	1000	0.90710	1.91990
[50 45]*	03:36:10	1000	0.96790	1.82720
[55 50]	04:06:05	1000	0.91230	1.90870
[60 55]	01:14:23	116	0.90690	2.13640
[65 60]	01:10:30	112	0.90550	2.12240
[70 65]	01:05:39	107	0.90220	3.08900
[75 70]	00:59:40	102	0.89020	3.44890
[80 75]	00:52:23	89	0.87040	3.41470
[85 80]	00:46:04	86	0.84440	3.39020
[90 85]	00:39:10	81	0.76600	3.31980
[95 90]	00:36:40	76	0.76120	3.30010
[100 95]	00:32:14	56	0.75890	3.05660

TABLE III: ANALYSIS OF NEURON VARIATION IN MLP-ANN WITH THREE HIDDEN LAYERS

Neuron number for three hidden layer MLP	Training time	Epoch (1000)	Coefficient of Correlation (r)	Standard Deviation (SD)
[15 10 5]	00:03:08	1000	0.90460	1.98040
[20 15 10]	00:07:32	1000	0.91590	1.93130
[25 20 15]	00:07:85	1000	0.91030	1.90090
[30 25 20]*	00:32:12	1000	0.91830	1.80390
[35 30 25]	01:19:52	1000	0.94450	2.42670
[40 35 30]	03:41:04	1000	0.89400	3.30380
[45 35 30]	03:56:45	1000	0.89010	3.23450
[50 45 40]	03:51:20	1000	0.86100	4.30100
[55 50 45]	04:40:22	1000	0.81890	4.87600
[60 55 50]	04:52:10	1000	0.80070	4.89010
[65 60 55]	05:51:49	1000	0.80020	4.56900
[70 65 60]	05:45:45	1000	0.79090	5.67800
[75 70 65]	06:57:19	1000	0.75900	5.61240
[80 75 70]	06:45:41	1000	0.75750	5.60010
[85 80 75]	06:41:56	1000	0.73890	5.55900
[90 85 80]	07:47:10	1000	0.73420	6.67900
[95 90 85]	07:32:57	1000	0.71890	6.56700
[100 95 90]	07:09:40	1000	0.68370	6.48200

*Highest prediction accuracy

A multi-layer perceptron neural network with one hidden layer can effectively predict signal power loss with an adequately trained dataset as the target values. This is clearly seen as 0.96870: 0.96790: 0.91830 for the coefficient of correlation for MLP-ANN with one hidden layer, two hidden layers, and three hidden layers, respectively. And their standard deviation from the actual value (target value) is 1.39040: 1.82720:1.80390 for MLP-ANN with one hidden layer, two hidden layers, and three hidden layers, respectively. The training time required to train the MLP-ANN with one hidden layer is also very minimal in comparison to the required training time for training MLP-ANN with two and three hidden layers, respectively.

V. Conclusions and Future Recommendations

In this research work, MLP-ANN has been trained with *one*, *two*, and *three* hidden layered neurons with neurons ranging from 5 to 100 neurons in the hidden layers to ascertain the best architectural composition of MLP-ANN in the prediction of signal power loss. Other training strategies, such as training using the early stopping method and Bayesian regularization algorithm for optimum training results, were adopted. The training dataset represents different signal propagation paths and conditions, and the results can make suitable generalization in different propagation situations. 1st order statistical measurement indices, the coefficient of correlation, and the standard deviation were used to ascertain the performance of the MLP-ANN training while training time was also considered.

The results conclude that for effective prediction of signal power loss using the MLP-ANN model, a network with complex architectural composition is not required as this leads to overfitting during network training resulting in high prediction errors and very high training time. Future work will investigate signal power loss prediction in indoor and in-built houses using a multi-layer perceptron artificial neural network with a single hidden layer.

Conflict of Interest

The authors declare no conflict of interest.

Authors Contributions

Virginia C. Ebhota (VCE) and Viranjay M. Srivastava (VMS) conducted this research. VCE trained and analyzed the model with data and wrote the paper, VMS has verified the result with the designed model. All authors had approved the final version.

References

[1] T. S. Rappaport, *Wireless Communications: Principles and Practice*: Publishing House of Electronics Industry, 2004.

[2] F. Ikegami, T. Takeuchi, and S. Yoshida, "Theoretical prediction of mean field strength for urban mobile radio," *IEEE Transaction on Antennas Propagation*, vol. 39, pp. 299-302, 1991.

[3] H. Andrej, K. Gorazd, and J. Tomaz, "A survey of radio propagation modeling for tunnels," *IEEE Communication Surveys & Tutorials*, vol. 16, pp. 658-669, 2014.

[4] S. Yelen, S. S. Seker, and F. C. Kunter, "Radio propagation path loss prediction of UMTS for an urban area," in *Proc. 14th Biennial IEEE Conference on Electromagnetic Field Computation*, Chicago, IL, USA, May 9-12, 2010, p. 1.

[5] V. C. Ebhota, J. Isabona, and V. M. Srivastava, "Improved adaptive signal power loss prediction using combined vector smoothing and neural network approach," *Progress in Electromagnetic Research C*, vol. 82, pp. 155-169, 2018.

[6] V. C. Ebhota, J. Isabona, and V. M. Srivastava, "Base line knowledge on propagation modelling and prediction techniques in wireless communication networks," *Journal of Engineering and Applied Sciences*, vol. 13, pp. 1919-1934, 2018.

[7] E. Ostlin, H. J. Zepernick, and Suzuki, "Macrocell path loss prediction using artificial neural networks," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 2735-2747, 2010.

[8] C. Stergiou and D. Siganos, "Neural networks," *Sunrise Journal*, vol. 4, no. 11, 1996.

[9] J. Isabona and C. C. Konyeha, "Urban area path loss propagation prediction and optimisation using Hata model at 800MHz," *Journal of Applied Physics*, vol. 3, pp. 08-18, 2013.

[10] C. Phillips, D. Sicker, and D. Grunwald, "A survey of wireless pathloss prediction and coverage mapping methods," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 255-270, 2013.

[11] S. Haykin, *Neural Network and Learning Machines*, 3rd Ed. New Jersey, USA: Pearson Prentice Hall, 2009.

[12] A. B. Tammam, A. Rabie, and K. S. Mustafa, "Neural network approach to model the propagation path loss for great tripoli area at 900, 1800, and 2100 MHz bands," in *Proc. 16th International Conference on Sciences and Techniques of Automatic Control & Computer Engineering,* Monastir, Tunisia, Dec. 21-23, 2015, pp. 793-798.

[13] K. Burse, R. N. Yadav, and S. C. Shrivastava, " Channel equalization using neural networks: A review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 3, pp. 352-357, 2010.

[14] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*, Boston: PWS Publishing, 1996.

[15] V. C. Ebhota, J. Isabona, and V. M. Srivastava, "Effect of learning rate on GRNN and MLP for the prediction of signal power loss in microcell sub-urban environment," *International Journal on Communications Antenna and Propagation*, vol. 9, no. 1, pp. 36-45, 2019.

[16] V. C. Ebhota, J. Isabona, and V. M. Srivastava, "Investigation and comparison of generalization ability of multi-layer perceptron and radial basis function artificial neural networks for signal power loss prediction,"

International Journal on Communications Antenna and Propagation, vol. 9, no. 1, pp. 46-54, 2019.
[17] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are nural approximators," *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.
[18] G. Cybenko, "Approximation by superpositions of sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303-314, 1989.
[19] M. O. Omae, E. N. Ndungu, P. L. Kibet, and H. Tarus, "Artificial intelligence approach to signal propagation modeling for outdoor to indoor wireless communication networks: A proposed study," in *Proc. Mechanical Engineering Conference on Sustainable Research and Innovation*, 2012, vol. 4, pp. 12-16.
[20] H. Memarian and S. K. Balasundram, "Comparison between multi-layer perceptron and radial basis function networks for sediment load estimation in a tropical watershed," *Journal of Water Resource and Protection*, vol. 4, no. 10, pp. 870-876, 2012.
[21] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431-441, 1963.
[22] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, pp. 415-447, 1992.
[23] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian regularization," in *Proc. International Joint Conference on Neural Networks*, Houston, TX, USA, June 12, 1997, pp. 1930-1935.
[24] M. Hassoun, *Fundamentals of Artificial Neural Networks*. Massachusetts: MIT Press, 1995.
[25] M. Ayadi, A. B. Zineb, and S. Tabbane, "A UHF path loss model using learning machine for heterogeneous networks," *IEEE Transaction on Antennas and Propagation*, vol. 65, no. 7, pp. 3675-3683, 2017.
[26] M. K. S. Alsmadi, K. B. Omar, and S. A. Noah, "Back propagation algorithm: The best algorithm among the multi-layer perceptron algorithm," *International Journal of Computer Science and Network Security*, vol. 9, no. 4, pp. 378-383, 2009.
[27] A. B. Zineb and M. Ayadi, "A multi-wall and multi-frequency indoor path loss prediction model using artificial neural networks," *Arabian Journal of Science and Engineering*, vol. 41, pp. 987-996, 2016.
[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by backpropagating errors," *Nature*, vol. 323, pp. 533-536, 1986.
[29] C. Zanchettin, T. B. Ludermir, and L. M. Almeida, "Hybrid training method for MLP: Optimization of architecture and training," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 4, pp. 1097-1109, 2011.
[30] M. T. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transaction on Neural Networks*, vol. 5, no. 6, pp. 989-993, 1994.
[31] M. T. Hagan and H. B. Demuth, "Neural networks for control," in *Proc. American Control Conference*, San Diego, CA, USA, June 2-4, 1999, pp. 1642-1656.
[32] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111-122, 2011.
[33] V. C. Ebhota, J. Isabona, and V. M. Srivastava, "Signal power loss prediction based on artificial neural networks in microcell environment," in *Proc. 3rd IEEE International Conference on Electro-Technology for National Development*, Owerri, Nigeria, Nov. 7-10, 2017, pp. 250-257.
[34] R. Haelterman, J. Degroote, D. V. Heule, and J. Vierendeels, "The quasi-newton least squares method: A new and fast secant method analyzed for linear systems," *SIAM Journal on Numerical Analysis*, vol. 47, no. 3, pp. 2347-2368, 2009.
[35] B. M. Wilamowski and H. Yu, "Improved computation for Levenberg Marquardt training," *IEEE Transction on Neural Networks*, vol. 21, no. 6, pp. 930-937, June 2010.
[36] C. Kanzow, N. Yamashita, and M. Fukushima, "Levenberg-Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints," *Journal of Clinical and Analytical Medicine*, vol. 172, no. 2, pp. 375-397, 2004.
[37] M. G. B. Blum, M. A. Nunes, D. Prangle, and S. A. Sisson, "A comparative review of dimension reduction methods in approximate bayesian computation," *Statistical Science*, vol. 28, no. 2, pp. 189-208, 2013.
[38] J. Isabona and V. M. Srivastava, "A neural network based model for signal coverage propagation loss prediction in urban radio communication environment," *International Journal of Applied Engineering Research*, vol. 11, no. 22, pp. 11002-11008, 2016.
[39] J. Isabona and V. M. Srivastava, "Hybrid neural network approach for predicting signal propagation loss in urban microcells," in *Proc. IEEE Region 10 Humanitarian Technology Conference*, Agra, India, Dec. 23-25, 2016, pp. 1-5.
[40] M. S. Mollel and M. Kisangiri, "Comparison of empirical propagation path loss models for mobile communication," *Computer Engineering and Intelligent Systems*, vol. 5, no. 9, pp. 1-4, 2014.
[41] S. Ghahramani, *Fundamentals of Probability*, 2nd Ed. New Jersey: Prentice-Hall, 2000.
[42] H. Anysz, A. Zbiciak, and N. Ibadova, "The influence of input data standardization method on prediction accuracy of artificial neural networks," *Procedia Engineering*, vol. 153, pp. 66-70, 2016.
[43] V. C. Ebhota, J. Isabona, and V. M. Srivastava, "Investigating signal power loss prediction in a metropolitan island using ADALINE and multi-layer perceptron back propagation networks," *Applied Engineering Research*, vol. 13, no. 18, pp. 13409-13420, 2018.

Dr. Virginia C. Ebhota received a Bachelor of Engineering degree in Electronic Engineering from the University of Nigeria, Nsukka (UNN), Master of Engineering from Enugu State University of Science and Technology, Enugu, Nigeria and a Doctoral degree in Electronic Engineering from Kwazulu-Natal, Durban, South Africa. Presently, she is working towards a post-doctoral research fellow with the Department of Electronic Engineering, University of Kwazulu-Natal, Durban South Africa.

She is an active member of IEEE. She is author/co-author of more than 15 scientific contributions, including articles in international refereed Journals and Conferences. Her major research area is system modeling using Artificial Intelligence.

Prof. Viranjay M. Srivastava is a Doctorate (2012) in the field of RF Microelectronics and VLSI Design, Master (2008) in VLSI design, and Bachelor (2002) in Electronics and Instrumentation Engineering. He has worked for the fabrication of devices and the development of circuit design. Presently, he is working in the Department of Electronic Engineering, Howard College, University of KwaZulu-Natal, Durban, South Africa. He has more than 18 years of teaching and research experience in the area of VLSI design, RFIC design, and Analog IC design. He has supervised various Bachelors, Masters, and Doctorate theses.

He is a Professional Engineer of ECSA, South Africa and Senior member of IEEE, IET, and member of IEEE-HKN, IITPSA. He has worked as a reviewer for several Journals and Conferences, both national and international. He is author/co-author of more than 250 scientific contributions, including articles in international refereed Journals and Conferences, and also the author of various books.