International Journal of Networks and Communications 2026, 15(1): 1-12
DOI: 10.5923/j.ijnc.20261501.01

Comparative Analysis of Machine Learning Algorithms for Good Quality of Service on Terrestrial and Satellite Network Systems Over Nigeria

Oladayo Gbolahan Ajileye[1], Joseph Sunday Ojo[1], Vincent Andrew Akpan[2,*]

[1]Department of Physics, The Federal University of Technology, Akure, Nigeria
[2]Department of Biomedical Engineering, The Federal University of Technology, Akure, Nigeria

Abstract Rain fade presents a major challenge to maintaining quality of service (QoS) in wireless and satellite communication systems, especially at higher frequency bands (Ku and above). Severe rainfall can significantly attenuate signals, leading to transmission failures. To mitigate this, accurate modelling of rain-induced attenuation is essential for real-time adaptation of power levels, coding, and modulation through decision support systems. This study introduces a hybrid modelling approach using multilayer perceptron neural network (MLPNN) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS), and compares the results with Synthetic Storm Techniques (SST), which simulate rain attenuation based on time-series rain rate data. A two-input, one-output model is developed to estimate rain attenuation more effectively. At a rain rate of 180 mm/hr, observed attenuation was 48 dB, while predicted values were 46.88 dB (MLPNN) and 35.92 dB (ANFIS), demonstrating the potential of AI-driven models to approximate real conditions with high accuracy. These results enable decision support systems to dynamically adjust satellite parameters, ensuring QoS and service-level agreement (SLA) compliance during adverse weather. The proposed approach offers a reliable algorithm for improving the robustness of satellite and terrestrial links under extreme weather conditions.

Keywords Comparative Analysis, Machine Learning, Quality of Service, Terrestrial and Satellite Network

1. Introduction

During a period of heavy rainfall, satellite and terrestrial microwave communications that operate at frequencies higher than 10 GHz which may have signal interruptions [1]. Thus, point-to-point (PP) connectivity may impede continuous content streaming [2]. Link errors must be fully eliminated, especially during live broadcasting [3]. The signal along wireless communication networks is weakened by rain, a natural occurrence [4,5]; hence, mitigating rain attenuation is required to maintain a constant stream of content. Based on this assumption, dynamic fade mitigation strategies may be used in conjunction with fade prediction models that can predict the link's condition. In light of this, steps may be taken to guarantee that the link will remain completely operational for communication services even during a storm. Many researchers have employed an artificial neural network (ANN) for rainfall forecasting and demonstrated that the ANN may provide acceptable results after training [1,6–9]. In this study, rain attenuation is predicted and classified using the adaptive neural fuzzy inference system (ANFIS), a neural network employing a feedforward multilayer perceptron neural network (MLPNN) trained with Levenberg-Marquardt algorithm (LMA) for dynamic rain attenuation mitigation.

This research aims to comparatively analyze adaptive techniques for good quality of service on terrestrial and satellite network systems in Nigeria, especially during intense storms. The objectives are to deduce attenuation using 5-year rain rate data from a tropical location in Akure Nigeria (using the synthetic storm technique and ITU-R attenuation model); formulate an adaptive technique to predict and control signal to noise using hybrid adaptive measures (fuzzy logic and an artificial neural network), and further compare a proposed adaptive mitigation measure to improve Quality of Service (QoS) during intense weather conditions.

The techniques presented in this research can be adapted to create a stable and effective algorithm that can effectively manage signal propagation for network service providers as well as manage the quality of signals in channels impacted by attenuation due to weather at high frequencies. It can also serve as a noticeable contribution to improving the signals transmitted via satellite networks and invariably aid in improving satellite services. Many studies have been carried out by researchers to improve the quality of service on

* Corresponding author:
vaakpan@futa.edu.ng (Vincent Andrew Akpan)
Received: Dec. 14, 2025; Accepted: Jan. 3, 2026; Published: Jan. 7, 2026
Published online at http://journal.sapub.org/ijnc

wireless and satellite networks. Crane and co-workers carried out a researched the prediction of the effect of rain on satellite communication in temperate regions [10]. His research shows that the effect can be estimated if the distribution of rain intensity is known in both time and space. Harb and colleagues investigated the quality of service improvement in weather-impacted satellites using the Markov model to predict weather attenuation that can maintain quality of service via an intelligent awareness control system [11–14]. Nomura and co-researchers proposed a learning method of fuzzy inference in which the inference rules express the input-output relation of the data and are obtained automatically from the data gathered by specialists [15]. In this method, triangular membership functions were tuned through experience. The learning speed and generalization capabilities of this method were higher than those of a conventional back propagation neural network. Dhafer and co-workers also investigated how to simplify the factors for an ad-hoc network on mobile phones using fuzzy techniques, and it was deduced that higher throughput does not usually mean a high quality of service [8]. Furthermore, the system developer expressed the knowledge acquired from the input-output data in the form of fuzzy inference rules. Eyob and colleagues worked on enhanced adaptive code modulation (ACM) for rainfall fade mitigation in Ethiopia, and the results show that a neuro-fuzzy inference system can be employed to enhance mitigation techniques while maintaining link availability [9]. This study aimed to adopt a hybrid-adaptive mitigation measure to maintain the quality of signals along the terrestrial and satellite propagation links in a tropical location. The list of abbreviations and their respective meaning are listed in Table 1.

Table 1. List of abbreviations and meaning

S/N	Abbreviation	Meaning
1	ITU-R	International Telecommunication Union – Radiocommunication Sector
2	MLPNN	Multilayer Perceptron neural network
3	QoS	Quality of Service
4	ANFIS	Adaptive neuro-fuzzy inference system
5	AI	Artificial intelligence
6	SLA	Service-level agreement
7	PP	Point-to-point
8	LMA	Levenberg-Marquardt algorithm
9	SHF	Super high frequency
10	EHF	Extremely high frequency
11	NNARX	Neural network autoregressive with exogenous input model
12	ANN	Artificial neural network
13	ML	Machine learning
14	AI	Artificial intelligence
15	ACM	Adaptive code modulation
16	SST	Synthetic storm techniques
17	TDL	Tapped delay line
18	MSE	Mean square error
19	TSK	Takagi-Sugeno Khan
20	FNN	Feedforward neural network
21	RA	Rain Attenuation
22	IoT	Internet-of-Things
23		

The remaining sections of the paper are as follows. Section 2 gives the background knowledge of the work and methodology adopted. The adaptive modelling schemes and model predictors are presented in Section 3. The implementation and simulation strategies are demonstrated in Section 4. Section 5 presents a detailed results and discussions. Section 6 concludes the paper with discussions on future directions.

2. Background Knowledge

2.1. Modelling Based on Machine Learning Using Experimental Data

The goal of machine learning (ML), a division of artificial intelligence (AI), is to create systems that can learn from the data they consume and enhance their performance. Artificial intelligence is a general term that describes robots or systems that can imitate human intellect. Every learning strategy is data-driven. The system is trained using data sets. These data sets are gathered and applied to training. In this study, the system received 2732 rows of data based on rain rate measurement based on vertically-looking micro rain radar as presented in Figure 1. To validate the data used, tipping bucket rain gauge measurements from archived data of the Communication Research Group, Physics Department, Federal University of Technology, Akure have been used. It comprises of five years of rain data of one minute integration time series from 2013–2018.

The cumulative distribution function and, later, the diurnal influence on the signal were determined by analyzing the time series rain rates. At both the Super High Frequency (SHF) and Extremely High Frequency (EHF) bands, rain-induced attenuations (RA) were deduced using the SST model. The model utilized in this study is human-derived and based on human observations and experiences. It is known as mechanisms for supporting decision-making in networks as discussed in the next sub-section. The typical perception of models is that they are human-derived and serve as a foundation for machine learning. Some machine learning techniques, however, create their models free of any human-derived structure.

To achieve this practically, a system that maps an input to an output needs to be trained. The neural network and fuzzy logic are two examples of such systems [16,17]. Systems for machine learning must be taught through training [18]. Learning Systems frequently employ models, which offer mathematics in order to learn a mapping. The learning machine's structure (models or data) must be modified once

the system receives an input and its associated output. This has some similarities to regression or curve fitting. When additional inputs are added, the system ought to be able to provide the right results if we have enough training pairs. Problems can occur if there are not enough training sets, if the training data is not diverse enough, or if it fails to encompass the entire range.

(a)

(b)

Figure 1. Setup of the micro rain radar: (a) Observatory centre and (b) micro rain radar station

Unsupervised learning and supervised learning are the two types of machine learning [19]. Specific training sets of data are applied to the system during supervised learning. Although the "training sets" were created by people, the learning is supervised. However, this does not necessarily imply that humans are actively verifying the outcomes. Labelling is the process of categorizing the system's outputs for a certain set of inputs. In other words, you clearly state which outputs or results are correct for every set of inputs. The generation of training sets might take some time. However, tremendous efforts must be made to guarantee that the training sets will provide acceptable training results; in order to ensure that the system will produce the right results when real-world data are presented. The whole range of anticipated inputs and planned outputs must be covered by them. Test sets are used to validate the findings after the training.

2.2. Time-Series Using Synthetic Storm Techniques

Synthetic Storm Techniques (SST) are physical-mathematical radio propagation method that can be used to generate reliable rain attenuation time series by converting a rain rate time series at a specified site into a rain attenuation time series. Local parameters such as rain rate, length of the signal path through the rain cell, and rain cell velocity at the site under investigation are required in the SST [20]. Moreover, by applying the SST method, rain attenuation time series at any frequency and polarisation can be generated for any slant path above approximately 10 degrees, as long as the hypothesis of isotropy of the rainfall spatial field holds in the long term. The SST method is very useful for designing communication satellite systems and improving their performance for calculating rain attenuation over a satellite path. The troposphere's vertical structure divides into two levels when it rains. A is regarded as the rain layer, and B is the melting layer. The expression is given as

$$\left.\begin{aligned} A(t) = \mathrm{k}_A R^{\alpha_A}(t) L_A + \\ r^{\alpha_B} k_B R^{\alpha_{BA}}(t)(L_B - L_A) \end{aligned}\right\} \quad (1)$$

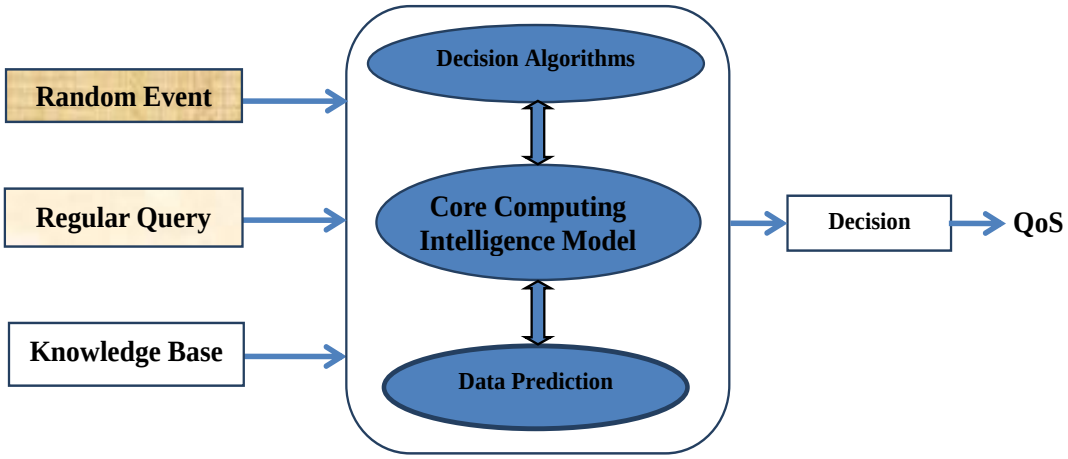


Figure 2. Block diagram of the proposed machine learning scheme for supporting decision-making in networks

where *A(t)* denotes the time series generated attenuation, $R^{\alpha A}(t)$ gives the time-series rain rate at rain layer *A*, $R^{\alpha B}(t)$ is the time-series rain rate at melting layer *B*, L_A and L_B gives the slant paths at layer *A* and *B*, respectively. α_A, and k_A are the constant that are estimated for layer *A* (for the temperature of precipitation is 20 ℃ as reported in the ITU-R Recommendation P.838-3 [21,22] and α_B, and k_B denotes the constant that are estimated for layer *B* (where the temperature of the precipitation is 0 ℃) [23].

2.3. Block Diagram Description of the Proposed Machine Learning Scheme for Good QoS

The proposed system makes use of different weather conditions (such as rain) occurring at various times are taken into account, supported by accurate background data, to enhance the effectiveness of our decision system. The block diagram of the proposed machine learning scheme uses random weather event, regular query and knowledge base in conjunction with ANFIS decision algorithms, core computing intelligent model and data prediction scheme based on neural network to determine good quality of service (QoS) as illustrated in Figure 2. This system utilizes the Levenberg-Marquardt algorithm within an Adaptive Neuro-Fuzzy Inference System (ANFIS) using neural network to perform predictions and decision-makings in ensuring consistent good quality of service (QoS).

The adopted system uses various random weather conditions, such as rain, which are queried for necessary precision at different times and supported by accurate background knowledge to enhance the effectiveness of good decision-making process.

3. Adaptive Modelling Schemes and Model Predictors

3.1. The Feedforward Multilayer Perceptron Neural Network (MLPNN)

The neural prediction scheme used in this paper is based on the series-parallel architecture shown in Figure 3 where the system is in parallel with the NN identification model (in the dashed box) and tapped delay line (TDL) denotes tapped delay line memory used to store temporal NN input information. It consists of a training algorithm and the NN model. Assuming that the input-output data pair Z^N of the system taken over *NT* period of time is available [19]:

$$Z^N = \{[U(N), Y(N)], \quad N = 1, 2, 3, \ldots\}, \tag{2}$$

where *N* is the number of data pair and *T* the sampling interval. Assuming also that the input-output data pair Z^N in (7) is available, and *m* and *n* are known. The inputs $\varphi(k) = [\varphi_{l_m}(k) \;\; \varphi_{i_n}(k)]$ to the NN, via an l_m-TDL and i_n-TDL, are the past *m*-inputs and *n*-outputs contained in $\varphi(k)$ which is obtained from Z^N defined in (2); where $l_m = p \times m$ and $i_n = q \times n$.

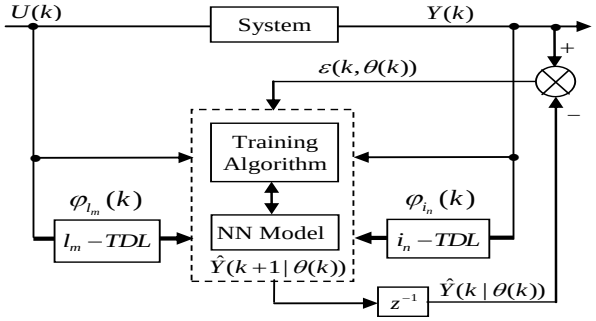


Figure 3. Series-parallel structure for neural network model identification

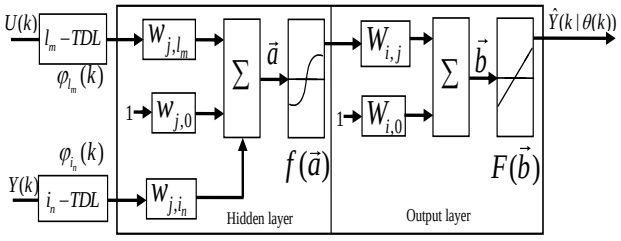


Figure 4. Multilayer perceptron neural network (MLPNN) model architecture

The NN considered of Figure 3 has a dynamic recurrent architecture as shown in Figure 4, since it contains temporary memories elements (TDL) and feedback, in this case, from the output of the system rather than the output of the NN model. Note that series-parallel structure shown in Figure 3 correspond to the autoregressive with exogenuous input (NNARX) model predictor. This is the so-called teacher forcing method where the network output is forced to follow the system outputs which leads to faster online training in real-time. With this method, the actual NN model, shown in Fig. 4, can be trained as a FNN.

The internal architecture for the proposed for the NN model is a MLP NN with two-layers (one hidden and one output layer) shown in Fig. 4. Decomposing $\varphi(k)$ in (14) into the input and output parts as $\varphi_{l_m}(k) = [U(k-d), \ldots, U(k-d-m)]^T$ and $\varphi_{i_n}(k) = [Y(k-1), \ldots, Y(k-n)]^T$ respectively, the output of the NN of Figure 3 can be expressed in terms of the network parameters of Figure 4 as

$$\hat{Y}(k \mid \theta(k)) = F_i\left(\sum_{j=1}^{n_h} W_{i,j} f_j(\vec{a}) + W_{i,0}\right) \tag{3}$$

where $\vec{a} = \sum_{l_m=1}^{n_m} w_{j,l_m} \varphi_{l_m}(k) + \sum_{i_n=1}^{n_n} w_{j,i_n} \varphi_{i_n}(k-1) + w_{j,0}$

and *j* is the number of hidden neurons; (w_{j,l_m} and w_{j,i_n}) and $W_{i,j}$ are the hidden and output weights respectively; $w_{j,0}$ and $W_{i,0}$ are the hidden and output biases; $F_i(\vec{b})$ is a linear activation function for the output layer and $f_j(\vec{a})$ is an hyperbolic tangent activation function for the hidden layer defined here as:

$$f_j(\vec{a}) = 1 - \frac{2}{e^{2\cdot\vec{a}} + 1} \tag{4}$$

Bias is interpreted as a weight acting on the input clamped to 1. Here, we use the term "joint weight" to imply both the network weights and biases that constitute $\theta(k)$.

The neural network output prediction problem then translates to a nonlinear minimization where the error estimates

$$\varepsilon(k, \theta(k)) = Y(k) - \hat{Y}(k \mid \theta(k)) \tag{5}$$

between the outputs of the true system output $Y(k)$ and the predictor output $\hat{Y}(k \mid \theta(k))$ in (3) is minimized in some sense and used to adjust and update $\theta(k)$ recursively to obtain an optimal parameter vector $\hat{\theta}(k)$. The minimization of (5) to obtain $\hat{\theta}(k)$ can be expressed as:

$$\hat{\theta}(k) = \arg\min_{\theta} J(Z^N, \varphi(k), \theta(k)) \tag{6}$$

where $J(Z^N, \varphi(k), \theta(k))$ is formulated as a mean square error (MSE) type cost function given as:

$$J(Z^N, \varphi(k), \theta(k)) = \frac{1}{2N} \sum_{k=1}^{N} [\varepsilon(k, \theta(k))]^2 \tag{7}$$

Several techniques for solving (5) exist in literature and different methods for adjusting and updating $\theta(k)$ have also been reported [19,24–26].

The training algorithm used here is the Levenberg-Marquardt algorithm (LMA) from the Deep learning Toolbox of the MATLAB [24]. The network training method is as follows. At time k, given Z^N, $\varphi(k)$, a small initial $\theta(k) = \theta_0$ as well as the available $U(k)$ and $Y(k)$; the training algorithm then computes the *a priori* output estimate $\hat{Y}^0(k+1 \mid \theta(k))$. At time $k+1$, $\theta(k+1)$ is available which is used to compute the *a posteriori* output estimate $\hat{Y}(k+1 \mid \theta(k))$ and the error (5) is used to adjust $\theta(k)$ until $\hat{\theta}(k)$ is obtained or certain stopping criteria is satisfied. Note that $\hat{Y}(k \mid \theta(k))$ in (5) is now given by (3) and we still seek the value of $\hat{\theta}(k)$ in (6).

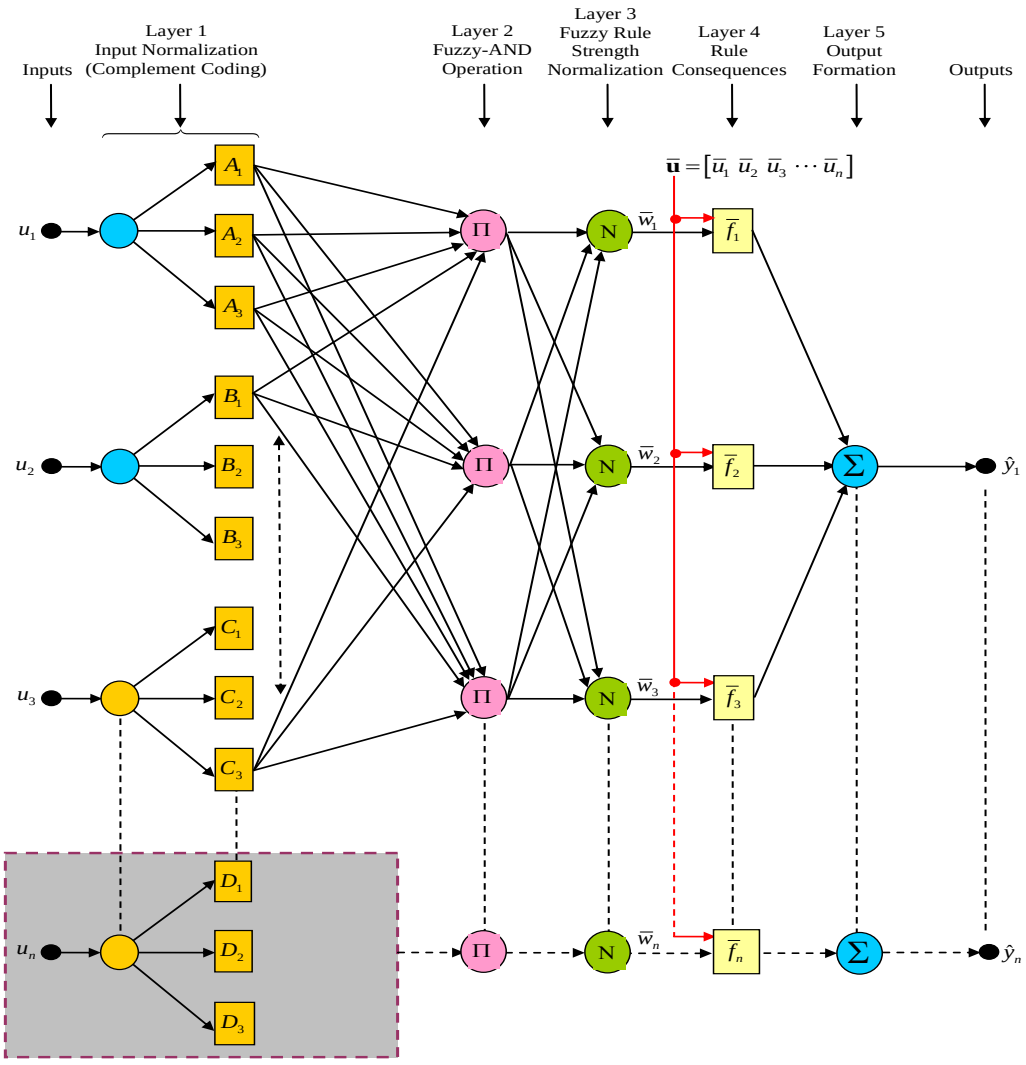


Figure 5. The architecture of the five-layer ANFIS

3.2. The Adaptive Neuro-Fuzzy Inference System (ANFIS)

The adaptive neuro-fuzzy inference system (ANFIS) was introduced to create a method for intelligent systems design whose behaviour or pattern inherent in a given dataset is unknown which could be obtained using neural network techniques from where fuzzy rule-based intelligence can be automatically generated and incorporated using the fuzzy inference system (engine). A typical architecture of a five-layer ANFIS is shown in Figure 5. The commonly and most widely used five-layer shown in Figure 5 ANFIS architecture including the one implemented in MATLAB® and Simulink® [24] which has been adopted for used in this study. For completeness, consistency and simplicity, each of these five layers is described below with their accompanying mathematical descriptions [27].

Layer No.1 - Inputs Normalization Layer (Complement Coding)

The nodes of this layer simply transmit the input signals to the next layer. The ANFIS uses the technique of complement coding to normalize the input training data. Complement coding is a normalization process that replaces an n-dimensional input vector $\mathbf{u}=[u_1,u_2,\ldots,u_n]$ with its 2n-dimensional complement coded form u' such that:

$$\mathbf{u}' \equiv [\bar{u}_1, 1-\bar{u}_1, \bar{u}_2, 1-\bar{u}_2, \ldots, \bar{u}_n, 1-\bar{u}_n] \tag{8}$$

where $[\bar{u}_1,\bar{u}_2,\ldots,\bar{u}_n]=\bar{\mathbf{u}}=\mathbf{u}/\|\mathbf{u}\|$. Complement coding helps avoiding the problem of category proliferation when using fuzzy-ART for data clustering. Having this in mind, we can write the I/O function of the first layer as follows:

$$\hat{y}_i^{(1)} = \left(\overline{u_i^{(1)}}, 1-\overline{u_i^{(1)}}\right), \qquad i=1,2,\ldots,Z^N \tag{9}$$

where Z^N is the number of input variables.

Layer No.2 - Fuzzy-AND Operation

These nodes represent a fuzzy set and they calculate the degree of activation of each membership function. Each node is characterized by the parameters of the fuzzy set it represents. Each node in this layer performs a fuzzy-AND operation. Similar to the other ANFIS networks, the T-norm operator of the algebraic product is selected. These results make each node's output to be the product of all its inputs:

$$\hat{y}_{k-j}^{(3)} = w_{k=j} = \prod_{i=1}^{Z^T} \hat{y}_{ij}^{(3)}, \qquad k(=j)=1,2,\ldots,Z^T \tag{10}$$

where Z^T is the number of output variables.

The output of each node in this layer represents the firing strength (or activation value) of the corresponding fuzzy rule. Note that the number of the fuzzy rules equals the number of input term nodes. The latter is common for all the input variables. Therefore, each fuzzy rule may be assigned an index *k*equal to the corresponding index *j* of the input term node, which is common for each input linguistic variable.

Layer No.3 - Normalization of Each Rule Firing Strength (Fuzzy Rules Strength Normalization Layer)

The nodes in this layer calculate the firing of the rule, usually through the product operator. The output of the k-th node in this layer is the firing strength of each rule divided by the total sum of the activation values of all the fuzzy rules. This results in the normalization of the activation values of all fuzzy rules:

$$\hat{y}_k^{(4)} = \overline{w_k} = \frac{\hat{y}_k^{(3)}}{\sum_{m=1}^{Z^R} \hat{y}_m^{(3)}}, \qquad k(=j)=1,2,\ldots,Z^R \tag{11}$$

where Z^R is the number of fuzzy-based rules.

Layer No. 4 – Rule Consequent Layer

The nodes of this layer correspond to the consequent of the rule. Each rule node in layer 3 has an equivalent node in this layer. The latter calculates the algebraic function of the Takagi-Sugeno Khan (TSK) rule. Both constant and linear function types are allowed. Each node k in this layer is accompanied by a set of adjustable parameters $a_{1k}, a_{2k}, \ldots, a_{N_{Inputs}k}, a_{0k}$ and implements the linear function:

$$\left.\begin{aligned} \hat{y}_k^{(5)} &= \overline{w_k f_k} \\ &= \overline{w_k}\left(a_{1k}\overline{u_1^{(1)}} + a_{2k}\overline{u_2^{(1)}} + +a_{Z^N}\overline{u_{Z^N}^{(1)}} + a_{0k}\right), \\ & k(=j)=1,2,\ldots,Z^R \end{aligned}\right\} \tag{12}$$

The weight $\bar{w}_k$ is the normalized activation value of the k-th rule calculated with the aid of (12). Those parameters are called the consequent parameters or linear parameters of the ANFIS system and are regulated by the training algorithm.

Layer No.5 - Output Layer

The single node of this layer produces the output of the network, performing the weighted average defuzzification process. For the proposed HANFA-ART system, this layer consists of one and only node that creates the network's output as the algebraic sum of the node's inputs:

$$\hat{y}^{(6)} = \sum_{k=1}^{N_{Rules}} \hat{y}_k^{(5)} = \sum_{k=1}^{Z^R} \overline{w_k}\,\overline{f_k} = \frac{\sum_{k=1}^{Z^R} w_k \overline{f_k}}{\sum_{k=1}^{Z^R} w_k} \tag{13}$$

Equations (8) through (13) shows how the input vector is fed through the network layer by layer to the predicted, estimated or classified output for a particular given problem.

4. Implementation and Simulation Strategies

The five-year rain rate data was evaluated using feedforward multilayer perceptron neural network (MLPNN)

trained with the Levenberg-Marquardt algorithm (LMA). Adaptive neuro-Fuzzy inference system (ANFIS), and neural network. As shown in Figure 3 is the neural architecture; the neural set-up comprises the two inputs (rain rate at layer A and rain rate at layer B) and output (Attenuation or quality of service (QoS)).

In this study, both the feedforward multilayer perceptron neural network (MLPNN) predictor of Figure 3 and the Adaptive neuro-fuzzy inference system (ANFIS) of Figure 5 were trained using the Levenberg-Marquardt algorithm (LMA). The 2,731 data was split into 70% (1,911) training data for both MLPNN and the ANFIS; 15% (410) was used for validation while 15% (410) was reserved for the trained network testing [17,19,25,26]. If the output prediction (QoS) are poor, the procedure is repeated by adjusting the hidden layer neuron for MLPNN, joint weight (network weights and biases) and the number of iteration for performance improvements while maintaining the same number of three past values for performance comparison until satisfactory performance was achieved.

The system is trained using the training data sets and validated using the validation data sets. These data sets are gathered and applied to training. In this instance, the system received 2732 new pieces of data. The rain rate and attenuation time series for frequency 12.25 GHz are shown in Figure 6. The relations show a trend whereby, as the rain rate increases, attenuation also rises. For example, at a rainstorm's 180.6 mm/hr over 1100 seconds time interval, attenuation was 50 dB, which had a significant impact on the propagating signal, while at a drizzle's rain rate categorization of 30.4 mm/hr, the attenuation was 5.4 dB.

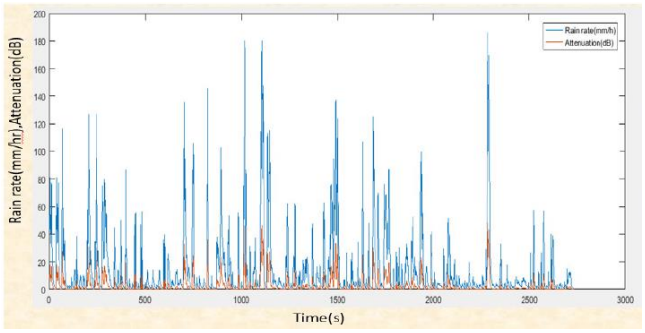


Figure 6. Time Series Rain Rate Data and Attenuation at 12.25GHz

5. Results and Discussion

The MLPNN and the ANFIS were trained using the LMA through simulation studies and the structure of the simulated MLPNN and the ANFIS networks are shown in Figure 7 and Figure 8 respectively. The LMA modifies the weight and bias parameters in accordance with LMA optimization scheme. In Figures 9 and 10, the Levenberg-Marquardt optimization procedure determines the update of the parameters using the adaptation parameter (Mu). Figure 6 displays the gradient and Mu-values of the trained network. Mu is the control parameter for the neural network training technique; its initial value was 0.001, and its maximum value is 1e10.

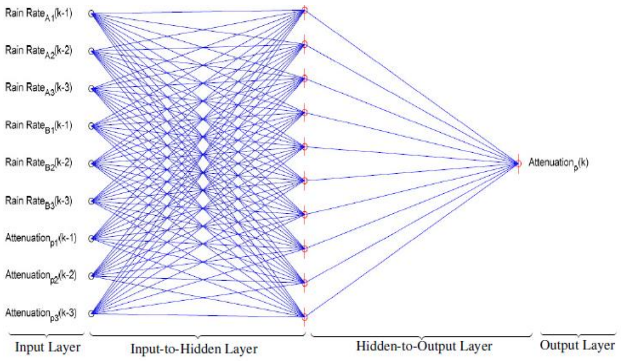


Figure 7. Structure of the simulated feedforward multilayer perceptron neural network (MLPNN)

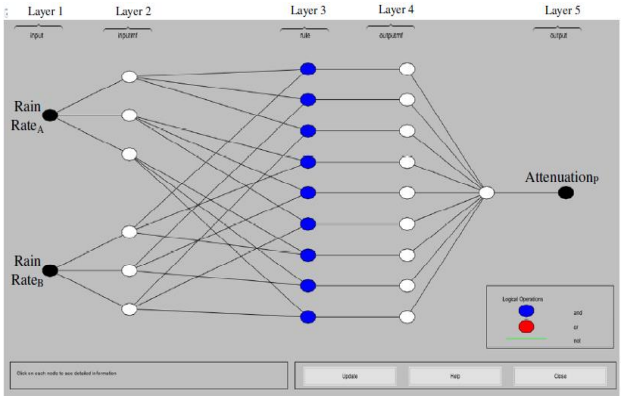


Figure 8. Structure of the simulated ANFIS network

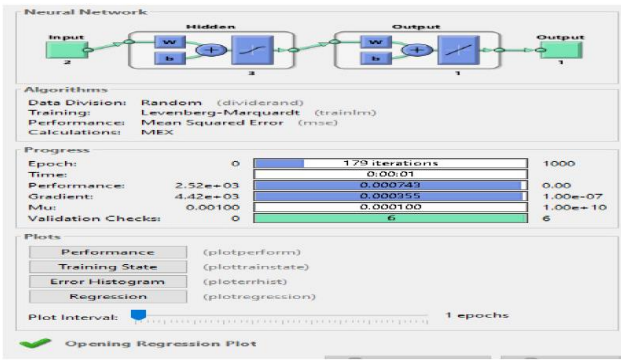


Figure 9. The graphical user interface (GUI) for the MLPNN training

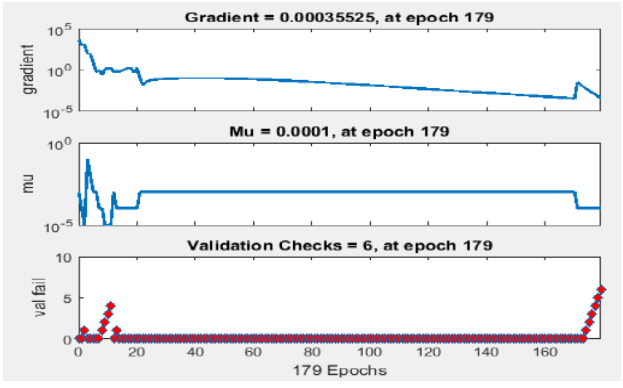


Figure 10. Convergence parameters for MLPNN training after 179 iterations

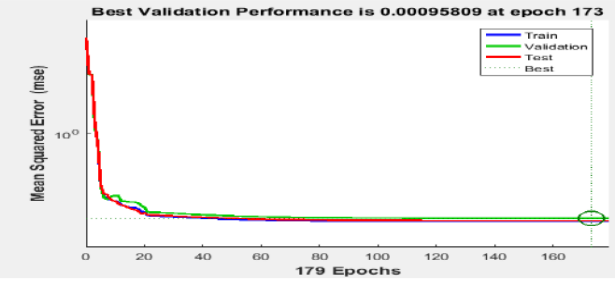


Figure 11. Convergence results for training, validation, testing based on mean squares error (MSE)

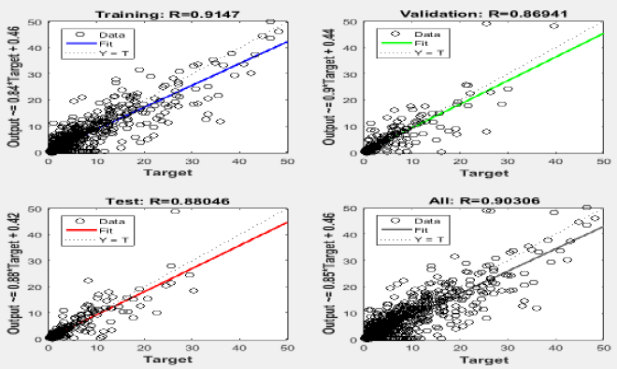


Figure 12. Linear regression ® fitting result for the MLPNN training

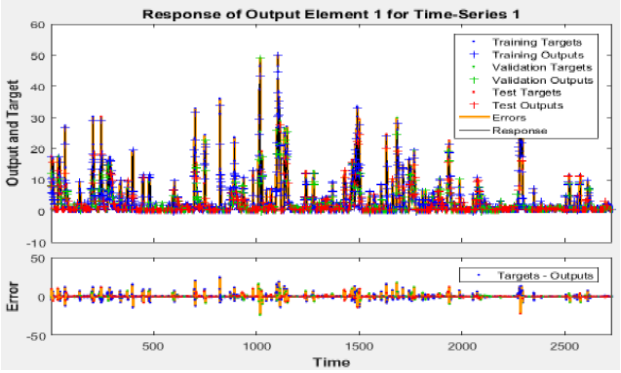


Figure 13. The MLPNN QoS output predictions and the prediction errors

The dotted-line attribute is the best-trained epoch as shown in Figure 11 which designates the iteration at which the validation performance was at its lowest. Before coming to an end, there were another six (6) repetitions of training. It is probable that some over-fitting occurred if the test curve grew considerably before the validation curve did. The creation of a regression plot, which depicts the relationship between the network's outputs and the targets (Figure 12), is the next stage in verifying or validating the network. The network outputs and targets would be precisely equal if the training was flawless, but in reality, this is rarely the case. For instance, using commands, we might make a regression plot. The first command figures out how the trained network will react to each input in the data set. The targets and outputs from the training, validation, and test subsets are extracted by the other instructions or commands. Three regression graphs are produced by the last command for training, testing, and validation. As shown in Figure 8, the three graphs reflect the training, validation, and testing data. Each plot's dashed line illustrates the perfect outcomes, which are outputs equal to targets. The solid line displays the best-fitting linear regression line between the outputs and the targets. The link between the outputs and the targets is shown by the R value. If R = 1, then the connection between the outputs and the targets is exactly linear. There is no linear relationship between outputs and targets if R is close to zero [24, 28]. Figure 13 shows the result of the trained data for the two layers and the attenuation. The regression results are all close to one.

With a very low mean square error (MSE) of 7e-7 for the train at 1911 samples and a regression rate close to 0.999 e-1, the trained MLPNN output time series is shown in Figure 14. The MSE of 6.386e-7 at 410 samples for validation and a 410 test sample MSE of 6.967e-7 are obtained from Figure 14, which clearly illustrates the value for the train, validation, and test.

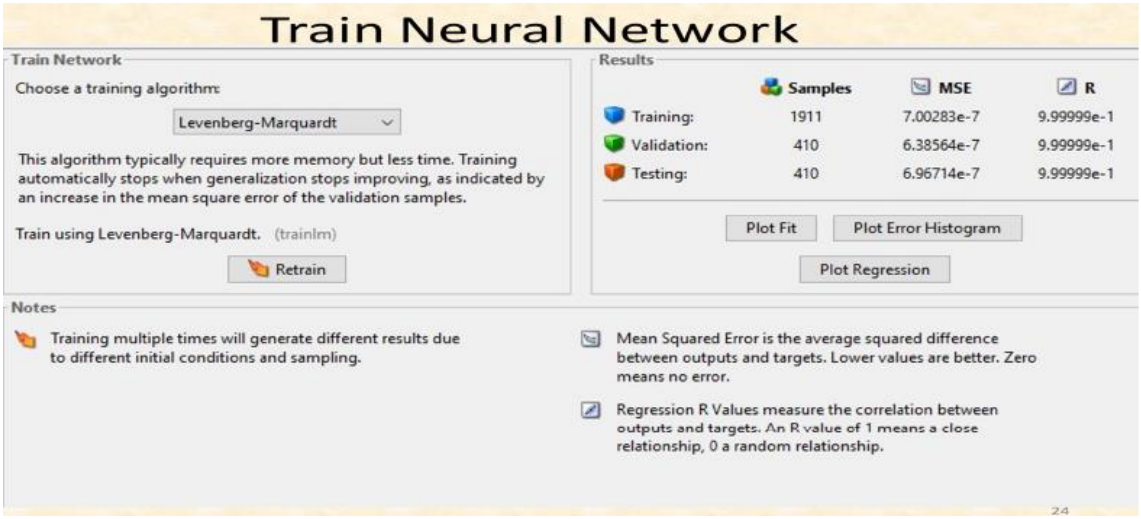


Figure 14. Performance metrics of the trained neural network

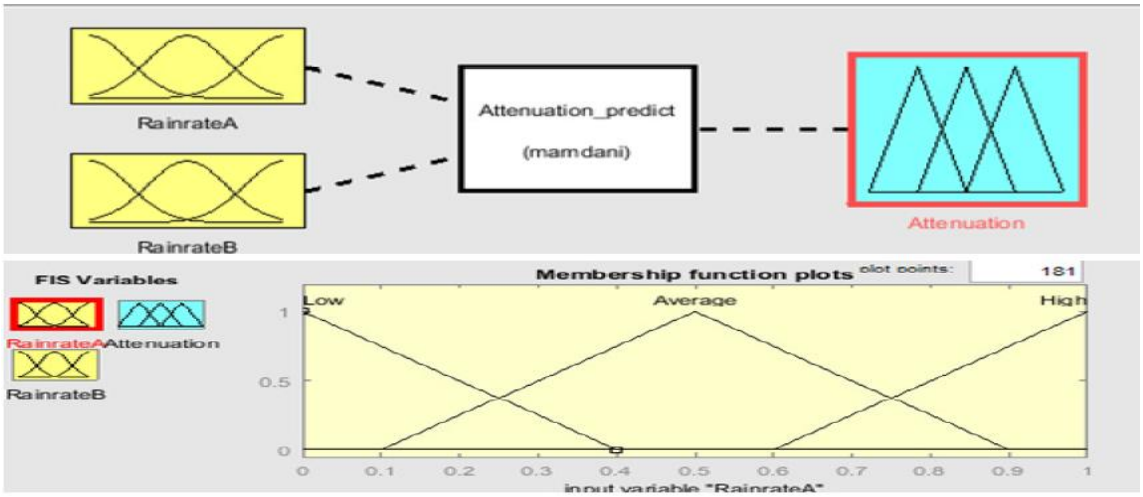


Figure 15. Inputs, rule editor, output and membership function editor of the ANFIS network

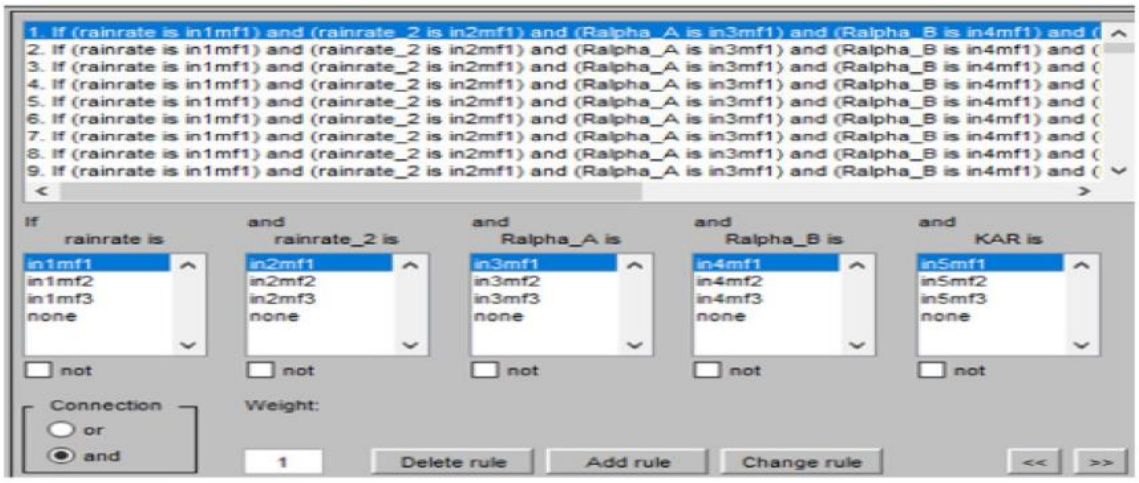


Figure 16. Automatically generated fuzzy rules by the trained ANFIS

The design of the adaptive neural fuzzy inference system with inputs (rain rate at layers A and B), outputs (attenuation), and corresponding membership functions are shown in Figure 15. The input parameters are the Rain rate at layers A and B, and the output value is Attenuation. Low, average, and high values are assessed for the input parameters.

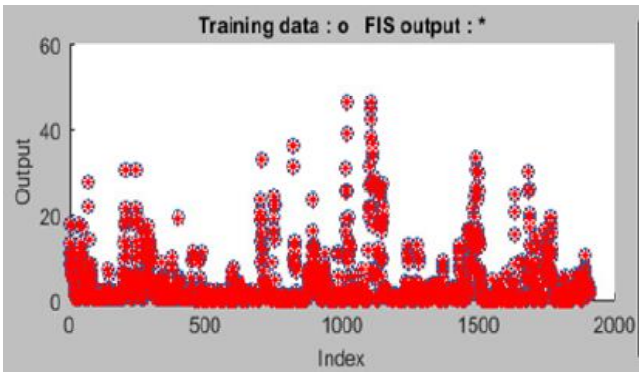
A comparable prediction model was created in a similar way to verify the research's central idea. With the same input and output data, we employed an Adaptive Neural Fuzzy Inference System (ANFIS). A neural network and a fuzzy inference system (FIS) are combined to create the Neuro-Fuzzy system. It has the advantages of both soft computing technologies together. By creating an if-then rule, it inherits the neural learning technique used to modify the membership function parameters from the neural network and the structure from FIS (Figure 16) [8,10]; nonetheless, sample data must be trained, tested, and checked to confirm the outcome. The trained FIS is confirmed with testing data that is different from the training data we used to train it, with 70% of the data—the 1910 sample data—used for training and the remaining 30% for testing the trained FIS network. Figure 17 demonstrates that there is a perfect match at the 1000 epoch, and Figure 18 shows that the training result has an error of 0.024287.

Performance Comparison of the Machine Learning Models

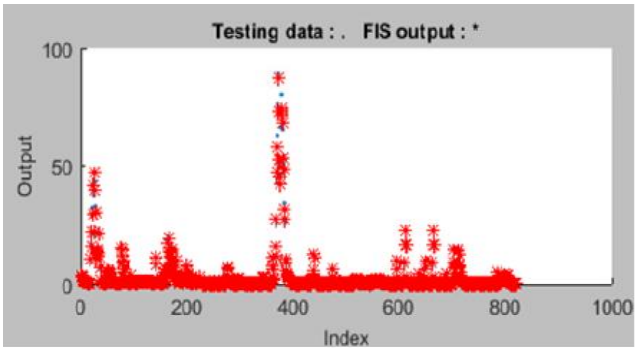
The number of iterations (or epochs) for both models—the neural network and the adaptive neural fuzzy inference system—was fixed at 1000. The neural model's best match was discovered after 179 iterations with a mean square error rate of 7e-7, whereas the ANFIS was 0.024287 after 1000 iterations. A table comparing the two models and their relative predicted outcomes based on error rates is shown in Table 2. The MLPNN model and ANFIS predictions for attenuation are compared to the actual attenuation in Table 2. For example, the predicted values for the multilayer perceptron neural network (MLPNN) and Adaptive Neural Fuzzy Inference System (ANFIS) are 10.79 dB and 9.31 dB, respectively, at actual attenuation of 10.76dB. Similarly, the MLPNN and the ANFIS predict values of 8.55 dB and 7.27 dB with an actual attenuation value of 8.56dB.

Figure 19 presents a plot comparison of the rain rate, attenuation, and predicted values during a time span of 500

seconds. For the values at the 500-second time period, there was a cluster. However, the predicted values also increased in tandem with the rising rain rate and attenuation values. According to MLPNN and ANFIS, the attenuation, for instance, is 48 dB at a rain rate of 180 mm/hr, while the predicted values are 46.88 dB and 35.92 dB, respectively. Similar to Figure 20, Figure 21compares the actual and predicted attenuation. At 10-second intervals, both the actual and predicted attenuations have a fair fix and good matches. The comparison of the logarithms in Figure 21 demonstrates the cluster and matches, especially at intervals of 1000 seconds. Back propagation neural network and adaptive neural fuzzy inference system projected appropriate values, as indicated in all attenuations for specified values. The two models actually predicted accurate values as corroborated in Figure 21.



(a)



(b)

Figure 17. The ANFIS output predictions: (a) training data and (b) test data

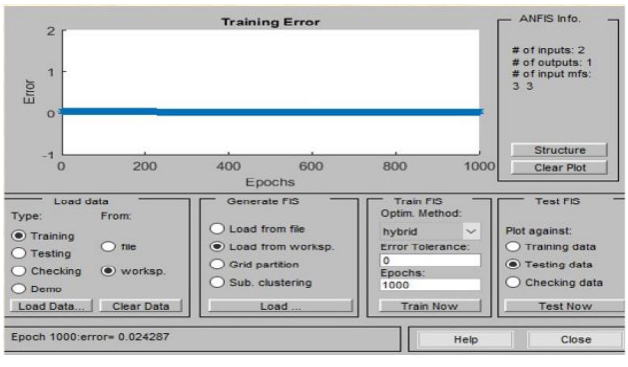


Figure 18. Training errors for the ANFIS network at 1000 iterations

Table 2. Predicted and actual values of attenuation

S/N	Actual Attenuation (dB)	MLPNN Predicted Attenuation (dB)	ANFIS Predicted Attenuation (dB)
1.	0.00000	0.023400	0.251763
2.	10.76467	10.79444	9.306781
3.	8.535769	8.54757	7.271931
4.	8.350319	8.359239	7.107184
5.	17.59825	17.56139	16.16896
6.	13.12695	13.13609	11.57359
7.	11.00689	11.03632	9.534007
8.	6.315339	6.290143	5.344835
9.	6.007914	5.978561	5.085714
10.	7.615696	7.612189	6.461414
11.	9.331598	9.353791	7.986856

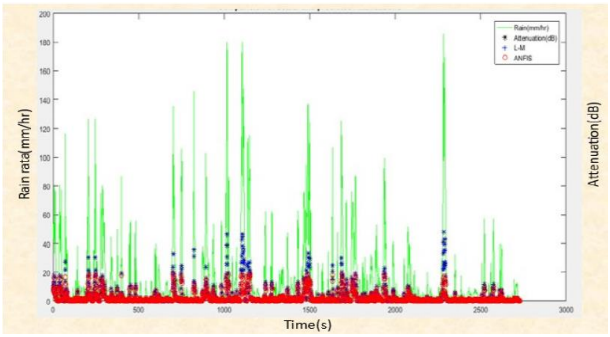


Figure 19. Comparison of actual rain rate, attenuation, and predicted attenuation

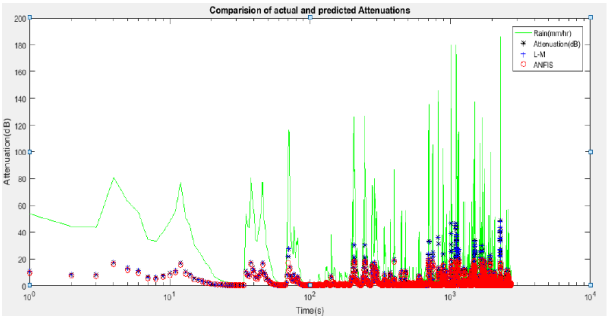


Figure 20. Comparison of actual rain rate, attenuation, and predicted attenuation

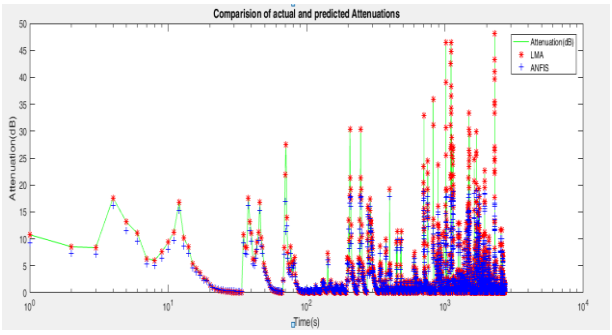


Figure 21. Logarithmic comparison of actual and predicted attenuation

6. Conclusions and Future Direction

6.1. Conclusions

Five-year rainfall data has been utilized in this study to assess various prediction methods and suggest mitigating strategies to enhance service quality and control signal to noise. Comparing the ANFIS and our actual output attenuation revealed an overall percentage difference of 0.2517%, whereas the Levenberg-Marquardt neural network and our actual output attenuation revealed a difference of 0.0234%. This suggests that, despite the fact that both MLPNN and ANFIS are beneficial for prediction purposes, attenuation prediction in neural networks is more accurate and suitable. This is corroborated in the results for both MLPNN and ANFIS, which exhibit the same pattern- and graphic-based trend. It was then deduced that the MLPNN result shows a more accurate and reliable prediction with lesser error and thus is a good decision support system for mitigating rain attenuation. With the predicted values, our model can accurately manage the loss of signal (attenuation) and provide necessary mitigation techniques for an effective signal to maintain quality of service, especially under intense weather conditions.

6.2. Future Direction

Future research should expand the scope of this study by incorporating longer-term and more diverse meteorological datasets to improve the robustness of attenuation prediction models. Extending rainfall data beyond five years, and integrating additional climatic variables such as temperature, humidity, cloud density, and wind speed, could further enhance model generalization under varying environmental conditions.

Furthermore, hybrid deep-learning architectures such as deep convolutional neural networks (Deep CNN), long short -term memory (LSTM) networks, random forest, support vector machine (SVM), and deep neuro-fuzzy systems should be investigated to determine their potential to outperform conventional MLPNN and ANFIS models in highly random weather scenarios. Future studies may also examine real-time adaptive learning techniques, enabling models to update continuously as new rainfall and attenuation data become available, thus improving accuracy during dangerous weather conditions.

Another promising direction is the development of intelligent, automated mitigation frameworks that combine predictive outputs with proactive countermeasures. These may include dynamic power control, adaptive modulation and coding, site diversity, and antenna beam optimization, triggered autonomously based on predicted attenuation levels. Implementing such systems within real communication networks will allow for end-to-end performance assessment and validation.

Finally, integrating satellite-based rainfall measurement data, IoT-enabled weather sensors, and GIS-based spatial analysis can support the creation of more broad attenuation maps. These tools will provide network designers and operators with better insights for infrastructure design, resilience planning, and long-term service quality improvement.

REFERENCES

[1] A. Dissanayake, J. Allnutt and F. Haidara, ‘A prediction model that combines rain attenuation and other propagation impairments along Earth–satellite paths’, *IEEE Transactions on Antennas and Propagation*, Vol. 45, No. 10, pp. 1546–1558, 1997.

[2] D. Chakraborty, ‘VSAT communications networks: An overview’, *Communications Magazine*, Vol. 26, No. 5, pp. 10–24, 1988. doi: 10.1109/35.449.

[3] G. Maral and J. M. Restrepo, ‘Satellite communications: fundamentals and future trends’, *International Journal of Satellite Communications and Networking*, Vol. 22, No. 1, pp. 3–19, 2004. doi: https://doi.org/10.1002/sat.762.

[4] J. S. Ojo, M. O. Ajewole and S. K. Sarkar, ‘Rain rate and rain attenuation prediction for satellite communication in Ku and Ka bands over Nigeria’, *Electromagnetics Research B*, Vol. 5, pp. 207–223, 2008. doi: 10.2528/PIERB08021201.

[5] J. S. Ojo and O. Falodun, ‘NECOP propagation experiment: Rain-rate distribution observations and prediction model comparisons’, *International Journal of Antennas and Propagation*, Vol. 2012, No. 913596, 2012. https://doi.org/10.1155/2012/913596.

[6] W. C. Daugherity, B. Rathakrishnan and J. Yen, ‘Performance evaluation of a self-tuning fuzzy controller’, In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 389–397, 1992. San Diego, CA.

[7] I. Adegbindin, P. Owolawi and M. Odhiambo, ‘Intelligent weather awareness technique for mitigating propagation impairment at SHF and EHF satellite network system in a tropical climate’, *SAIEE Africa Research Journal*, Vol. 107, No. 3, pp. 136–145, 2016.

[8] R. Z. Dhafer, S. Thulfiqar, A. Aldeen and A. Al-Wahab, ‘Simplified the QoS factor for the ad hoc network using fuzzy technique’, *International Journal of Communications, Network and System Sciences*, Vol. 6, pp. 381–387, 2013. doi: https://doi.org/10.4236/ijcns.2013.69041.

[9] M. W. Eyob and D. D. Feyisa, ‘Enhanced adaptive code modulation for rainfall fade mitigation in Ethiopia’, *EURASIP Journal on Wireless Communications and Networking*, Vol. 2022, No. 19, 2022. https://doi.org/10.1186/s13638-021 -02085-0.

[10] R. K. Crane, ‘Prediction of the effects of rain on satellite communication systems’, *Proceedings of the IEEE*, Vol. 65, No. 3, pp. 456–474, 1977. doi: https://doi.org/10.1109/PROC. 1977.10522.

[11] K. Harb, A. Srinivasan, B. Cheng and C. Huang, ‘Prediction method to maintain QoS in weather-impacted wireless and satellite networks’, In *Proceedings of the IEEE international conference on Systems, Man and Cybernetics (SMC)*, pp. 4008–4013, 2007. doi: https://doi.org/10.1109/ICSMC.2007. 4414255.

[12] K. Harb, A. Srinivasan, B. Cheng and C. Huang, 'QoS in weather-impacted satellite networks', In proceedings of the IEEE *Pacific Rim Conference on Communications, Computers and Signal Processing.* (PACRIM 2007) pp. 178–181, 2007.

[13] K. Harb, A. Srinivasan, B. Cheng and C. Huang, 'Intelligent weather aware scheme for satellite systems', *IEEE International Conference on Communications (ICC),* pp. 1930–1936, 2008. doi: 10.1109/ICC.2008.370.

[14] K. Harb, F. R. Yu, P. Dakhal and A. Srinivasan, 'An intelligent QoS control system for satellite networks based on Markovian weather prediction', *IEEE Vehicular Technology Conference (VTC 2010- Fall), Ottawa,* Ontario, Canada, pp. 1–5, 2010. doi: 10.1109/VETECF.2010.5594097.

[15] H. Nomura, I. Hayashi and N. Wakami, 'A learning method of fuzzy inference rules by descent method',*1992 IEEE International Conference on Fuzzy Systems,* pp. 203–210, 1992. San Diego, CA.U.S.A.

[16] J. Barron, 'Putting fuzzy logic into focus', *Byte*, 111–118. Dealing with ambiguous data, desktop fuzzy-logic applications deliver precise results; Pubblicato in Italia da "Metanetwork", n.2, Inverno 1993-1994, a cura di Tommaso Tozzi e Nazario Renzoni. Trattodallarivista "Byte", Ottobre, 1993, U.S.A. doi: https://www.strano.net/wd/fm_fz/fuzzy001.htm.

[17] D. I. Brubaker, 'Fuzzy-logic basics: Intuitive rules replace complex math', *EDN*, pp. 111–116, 1992. Available [Online]: https://www.pzs.dstu.dp.ua/logic/bibl/practical_approach.pdf

[18] B. Kosko, 'Neural Networks and Fuzzy Systems: A Dynamic System Approach to Machine Intelligence,' Prentice-Hall, Englewood Cliffs, New Jersey, U.S.A., 1992.

[19] V. A. Akpan, '*Development of new model adaptive predictive control algorithms and their implementation on real-time embedded systems'*, Ph.D. Dissertation, 517 pages, 2011. Available [Online]: http://invenio.lib.auth.gr/record/127274/files/GRI-2011-7292.pdf & http://invenio.lib.auth.gr/record/127274?ln=el.

[20] E. Matricciani, 'Physical-mathematical model of the dynamics of rain attenuation based on rain rate time series and a two-layer vertical structure of precipitation', *Radio Science*, Vol. 31, No. 2, pp. 281-295, 1996. doi: https://doi.org/10.1029/95RS03129.

[21] ITU-R P.838-3, 'International Telecommunication Union, *Recommendation ITU-R P.838-3: Specific attenuation model for rain for use in prediction methods*', Geneva: ITU, 2003. Available [Online]: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.838-3-200503-I!!PDF-E.pdf.

[22] ITU-R P.838-3, 'International Telecommunication Union, *Recommendation ITU-R P.838-3: Specific attenuation model for rain for use in prediction methods'*, Geneva: ITU, 2005. Available [Online]: https://www.itu.int/rec/R-REC-P.838/en.

[23] D. Maggiori, 'Computed transmission through rain in the 1–400 GHz frequency range for spherical and elliptical drops and any polarization', *Alta Frequenza (Italy)*, Vol. 50, pp. 262–273, 1981.

[24] The MathWorks, Inc. (2025) *MATLAB® and Simulink® 2025*. Natick, MA. doi: https://doi.org.mathworks.com.

[25] V. A. Akpan and G. D. Hassapis, 'Training dynamic feedforward neural networks for online nonlinear model identification and control applications', *International Reviews of Automatic Control: Theory & Applications*, Vol. 4, No. 3, pp. 335–350, 2011.

[26] V. A. Akpan and G. D. Hassapis, 'Nonlinear model identification and adaptive model predictive control using neural networks', *ISA Transactions*, Vol. 5, No. 2, pp. 177– 194, 2011. doi: https://doi.org/10.1016/j.isatra.2010.12.007.

[27] V. A. Akpan and J. B. Agbogun, 'A hybrid adaptive neural–fuzzy algorithms based on adaptive resonant theory with adaptive clustering algorithms for classification, prediction, tracking and adaptive control applications', *American Journal of Intelligent Systems*, Vol. 12, No. 1, pp. 9–33, 2022. doi: http://article.sapub.org/10.5923.j.ajis.20221201.02.html.

[28] Y. P. S. Foo and Y. Takefuji, 'Integer linear programming neural networks for job-shop scheduling', In *Proceedings of the IEEE International Conference on Neural Networks, V*ol. 2, pp. 341–348, 1988.