This paper is available online at www.praiseworthyprize.org
International Journal on Communications Antenna and Propagation (I.Re.C.A.P.), Vol. 13, N. 5
ISSN 2039 – 5086
October 2023

Influence of Learning Rate Hyper-Parameter and Early Stopping Training Technique in Training Vanilla Neural Network Model for Effective Signal Power Loss Prediction

Virginia C. Ebhota, Thokozani Shongwe

Abstract – *In this research work, the authors employed a Vanilla Neural Network (VNN) model to examine the influence of the Learning Rate (LR) hyper-parameter, early stopping training technique, Levenberg-Marquardt (LM) and Bayesian Regularization (BR) training algorithms in the training and prediction of signal power loss using a measured dataset from a Long Term Evolution (LTE) micro-cell environment. First order statistical performance indices, including the Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Regression (R), were adopted for interpreting and analyzing the results. The LR hyper-parameters were sequentially selected cyclically from 0.002 to 1.00, while the early stopping training technique was chosen at a ratio of 70%:15%:15% for training, testing, and validation of the neural network model during network training. The authors examined the neural network training and its predictive abilities of the measured signal power by training the model with LM and BR training algorithms and applying varied LR values and the early stopping training technique. Comparative analysis was also conducted by training the VNN model without the application of LR hyper-parameter and the early stopping training technique. The LR hyper-parameter is a distinct training parameter that, when efficiently applied, improves network convergence, while the application of training techniques such as early stopping minimizes over-fitting during network training. The training result outputs demonstrate the effectiveness of the VNN model when applying a very small LR of 0.002. The best prediction results of the VNN model were observed when using an LR of 0.002, with an R value of 0.9922, performance MSE of 1.48, and RMSE of 1.6790 while training the VNN model with the BR algorithm and applying the early stopping training technique. When training the VNN model with an LR of 0.002 using the LM algorithm, an R value of 0.9816, performance MSE of 5.060, and RMSE of 2.5619 were obtained. The worst prediction results were computed when training the VNN model without the application of LR and the early stopping training technique, resulting in an R value of 0.97480, performance MSE of 6.3, and RMSE of 3.0167. However, when training the same VNN model without the application of LR and the early stopping training technique using the BR training algorithm, an R value of 0.9910, performance MSE of 3.3, and RMSE of 1.8045 were obtained. This strongly demonstrates that the BR training algorithm is not only a good training algorithm but also an excellent training technique.*

***Keywords**: Signal Power Loss, Vanilla Neural Network Model, Learning Rate, Early Stopping Training Technique, Bayesian Training Algorithm, Levenberg-Marquardt Training Algorithm*

Nomenclature

ANN	Artificial Neural Network
AI	Artificial Intelligence
BS	Base Station
BR	Bayesian Regularization
GPS	Global Positioning System
GSM	Global System for Mobile communication
LM	Levenberg-Marquardt
LTE	Long Term Evolution network
MATLAB	Matrix Laboratory
ML	Machine Learning
MSE	Mean Squared Error
MS	Mobile Station
PL	Path-Loss
RF	Radio Frequency
RMSE	Root Mean Square Error
RSRP	Reference Signal Receive Power
R	Correlation coefficient
SD	Standard Deviation
TEMS	Test Mobile System software
VNNs	Vanilla Neural Networks
f	Frequency

Available online by October 31st, 2023
https://doi.org/10.15866/irecap.v13i5.23606

d	Distance
C	Speed
λ	Damping factor
δ	Weight update
β	Performance index
α	Decay rate
E	Error
E_t	Sum squared weight
E_d	Sum squared error
J	Jacobian matrix
η	Learning Rate parameter

I. Introduction

Despite the accomplishments of various Artificial Neural Network (ANN) models in different applications, there is still ongoing research on understanding their internal principles [1], [2]. It is crucial to have a thorough understanding of their hyper parameters and how to effectively use them for optimal performance during network training. Some hyper parameter values are defined prior to training, while others are selected empirically [3]. Determining appropriate hyper-parameters before training is important for achieving good training parameters. However, optimizing some hyper-parameters can be challenging as they often require physical search [3]. Additionally, the complex inter-relationship between hyperparameters makes it difficult to select suitable values. For example, a particular value for weight decay may work well with a specific Learning Rate (LR) for one ANN model but not for another. Understanding the effective application of hyperparameters such as LR and the necessary training strategies is vital for efficient convergence during ANN model training [4], [5].

A feed-forward Vanilla Neural Network (VNN) can be a suitable model for solving non-linear problems like predicting signal power loss. However, understanding the architectural compositions and selecting and tuning the hyper-parameters correctly are essential for optimal performance [6]. While there are default hyper-parameters and helpful rules of thumb, physically selecting and combining hyperparameters like LR, spread factor, and input delay factors is crucial for effective ANN model training [7].

The appropriate selection of LR as a hyper parameter for optimal ANN model training has been studied by various authors. In [8], the authors developed an approach using logistic regression and ANN, implementing the Improved Prediction System (IPS) with data mining steps for ball match prediction. The results showed excellent dataset prediction using a high LR. With an LR of 0.2, the prediction accuracy was 85%, while with an LR of 0.5, the accuracy improved to 95%.

In [9], the authors developed an ANN model to locate hidden patterns and bonds within a dataset. The results showed poor training of the dataset and low parameterization, with a highest prediction rate of 75% and a lowest prediction rate of 37.5% using the best LR of 0.2. A hybridized technique was utilized in the development of ANN Model and Linear Regression, investigation on employing Data mining phase: The efficient prediction of a dataset can be achieved through the application of low Learning Rate (LR) of 0.2, resulting in a prediction efficiency of 90.3% [8].

In this research work, we explore the technique of LR tuning, which involves tuning LR values during VNN (Vanilla Neural Network) model training to determine the best fit LR value for predicting signal power loss. To train the VNN model, we utilized a dataset from a Long Term Evolution (LTE) micro-cell environment operating at a frequency band of 1900 MHz. The performance of the VNN model was analyzed by applying various tuned values of LR during the model training. The accuracy of the predictions of signal power loss was evaluated using first-order statistical performance indices. The importance of training the VNN model with an appropriate LR hyper-parameter value was established by comparing the trained VNN model with different LR values, as well as a VNN model trained without the application of LR. The neural network training employed two back propagation algorithms: the Levenberg-Marquardt (LM) and the Bayesian Regularization (BR) training algorithms. The main objective of this research work was to determine the optimal LR value for effectively training the VNN model and enhancing the prediction of signal power loss. The efficiency of various LR hyper-parameter values for achieving VNN model convergence during network training was examined to ensure optimized network generalization. Learning rate tuning is crucial for the optimal functioning of the ANN model during neural network training, regardless of the gradient descent algorithm used [10]. This is because the LR hyper-parameter remains one of the most important hyper-parameters, as it significantly affects the overall performance of the VNN model [5].

The structure of this research work is as follows: Section I provides an introduction. Section II briefly discusses the VNN model, the learning rate, and the training paradigm. Section III details the methodology of dataset collection, VNN model learning and training. S

Section IV presents the results analysis using tables and graphs. Section V concludes with recommendations and suggestions for future work.

II. Background Study

This section delves into the fundamentals of the neural network learning paradigm.

II.1. The Neural Network Learning Paradigm

During neural network training, understanding the patterns in the dataset is crucial for artificial intelligence systems [12]. In the production phase, training the ANN model can take various forms, such as using a combined learning approach or following specific guidelines. A static network refers to a production phase where learning does not continue, while a dynamic system refers to a network that can continue learning during production [13]. Artificial neural network learning can be supervised, unsupervised, or a combination of both, depending on how the dataset is presented to the network [14], [15]. Different learning rules are applied for supervised and unsupervised learning, resulting in different outputs. ANNs are highly interconnected neurons that can perform specific tasks in a short amount of time. They are trained using learning rules, which are techniques used in neural network training. Learning algorithms are mathematical techniques used to update the weights of the neural network during training [3].

Each learning rule offers various algorithms, and different learning algorithms can be used with different learning rules.

Supervised learning involves building predictive models that require a defined training and validation protocol to ensure accurate predictions. When training a neural network with a large dataset, a suitable learning rate is necessary to avoid hindering learning and generalization. This is because having too many neurons or a high volume of computation can exceed the input and vector space dimensionality. Therefore, training ANNs requires careful input selection and training to prevent a poorly trained network, even with a well-designed network. The learning and training of ANNs are stochastic optimization processes that start with a random set of neural network parameters, searching for the optimal direction during training. This is known as gradient descent, where the optimal direction is determined using a random subset of the training dataset.

The Learning Rate (LR) hyper-parameter controls the step size in weight updates during neural network training [12], [22]. Using a small LR during VNN model training can help prevent missing local minima and improve network accuracy by slowing down the downward slope movement. However, this can result in longer convergence times as the network may get stuck on plateau regions [12], [16]. Typically, the LR is randomly designated during the programming of neural network training, making it difficult to determine the appropriate LR value for a specific problem.

The LR is defined by the Equation (1) as given [12]:

$$\Delta w_{id} = \eta \frac{\partial E}{\partial w_{id}} \eta O_K \delta_K \tag{1}$$

where η is the LR parameter, w is the weight of the neuron, $_{id}$ is input dataset and E is error.

An efficient training of the VNN model can be achieved by appropriately applying the LR through proper selection. This can be done either through random selection or LR estimation through dataset training, starting with a small LR and gradually increasing it exponentially to find a robust LR for the particular problem [17]. The change of LR at each iteration according to the cyclic function can also be studied [18].

Each cycle consists of a static length in terms of iteration number, and this technique allows for variation of LR between reasonable boundary values during network training.

II.2. Vanilla Neural Network Model and Training Algorithms

The VNN model is a feed-forward neural network that consists of interconnected neurons with a non-linear mapping between the input and output vectors. Each neuron has weights and an output signal, which is determined by the inputs and modified using a non-linear activation function [19], [20]. By selecting appropriate weights, activation functions, and hyper-parameters during training, the VNN model can approximate smooth and measurable functions between the input and output vectors [20]. The VNN model learns through the use of a dataset, where inputs are associated with their corresponding outputs [1]. It is composed of an input layer, one or more intermediate layers, and an output layer [19], [20].

During training, the VNN model uses the back propagation algorithm to adjust the weights for proper input-output mapping. However, the predicted output of the VNN model may not always match the known output from the input dataset, resulting in an error signal. This error signal represents the difference between the actual values and the predicted values [11], [21]. To improve the prediction ability of the VNN model, the appropriate Learning Rate (LR) hyper-parameter is applied during training. Additionally, using the right training algorithm enhances the prediction results. In this study, two back propagation training algorithms, the LM and BR algorithms, were used in the VNN training. Various values of LR were tuned to find the most suitable value for optimal prediction of signal power with minimal error.

The LM training algorithm [23] is a back propagation algorithm that improves the speed of second-order training without requiring the approximation of the Hessian matrix. On the other hand, the BR training algorithm minimizes the linear permutation of squared error and weight combination, defining the precise combination for a well-generalized neural network model. The BR algorithm is used for weight and bias updates in accordance with the LM training algorithm [13], [14].

This can be expressed as:

$$\left[J^t J + \lambda I\right]\delta = J^t E \tag{2}$$

where J is Jacobian matrix, λ is damping factor, δ is unknown weight update vector, and $J^t J$ is the approximate Hessian. The λ modification for the optimization process occurs at every iteration. With the introduction of two Bayesian hyper-parameters, namely alpha and beta, there is an increase in the cost function.

These hyper-parameters search for the minimum error by exerting the smallest weight and indicate the fitting direction for the learning process, which is minimal error and weights. Consequently, the cost function turns out to be:

$$F = \beta E_d + \alpha E_s \tag{3}$$

where E_d and E_s are sum squared error and sum of squared weight respectively.

During neural network training, it is important to avoid over-learning as it can lead to over-fitting and hinder the network's ability to generalize effectively. The weight capability of a neural network plays a crucial role in achieving proper convergence and optimal operation of the dataset [14]. Several factors, including the network configuration, hyper-parameter selection (such as learning rate), type of training instances, and complexity of the problem being addressed, define the network's ability to train and generalize well. Additionally, the architectural composition of the Virtual Neural Network (VNN) model also contributes to the overall performance and generalization capability of the network. If the architecture is overly large, the network may end up memorizing the training dataset, resulting in over-fitting and poor generalization. Therefore, it is essential to choose an appropriate network size for a specific problem to minimize transmission overhead.

To minimize and overcome over fitting during VNN model training, various network training techniques can be applied. These include the early stopping technique, Bayesian regularization technique, and tuning the number of intermediate layer neurons [14], [21]. The early stopping technique involves dividing the dataset into three sets: the training set, the testing set, and the validation set. This technique allows for the training instances to stop as soon as over fitting occurs during network training. The Bayesian regularization technique helps reduce over fitting by utilizing the squared error and sum of squared weight from the BR training algorithm. This technique improves the network's generalization ability. Additionally, selecting the appropriate number of intermediate layer neurons assists with cross-validation and ensures the accuracy of the VNN [21].

III. Methodology

The dataset was measured in an LTE micro-cell environment operating at a frequency band of 1900 MHz through a drive test. Various measurement tools were used, including a Global Positioning System (GPS) device, a laptop, LTE proficient mobile phones with Test Mobile System (TEMS) software, a compass, and network scanners. These tools were used to generate and evaluate datasets that contain signal power values from the transmitter. The acquired datasets were normalized using an Excel spreadsheet and trained using the deep learning toolbox in MATLAB 2022b software. The laptop and LTE proficient mobile phones with TEMS software provided convenient access, extraction, and recording of the signal dataset. The GPS and compass were used to match the user equipment locations with the transmitter. The log files from the drive test provided the Reference Signal Received Power (RSRP) in dBm. The VNN model training utilized the early stopping training technique, with a ratio of 70% for training, 15% for testing, and 15% for validation. This approach was adopted to prevent over fitting and improve the network's generalization during training, ultimately enhancing the predictive capability of the VNN model. Different Learning Rates (LR) were tested through tuning, and the same datasets were used to train the VNN model without LR and without early stopping training technique for comparison purposes. Additionally, the LM and BR training algorithms were employed to train the VNN model, and their performance during training was compared. The input datasets were normalized in an Excel spreadsheet using the provided expression:

$$N_v = \frac{\left[O_V - m_{\min}\right]}{\left[M_{\max} - m_{\min}\right]} \tag{4}$$

where O_V and N_v are the dataset original and normalized values respectively, $M_{\max}$ and $m_{\min}$ are the dataset maximum and minimum values respectively.

III.1. Optimizing Neural Networks Through Learning Rate Tuning

Randomly selecting a LR hyper-parameter value for efficient ANN model training is a challenging task. This is because different values of LR work well for different neural network models, and the underlying problem that the neural network model is solving must also be considered. However, it is possible to select an adequate LR hyper-parameter value for efficient training of a specific neural network model, such as the VNN model, through tuning of LR values [12]. In this research work, various values of LR hyper-parameter were used for training the measured dataset using the VNN model. The singular influence of these values on the VNN model's prediction of the measured dataset signal power was analyzed. The LR values ranged from 0.002 to 1.0 and were increased in steps of 0.002. The effect of the LR values on the training performance of the VNN model was examined using first order statistical performance indices, including the Root Mean Square Error (*RMSE*), coefficient of regression (*R*), and the performance Mean Squared Error (*MSE*). Additionally, notes were taken on the training time and the training epoch while using the LM and BR training algorithms, respectively. The results of the exerted first order performance indices are expressed [20]:

$$RMSE = \sqrt{MSE} = \frac{1}{K_{measured}} \sqrt{\sum_{k=1}^{k_{measured}} \left[t_k - y_k\right]^2} \tag{5}$$

$$MSE = \frac{1}{k_{measured}} \sum_{k=1}^{k_{measured}} (t_k - y_k)^2 \quad (6)$$

$$R = \frac{K_{measured(\sum t_k - y_k) - (\sum t_k)(\sum y_k)}}{\sqrt{\left[K_{measured}\left(\sum t_k^2 - (\sum t_k)^2\right]\left[K_{meausred}\left(\sum y_k^2\right) - (\sum y_k)^2\right]}} \quad (7)$$

t_k is the measured values while y_k is the predicted value and K ranges from 1 to .., representing measured signal power values.

III.2. Learning Rate Tuning to Examine the VNN Model's Generalization Capability

The main goal of the study was to determine whether the VNN model could effectively train and predict the observed dataset with little error and no over fitting. Two scenarios were looked at: one with the LR hyper parameter applied and the other without. Additionally, the effect of early ending training on the VNN model's capacity for generalization was looked into.

IV. Results of the Experiment and Discussion

The tabulated training outcomes are currently being discussed.

IV.1. Enhancing the Performance of Neural Networks by Adjusting the Learning Rate

The VNN model trainings were carried out applying LR tuning to examine the predictive ability of the VNN model on the measured dataset based on variations in the LR hyper-parameter values. The impact of the LR and application of the early stopping training technique on the performance output results from the VNN model training on prediction of the signal power was compared with the VNN training output results without the application of the LR and without the application of the early stopping training technique for the VNN model training.

The output result analysis and comparisons were made using first-order statistical performance indices, the *RMSE*, the *MSE*, and the *R*. The *MSE* value is low and closer to zero with a well-trained neural network, meaning that the predicted values of the VNN model are closer to the measured dataset as the value approaches zero. The *R* shows the strength of the relationship between the measured values and the predicted values, with *R* closer to +1 meaning a strong relationship. The *RMSE* is the mean error magnitude between the measured dataset and the predicted values.

The *R* graphs from the VNN model training were also examined, considering the training, testing, and validation results. The performance output results gotten from the VNN model exerting LM and BR training algorithms on application of the LR hyper parameter and on non-application of the LR hyper parameter are shown in the tables below. The tuning of LR values was done with LR values of 0.002, 0.004, 0.006, 0.008, 0.010, 0.012, 0.014, 0.016, 0.018, 0.20, and 1.00.

The best performance result outputs on training the VNN model with LR are highlighted in red, while the best result outputs on training the VNN model without LR are highlighted in blue for the exerting LM training algorithm and early stopping training technique in Tables I and II, respectively. On training the VNN model with the BR training algorithm, the best performance output results on training the VNN model with LR are highlighted in green, while the best performance output results without application of LR are highlighted in purple, all trained with the early stopping training technique in Tables III and IV, respectively. Training the VNN model with and without the early stopping training technique, with and without the LR hyper parameter, were also examined.

The best performance result outputs on training the VNN model without early stopping training technique using LM training algorithm are highlighted in yellow in Table V, and on training the same VNN model without early stopping technique using the BR training algorithm, the best performance result output is highlighted in dark red in Table VI. A constant neuron number was exerted in the intermediate layer of the VNN model for all training instances.

IV.2. Analysis of Performance Output Results Using Tables and Graphs

The performance output results of the VNN model trainings are shown in the tables and graphs below.

TABLE I
VNN MODEL TRAINING PERFORMANCE RESULT OUTPUTS AT DIFFERENT LR VALUES EXERTING THE LM TRAINING ALGORITHM AND ON APPLICATION OF THE EARLY STOPPING TRAINING TECHNIQUE

Learning Rate	Epoch (Iteration) Maximum= 1000	Time (seconds)	Performance (MSE)	Root mean Squared Error (RMSE)	Coefficient of Regression (R)	Validation check
0.002	18	00:00:02	5.060	2.5619	0.9818	6
0.004	8	00:00:01	5.260	2.6578	0.9804	6
0.008	10	00:00:01	5.480	2.6903	0.9799	6
0.010	11	00:00:01	5.600	2.7235	0.9794	6
0.012	10	00:00:01	6.180	2.7588	0.9789	6
0.014	12	00:00:01	6.480	2.7639	0.9788	6
0.016	8	00:00:01	5.240	3.0391	0.9744	6
0.018	8	00:00:01	5.590	3.0489	0.9743	6
0.020	10	00:00:01	6.340	3.2135	0.9723	6
0.022	8	00:00:01	7.200	3.2154	0.9717	6
0.024	7	00:00:01	7.140	3.2529	0.9705	6
0.026	7	00:00:01	4.750	3.3466	0.9691	6
0.028	7	00:00:01	6.630	3.6238	0.9653	6
0.200	7	00:00:01	6.400	4.0752	0.9560	6
1.000	**7**	**00:00:01**	**6.170**	**4.1355**	**0.9528**	**6**

TABLE II
VNN MODEL BEST TRAINING PERFORMANCE RESULT OUTPUTS WITHOUT LR VALUE EXERTING LM TRAINING ALGORITHM AND ON APPLICATION OF EARLY STOPPING TRAINING TECHNIQUE

Training algorithm	Epoch (Iteration) Maximum= 1000	Time (seconds)	Performance (MSE)	Gradient	Root mean Squared Error (RMSE)	Regression (R)	Validation check
LM	8	00:00:01	6.222	7.73	2.9593	0.9759	6

Comparing the training results of Tables I and II, the various results, which are the VNN model training results outputs on training the model with LR and on training the model without LR, respectively, exerting the LM training algorithm, show that the best training results for both instances are seen on application of a very small LR of 0.002 among other tuned LR values.

The best performance result output on application of LR gives *R* of 0.9818, *RMSE* of 2.5619, and *MSE* of 5.060. The training time was 00:00:02. On training the VNN model without application of LR, as shown in Table II, the best performance result outputs gave *R* of 0.9759, *RMSE* of 2.9593, and *MSE* of 6.222. The results from Table I in comparison to the results from Table II show higher *R*, lower *MSE*, and lower *RMSE* on training the VNN model using the LR hyper parameter in Table I in comparison to training the VNN model without the LR hyper parameter as shown in Table II. Both trainings were carried out using the LM training algorithm and the early stopping training technique.

Performance output results of training the VNN model with the LR hyper parameter and without the LR hyper parameter on application of the BR training algorithm for both training instances are shown in Tables III and IV, respectively. The results from both training instances show the best VNN model training with a small value of LR, which is 0.002 among other considered LR values.

TABLE III
VNN MODEL TRAINING PERFORMANCE RESULT OUTPUTS AT DIFFERENT LR EXERTING BR TRAINING ALGORITHMS AND ON APPLICATION OF EARLY STOPPING TRAINING TECHNIQUE

Learning Rate	Epoch (Iteration) Maximum= 1000	Time (seconds)	Performance (MSE)	Root Mean Squared Error (RMSE)	Coefficient of Regression (R)
0.002	1000	00:00:14	1.48	1.6790	0.9922
0.004	1000	00:00:14	2.86	1.7832	0.9913
0.008	1000	00:00:14	2.83	1.8461	0.9907
0.010	100	00:00:14	3.38	1.9425	0.9896
0.012	1000	00:00:09	2.73	19510	0.9895
0.014	1000	00:00:09	3.22	1.9603	0.9894
0.016	1000	00:00:09	3.24	1.9612	0.9894
0.018	1000	00:00:09	3.22	2.0295	0.9887
0.020	1000	00:00:09	3.39	2.0563	0.9883
0.022	1000	00:00:10	4.59	2.0698	0.9882
0.024	1000	00:00:09	3.39	2.0939	0.9879
0.026	1000	00:00:09	5.28	2.1366	0.9874
0.028	1000	00:00:10	3.73	2.1695	0.9870
0.200	1000	00:00:10	3.83	2.2204	0.9865
1.000	1000	00:00:12	3.39	3.1722	0.9731

TABLE IV
VNN MODEL BEST TRAINING PERFORMANCE RESULTS OUTPUTS WITHOUT LR EXERTING BR TRAINING ALGORITHM ON APPLICATION OF EARLY STOPPING TRAINING TECHNIQUE

Learning Rate	Epoch (Iteration) Maximum= 1000	Time (seconds)	Performance (MSE)	Root Mean Square Error (RMSE)	Coefficient of Regression (R)	Validation check
0.002	13	00:00:01	5.99	2.6631	0.9810	6
0.004	11	00:00:02	5.85	2.6931	0.9799	6
0.006	9	00:00:01	6.04	2.7192	0.9795	6
0.08	11	00:00:01	5.64	2.7359	0.9793	6
0.010	9	00:00:01	5.36	2.8086	0.9783	6
0.012	9	00:00:01	5.68	2.8131	0.9780	6
0.014	19	00:00:01	4.79	2.8215	0.9779	6
0.016	10	00:00:01	6.04	2.8358	0.9778	6
0.018	12	00:00:01	6.52	2.8445	0.9777	6
0.020	8	00:00:01	5.68	2.8713	0.9772	6
0.022	10	00:00:01	6.52	2.9145	0.9765	6
0.024	9	00:00:01	7.27	2.9182	0.9764	6
0.026	9	00:00:01	6.98	3.0025	0.9752	6
0.028	9	00:00:01	4.46	3.0441	0.9742	6
0.200	12	00:00:01	4.99	3.2353	0.9709	6
1.000	8	00:00:01	4.92	3.3343	0.9695	6

Training the VNN model with LR gives *R* of 0.9922, *RMSE* of 1.6790, and *MSE* of 1.48. The training time is 00:00:02. On training the VNN model without applying LR, *R* gives 0.9842, *RMSE* is 2.3862, and *MSE* is 5.11.

Both training instances were carried out using the early stopping technique. The results show the highest *R*, lowest *RMSE*, and lowest *MSE* on application of a low LR of 0.002 for training the VNN model in comparison to training the VNN model without application of LR.

On comparison of the best trained performance outputs results exerting the LM and the BR training algorithms, respectively, training the VNN model with the LM algorithm gave *R* of 0.9818, *RMSE* of 2.5619, and *MSE* of 5.060, while the best training of the VNN model with the BR algorithm gave *R* of 0.9922, *RMSE* of 1.6790, and *MSE* of 1.48. Both training instances were done exerting the early stopping training technique, and their results on training with 0.002 LR These results clearly demonstrate excellent VNN model training and prediction of the measured dataset, which is the signal power on application of not just the LR hyper-parameter, but training the VNN model with the BR training algorithm also improves its predictive ability.

As it has been ascertained from Tables I to IV, the application of the LR hyper parameter as well as the BR training algorithm improves VNN model prediction capability.

Further training of the VNN model was carried out without the application of the early stopping training technique initially used to ascertain and compare the model's performance on training with early stopping training techniques and on training without the early stopping technique.

The performance result outputs gave VNN model training results without early stopping training techniques on training with both LM and BR training algorithms, respectively, with an *R* of 0.9810. *RMSE* of

2.6631 and *MSE* of 5.99 with a training time of 00:00:02 on exerting the LM training algorithm as shown in Table V; on exerting the BR training algorithm, the performance output results are *R* of 0.9221, *RMSE* of 1.6970, *MSE* of 3.2, and a training time of 00:00:01 as shown in Table VI.

On comparison of the results of Tables V and VI, there are better training results without the application of the early stopping training technique when training the VNN model with the BR training algorithm in comparison to training with the LM training algorithm.

The output results of VNN model training with the BR training algorithm were very high in comparison to training the model with the LM training algorithm, showing that the BR training algorithm is also a good training technique that helps to overcome over-fitting and improve network generalization during neural network training.

The VNN model training results exerted by the LM and the BR training algorithms, as recorded in Tables I to VI, show that very few iterations are required for training using the LM training algorithm in comparison to the BR training algorithm, which requires 1000 iterations for each training for every trained instance. This is the batch number required for completion of each epoch, as the training algorithm iteration showed the pass number required for an optimal result output.

TABLE V
VNN MODEL TRAINING PERFORMANCE RESULTS OUTPUTS AT DIFFERENT LR EXERTING LM ALGORITHMS WITHOUT APPLICATION OF THE EARLY STOPPING TRAINING TECHNIQUE

Training algorithm	Epoch (Iteration) Maximum= 1000	Time (seconds)	Performance (MSE)	Root mean Squared Error (RMSE)	Regression (R)
BR	1000	00:00:07	5.11	2.3862	0.9842

TABLE VI
VNN MODEL TRAINING PERFORMANCE RESULT OUTPUTS AT DIFFERENT LM EXERTING BR TRAINING ALGORITHMS WITHOUT APPLICATION OF EARLY STOPPING TRAINING TECHNIQUE

Learning Rate	Epoch (Iteration) Maximum= 1000	Time (seconds)	Performance (MSE)	Root Mean Squared Error (RMSE)	Coefficient of Regression (R)
0.002	1000	00:00:10	3.2	1.6907	0.9921
0.004	1000	00:00:10	3.74	1.7431	0.9917
0.006	1000	00:00:10	4.22	1.8229	0.9908
0.008	1000	00:00:10	3.3	1.8824	0.9902
0.010	1000	00:00:10	2.74	1.8871	0.9902
0.012	1000	00:00:10	3.92	1.9276	0.9899
0.014	1000	00:00:11	3.61	1.9271	0.9898
0.016	1000	00:00:12	3.06	1.9400	0.9896
0.018	1000	00:00:09	2.18	1.9651	0.9895
0.020	1000	00:00:10	3.2	1.9622	0.9894
0.022	1000	00:00:11	3.39	1.9889	0.9891
0.024	1000	00:00:12	3.31	1.9889	0.9890
0.026	1000	00:00:12	2.63	2.0330	0.9886
0.028	1000	00:00:29	2.58	2.0572	0.9884
0.200	1000	00:00:30	1.93	2.1498	0.9874
1.000	1000	00:00:10	5.2	2.1605	0.9872

This shows that the LM training algorithm improves neural network training speed, resulting in just a few epochs for its effective training in comparison to the BR training algorithm, which required up to 1000 iterations for its effective training, which results in more training time. However, the BR training algorithm minimizes squared error, thus ensuring a more generalized VNN model for more efficient results.

Tables VII and VIII show the VNN model training without the application of the LR hyper-parameter and without the application of the early stopping training technique exerting the LM training algorithm with results shown in Table VII and the BR training algorithm with results shown in Table VIII. The results in comparison with training results from Tables I to VI indicate the importance of applying both the LR hyper parameter and the early stopping training technique in VNN model training, as their prediction results show improved efficient prediction of the signal power that is far above the prediction results in Tables VII and VIII, respectively. On comparison of the results in Tables VII and III, respectively, the results clearly demonstrate the effectiveness of the BR training algorithm in VNN model training and also show the BR training algorithm as not only a training algorithm but a good training technique, as the *R* of Table VII is 0.9910 using the BR training algorithm for VNN model training even without LR and early stopping training technique, which shows huge closeness to the *R* of 0.9922 recorded in Table III as the best VNN model prediction results when training the model with the BR training algorithm on application of both LR hyper parameter and early stopping training technique. The results of Tables IX to XII are regression results detailing the training, testing, and validation result values of the VNN model training on application of both LR and early stopping training techniques employing the LM and BR training algorithms, as shown in Table IX with the best trained results highlighted in red and in Table X with the best trained results highlighted in blue, respectively.

TABLE VII
VNN MODEL TRAINING PERFORMANCE RESULTS WITHOUT LR AND EARLY STOPPING TRAINING TECHNIQUES EXERTING THE LM TRAINING ALGORITHM

Training algorithm	Epoch (Iteration) Maximum= 1000	Time (seconds)	Performance (MSE)	Root mean Squared Error (RMSE)	Regression (R)	Validation
LM	8	00:00:01	6.3	3.0167	0.9748	6

TABLE VIII
VNN MODEL TRAINING PERFORMANCE RESULT WITHOUT LR AND EARLY STOPPING TRAINING TECHNIQUE EXERTING BR TRAINING ALGORITHM

Training algorithm	Epoch (Iteration) Maximum= 1000	Time (seconds)	Performance (MSE)	Root mean Squared Error (RMSE)	Regression (R)
BR	1000	00:00:09	3.3	1.8045	0.9910

TABLE IX
DETAILS OF VNN MODEL REGRESSION PERFORMANCE RESULTS AT DIFFERENT LR EXERTING LM TRAINING ALGORITHMS ON APPLICATION OF THE EARLY STOPPING TRAINING TECHNIQUE

Learning Rate	Training	Test	Validation	All
0.002	0.98555	0.96969	0.97586	0.98183
0.004	0.98665	0.96876	0.96523	0.98041
0.008	0.98096	0.97930	0.97809	0.97995
0.010	0.98463	0.95425	0.9755	0.97938
0.012	0.98142	0.97095	0.96939	0.97890
0.014	0.97996	0.97306	0.97763	0.97878
0.016	0.98290	0.96199	0.94867	0.97435
0.018	0.98192	0.95225	0.95283	0.97432
0.020	0.97619	0.94750	0.96760	0.97321
0.022	0.98028	0.93509	0.96960	0.97170
0.024	0.97656	0.95521	0.95414	0.97054
0.026	0.97446	0.95541	0.95692	0.96914
0.028	0.96060	0.97376	0.98433	0.96581
0.200	0.95426	0.95646	0.94233	0.95601
1.000	0.95115	0.94122	0.97587	0.95279

TABLE X
DETAILS OF VNN MODEL REGRESSION PERFORMANCE RESULTS AT DIFFERENT LR EXERTING BR TRAINING ALGORITHMS ON APPLICATION OF THE EARLY STOPPING TRAINING TECHNIQUE

Learning Rate	Training	Test	All
0.002	0.99283	0.98940	0.99222
0.004	0.99440	0.98577	0.99127
0.008	0.99455	0.97143	0.99071
0.010	0.99305	0.97002	0.98964
0.012	0.99413	0.96716	0.98952
0.014	0.99363	0.97126	0.98943
0.016	0.99350	0.95576	0.98937
0.018	0.99386	0.96253	0.98866
0.020	0.99477	0.93692	0.98833
0.022	0.99177	0.97455	0.98816
0.024	0.99358	0.95189	0.98794
0.026	0.98996	0.97239	0.98739
0.028	0.99384	0.93469	0.9870
0.300	0.99419	0.93618	0.98653
1.000	0.99333	0.85481	0.97310

TABLE XI
DETAILS OF VNN MODEL REGRESSION RESULTS AT DIFFERENT LR EXERTING LM TRAINING ALGORITHMS WITHOUT APPLICATION OF THE EARLY STOPPING TRAINING TECHNIQUE

Learning Rate	Training	Test	Validation	All
0.002	0.98554	0.96690	0.97428	0.98095
0.004	0.98287	0.97370	0.96948	0.97993
0.006	0.98259	0.97140	0.97204	0.97948
0.008	0.98081	0.98250	0.96993	0.97931
0.010	0.98243	0.96009	0.98299	0.97829
0.012	0.97961	0.96411	0.98144	0.97801
0.014	0.98617	0.97402	0.94417	0.97794
0.016	0.97928	0.97643	0.97763	0.9778
0.018	0.97955	0.98391	0.96879	0.97768
0.020	0.98244	0.94770	0.98295	0.97724
0.022	0.9782	0.96790	0.97521	0.97652
0.024	0.97795	0.98211	0.96724	0.97636
0.026	0.98125	0.96561	0.96561	0.97524
0.028	0.97743	0.96723	0.97542	0.97424
0.300	0.97364	0.96654	0.95725	0.97087
1.000	0.97170	0.94718	0.97136	0.96952

The best trained results were obtained with the application of the 0.002 LR hyper parameter for both training using the BR and LM training algorithms. Tables XI and XII show details of the regression results on the application of LR but training the VNN model without the early stopping training technique for training employing the LM and the BR training algorithms, respectively. The results of Tables XI and XII in comparison to the results in Tables IX and X show the essence of the application of the early stopping training technique as it helps in the reduction of network over-fitting, thereby improving network generalization.

However, on comparison of the best predicted results in Table X with R of 0.99222 on training the VNN model with both LR and early stopping training technique (highlighted in blue color) and R of 0.99211 on training the VNN model with LR but without early stopping training technique (highlighted in green color), both instances trained on application of the BR training algorithm, the two results show the effectiveness of the BR training algorithm as well as a good training technique, and the two results show closeness.

Tables XIII to XVI show the details of the regression results on training the VNN model without the LR hyper parameter but with the early stopping training technique, as shown in Tables XIII and XIV, respectively, and training the VNN model without the application of LR and without the application of the early stopping training technique, as shown in Tables XV and XVI, respectively, with clearly demonstrated results showing a decline in the prediction of signal power loss, which is the measured dataset.

TABLE XII
DETAILS OF VNN MODEL REGRESSION RESULTS AT DIFFERENT LR-EXERTING BR TRAINING ALGORITHMS WITHOUT APPLICATION OF THE EARLY STOPPING TRAINING TECHNIQUE

Learning Rate	Training	Test	All
0.002	0.99372	0.98459	0.99211
0.004	0.99425	0.97411	0.99167
0.006	0.99397	0.95616	0.99082
0.008	0.99441	0.96228	0.99023
0.010	0.99392	0.96376	0.99019
0.012	0.99334	0.97958	0.98988
0.014	0.99329	0.97256	0.98982
0.016	0.99418	0.96693	098964
0.018	0.99454	0.96796	0.98948
0.020	0.99328	0.97175	0.98936
0.022	0.99220	0.96824	0.98909
0.024	0.99352	0.95712	0.98896
0.026	0.99388	0.95117	0.98857
0.028	0.99391	0.95637	0.98838
0.300	0.99518	0.94891	0.98743
1.000	0.9908	0.96965	0.9872

TABLE XIII
DETAILS OF VNN MODEL REGRESSION RESULTS WITHOUT LR EMPLOYING THE LM TRAINING ALGORITHM ON APPLICATION OF THE EARLY STOPPING TRAINING TECHNIQUE

Training algorithm	Training	Test	Validation	All
LM	0.97541	0.97516	0.98051	0.97594

TABLE XIV
DETAILS OF VNN MODEL REGRESSION RESULTS WITHOUT APPLICATION OF LR EMPLOYING THE BR TRAINING ALGORITHM ON APPLICATION OF THE EARLY STOPPING TRAINING TECHNIQUE

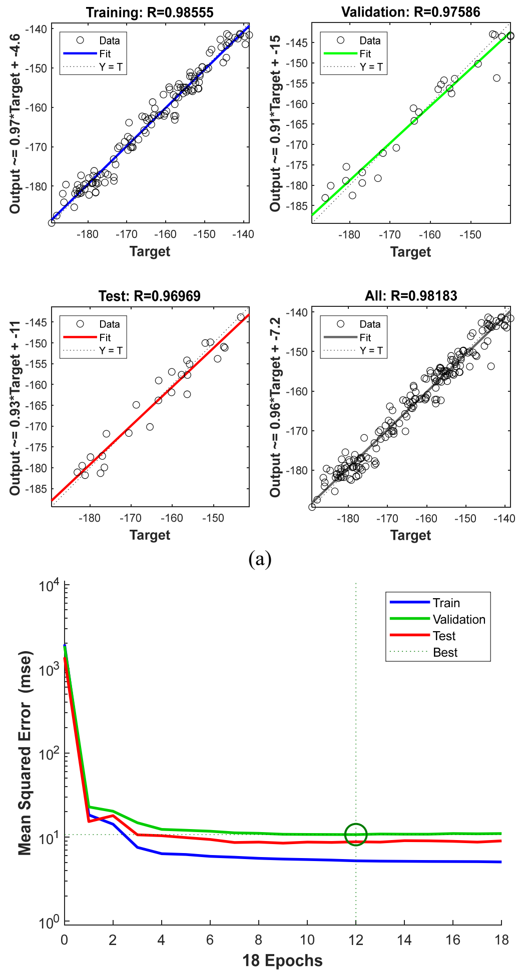
Training algorithm	Training	Test	All
BR	0.9860	0.97314	0.98421

TABLE XV
DETAILS OF VNN MODEL REGRESSION RESULTS WITHOUT APPLICATION OF LR AND WITHOUT EARLY STOPPING TRAINING TECHNIQUES EMPLOYING THE LM TRAINING ALGORITHM

Training algorithm	Training	Test	Validation	All
LM	0.97511	0.97913	0.97179	0.97480

TABLE XVI
DETAILS OF VNN MODEL REGRESSION RESULTS WITHOUT APPLICATION OF LR AND WITHOUT EARLY STOPPING TRAINING TECHNIQUES EMPLOYING THE BR TRAINING ALGORITHM

Training algorithm	Training	Test	All
BR	0.99408	0.97222	0.99103



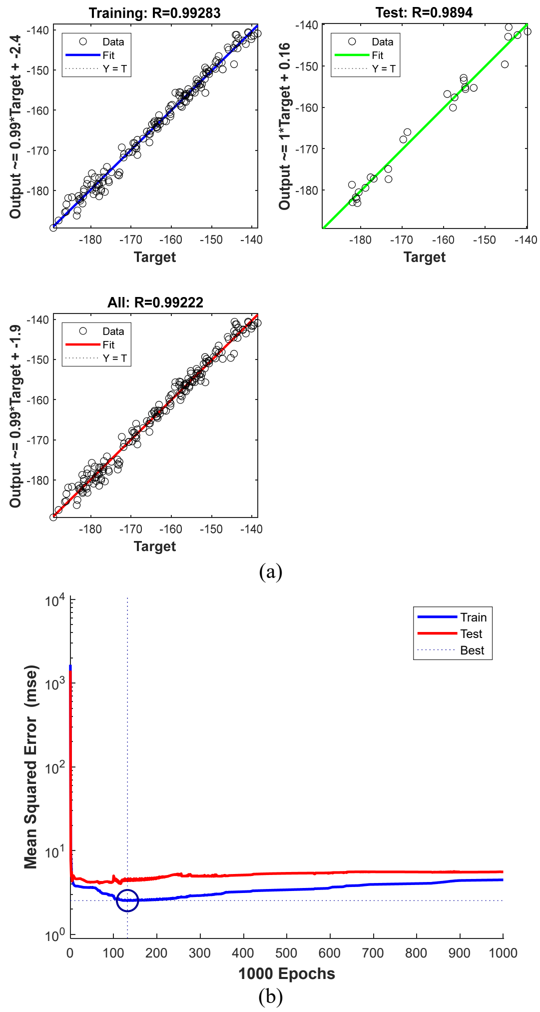
Figs. 1. Best VNN model training and prediction (a) regression (b) performance MSE graphs training with 0.002 LR employing the LM training algorithm and on application of the early stopping training technique

IV.3. Generalization Ability of the VNN Model from the Results Analysis and Graphical Representations

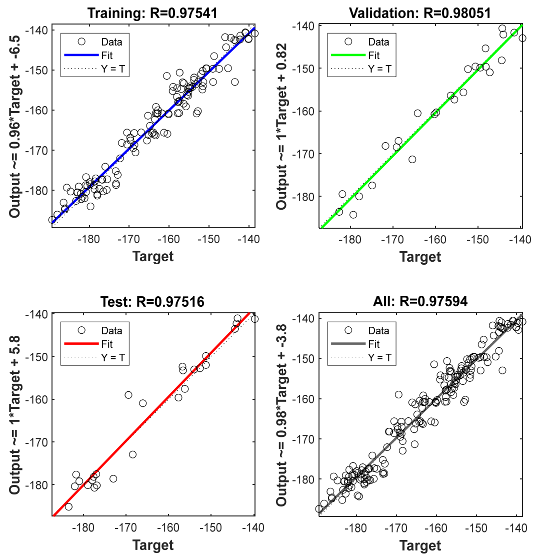
From the VNN model training result outputs of the various investigated instances, the overall training and prediction results from each of the training instances, as recorded in various tables, show that in training the VNN model, a small LR hyper parameter is required. The best VNN model training and prediction results were all gotten at LRs of 0.002, both with training employing LM, BR, and early stopping training techniques and without training using early stopping training techniques.

On gradual increment of the LR value during the VNN model training, there was a decline in the predictive ability of the VNN model, as observed clearly in the regression results as well as in other utilized indices of the first-order statistical performance. Optimal prediction results were seen on application of a small LR of 0.002, on application of the early stopping training technique, and on application of the BR training algorithm, while the worst prediction results were generally recorded for all instances on application of a LR of 1.00.

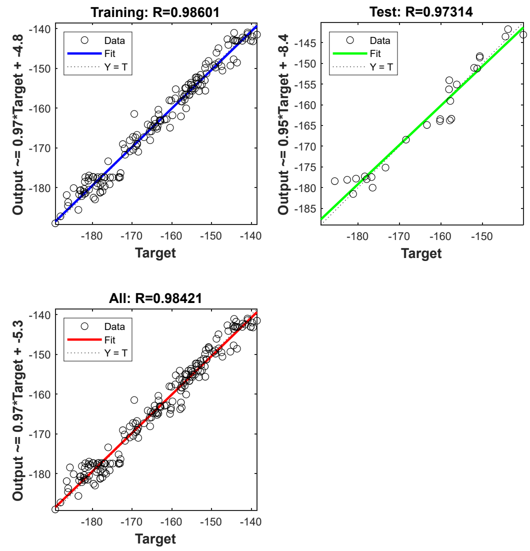
The graphs of the best VNN model training results on application of LR and early stopping training technique using LM and BR training algorithms are shown in Figs. 1 and 2, respectively, and on training the VNN model with LR but without early stopping training technique, the graphs are shown in Figs. 3 and 4 using LM and BR training algorithms, respectively.



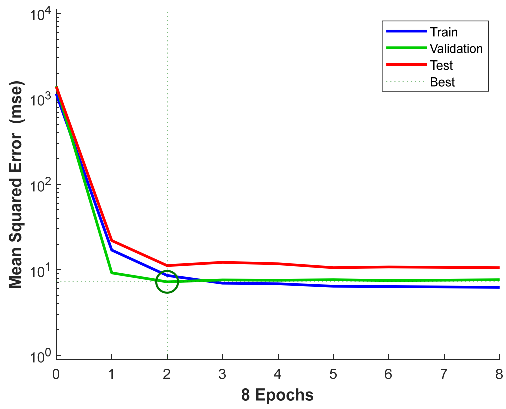
Figs. 2. VNN model training and prediction (a) regression (b) performance MSE graphs training with 0.002 LR employing the BR training algorithm and on application of the early stopping training technique



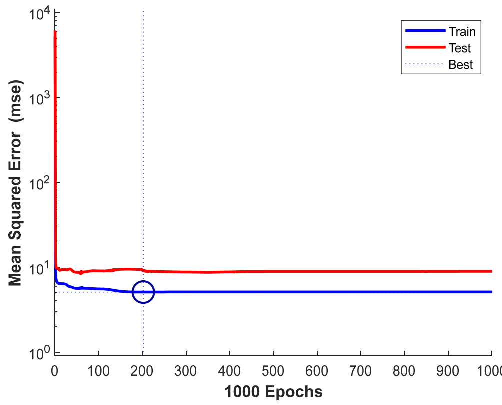
(a)



(a)



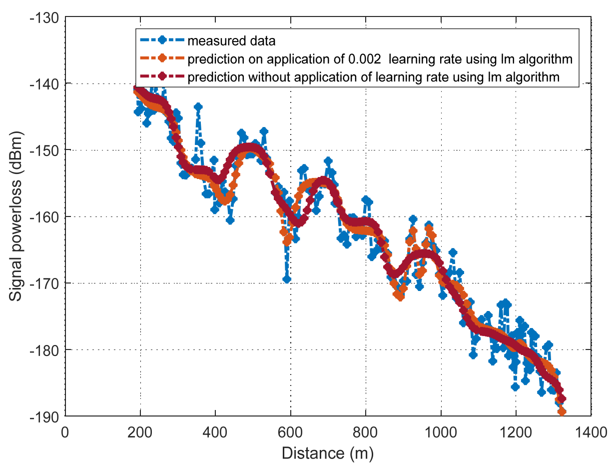
(b)



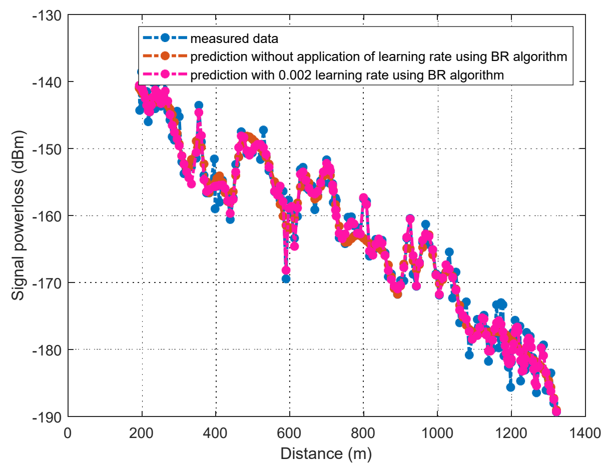
(b)

Figs. 3. VNN model training and prediction (a) regression (b) performance MSE graphs training without LR employing the LM training algorithm and on application of the early stopping training technique

Figs. 4. VNN model training and prediction (a) regression (b) performance MSE graphs training without LR employing the BR training algorithm and on application of the early stopping training technique

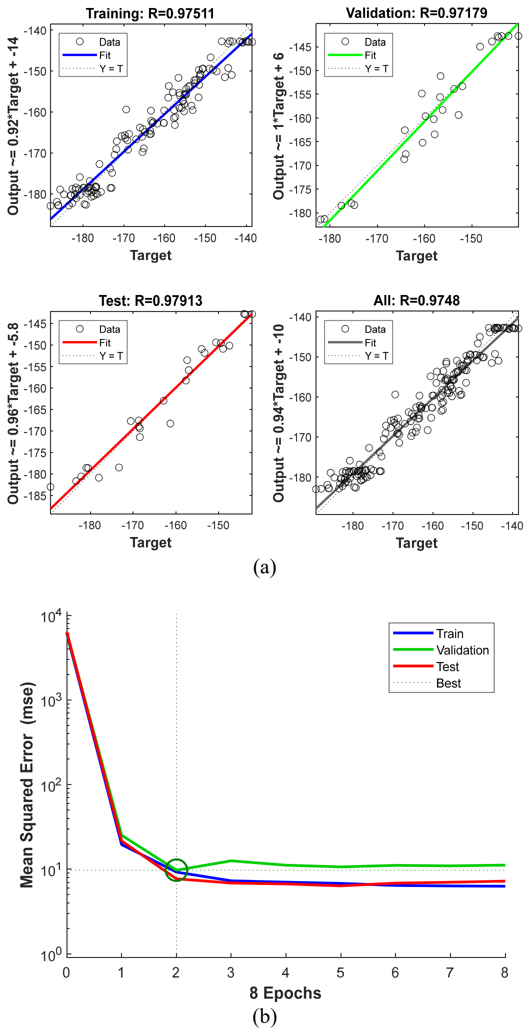


(a)

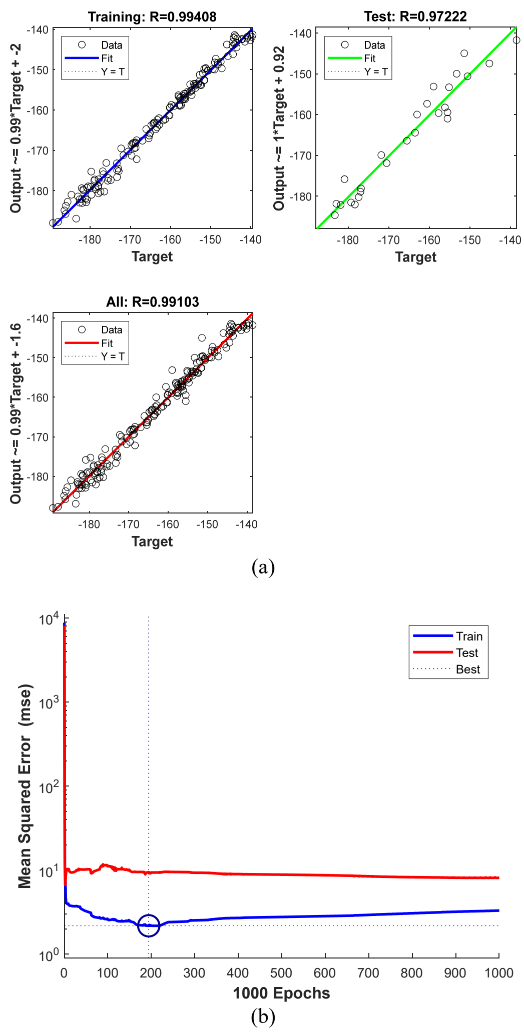


(b)

Figs. 5. Prediction graphs of (a) training with a LR of 0.002 and without application of LR using the LM training algorithm and (b) training with a LR of 0.002 and without application of LR using the BR training algorithm



Figs. 6. VNN model training and prediction (a) regression (b) performance MSE graphs training without LR employing the LM training algorithm and without application of the early stopping training technique



Figs. 7. VNN model training and prediction (a) regression (b) performance MSE graphs training without LR employing the BR training algorithm and without application of the early stopping training technique

The prediction without the application of the LR hyper-parameter using the BR training algorithm represented in brown color in Fig. 5(a) has a more progressive prediction of the measured dataset, i.e., the signal power, in comparison to the prediction on application of the best LR of 0.002 and on non-application of LR employing LM represented in brown and dark red colors, respectively, in Fig. 5(a). This buttresses the fact that the BR training algorithm is also a very excellent VNN training technique that minimizes over-fitting during neural network training and ensures a better generalized network for an optimized result.

Examining Figs. 2(a) and (b), Figs. 3(a) and (b), Figs. 4(a) and (b), Figs. 6(a) and (b), and Figs. 7(a) and (b), the performance *MSE* for all the VNN training employing the BR training algorithm is less, or approximately 101, while the performance *MSE* is more, or approximately 103, for all the VNN training using the LM training algorithm. The performance *MSE*, when closer to zero, indicates a well-trained neural network model. The worst VNN model training among the graphs is seen in Figs. 6(a) and (b), which have a regression result of 0.9748 and a performance *MSE* close to 10^4 on the VNN model training using the LM training algorithm without the application of the LR hyper-parameter and without the application of the early stopping training technique. For effective neural network training and prediction optimization for improved result outputs with minimal prediction error, the application of an adequate LR hyper parameter is of utmost importance. The VNN model shows good prediction results on application of very small LR, so huge LR leads to over-learning of the neural network model. Of importance is also the good selection of the training algorithm and the training techniques for a particular underlying problem for efficient network optimization. If the LM training algorithm is to be used for VNN model training, efficient hyperparameters should be tuned to balance its impediments to ensure optimal prediction results.

Additionally, the BR training algorithm from the training results is also an excellent training technique that performs well in the absence of other training techniques, such as the application of early stopping techniques during neural network training, as has been proven by the prediction result outputs from this research work.

V. Conclusion

The authors examined the effect of the LR hyper parameter and early stopping training technique in training a VNN model for an excellent generalized neural network that enhances prediction of signal power. The impact of two training algorithms, the LM and the BR training algorithms, as well as the early stopping training technique for efficient and optimized network prediction, were all examined.

Results showed an optimized VNN model prediction of the measured dataset on application of a small LR of 0.002. A progressive increase in LR leads to over-learning of the neural network and thus poor network generalization.

The early stopping training technique is a training technique that enhances the VNN model's training efficiency when efficiently applied. The training result outputs on training with the early stopping training technique and without training with the early stopping training technique demonstrated improved VNN model prediction ability on application of the early stopping training technique. However, the application of the BR training algorithm even without the early stopping training technique shows good VNN model prediction ability. This demonstrates that the BR training algorithm is also an excellent training technique that enhances network generalization by reducing network over fitting during neural network training.

Application of adequate LR hyper-parameter, early stopping training technique, and employing BR training technique ensure optimal prediction results with very high R results, low performance *MSE*, and low *RMSE* in comparison to the VNN model training using LM training algorithm, without application of early stopping training technique and without LR hyper-parameter, or on application of a high LR parameter, which leads to network over-learning and over-fitting.

Future works will examine the impact of other neural network hyper-parameters such as input delay, spread factor, etc. in optimizing the prediction efficiency of neural network models for efficient prediction of signal power with minimal error.

References

[1] 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," *International Journal of Applied Engineering Research* vol. 13, pp. 13409-13420, 2018.

[2] M.M. Rathore, A. Paul, S. Rho, M. Khan,. S. Vimal,. S.A.Shah," Smart traffic control: Identifying driving-violations using fog devices with vehicular cameras in smart cities". *Sustain Cities Soc*., vol.71, pp.102986, 2021.

[3] J. Isabona, A.L. Imoize, S. Ojo, O. Karunwi, , Y. Kim, L. Cheng-Chi, L. Chun-Ta, "Development of a Multilayer Perceptron Neural Network for Optimal Predictive Modeling in Urban Microcellular Radio Environments" *Appl.Sci*. vol. 12, pp.1-27, 2022.

[4] Ebhota, V., Srivastava, V., Prediction of Signal Power Loss (at Micro-Cell Environments) Using Generalized Adaptive Regression and Radial Basis Function ANN Models Built on Filter for Error Correction, (2021) *International Journal on Communications Antenna and Propagation (IRECAP)*, 11 (2), pp. 94-105.
doi:https://doi.org/10.15866/irecap.v11i2.19964

[5] Jiakai Wei, "Forget the learning rate, decay loss," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 3, pp. 267-272, June 2019.

[6] C. Nguyen, A.A. Cheema, "A deep neural network-based multi-frequency path loss prediction model from 0.8 GHz to 70 GHz". *Sensors*, vol. 21, pp5100, 2021.

[7] K.C Igwe, O.D. Oyedum, A.M. Aibinu, M.O. Ajewole, and A.S.Moses, "Application of artificial neural network modeling techniques to signal strength computation" *Journal of Heliyon*, vol. 7, pp. 1-9, 2021.

[8] O. U. Anyama and C. P. Igiri, "An application of linear regression & artificial neural network model in the NFL result prediction," *International Journal of Engineering Research & Technology,* vol. 4, pp. 457-461, 2015.

[9] C. P. Igiri and E. O. Nwachukwu, "An improved prediction system for football a match result," *Journal of Engineering,* vol. 4, pp. 12-20, 2014.

[10] Virginia C. Ebhota and Viranjay M. Srivastava, "Enhanced Error Reduction of Signal Power loss during Electromagnetic propagation; An Architectural Composition and Learning Rate Selection", *Journal of Communication (JCM)*, vol. 16, no 10, 2021.

[11] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *Neural Networks:Tricks of the Trade,* vol. 7700, pp. 437-478, 2012.

[12] H. White, "Learning in neural networks: A statistical perspective," *Neural Computation,* vol. 1, pp. 425- 464, 1989.

[13] Virginia C. Ebhota and Viranjay M. Srivastava, "Performance Comparison of Non-Linear Median Filter Built on MLP-ANN and Conventional MLP-ANN: Using Improved Dataset Training in Micro-Cell Environment" *Journal of Communication (JCM)*, vol.16, no 11, pp.508-515, 2021.

[14] J. Isabona and V. M. Srivastava, "Hybrid neural network approach for predicting signal propagation loss in urban microcells," *Proceedings of the 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Agra, India, 2016.

[15] V. C. Ebhota, J. Isabona, and V. M. Srivastava, "Environment-adaptation based hybrid neural network predictor for signal propagation loss prediction in cluttered and open urban microcells," *Wireless Personal Communication,* pp. 1-14, 2018.

[16] A. Neskovic and N. Neskovic, "Microcell electric field strength prediction model based upon artificial neural networks," *International Journal Electronics and Communication,* vol. 64, pp. 733-738, 2010.

[17] P. S. Sotirios and K. Siakavara, "Mobile radio propagation path loss prediction using artificial neural networks with optimal input information for urban environments," *International Journal of Electronics and Communications* pp. 1453-1463, 2015.

[18] 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," *16th International Conference on Sciences and Techniques of Automatic Control & Computer Engineering* pp. 793-798, 2015.

[19] V. C. Ebhota, J. Isabona, and V. M. Srivastava, "Signal power loss prediction based on artificial neural networks in microcell environment,," *3rd IEEE International Conference on Electro-Technology for National Development (IEEE NIGERCON)*, Owerri, Nigeria, 7-10 Nov. 2017, pp. 250-257.

[20] Virginia C. Ebhota, Joseph Isabona, and Viranjay M. Srivastava, "Improved adaptive signal power loss prediction using combined vector statistics based smoothing and neural network approach," *Int. J. on Progress in Electromagnetics Research C (PIER C*), vol. 82, pp. 155-169, March 2018.

[21] Joseph Isabona and Viranjay M. Srivastava, "Coverage and link quality trends in suburban mobile broadband HSPA network environments," *Wireless personal Communication (WPC),* vol. 95, no. 4, pp. 3955-3968, Aug. 2017.

[22] Zhang, S., Sakulyeva, T., Pitukhin, E., Doguchaeva, S., Neuro-Fuzzy and Soft Computing - A Computational Approach to Learning and Artificial Intelligence, (2020) *International Review of Automatic Control (IREACO)*, 13 (4), pp. 191-199. doi:https://doi.org/10.15866/ireaco.v13i4.19179

[23] Anggriawan, D., Amsyar, A., Firdaus, A., Wahjono, E., Sudiharto, I., Prasetyono, E., Tjahjono, A., Real Time Harmonic Load Identification Based on Fast Fourier Transform and Artificial Neural Network, (2021) *International Review of Electrical Engineering (IREE)*, 16 (3), pp. 220-228. doi:https://doi.org/10.15866/iree.v16i3.18131

Authors' information

Department of Electrical and Electronic Engineering Science, University of Johannesburg, P.O. Box 524, Auckland Park, 2006, Johannesburg, South Africa.

E-mails: engrchikaugwu@yahoo.com
tshongwe@uj.ac.za

Dr. **Virginia C. Ebhota** received a Bachelor of Engineering degree in Electronic Engineering from the University of Nigeria, Nsukka (UNN), a Master of Engineering from the 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 a post-doctoral research fellow with the department of electrical and electronic engineering at the University of Johannesburg, South Africa. She has published over seventeen (17) peer-reviewed articles in journals and at IEEE conferences. Her major research area is system modeling using artificial intelligence (AI), such as artificial neural networks, etc.

Thokozani Shongwe received the B. Eng degree in Electronic Engineering from the University of Swaziland, Swaziland, in 2004; the M. Eng degree in Telecommunications Engineering from the University of the Witwatersrand, South Africa, in 2006; and the D. Eng degree from the University of Johannesburg, South Africa, in 2014. He is currently an Associate Professor of Telecommunications and the Head of the School of Electrical and Electronic Engineering at the University of Johannesburg. He is a recipient of the 2014 University of Johannesburg Global Excellence Status (GES) award, which was awarded to him to carry out his postdoctoral research at the University of Johannesburg. In 2016, Prof. T. Shongwe was a recipient of TWAS-DFG Cooperation Visits Programme funding to do research in Germany.

Other awards that he has received in the past are: the post-graduate merit award scholarship to pursue his master's degree at the University of the Witwatersrand in 2005, which is awarded on a merit basis; In 2012, Prof. Shongwe (and his co-authors) received an award for the best student paper at the IEEE ISPLC 2012 (power line communications conference) in Beijing, China. Prof. T. Shongwe's research fields are digital communications and error-correcting coding. His research interests are in power-line communications, cognitive radio, smart grids, visible light communications; machine learning; and artificial intelligence.

Prof. T. Shongwe is the co-founder of a research group at the University of Johannesburg called Artificial Intelligence for Electrical Engineering Applications (AI for EE Applications). This research group is currently composed of five staff members, two postdoctoral researchers working in the fields of communications and machine learning, five doctoral students, and ten master's degree students working in the fields of power line communications, visible light communications, application of ML in PLCs, power systems, agriculture, and object tracking.