Microstrip Line Fed Dielectric Resonator Antenna Optimization using Machine Learning Algorithms

Om Singh1, Manjula R. Bharamagoudra2, Harshit Gupta3, Ajay Kumar Dwivedi4,Pinku Ranjan3, Anand Sharma1\*

1Department of Electronics and Communication Engineering, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, Uttar Pradesh, India

2Department of Electronics & Communication Engineering, School of Electronics and Communication, REVA University, Bangalore, India

3Atal Bihari Vajpayee-Indian Institute of Information Technology and Management,

Gwalior, India

4Department of Electronics & Communication Engineering, Nagarjuna College of Engineering and Technology, Bangalore, India

Email: omsingh6903@gmail.com, manjularb@rediffmail.com, mtcn\_202005@iiitm.ac.in, er.ajaydwivedi@gmail.com, pinkuranjan@iiitm.ac.in, **anandsharma@mnnit.ac.in\***

**Abstract: In this communication, a microstrip line fed dielectric resonator antenna is optimized using various Machine learning-based models. Different ML algorithms such as ANN (artificial neural network), KNN (K-Nearest Neighbors) , XG Boost (Extreme Gradient Boosting), Random Forest, and Decision Tree are used to optimize the proposed antenna design within the frequency band 3.3-3.65 GHz. |S11| of the proposed antenna is predicted by using various ML algorithms. Dataset for the same is created through HFSS EM (Electromagnetic) simulator by varying the radius, height of DRA (Dielectric Resonator Antenaa) as well as the width of microstrip line and conformal strip. Predicted results from all these models are quite close to the actual one except ANN. To overcome the problem of ANN, Knowledge-Based Neural Network techniques (KBNN) are implemented. All these ML algorithms are authenticated by practically constructing and measuring the proposed antenna. Fabricated antenna results have a good agreement with the values predicted by ML algorithms.**

**Keywords- Artificial neural network (ANN), Random Forest, XG boost, K nearest neighbor (KNN), Knowledge-based neural network (KBNN), Dielectric Resonator Antenna, Microstrip Line**

1. **Introduction**

An antenna is one of the most important component of a wireless communication system. In the modern cellular communication world, an efficient antenna is required which can support high data rate, wider bandwidth, and high-frequency range [1]. For the designing of such an efficient radiator, its different design parameters must be fully optimized. HFSS and CST-MWS are EM simulators, which are widely used for the said purpose. But, the problem with these simulators is that it takes lot of time while simulating the process. The time required for simulation increases exponentially if the value of frequency is high. For lower frequency values such as sub-6 or below 10 GHz simulation is less, but for mm-wave and for frequency range more than 20GHz it will late large amount of time. To overcome the problem of more computational time required by simulation tools, various ML models come into the picture for forecasting the antenna performance at a certain frequency with a very less amount of time [2].

In the open literature, some researchers have focused on this problem and predicted the antenna performance through different ML algorithms. Gao et.al. utilized the Gaussian process, as well as support vector machine for optimization of Yagi Uda, shaped printed antenna and dual-mode printed antenna [3]. Dadashzadeh et.al. obtained the optimum position of shorting pin in a microstrip array through a clustering approach for getting desired polarization characteristics [4]. Lee et.al. proposed the optimization of mutual coupling parameters through ANN in the case of antenna arrays [5]. Sharma et.al. proposed the optimization of T-shaped monopole through LASSO (Least Absolute Shrinkage and Selection Operator), ANN, and KNN techniques. 450 data set points have been taken to train the model within two different frequency bands i.e. 2.5 GHz and 5.5 GHz [6]. J. P. Jacobs proposed the optimization of a U-shaped slot-loaded microstrip antenna through Gaussian Process Regression based ML algorithms. The aforementioned antenna supports dual frequency bands i.e. 2.75 GHz and 3.75 GHz respectively [7]. Sharma et.al. proposed a compact dual notch loaded patch antenna, which is optimized through Gaussian Process Regression(GPR) based ML model [8].

In this article, the microstrip line fed dielectric resonator antenna is optimized through different ML models such as ANN, KNN, random forest, XG Boost, and decision tree. Two important aspects of the proposed article are: (i) the ML models are applied for the optimization of dielectric resonator antenna, it is the first time; and (ii) the problem associated with ANN is removed by using the KBNN algorithm. The data set is created through the HFSS EM simulator by varying the radius, the height of DRA as well as the width of the microstrip line, and the conformal strip on DRA. Finally, the predicted data is compared with the measured result. The proposed antenna shows a good predicted value within the frequency range of 3.3-3.65 GHz. For better understanding, the proposed article is divided into subsections: Antenna geometrical layout; antenna optimization through ML models; experimental verification, and final conclusion.

1. **Antenna Geometrical Layout**

The proposed antenna structure is shown in Fig.1. Alumina-made cylindrical ceramic is fed through a microstrip line along with a vertical conformal strip. The permittivity of cylindrical ceramic is 9.8. Microstrip line has been printed over FR4 substrate having permittivity 4.4. 3D view and feeding structure of fabricated antenna are shown in Fig. 2. The optimized dimension of the proposed radiator is given in the caption of Fig.1.

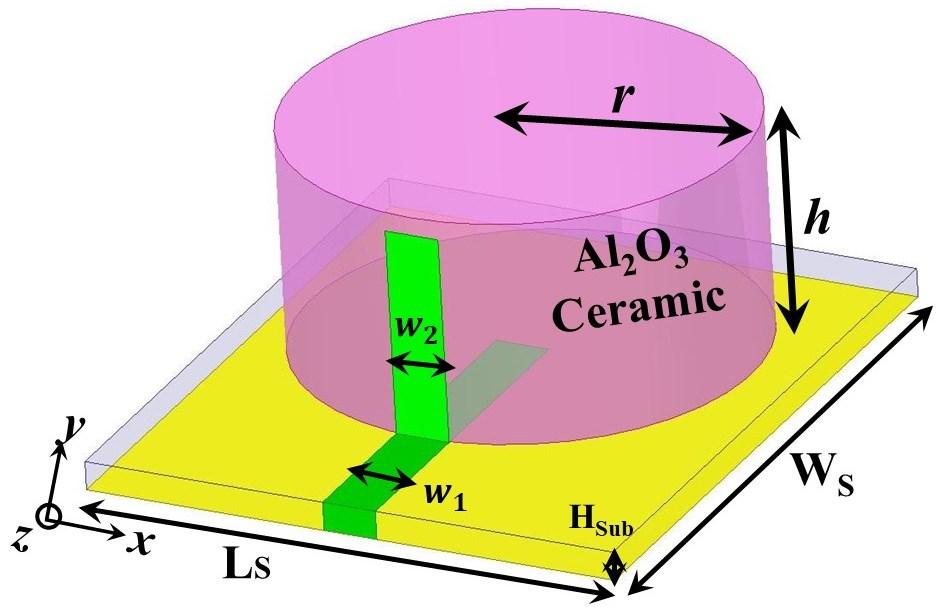


Fig. 1 Microstrip Line fed cylindrical DRA; LS= 30mm; WS= 35mm; w1=3.0 mm; w2=3.0 mm; HSub=1.6 mm; r=12.5 mm; h=12.5 mm

|  |  |
| --- | --- |
| C:\Users\hp\Downloads\IMG-2410.jpg  (a) | C:\Users\hp\Downloads\IMG-2419.jpg  (b) |

Fig. 2 Pictures of Fabricated Antenna (a) Feeding Structure (b) 3D View

1. **Optimization of Proposed Antenna through ML Models**

For the optimization through ML models, cylindrical DRA has been designed on the Ansys HFSS EM simulator. Few design parameters, which directly affect the reflection coefficient of proposed DRA, are chosen such as height, the radius of DRA, w1 (width of the feed line), and w2 (width of conformal strip). The range and step size selected for *h, r, w1 and w2* are 13mm -14mm: 0.2mm, 13mm-14.2mm: 0.4mm, 2mm-3mm: 0.5mm, 2mm-3mm: 0.5 mm respectively. These ranges are selected by considering the optimized values which were obtained from HFSS optimetrics and the proposed dimensions.

216 parameter combinations are generated with these ranges, which mean HFSS will have to perform 216 parameter operations, which increase the time complexity of HFSS. This process of generating the dataset with such minor iteration count itself shows the inefficiency of using such simulation software for parameter explorations or for complex antenna design optimization. An ML model requires a dataset for its training, on the basis of whether the dataset is labeled or labeled these models can be categorized into supervised and unsupervised models. The dataset which is generated in our study is labeled, so the model undergoes supervised learning. Learning good values for all the weights and the bias from labeled samples is all that training a model entails. A machine learning algorithm generates a model in supervised learning by studying numerous examples and tries to find a model that minimizes loss; this process is known as empirical risk minimization. For Machine Learning models, the generated training dataset has to be separated into training and test sets. This separation is helpful in testing the models for the test data set on which it is never trained on.

It helps in better evaluation of the prediction efficiency of each. For each model, randomly 30% of samples are picked up for a test-set. Later, the training data frames are utilized for the training of each model. The value of K = 5 was chosen to minimise the cross-validation error when using the sklearn python library to implement KNN, Decision-Tree, and Random-Forest. The training process of these models is quick and simple.

Implementation of ANN has been done using *keras* library, which internally uses *tensor-flow* library. There is one input layer with the dimension of 5, as we have five input features for the model with a ‘relu’ activation function. 2 hidden layers are added with the same ‘relu’ activation function. The output layer of unit-1 provides the final result along with the Adam optimizer. Mean squared error (MSE)is used as the loss function for the model. Fig 3 represents the layered structure of ANN model.

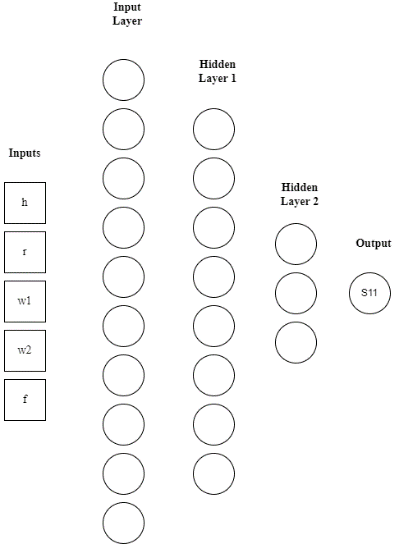


Fig 3: Input, Output and Hidden Layers of ANN

Training of ANN models involves certain hyper-parameters (parameters which control the learning of a model), two of them are epochs and batch-size. Epocs represent the number of iterations or passes of the model through the entire dataset and batch-size is the size of the training data which have to be worked upon before updating the model’s internal parameters.

Our is carried on for 100 epochs with a batch size of 10. Training of an ANN model is the time-taking process for machines with less compute potential. The training process is also analyzed to have a clear picture of how the model trains itself with the input training data set. Fig. 4 (a) and Fig. 4 (b) display the model loss and model validation loss variation for the ANN model. From Fig. 4 (a) and 4 (b), it can be observed that model loss and model validation loss decrease as the number of epochs increases.

|  |  |
| --- | --- |
| (a) | (b) |

Fig. 4 Loss variation for ANN Model with epoch (a) Model Loss variation (b) Model Validation Loss

The training loss is a metric that measures how well a deep learning model matches the training data. It evaluates the model's error on the training set. The training loss is calculated computationally by adding the sum of mistakes for each sample in the training set, and it is measured after each batch. The training loss is commonly illustrated by plotting a curve. Validation loss is a statistic for evaluating a deep learning model's performance on the validation set. The validation set is a subset of the dataset put aside to test the model's performance. The validation loss is determined from the sum of the mistakes for each sample in the validation set, similar to the training loss. After each epoch, the validation loss is also calculated. This indicates whether the model requires additional tuning or tweaks. To do so, a learning curve for the validation loss is frequently shown.

After preparing the data set, each ML model can now predict the values of |S11| for a given antenna parameter in a particular frequency band. It’s better to test the model with a data set with which it is not trained. By doing so, we can understand its precision better and analyze the prediction accuracy. If the predictions made are not up to the mark, then any of the following could be the case:

* The training data could be inappropriate or of poor quality.
* A model might require hyper-parameter tuning or some cross-validation.
* The choice of the input features could be wrong.

There might be other reasons even, but for our use case, we tried working on the above-mentioned issues in case of poor model prediction. Prediction metrics provide evaluative and comparative numbers, which helps in opting out the best possible algorithm for a specific use case. The two most relied metrics are the R2-score and MSE (mean square error) [9]:

n = number of data points

= observed values

= predicted values

Table-1 contains these accuracy metrics for predictions made by each ML technique. To obtain these values, “*sklearn metrics”* has been used. These metrics provide us a picture of the prediction effectiveness and help in improving the training dataset or tuning the model. Table 1 shows that the ANN model has the highest error i.e 35.34, whereas the XG Boost model has the lowest error with 13.79. Fig. 5 shows the comparison of predicted and the actual result obtained from different ML models i.e. ANN, KNN, XGBoost, Random Forest and Decision Tree. In Fig. 5, the X = Y line represents the ideal case which means the density of points near the line presents a better accuracy result. From Fig. 5, it can be said that there is a close relationship between actual and predicted values for each ML model except ANN. It can also be observed that for smaller values of S11 ML models perform well. But, for the values of |S11| near 40-50 dB, there is a minor deviation from the actual values. The reason for better model accuracy for lower negative values of |S11| could be the simpler relation between the parameters and S11 for lower negative values as compared to higher values.

Table-1 Prediction Metrics Obtained from different ML Models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | R2 Score | Mean Absolute Error | Mean Squared Error | Maximum Error |
| ANN | 0.94 | 0.42 | 1.02 | 35.34 |
| KNN | 0.98 | 0.11 | 0.31 | 30.51 |
| Decision Tree | 0.97 | 0.16 | 0.51 | 29.12 |
| Random Forest | 0.99 | 0.13 | 0.32 | 30.03 |
| XG Boost | 0.98 | 0.18 | 0.33 | 13.79 |

|  |  |
| --- | --- |
| (a) | (b) |
| (c) | (d) |
| (e)  Fig. 5 Comparison of Predicted value with actual one (a) ANN (b) Decision Tree (c) KNN (d) Random Forest (e) XG Boost  Table-2 Five Different Random Data Samples for comparison of HFSS results with results predicted from ML models | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sample | Height (mm) | Radius (mm) | Freq (GHz) | W1 (mm) | W2 (mm) |
| D1 | 14.10 | 14.55 | 2.42 | 2.25 | 2.6 |
| D2 | 13.04 | 13.30 | 1.84 | 2.3 | 2.7 |
| D3 | 13.19 | 13.74 | 1.95 | 2.7 | 2.45 |
| D4 | 14.42 | 13.12 | 2.41 | 2.4 | 2.8 |
| D5 | 13.86 | 14.57 | 2.56 | 2.8 | 2.1 |

Now, to validate the different ML models, randomly take five different datasets and compare the result with HFSS EM simulator. Table-2 lists five different data sets for comparing HFSS obtained value with predicted values from different models. These samples are chosen such that the S11 values are in moderate ranges for testing purposes. Fig. 6 shows the result assessment from HFSS and that from the machine learning models (for all five dataset listed in Table-2). It is clearly observed that the models are found to be acting well. It is perceived from Fig. 6 that the error percentage is in the range 0-7% for all the models.

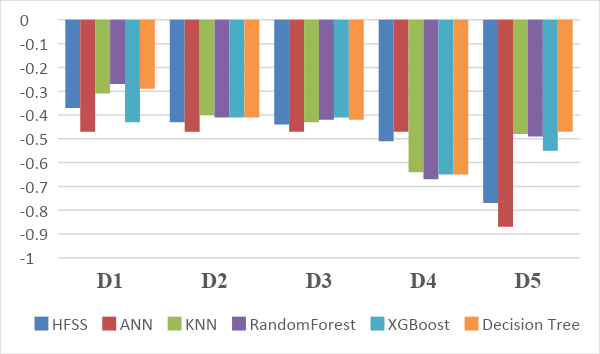


Fig.6 Graph representing the S11 comparison for each Dataset

The accuracy and prediction capability of ANN should be way more than other implemented ML algorithms, however, in the accuracy analysis, the results are not as expected. Primarily, the low accuracy results of ANN are a reason for less training data-set on which the model was trained. ANN can map complex data relations because of different implied layers. But, the data required for training should have large samples. Generating large training samples through HFSS is not a feasible job. As each iteration requires heavy computation, it makes the process highly inefficient.

However, when the training data is difficult to obtain, the design of experiment (DOE) techniques can also be implied [10], but the data requirement could still be too large. To tackle such issues, some existing models are required. It requires additional prior knowledge and is known as knowledge-based neural network (KBNN). This prior knowledge can be implemented using two different methods: *prior knowledge input* (PKI) and *source difference* [10]*.* In the PKI method, the ANN is trained to map the same input-output relation but along with the training data and other inputs are also added. Additional data is generated using the empirical models, so for PKI the additional training data is generated using *KNN, Decision-Tree, Random-Forest and XGBoost* models. In the source difference method, the ANN is trained on the error (difference) between the HFSS output and the source-model output. The mapping of this error matrix is simpler thus requiring less training. Considering λ as a design parameter vector, for a particular vector the S (λ) is the S11 obtained from HFSS. Similarly, F (λ) obtained from the empirical model. Further, the difference between both results gives the error matrix:

The ANN is now trained on the E (λ), the final required value of S11 can be obtained by summation of the ANN prediction with the empirical model’s output for a parameter vector:

Training efficiency of KBNN supports high dimensional parametric sweeps and design space exploration. Fig 7 (a) and Fig 7 (b) represent the accuracy plot of PKI and Source Difference KBNN models. From Fig. 7, it is clearly observed that there is good relation between predicted and actual value. Table-3 contains the prediction metrics, and provides a comparison of the accuracy performance. Both the KBNN techniques provide better accuracy results as compared to ANN as evident from its MSE and R2 Score. The accuracy results of both the SD and PKI techniques are in close similarity.

|  |  |
| --- | --- |
| (a) | (b) |

Fig.7 Comparison of Predicted value with actual one (a) KBNN-PKI (b) KBNN-SD

Table-4 MSE and R2 score for ANN, KBNN-PKI and KBNN-SD

|  |  |  |
| --- | --- | --- |
| ML Models | MSE | R2 |
| ANN | 1.02 | 0.94 |
| SD KBNN | 0.51 | 0.97 |
| PKI KBNN | 0.73 | 0.96 |

The training of each model gets compute intensive, however once the model is trained the results could be generated easily without the utilization of much compute resources. So once we have a model for the antenna, any number of design iterations can be done with far too little compute-resource as compared to HFSS simulation as each iteration in this case would require entire simulation of the antenna which will be requiring high compute power and thus will be a time consuming process. Table-4 provides a comparison between algorithms in terms of training time. Neural network based models are taking more time because of their layered structure and a complicated learning algorithm[11]. The source difference model is less intensive than ANN as the training of the difference matrix is simpler and less complicated than actual input-output mapping. The PKI model took most of the training time as it involves training with a larger dataset in order to increase the accuracy.

Table-4 Training time for ML models

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | KNN | Decision Tree | Random Forest | XG Boost | ANN | PKI | Source Difference |
| Training Time (sec) | 0.497 | 0.351 | 1.7 | 2.7 | 400 | 2485 | 200 |

1. **Experimental Outcomes**

In this section, the predicted outcome is compared with measured results. The reflection coefficient (|S11|) of the proposed antenna is measured through Keysight make VNA E5071C. Fig. 8 shows the measured and simulated |S11| variation with respect to frequency. It is clearly observed from Fig. 8 that there is a good correlation between measured and simulated |S11|, and the final dimensions of the fabricated antenna are LS= 30mm; WS= 35mm; w1=3.0 mm; w2=3.0 mm; HSub=1.6 mm; r=12.5 mm; h=12.5 mm. The proposed antenna operates over the frequency range of 3.3-3.65 GHz. In order to find the accountability of resonance at 3.5 GHz, the E-field pattern is displayed in Fig.9. It is perceived from Fig. 9 that HEM11δ mode is generated in cylindrical-shaped ceramic [12]. It is due to the microstrip line feed, which acts as a magnetic dipole [13]. Mathematically, the resonance of the HEM11δ mode can also be verified by using the following empirical formulation [13]:

In eqn. (5), the ‘r’ and ‘h’ is the radius and height of cylindrical ceramic respectively. From eqn. (5), the resonant frequency is found to be 3.35 GHz, which is close to the simulated one.

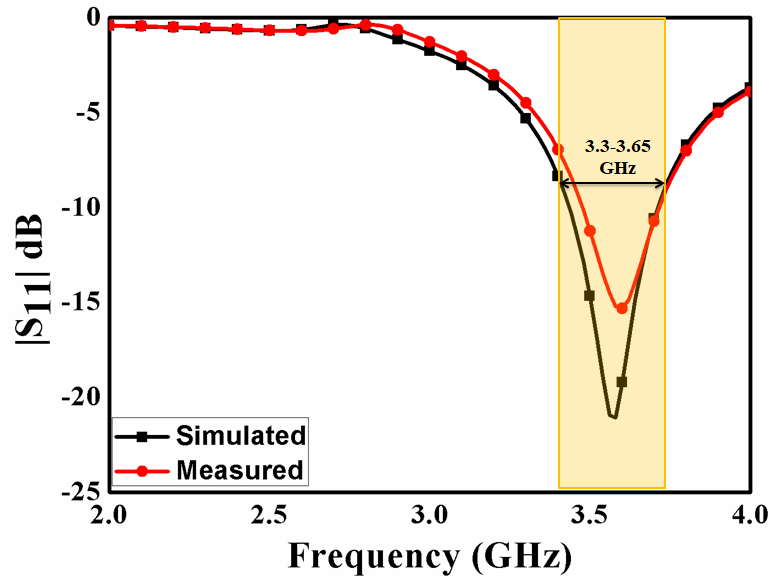


Fig. 8 Measured/Simulated |S11| of the proposed antenna

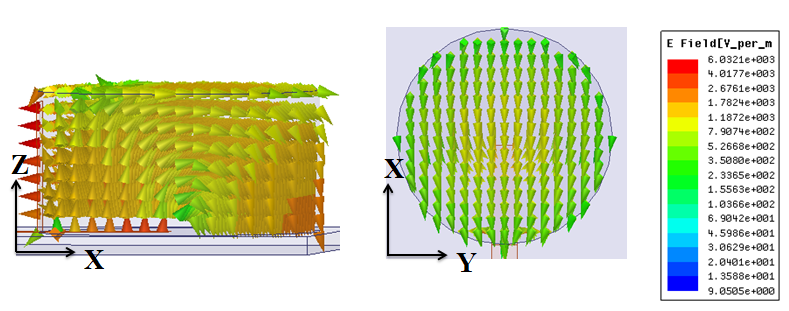


Fig. 9 E-field variation of proposed antenna at 3.5 GHz

Fig. 10 shows the measured/simulated gain variation of the proposed antenna. Gain is measured with the assistance of the two antenna method [14]. From Fig.10, it is observed that the value of gain is about 3.5 dBi within the operating frequency band. Fig. 11 shows the radiation pattern of the proposed antenna in the XZ plane and YZ plane at 3.5 GHz. From Fig. 11, it is clearly observed that broadsided radiation pattern is obtained, which is due to HEM11δ mode in CDRA [12]. There is a good co-pol to cross-pol difference in both the plane.

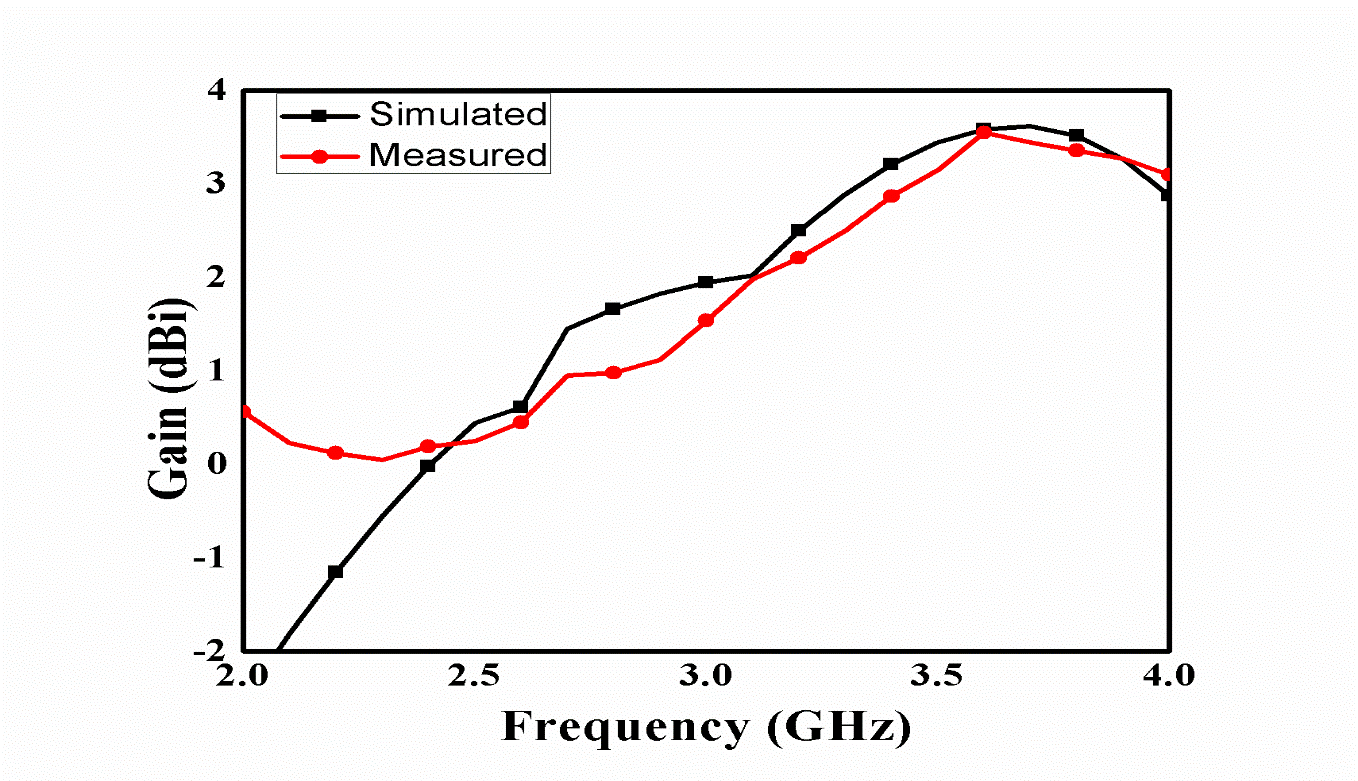


Fig.10 Measured/Simulated Gain variation of proposed antenna

|  |  |
| --- | --- |
| C:\Users\hp\Downloads\E plane co and cross 3.5 GHz.TIF  (a) | C:\Users\hp\Downloads\H plane co and cross 3.5 GHz.TIF  (b) |

Fig. 11 Far-field pattern of proposed antenna at 3.5 GHz (a) XZ plane (b) YZ plane

1. **Conclusion**

In this article, the microstrip line fed dielectric resonator antenna is optimized through various ML algorithms such as ANN, KNN, random forest, and decision tree. It is the first time when ML models are used to optimize the Dielectric resonator-based radiators. Dataset for the same is created through the HFSS EM simulator. The problem associated with ANN is rectified by using a knowledge-based neural network (KBNN). There is a close prediction obtained with these ML models with actual values (obtained from HFSS). Experimental results further validate the ML models. The proposed antenna works within 3.3-3.65 GHz. The optimized design shows stable radiation characteristics within the operating frequency range and confirms its application for the WiMAX frequency band.

**References**

1. Tao Tang, Tao Hong, Cong Liu, “Design of 5G Dual-Antenna Passive Repeater Based On Machine Learning”, 15th International Wireless Communications & Mobile Computing Conference (IWCMC), 2019, DOI: 10.1109/IWCMC.2019.8766614
2. Hilal El Misilmani, Tarek Naous, Sala Al Khatib, “A Review on the Design and Optimization of Antennas Using Machine Learning Algorithms and Techniques”, International Journal of RF and Microwave Computer-Aided Engineering, 30, 2020, 1-7
3. Jing Gao , Yubo Tian , and Xuezhi Chen, “Antenna Optimization Based on Co-Training Algorithm of Gaussian Process and Support Vector Machine” IEEE Access, 8, 2020, 211380- 211390
4. G. Dadashzadeh, M. Kargar, Y. Torabi, and B. Rahmati, “Broad-band and wide scan phased array element design using data mining,” Appl. Comput. Electromagn. Soc. J., vol. 31, no. 3, pp. 244–251, 2016.
5. K.-C. Lee and T.-N. Lin, “Application of neural networks to analyses of nonlinearly loaded antenna arrays including mutual coupling effects,” IEEE Trans. Antennas Propag., vol. 53, no. 3, pp. 1126–1132, Mar. 2005.
6. Yashika Sharma, Hao Helen Zhang, and Hao Xin “Machine Learning Techniques for Optimizing Design of Double T-shaped Monopole Antenna”, IEEE Transactions On Antennas and Propagation, 68, 2020, 5658 – 5663
7. J.P. Jacobs, “Efficient resonant frequency modeling for dual-band microstrip antennas by gaussian process regression” IEEE Antennas Wirel. Propag. Lett. 2015; 14:337–41
8. Kanhaiya Sharma and Ganga Prasad Pandey, “Efficient modelling of compact microstrip antenna using machine learning” AEU-International Journal of Electronics and Communication, 135, 2021
9. S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition. Pearson Education, 2016
10. P. Watson and K. C. Gupta, EM-ANN models for via interconnects in microstrip circuits, IEEE MTT-S Int Microwave Symp Dig, 1996, pp. 1819-1822.
11. Zainud-Deen, S. H., I. I. Ibrahim, Sabry MM Ibrahem, and A. S. Hassan. "Neural network-based CAD model for the design of hemispherical dielectric resonator antenna with a concentric conductor." In Proceedings of the Twenty Third National Radio Science Conference (NRSC'2006), pp. 1-7. IEEE, 2006.
12. D. Kajfez, A.W. Glisson, J. James, Computed Modal Field Distributions for Isolated Dielectric Resonators, IEEE Trans. Microw. Theory Tech. 32 (1984) 1609–1616.
13. R.K. Mongia, P. Bhartia, Dielectric resonator antennas—a review and general design relations for resonant frequency and bandwidth, Int. J. Microw. Millimeter-Wave Comput. Eng. 4 (1994) 230–247.
14. G. A. Stutzman, W.L. and Thiele, Antenna Theory and Design, 2nd ed. Hoboken, NJ, USA: Wiley, 1998.