Machine Learning-Based Antenna Performance Prediction with Data Processing Enhancement

Duong Thi Thanh Tu
Faculty of Telecommunication 1
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
tudtt@ptit.edu.vn

Nguyen Thanh Dat
Faculty of Telecommunication 1
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
DatNT.B21VT119@stu.ptit.edu.vn

Phan Huu Phuc
Faculty of Telecommunication 1
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
PhucPH.B21VT351@stu.ptit.edu.vn

Nguyen Viet Hung
Faculty of Telecommunication 1
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
nvhung_vt1@ptit.edu.vn

Do Nguyen Minh Quan
Faculty of Telecommunication 1
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
QuanDNM.B23VT349@stu.ptit.edu.vn

Tran Phuong Khanh Nhi
Faculty of Telecommunication 1
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
NhiTPK.B25KD074@stu.ptit.edu.vn

Dang Quang Hung
Faculty of Telecommunication 1
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
HungDQ.B23VT189@stu.ptit.edu.vn

Mai Trung Hieu
Faculty of Telecommunication 1
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
HieuMT.B23VT153@stu.ptit.edu.vn

***Abstract*— Artificial intelligence (AI), particularly machine learning (ML), has become a key technology in antenna design due to its ability to model nonlinear relationships between antenna dimensions and performance. ML enables accurate prediction of antenna performance parameters based on input dimensions, reducing optimization time and the number of simulations compared to traditional methods. However, some existing studies still face limitations such as small datasets, high prediction errors, or complex and manual data processing procedures. This study proposes a data processing method on Google Colab to rapidly and accurately construct training datasets for machine learning. Based on a dataset of 15,000 samples created by this method, the Gradient Boosting (GB) model is used to predict antenna performance with a mean squared error (MSE) of 0.2914, demonstrating high prediction accuracy and low error, thereby significantly reduce simulation and antenna design optimization time.**

***Keywords*— *AI, ML, GB, Data processing, Google Colab*.**

I. INTRODUCTION

In recent years, artificial intelligence (AI) has rapidly developed and become a crucial technology in various engineering fields, including antenna design. AI possesses the capability to process and analyze large volumes of data, thereby facilitating the optimization of design processes. ML, an important branch of AI, has emerged as a prominent trend in antenna design. The application of ML enables the modeling of complex nonlinear relationships between antenna dimensions and antenna characteristics. Consequently, ML models can accurately predict antenna characteristics, substantially reducing optimization time and the number of simulations required compared to traditional methods, thus improving accuracy and reducing costs.

Numerous recent studies have applied ML to antenna optimization; however, these works still face certain limitations, such as small dataset sizes [1], [2], [4] or high mean squared error (MSE) [1], [6]. Additionally, some studies involve complex, manual, or multi-step data processing procedures, which are time-consuming and prone to errors [7], [8].

In this research, a data processing method is proposed to construct high-quality training datasets for machine learning. S_{11} simulation results obtained from CST software are exported as .txt files and processed on the Google Colab platform to rapidly and accurately extract antenna dimensions and performance parameters. The final dataset is saved in .csv format. This method not only significantly reduces the time required to build training datasets but also ensures high-quality, consistent input data for ML models. Based on the constructed dataset, a Gradient Boosting model is employed to predict antenna performance parameters from antenna dimensional parameters, achieving a MSE value of only 0.2914. This demonstrates high prediction accuracy with low error, thereby greatly reducing simulation and antenna design optimization time.

II. MACHINE LEARNING-BASED ANTENNA PERFORMANCE PREDICTION

In antenna design, machine learning is increasingly asserting its important role due to its ability to accurately predict antenna characteristics based on antenna dimensions, or conversely, to predict antenna dimensions based on desired performance parameters. However, the process of constructing training datasets is often time-consuming, as traditional simulation data collection and processing typically rely on manual operations, which likely to cause errors.

In this section, in addition to presenting the application of machine learning models for antenna performance prediction, a data processing method from CST software is proposed. S_{11} simulation results are exported as text files and processed on the Google Colab platform to rapidly and accurately extract both input dimensional parameters and performance parameters, which are then saved as .csv files for machine learning training. This method replaces a manual dataset construction process that typically requires several days with an automated process completed in minutes, thereby substantially improving efficiency, data quality, and scalability for machine learning–based antenna design.

A. Antenna design

To construct the dataset for the rectangular patch antenna, CST Microwave Studio simulation software is utilized. The antenna is designed on a commonly used Rogers RO4003C substrate with a dielectric constant of $\varepsilon_r = 3.55$, a loss tangent of 0.0027, and a thickness of $h = 0.8$ mm as shown in Fig.1.

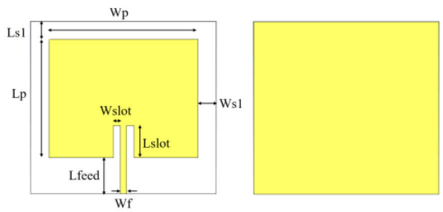

Fig. 1. Rectangular patch antenna

The effective dielectric constant ε_{eff} is calculated as follows [1]:

$$\varepsilon_{eff} = \begin{cases} \frac{\varepsilon_r+1}{2} + \frac{\varepsilon_r-1}{2}\left(\frac{1}{\sqrt{1+\frac{12h}{w}}}\right), \text{for } \frac{w}{h} < 1 \\ \frac{\varepsilon_r+1}{2} + \left(\frac{\varepsilon_r-1}{2\sqrt{1+\frac{12h}{w}}}\right), \text{for } \frac{w}{h} > 1 \end{cases} \quad (1)$$

Where ε_r is the dielectric constant of the substrate material. The width and length of the radiating patch of the rectangular microstrip antenna are determined as follows [2]:

$$W = \frac{c}{2f_0\sqrt{\frac{\varepsilon_r+1}{2}}} \quad (2)$$

$$L = L_{eff} - 2\Delta L \quad (3)$$

Where W and L are the width and length of the radiating patch, c is the speed of light in vacuum and f_0 is the operating frequency. L_{eff} is the effective length, and ΔL is the extended length of the antenna calculated by the following formula [2]:

$$L_{eff} = \frac{c}{2f_0\sqrt{\varepsilon_{ff}}} \quad (4)$$

$$\Delta L = 0.412h\frac{(\varepsilon_{eff}+0.3)\left(\frac{w}{h}+0.264\right)}{(\varepsilon_{eff}-0.258)\left(\frac{w}{h}+0.8\right)} \quad (5)$$

B. Data acquisition

The methodology for constructing the antenna design dataset is illustrated in Fig. 2:

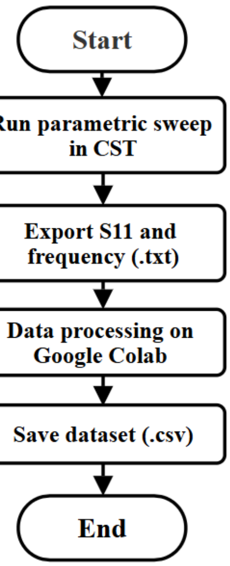

Fig. 2. Data acquisition workflow

The antenna design data is generated using the Parameter Sweep function in CST, as shown in Fig. 3. The Parameter Sweep process systematically varies the antenna dimension parameters within a defined range. After the simulation process, the resulting S_{11} data is presented in Fig. 4:

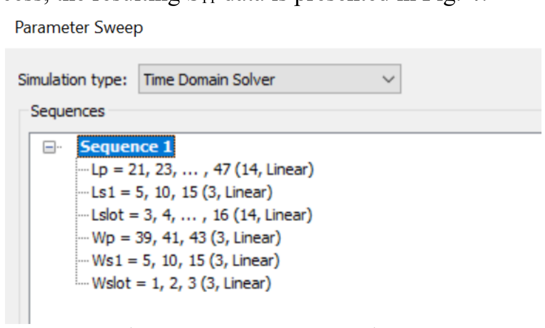

Fig. 3. Parameter Sweep setup in CST

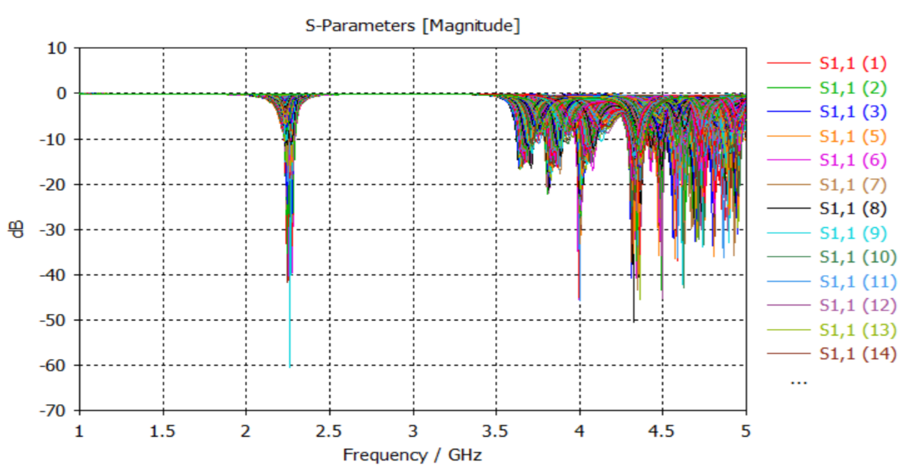

Fig. 4. Simulated S_{11} results

```
*Lp35 - Notepad
File  Edit  Format  View  Help
#Parameters = {Lfeed=7; Lp=35; Ls=40; Ls1=5; Lslot=3; Wfeed=1.72; Wp=39; Ws=44; Ws1=5; Wslot=1;
#"Frequency / GHz"      "S1,1 (1) [Magnitude]"
#----------------------------------------
1.0000000000000 -0.057343943531868
1.0039999485016 -0.057909714732463
1.0080000162125 -0.055595056700367
1.0119999647141 -0.050659304856851
1.0160000324249 -0.043617232800324
1.0199999809265 -0.035175296225349
1.0240000486374 -0.026158591032456
1.0279999971390 -0.017441986084516
1.0319999456406 -0.0098544813024220
1.0360000133514 -0.0041029007752037
1.0399999618530 -0.00071823537271193
1.0479999780655 -0.0019998608122847
1.0520000457764 -0.0065116589676110
1.0559999942780 -0.013100894330289
1.0599999427795 -0.021137696056190
1.0640000104904 -0.029862908939310
1.0679999589920 -0.038450885714412
1.0720000267029 -0.046097485061600
1.0759999752045 -0.052084881339198
1.0800000429153 -0.055867083704605
1.0839999914169 -0.057105532141519
```

Fig. 5. Example of exported S_{11} data from CST

No.	Lp	Ls	Lslot	Wp	Ws	Wslot	f1	s11_1	bw_1	f2	s11_2	bw_2	f3	s11_3	bw_3	f4	s11_4	bw_4
1	21	26	3	39	44	1	3.648	-49.379	0.074	0	0	0	0	0	0	0	0	0
2	21	26	3	41	46	1	3.640	-16.394	0.057	0	0	0	0	0	0	0	0	0
3	21	26	3	43	48	1	0	0	0	0	0	0	0	0	0	0	0	0
4	21	26	3	39	44	2	3.656	-18.331	0.062	0	0	0	0	0	0	0	0	0
5	21	26	3	41	46	2	3.640	-10.164	0.011	0	0	0	0	0	0	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14996	47	62	16	41	56	2	1.696	-16.997	0.016	3.276	-26.148	0.028	3.856	-21.959	0.028	4.452	-17.142	0.042
14997	47	62	16	43	58	2	1.692	-15.469	0.015	3.276	-25.617	0.028	3.692	-22.342	0.026	4.272	-19.657	0.039
14998	47	62	16	39	54	3	1.708	-12.158	0.011	3.260	-12.737	0.023	4.060	-20.148	0.029	4.792	-10.663	0.028
14999	47	62	16	41	56	3	1.704	-11.127	0.008	3.260	-12.679	0.023	3.872	-22.071	0.028	4.576	-11.480	0.032
15000	47	62	16	43	58	3	1.700	-10.301	0.004	3.256	-12.107	0.021	3.708	-21.813	0.025	4.392	-13.196	0.038

Fig. 6. Data processed by proposed method

The S_{11} result data is exported in .txt file format as shown in Fig. 5. Each data sample includes antenna dimension parameters along with the corresponding resonant frequency and reflection coefficient values. The data file is then processed on the Google Colab platform to extract the input dimension parameters and performance metrics for each antenna sample. The complete dataset is subsequently saved in .csv format as shown in Fig. 6, serving as input for machine learning training.

The dataset collected using this method consists of 15,000 samples, with each sample containing six key antenna dimension parameters that directly affect the resonant frequency and bandwidth. Using this dataset, machine learning models are trained to predict antenna performance based on input dimensions, thereby providing highly accurate predictions of resonant frequency and bandwidth, and significantly reducing optimization time compared to traditional methods.

C. Machine learning workflow for antenna performance prediction

In this research, eight different regression models are selected to train and predict antenna performance parameters. These machine learning models are chosen for their ability to model complex nonlinear relationships between independent variables (antenna dimensions) and dependent variables (antenna characteristics).

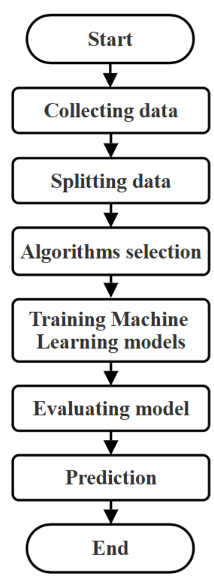

Fig. 7. Machine Learning for antenna performance prediction workflow

The workflow for building machine learning models for antenna performance parameter prediction is detailed in Fig. 7. First, antenna dimension and performance data are simulated and collected using CST Microwave Studio. Next, the dataset is split into two parts: 80% for training and 20% for testing. This split ensures that the model not only memorizes the training data but can also accurately predict unseen samples, reflecting the model's true capability in new antenna designs. Subsequently, regression models including Linear Regression, Support Vector Machine, K-Nearest Neighbor, Decision Tree, Random Forest, Gradient Boosting, Extreme Gradient Boosting, and Artificial Neural Network are applied. Finally, the trained models are evaluated on the test set to select the best model for antenna design.

C. Algorithms selection

To model the complex nonlinear relationships between antenna dimensions and performance parameters, eight different machine learning models were employed: Linear Regression, Support Vector Machine, K-Nearest Neighbor, Decision Tree, Random Forest, Gradient Boosting, Extreme Gradient Boosting, and Artificial Neural Network. Each model has distinct characteristics and strengths in handling nonlinear data, thereby enhancing the accuracy and generalization capability of the prediction system.

- Linear Regression: models proportional relationships between independent and dependent variables using a linear approach; assumes normally distributed errors and homoscedasticity [9].

- Support Vector Machine: effectively models nonlinear relationships by using kernel functions to project data into high-dimensional spaces. It performs well with limited samples and maintains strong resistance to overfitting due to its margin-maximization principle [10].

- K-Nearest Neighbor: determines the label of a new sample based on the K nearest points in the feature space. The label is assigned according to the majority voting principle among these neighbors. The value of K influences model performance: a small K is sensitive to noise, while a large K reduces sensitivity to local data patterns [11].

- Decision Tree: a supervised learning algorithm that utilizes a branching structure to perform classification or regression. Each node represents a data-splitting condition, making the model highly interpretable and intuitive [12].

- Random Forest: an ensemble model that combines multiple decision trees using bootstrap aggregation, which helps reduce variance and improve prediction accuracy [13].

- Gradient Boosting: a sequential training method of multiple weak learners, where each subsequent model corrects the errors of the previous one to minimize the loss function and enhance overall performance [14].

- Extreme Gradient Boosting: an advanced version of Gradient Boosting, capable of handling large-scale data, offering high training speed and superior performance, and is widely applied in both regression and classification tasks [15].

- Artificial Neural Network: a machine learning approach inspired by biological neural networks. Its structure consists of interconnected layers of neurons, where data is propagated from the input layer through hidden layers to the output layer. The model learns by adjusting connection weights to optimize prediction capability and approximate complex nonlinear relationships between input and output [16].

D. Evaluation metrics

To evaluate the performance of machine learning models, this study employs regression metrics including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared (R^2), and Explained Variance Score. These metrics provide a comprehensive assessment of model accuracy, generalization, and fit to simulated data. According to [17], the above evaluation metrics are defined as follows:

- Mean Squared Error (MSE): a widely used regression metric, MSE calculates the average of the squared differences between actual and predicted values. Lower MSE values indicate better model performance. The formula for MSE is expressed as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{6}$$

where: Y_i is the actual value and, $\hat{Y}_i$ is the predicted value for the i-th sample.

Root Mean Squared Error (RMSE) is the square root of MSE, expressing error in the same units as the target variable. It is sensitive to large errors and provides insight into model stability. Lower RMSE values indicate higher prediction accuracy. The formula for RMSE is expressed as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2} \tag{7}$$

Due to its sensitivity to large errors, the RMSE provides a clearer indication of the model's stability. A smaller RMSE value indicates higher prediction accuracy of the model.

Mean Absolute Error (MAE) is the average of the absolute differences between predicted and actual values, treating all errors equally. It is less sensitive to outliers than RMSE. The formula for MAE is expressed as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{8}$$

R-squared (R^2) represents the proportion of the variance in the dependent variable that is explained by the regression model, thereby indicating how well the model fits the actual data. Let $\bar{Y}$ denote the mean of the observed values Y_i, the R-squared formula is expressed as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} \tag{9}$$

$R^2 = 0$ indicates that the model fails to explain any of the variability in the data, while $R^2 = 1$ means that the model perfectly explains the variability of the dependent variable. However, in practice, achieving a perfect R^2 is extremely rare and often suggests overfitting.

Explained Variance Score is a commonly used regression metric that measures the extent to which a model can account for the variance in the target variable. In other words, it reflects the model's ability to capture the range of variation of the dependent variable, thereby indicating how well the model fits the observed data. The formula for the Explained Variance Score is expressed as follows:

$$\text{Variance Score} = 1 - \frac{\text{Var}(Y_i - \hat{Y}_i)}{\text{Var}(Y_i)} \tag{10}$$

E. Result

TABLE I. COMPARISON OF MACHINE LEARNING MODELS

Algorithms	MSE	RMSE	MAE	R-squared	Var Score
Linear Regression	0.861584	0.928215	0.629906	0.156248	0.156726
Support Vector Machine	0.568739	0.754148	0.373921	0.442634	0.454233
K-Nearest Neighbor	0.510844	0.714733	0.349556	0.498060	0.498381
Decision Tree	0.391175	0.625439	0.257077	0.616245	0.616401
Random Forest	0.291573	0.539975	0.248159	0.713733	0.713837
Gradient Boosting	0.291371	0.539788	0.279197	0.714053	0.714155
Extreme Gradient Boosting	0.291814	0.540198	0.279138	0.713593	0.713699
Artificial Neural Network	0.319222	0.564997	0.299310	0.683862	0.685793

Table I shows that the Gradient Boosting algorithm achieves more accurate prediction results than other regression models, with the lowest mean squared error of just 0.291371 and the highest R-squared value of 0.714053. The key advantage of Gradient Boosting lies in its additive training mechanism, where multiple weak learners are built sequentially to iteratively reduce the residual errors of previous models. This enables the model to capture complex nonlinear relationships between antenna dimensions and performance more effectively.

To evaluate the accuracy of the Gradient Boosting model, validation is performed by comparing the model's predicted performance results with simulation data from CST. Table II summarizes the input dimension parameters extracted from the test set, while Table III presents the comparison between the predicted performance metrics and the corresponding simulated values, demonstrating the consistency between the model and CST simulations.

TABLE II. ANTENNA DIMENSIONS FROM SIMULATION

Antenna dimensions from simulation									
No.	Lp	Ls	Lslot	Lfeed	Wp	Ws	Wslot	Wfeed	h
1	21	31	14	10	39	54	3	1.72	0.8
2	23	38	3	10	43	58	3	1.72	0.8
3	39	49	5	10	39	49	2	1.72	0.8
4	47	62	5	10	43	48	2	1.72	0.8
5	47	62	5	10	39	54	1	1.72	0.8

TABLE III. ANTENNA CHARACTERISTICS FROM SIMULATION AND PREDICTION

Antenna characteristics from simulation												
No.	F_1	S_{11_1}	BW_1	F_2	S_{11_2}	BW_2	F_3	S_{11_3}	BW_3	F_4	S_{11_4}	BW_4
1	3.476	-14.614	0.0608	0	0	0	0	0	0	0	0	0
2	3.364	-16.408	0.0421	0	0	0	0	0	0	0	0	0
3	4.064	-15.760	0.0362	0	0	0	0	0	0	0	0	0
4	3.648	-13.594	0.0213	0	0	0	0	0	0	0	0	0
5	3.34	-33.977	0.0285	3.98	-26.224	0.0324	0	0	0	0	0	0
Antenna characteristics from prediction												
No.	F_1	S_{11_1}	BW_1	F_2	S_{11_2}	BW_2	F_3	S_{11_3}	BW_3	F_4	S_{11_4}	BW_4
1	3.3325	-16.462	0.054	0	0	0	0	0	0	0	0	0
2	3.0814	-17.774	0.042	0	0	0	0	0	0	0	0	0
3	3.9037	-15.134	0.028	0	0	0	0	0	0	0	0	0
4	3.4398	-13.901	0.019	0	0	0	0	0	0	0	0	0
5	3.3702	-20.573	0.024	3.852	-25.920	0.0303	0	0	0	0	0	0

The comparison results in Table III demonstrate that the Gradient Boosting model is capable of accurately learning and predicting performance parameters from input dimension parameters, with only minor deviations from the simulation values. As such, the model can be utilized to replace traditional simulation methods that are time-consuming and costly. The zero values observed in columns F_2, S_{11_2}, BW_2, etc., indicate that no additional resonances with a reflection coefficient lower than -10 dB were detected in the dataset.

Table IV compares the proposed Gradient Boosting-based method with other machine learning approaches for antenna optimization reported in recent studies. The proposed method enables the optimization of antennas with compact size, complex structures, and low MSE.

TABLE IV. COMPARISON WITH RECENT METHODS

No.	Algorithms used	Dataset	MSE	MAE	RMSE	R-squared	Var Score
[1]	Decision Tree Random Forest Support Vector Regression Artificial Neural Network	215	5.556 3.45 5.317 4.39	-	-	-	-
[2]	Linear Regression Random Forest Support Vector Regression Gaussian Process Regression	164	0.0759 0.0857 0.1354 0.0138	0.0741 0.0931 0.1040 0.0639	0.1136 0.1028 0.0992 0.0698	0.7491 0.8233 0.7308 0.9287	0.7957 0.8453 0.7675 0.9377
[3]	Decision Tree XGBoost Extra Trees Regression Random Forest Gradient Boosting	-	0.0839 0.0732 0.0701 0.0789 0.0660	0.0866 0.0683 0.0450 0.0619 0.0494	0.0852 0.0713 0.0845 0.0765 0.0413	0.4289 0.5901 0.8213 0.6630 0.9842	0.5085 0.6288 0.8568 0.7117 0.9847
[4]	K-Nearest Neighbor Decision Tree Random Forest XGBoost Artificial Neural Network	216	0.31 0.51 0.32 0.33 1.02	0.11 0.16 0.13 0.18 0.42	-	0.98 0.97 0.99 0.98 0.94	-
[5]	Artificial Neural Network Decision Tree	35,035	1.892 0.036	-	3.017 0.487	-	-
This work	Gradient Boosting	15,000	0.291371	0.279197	0.539788	0.714053	0.714155

III. CONCLUSION

This project proposes an effective method for antenna design and optimization. A data processing approach on Google Colab was developed, enabling precise extraction of antenna dimension and performance values from CST. This process facilitates the creation of a comprehensive dataset for training machine learning models. Utilizing the Gradient Boosting and a dataset of 15,000 samples, the model is capable of predicting antenna parameters with a MSE of 0.2914, demonstrating the high prediction accuracy of the proposed approach. This result highlights the method's potential to significantly improve the efficiency and reliability of antenna design and optimization compared to traditional simulation-based approaches.

REFERENCES

[1] N. Kurniawati, "Predicting rectangular patch microstrip antenna dimension using machine learning," Journal of Communications, vol. 16, pp. 394–399, Sep. 2021.

[2] M. S. Yahya, S. Soeung, S. K. A. Rahim, U. Musa, S. S. B. Hashwan, and M. A. Haque, "Machine learning-optimized compact frequency reconfigurable antenna with RSSI enhancement for long-range applications," IEEE Access, vol. 12, pp. 10970–10987, 2024.

[3] M. A. Haque et al., "Multiband THz MIMO antenna with regression machine learning techniques for isolation prediction in IoT applications," Scientific Reports, vol. 15, no. 1, p. 7701, 2025.

[4] O. Singh et al., "Microstrip line fed dielectric resonator antenna optimization using machine learning algorithms," Sādhanā, vol. 47, no. 4, p. 226, 2022.

[5] B. Maity and S. K. Nayak, "Compact quad-band CP series-fed circular slit microstrip array antenna using machine learning," IEEE Access, vol. 10, pp. 116650–116661, 2022.

[6] R. Jain, V. V. Thakare, and P. K. Singhal, "Design and comparative analysis of THz antenna through machine learning for 6G connectivity," *IEEE Latin America Transactions*, vol. 22, no. 2, pp. 82–91, 2024.

[7] M. A. Haque et al., "Dual band antenna design and prediction of resonance frequency using machine learning approaches," Applied Sciences, vol. 12, no. 20, p. 10505, 2022.

[8] A. Garima, B. Bansal, and S. Sharma, "MATLAB-CST interfacing for a microstrip patch antenna," Indian Journal of Science and Technology, vol. 10, no. 44, pp. 1–5, 2017, doi: 10.17485/ijst/2017/v10i44/120576.

[9] J. Manasa, R. Gupta, and N. S. Narahari, "Machine learning-based prediction of house prices using regression techniques," in Proc. 2nd Int. Conf. Innov. Mech. Ind. Appl. (ICIMIA), Mar. 2020, pp. 624–630.

[10] E. Yoshino, F. I. Kurniadi, and B. Juarto, "Forecasting rice production in Indonesia using regression techniques: A comparative analysis of support vector machine, linear regression, and XGBoost regression," in Proc. 10th Int. Conf. ICT Smart Soc. (ICISS), Sep. 2023, pp. 1–5.

[11] V. Kumar, N. Kedam, K. V. Sharma, D. J. Mehta, and T. Caloiero, "Advanced machine learning techniques to improve hydrological prediction: A comparative analysis of streamflow prediction models," Water, vol. 15, no. 14, p. 2572, 2023.

[12] M. A. Haque, M. A. Zakariya, S. S. Al-Bawri, Z. Yusoff, M. Islam, D. Saha, and L. C. Paul, "Quasi-Yagi antenna design for LTE applications and prediction of gain and directivity using machine learning approaches," Alexandria Engineering Journal, vol. 80, pp. 383–396, 2023.

[13] D. Borup, B. J. Christensen, N. S. Mühlbach, and M. S. Nielsen, "Targeting predictors in random forest regression," International Journal of Forecasting, 2022, doi: 10.1016/j.ijforecast.2022.02.010.

[14] J. Velthoen, C. Dombry, J.-J. Cai, and S. Engelke, "Gradient boosting for extreme quantile regression," Extremes, vol. 26, no. 4, pp. 639–667, 2023, doi: 10.1007/s10687-023-00473-x.

[15] T. M. T. Lei, S. C. W. Ng, and S. W. I. Siu, "Application of ANN, XGBoost, and other ML methods to forecast air quality in Macau," *Sustainability*, vol. 15, no. 6, p. 5341, 2023, doi: 10.3390/su15065341.

[16] Y. Chen, L. Song, Y. Liu, L. Yang, and D. Li, "A review of artificial neural network models for water quality prediction," Applied Sciences, vol. 10, no. 17, p. 5776, 2020.

[17] M. M. Alam, N. A. T. Yusof, A. A. M. Faudzi, M. R. I. Tomal, M. E. Haque, and M. S. Rahman, "Machine learning-based approach for bandwidth and frequency prediction of circular SIW antenna," Journal of King Saud University–Engineering Sciences, vol. 37, no. 4, pp. 1–19, 2025.