Enhancing Antenna Design Through Machine Learning Evaluation of Predictive Models

[1]Swabi Mishra, [2]Dr. Girraj Prasad Rathor

[1]Department of Electronics and Communication Engineering, Technocrats Institute of Technology, Bhopal, (M.P.)
[2]Department of Electronics and Communication Engineering, Technocrats Institute of Technology, Bhopal, (M.P.)

* Corresponding Author: Swabi Mishra

Abstract: *Antennas are essential in radio engineering, facilitating the transmission and reception of radio signals. This paper explores various types of antennas, including dipole, loop, and parabolic antennas, and their applications in communication systems. The focus is on the microstrip patch antenna (MPA), valued for its compact size, low cost, and minimal profile. The study investigates the use of machine learning techniques to optimize antenna design, evaluating four predictive models—Decision Tree (DT), Random Forest (RF), Support Vector Regression (SVR), and Artificial Neural Networks (ANN)—through MATLAB simulations. The effectiveness of these models is measured using Mean Squared Error (MSE) metrics. Findings highlight the most accurate machine learning approach for optimizing antenna design.*

Keywords: *Antenna Design, Microstrip Patch Antenna, Machine Learning, Decision Tree, Random Forest, Support Vector Regression, Artificial Neural Networks.*

I. INTRODUCTION

An antenna or aerial in radio engineering is a specialized transducer, designed by an array of conductors which are connected electrically to the transmitter or receiver. The main function of an antenna is to transmit & receive radio waves equally within all horizontal directions Antennas are available in different types and shapes. The small antennas can be found on the roof of homes to watch TV and big antennas capture signals from different satellites which are away millions of miles. Antennas move vertically & horizontally to capture & transmit the signal.

We have covered the properties of antennas, and now we will discuss the different types of antennas used for various applications. These include dipole antennas, which are commonly used for radio and television broadcasting; loop antennas, often employed in direction-finding; and parabolic antennas, which are essential for satellite communications and radar systems. Each type of antenna has unique characteristics that make it suitable for specific purposes.

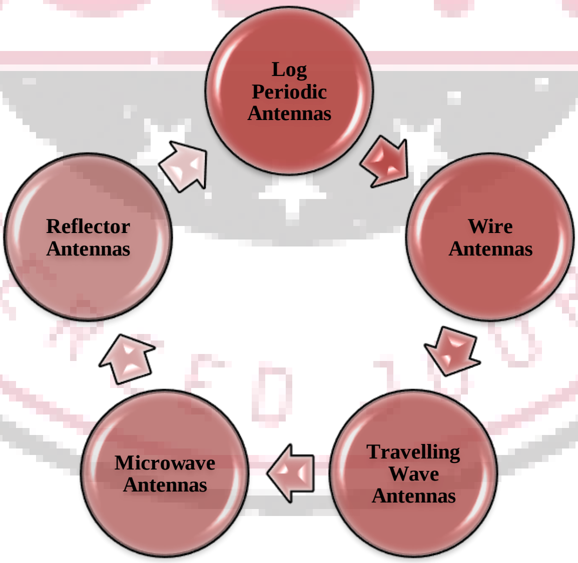

Figure 1 Different Types of Antennas

A. Rectangular Patch Microstrip Antenna

Antennas are an essential component of the telecommunications industry. To put it more simply, it is a transducer that changes radio signals into electrical energy and electrical energy into radio signals. Wireless communication technology enables the transmission and reception of signals among people who live in geographically inaccessible locations so that they can communicate with one another. Nowadays, a wide variety of applications make use of the microstrip patch antenna

(MPA), which is popular due to its low volume, low cost, and low profile. The performance of the microstrip patch antenna can be improved by optimizing its design for a number of different factors.

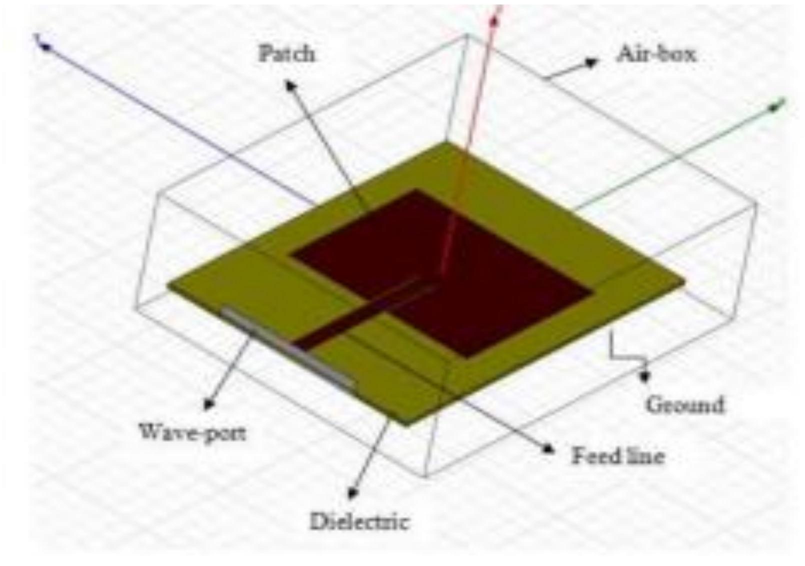

Figure 2 Microstrip patch antenna [3]

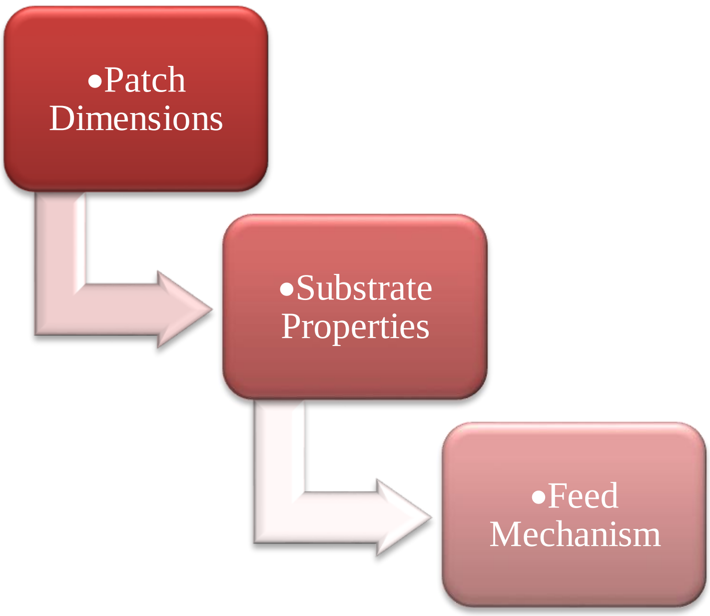

Figure 3 Design Parameters performance of a rectangular patch microstrip antenna

B. Challenges in Antenna Dimensioning

Challenges in antenna dimensioning are multifaceted, involving technical, physical, and environmental considerations. Technically, achieving the desired frequency, gain, and bandwidth while maintaining a compact size is complex. Physical constraints such as space limitations and the material properties of the antenna further complicate the design process. Environmental factors, including signal interference and the impact of weather conditions, can affect performance and necessitate adjustments in dimensioning. Additionally, the integration of antennas into devices requires careful balancing to avoid compromising the functionality of other components. Addressing these challenges requires innovative engineering solutions, rigorous testing, and a thorough understanding of electromagnetic principles.

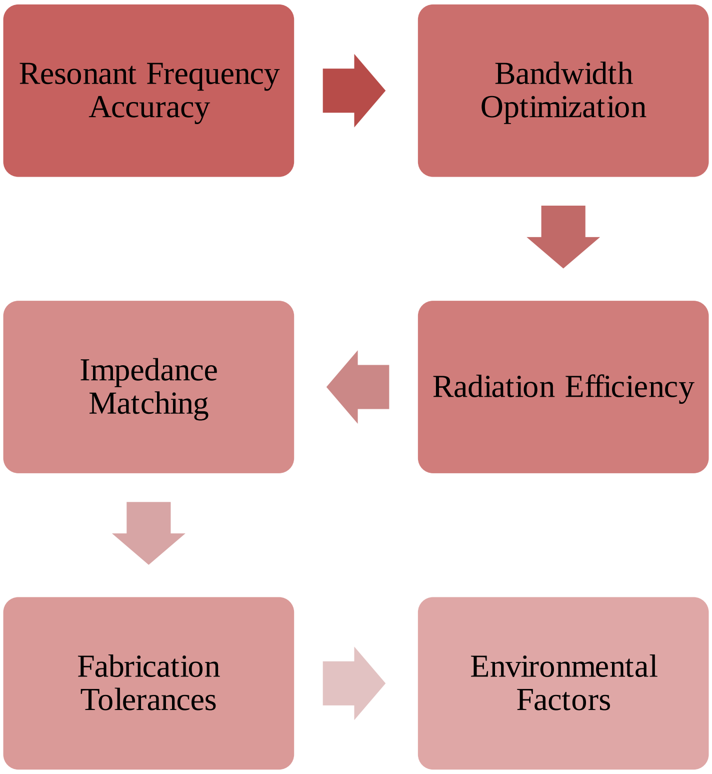

Figure 4 Dimensions for a rectangular patch microstrip antenna Common challenges

C. Machine Learning in Antenna Design

Machine learning plays a transformative role in engineering and design, offering powerful tools for optimizing complex systems. In antenna design, machine learning can streamline the process of determining optimal dimensions by analysing vast amounts of data and identifying patterns that traditional methods might overlook. This approach significantly reduces the time and effort required for trial-and-error testing. Advantages of using machine learning for antenna dimensioning include enhanced accuracy in predicting performance outcomes, the ability to handle multifaceted design constraints, and improved efficiency in achieving the best possible antenna characteristics, ultimately leading to more innovative and effective designs. Machines are increasingly acquiring human-like abilities such as problem-solving, decision-making, and learning. Machine learning (ML) automates analytical model building through data analysis, while deep learning (DL), a subset of ML, helps machines process data to mimic human behavior. ML and DL optimize antenna performance, making design processes more efficient and rapid. These technologies have become pivotal in recent research, enhancing various antenna design fields like millimeter wave, body-centric, terahertz, satellite, UAV, GPS, and textile antennas. For instance, body-centric antennas support wearable communication devices, terahertz frequencies aid in spectroscopy, and satellite antennas facilitate global communication [4]. Over the past few decades, antennas and their systems have rapidly advanced due to significant changes in their geometric and material profiles to meet modern applications like body-centric communications and multiband operations for 2G/3G/4G/5G [5-6]. Typically, antenna design follows established guidelines based on design experience.

II. LITERATURE REVIEW

Kurniawati, N., et al. (2021) [25] This study focuses on optimizing the design of rectangular patch microstrip antennas by incorporating machine learning techniques. Traditionally, antenna design relies on trial-and-error methods to meet desired parameters, but this research applies machine learning to predict antenna dimensions. Using simulation data from antennas with various dimensions, four algorithms—Decision Tree, Random Forest, Support Vector Regression (SVR), and Artificial Neural Networks (ANN)—were evaluated. Among them, Random Forest with 15 estimators achieved the best performance, showing the potential of machine learning to enhance the design process beyond traditional simulation software capabilities.

Sharma, K., & Pandey, G. P. (2020) [26] This paper explores the application of machine learning in designing a compact dual-band H-shaped rectangular microstrip antenna operating in two frequency ranges: 0.75–2.20 GHz and 3.0–3.44 GHz. The study utilizes an Artificial Neural Network (ANN) model developed from simulation data to predict antenna shape. Comparison with a mathematical model showed that the ANN-based approach offered superior prediction accuracy, demonstrating the effectiveness of machine learning in antenna design.

Y. Sharma, et al. (2022) [27] The article proposes a new machine learning-based optimization technique for complex antenna designs, specifically focusing on monopole antennas with spatially dependent dielectric materials. Gaussian Process (GP) regression and Artificial Neural Networks (ANN) were used to map dielectric constant values to gain patterns. The results were compared with heuristic methods like Genetic Algorithms (GAs), showing that the machine learning approach efficiently handled high-dimensional, nonlinear problems and provided accurate optimization results.

Nakmouche, M. F., et al. (2021) [28] This research develops a high-gain Frequency Selective Surface (FSS) reflector-backed monopole antenna for 5G applications, utilizing machine learning to determine optimal parameters for the FSS reflector and monopole ground dimensions. The study involved simulations and measurements to verify performance in the 6 GHz band. The ML-optimized design demonstrated improved size and gain efficiency compared to existing designs, showing the effectiveness of integrating FSS with machine learning for enhanced antenna performance.

Gupta, S. H., et al. (2022) [29] This paper presents a compact microstrip patch antenna used for lung cancer detection. The antenna, operating at the ISM band, was tested on phantoms of healthy and cancerous lungs. The study created a dataset from various cancer stages and used the Random Forest algorithm to classify and differentiate between healthy and cancerous states with high accuracy (93.75%). This highlights the potential of microstrip patch antennas in medical diagnostics.

L. P. Shi, et al. (2021) [30] The study addresses the challenge of time-consuming full-wave simulations for graphene reconfigurable reflectarray antennas by proposing a deep learning-based prediction method. Convolutional Neural Networks (CNNs) were used to predict electromagnetic responses from discretized input data. The CNN method achieved over 99% accuracy and significantly reduced computation time compared to traditional methods, demonstrating its efficiency in predicting complex antenna characteristics.

Pramudita, A. A., et al. (2019) [31] This study investigates the impact of antenna dimensions on the footprint in Ground Penetrating Radar (GPR) applications. It examines ultra-wideband (UWB) antennas and their footprint variations based on size, finding that antenna dimensions significantly affect detection results. The research proposes a method for controlling antenna footprint to adapt to different soil conditions, validated through simulations and experiments.

D. Mathur, et al. (2014) [32] The paper introduces a method for quickly estimating the dimensions of rectangular microstrip antennas (RMSA) based on the concept of "Equivalence of Design." By relating the classical extension in length to the patch dimensions and substrate thickness, the method allows for efficient parameter estimation and design transformation. Validation through simulation and measurement demonstrated accurate predictions, highlighting the practicality of the proposed estimation approach.

M. Fallahpour and R. Zoughi (2018) [33] This study explores antenna miniaturization techniques and their effects on bandwidth and efficiency. It discusses how reducing antenna size impacts radiation quality and impedance bandwidth, while recent investigations aim to minimize antenna size without compromising performance. The research contributes to understanding how electrical and physical properties can be adjusted to achieve miniaturized antennas with acceptable operational characteristics.

Alluri, S., & Rangaswamy, N. (2020) [34] The article presents a compact super wideband (SWB) antenna with high bandwidth dimension ratio (BDR) for future wireless applications. The antenna features a unique design with a tapered microstrip-fed circular ring and elliptical spokes. Experimental results show a SWB from 2.3 to 34.8 GHz and a high BDR, demonstrating the antenna's suitability for both civilian and military applications due to its compact size and broad bandwidth.

IV. OBJECTIVES

- To optimize Antenna Design Using MATLAB as simulation technique by selecting the parameters from (base-paper author name, year)
- To develop four Machine Learning models (DT, Random Forest, SVR, ANN) to evaluate Mean Square Error.
- To compare the values of MSE with base paper

IV METHODOLGY

Antenna Design and Simulation

The design and simulation of a rectangular patch microstrip antenna using MATLAB involved a detailed process aimed at accurately modeling the antenna's electromagnetic properties and evaluating its performance. The simulation process consisted of several key steps to ensure the design was optimized for the desired operating conditions, these steps are presented.

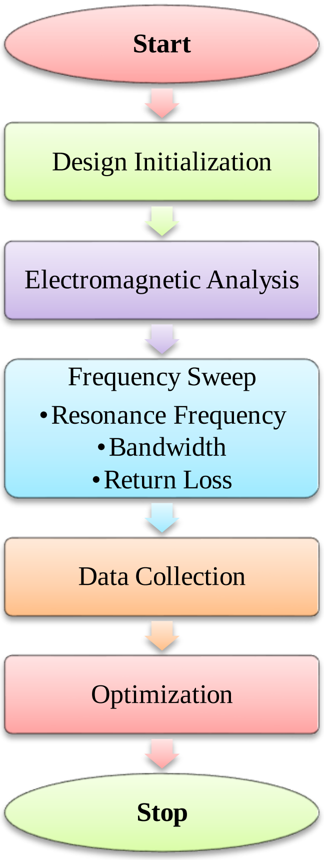

Figure 5 Flow chart of proposed methodology

RECTANGULAR PATCH ANTENNA SIMULATION

In this research, MATLAB was used as the primary simulation software to conduct a comprehensive study on rectangular patch microstrip antennas. MATLAB's advanced computational and simulation capabilities, including the implementation of numerical techniques such as the Finite Integration Technique (FIT) and the Transmission Line Matrix (TLM) method, made it an ideal choice for this high-frequency component analysis.

Table 4.1: Antenna Dimension and Values

Parameter	**Description**	**Value**
εr	Dielectric Constant	4.3
Hs	Substrate Thickness	1.6 mm
Wg	Substrate Width	76 mm
Lg	Substrate Length	58 mm
Ht	Copper-plate Thickness	0.035 mm
gpf	Gap Width	1 mm
Wf	Transmission Line Width	3.137 mm
fi	Transmission Line Length	8.85 mm
W	Patch Antenna Width	19-63 mm
L	Patch Antenna Length	10-54 mm

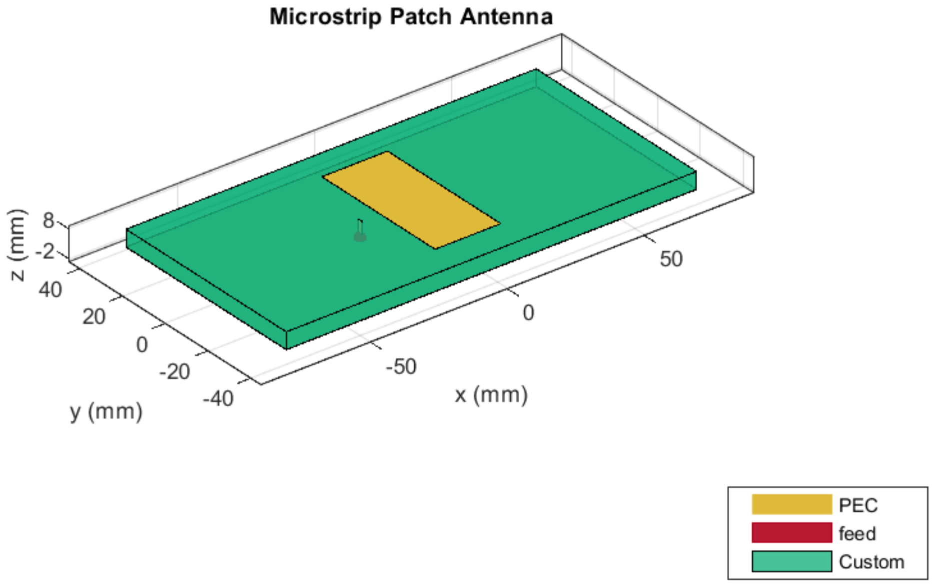

Figure 6 Rectangular Microstrip Patch Antenna

Machine Learning Algorithm Implementation

To make predictions based on the dataset generated from the antenna simulations, four advanced machine learning algorithms were employed: Decision Tree, Random Forest, Support Vector Regression (SVR), and Artificial Neural Networks (ANN). These algorithms were specifically chosen for their robust capability to handle regression tasks, which are essential for predicting numerical values. Given that the desired output involves predicting continuous numerical values, regression methods are the most appropriate for this purpose. Furthermore, these algorithms are proficient in handling non-linear data, which is a common characteristic in complex engineering problems such as antenna performance prediction.

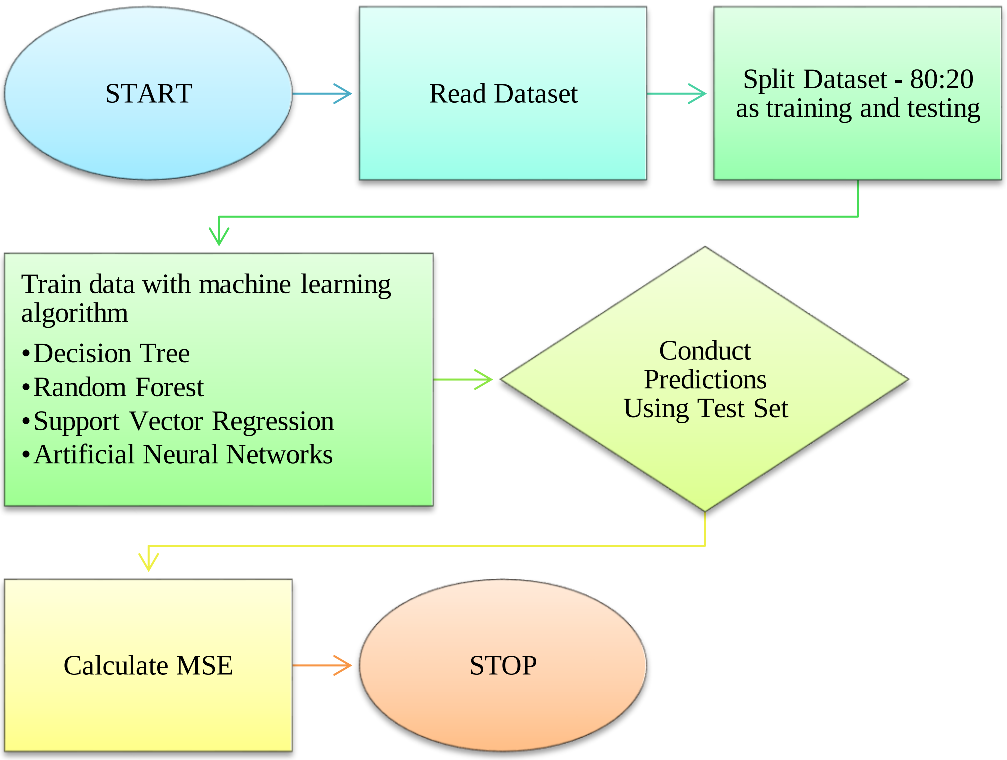

Figure 7 Machine learning algorithm implementation flowchart

Optimizers

Stochastic Gradient Descent (SGD)

SGD is a simple yet powerful optimizer that updates the model parameters using the gradient of the loss function with respect to each training example. It iteratively adjusts the parameters in the direction that minimally reduces the error, one data point at a time. This makes SGD particularly efficient for large datasets, although it can be noisy and may require careful tuning of the learning rate.

Adadelta

Adadelta is an extension of SGD that adapts the learning rate dynamically. Unlike SGD, which updates parameters in a single time step, Adadelta uses a moving window of gradient updates to calculate more robust step sizes. This adaptation overcomes the need to set a learning rate manually and allows for automatic adjustment of the learning rate based on the historical gradients, making it effective in handling diverse datasets.

Adam (Adaptive Moment Estimation)

Adam combines the advantages of two other extensions of SGD: AdaGrad and RMSProp. Adam maintains a running average of both the gradients and the squared gradients. This helps in adapting the learning rate for each parameter separately, allowing for efficient training of deep neural networks. Adam's adaptability to various data characteristics and its robustness to sparse gradients make it a popular choice for many machine learning applications.

AdaGrad

AdaGrad is another variant of SGD that adapts the learning rate based on the frequency of updates for each parameter. Parameters that receive frequent updates have their learning rates reduced, whereas infrequent updates lead to larger learning rates. This adaptive nature helps in optimizing models with sparse data and can significantly improve convergence rates.

Loss Function

In regression tasks, the loss function quantifies the error between the predicted values and the actual values. A lower loss function value indicates better model performance. Several loss functions are used to measure this error, including Sum of Errors (SE), Sum of Absolute Error (SAE), Sum of Squared Error (SSE), and Mean Squared Error (MSE).

Mean Squared Error (MSE)

MSE is a commonly used loss function for regression tasks due to its simplicity and effectiveness. It measures the average of the squared differences between the predicted values and the actual values. MSE is particularly advantageous because it penalizes larger errors more heavily, providing a clear metric for model optimization. Additionally, as the dataset size increases, the aggregate error decreases, making MSE a reliable metric for evaluating model performance.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2$$

where:

- n is the number of data points.
- yi is the actual value for the iii-th data point.
- $\widehat{y_i}$ is the predicted value for the iii-th data point.

The MSE value provides a single scalar that represents the average squared error of the predictions, making it easy to interpret the model's performance.

V. RESULT AND DISCUSSION

In this chapter, the Mean Squared Error (MSE) results for each of the four predictive algorithms—Decision Tree, Random Forest, Support Vector Regression (SVR), and Artificial Neural Network (ANN)—are presented. The MSE is a crucial performance metric that assesses the accuracy of these algorithms by measuring the average squared difference between their predicted values and the actual values obtained from the simulation data. To compute the MSE, the predicted values generated by each algorithm are compared against the corresponding actual values from the simulation. The differences between these values are squared to ensure that all differences are positive and to emphasize larger errors more than smaller ones. The squared differences are then averaged across all data points to produce the MSE for each algorithm.

The MSE serves as a valuable indicator of each algorithm's performance. A lower MSE signifies that the predictions are closer to the actual values, indicating higher accuracy and effectiveness of the model. In contrast, a higher MSE suggests greater discrepancies between the predicted and actual values, reflecting lower predictive accuracy. By comparing the MSE results across the Decision Tree, Random Forest, SVR, and ANN algorithms, we can determine which algorithm performs best in terms of prediction accuracy.

A Decision Tree Mean Squared Error Analysis

In the Decision Tree algorithm, the random state parameter plays a crucial role in determining the reproducibility and stability of the model's predictions. By setting different random state values, we can observe variations in the model's performance, as shown in Fig. 5.1, which displays the Mean Squared Error (MSE) values for a range of random state settings.

The plot reveals a fluctuating pattern in MSE as the random state changes. The MSE values oscillate between approximately 0.04460 and 0.04485. Several local minima and maxima are observed throughout the range. Notably, the MSE reaches its lowest points at random states around 5, 15, 25, 35, and 45. Conversely, the MSE peaks are observed at random states around 2, 12, 22, 32, and 42.

This fluctuation suggests that the random state parameter significantly influences the model's performance, with some values resulting in more accurate predictions (lower MSE) and others leading to less accurate predictions (higher MSE). The variability indicates the sensitivity of the Decision Tree algorithm to the randomness introduced during model training and data splitting.

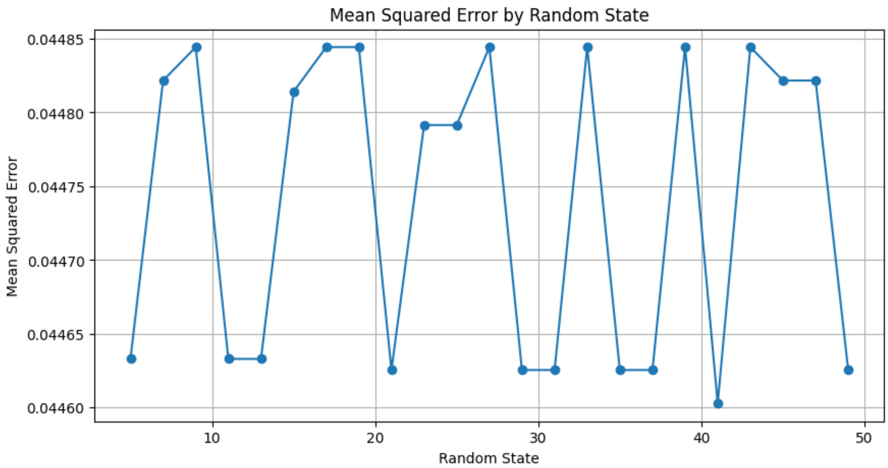

Figure 8 Decision tree for MSE

Beyond a random state of 20, the MSE values fluctuate, reflecting the inherent variability introduced by the random state. These fluctuations occur because the random state parameter influences how the data is split into training and testing sets and how the decision tree is constructed. Small changes in the random state can lead to different splits and tree structures, resulting in varying levels of prediction accuracy. Interestingly, the MSE reaches its lowest value when the random state is set to 50, suggesting that this specific configuration provides the most accurate predictions for the given data. The fluctuating results highlight the sensitivity of the Decision Tree algorithm to the random state parameter, emphasizing the importance of careful tuning and selection of this parameter to achieve optimal model performance.

B. Random Forest MSE Result

The Mean Squared Error (MSE) results for the Random Forest regression algorithm, as the number of estimators (decision trees) varies from 1 to 50. The x-axis represents the number of estimators, while the y-axis shows the corresponding MSE values. The graph in Fig. 5.2 shows a clear trend of decreasing MSE as the number of estimators increases from 1 to around 15. Initially, the MSE starts at approximately 0.035 and drops sharply to around 0.029. This decline indicates that adding more estimators improves the model's accuracy, likely due to the increased diversity and ensemble effect of combining multiple decision trees.

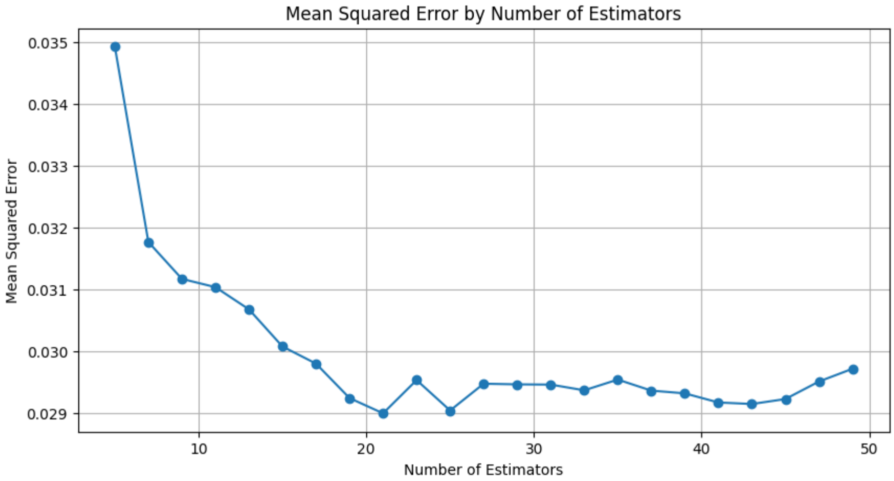

Figure 9 Mean Square Error for Random Forest

After reaching this minimum MSE around 15 estimators, the MSE values remain relatively stable, fluctuating slightly around the 0.029 mark. There is a minor increase in MSE as the number of estimators reaches 50, suggesting a possible overfitting or diminishing returns in model performance improvement. This stability indicates that beyond a certain point, adding more estimators does not significantly enhance the model's predictive accuracy.

The different MSE results for varying numbers of estimators highlight the inherent nature of the Random Forest regression algorithm. In Random Forests, each estimator (or decision tree) acts as an independent sampler of the dataset, creating a diverse set of models that contribute to the final prediction. This diversity is generally beneficial up to a point, beyond which the addition of more estimator's yields diminishing returns or even slight performance degradation due to potential overfitting.

C. Result for MSE in Support Vector Regression

Mean Squared Error (MSE) results for various configurations of the Support Vector Regression (SVR) model are presented in Table 5.1, focusing on different values of the hyperparameters Epsilon and Gamma.

Epsilon: This parameter defines the margin of tolerance where no penalty is given to errors. In simpler terms, it's the threshold within which the errors are ignored when fitting the model. Smaller values of Epsilon allow the model to be more sensitive to small errors, potentially capturing more details in the data but also risking overfitting.

Gamma: This parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. In other words, Gamma determines the weight of each data point on the decision boundary. Higher values of Gamma can lead to a model that fits the training data more closely, but it can also increase the risk of overfitting.

Table 5.1: Support Vector Regression: MSE Results for Different Epsilon and Gamma Values

	Epsilon	Gamma	MSE
0	0.01	0.0001	0.023610
1	0.01	0.0010	0.026248
2	0.01	0.0100	0.028902
3	0.01	0.1000	0.043129
4	0.01	1.0000	0.022314
5	0.10	0.0001	0.022606
6	0.10	0.0010	0.023468
7	0.10	0.0100	0.026289

8	0.10	0.1000	0.031320
9	0.10	1.0000	0.022131
10	1.00	0.0001	0.021970
11	1.00	0.0010	0.021970
12	1.00	0.0100	0.021970
13	1.00	0.1000	0.021970
14	1.00	1.0000	0.021970

The table shows the MSE values for each combination of Epsilon and Gamma, indicating how well each configuration performs in terms of prediction accuracy. For example, with Epsilon set to 0.01 and Gamma set to 0.0001, the MSE is 0.023610. As Gamma increases to 1.0000 with the same Epsilon, the MSE decreases to 0.022314, suggesting that the model's accuracy improves with a higher Gamma in this instance.

Across all rows, a trend can be observed: as Epsilon increases from 0.01 to 1.00, the MSE generally decreases, especially when Gamma is fixed at a lower value. The lowest MSE observed is 0.021970, occurring consistently when Epsilon is 1.00, regardless of the Gamma value. This consistency indicates that the model achieves optimal performance under these settings, highlighting the importance of selecting appropriate hyperparameter values.

D. Artificial Neural Networks: MSE Result Analysis

The graph given in Fig. 5.3 illustrates the relationship between the number of hidden layers in an ANN and the corresponding Mean Squared Error (MSE) values. The x-axis represents the number of hidden layers, while the y-axis shows the MSE values.

One Hidden Layer: The MSE starts at around 1.045. This value represents the model's prediction error with a simple architecture consisting of only one hidden layer.

Two Hidden Layers: As the number of hidden layers increases to two, the MSE decreases significantly to the lowest value observed in the graph, approximately 1.030. This indicates that adding a second hidden layer improves the model's ability to fit the data, reducing the prediction error.

Three Hidden Layers: However, when a third hidden layer is introduced, the MSE rises sharply to around 1.050, suggesting that adding this layer does not necessarily improve the model's performance and might lead to overfitting or increased complexity without significant gains in accuracy.

Four Hidden Layers: The MSE continues to increase slightly with four hidden layers, reaching the highest value in the graph, approximately 1.052. This further increase in MSE indicates that the model's complexity might be growing excessively, leading to poorer generalization on the test data.

Five Hidden Layers: Finally, with five hidden layers, the MSE drops again to around 1.040, suggesting a potential improvement in performance. However, this MSE is still higher than the minimum observed with two hidden layers, indicating that the optimal number of hidden layers might be less than five.

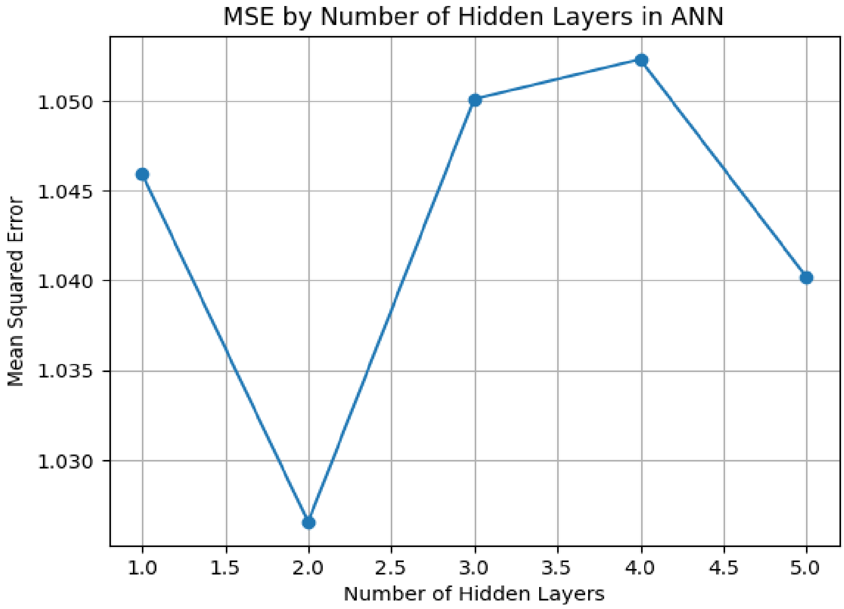

Figure 10 MSE by Number of Hidden Layers in ANN

The fluctuations in MSE values across different numbers of hidden layers highlight the importance of carefully tuning the ANN architecture. While adding more layers can potentially increase the model's capacity to learn complex patterns, it can also lead to overfitting if the model becomes too complex relative to the available data.

In practice, selecting the right number of hidden layers involves balancing the trade-off between model complexity and prediction accuracy. The graph suggests that in this particular case, the optimal architecture may include around two hidden layers, as this configuration resulted in the lowest MSE.

E. Comparative Analysis

In predictive modeling, evaluating the performance of different algorithms is crucial for selecting the best approach for a given problem. Mean Squared Error (MSE) is a common metric used to assess the accuracy of a model's predictions, with lower values indicating better performance. The following table compares the MSE of various predictive algorithms—Decision Tree, Random Forest, Support Vector Regression (SVR), and Artificial Neural Networks (ANN)—under different conditions, contrasting the base work with the proposed improvements. This comparison provides insights into the effectiveness of each technique and highlights the potential benefits of optimizing algorithm parameters.

Table 5.2: Comaprative Analysis of MSE Values for all Algorithms

Techniques	**Condition**	**Base Work MSE**	**Proposed Work MSE**
Decision Tree	Random State, 50	5.556	0.44
Random Forest	Estimators; 15	3.45	0.29
SVR	Gamma-0.0001; epsilon -1	5.317	0.25
ANN	Hidden layers 4; Optimizer ADAM	4.319	1.05

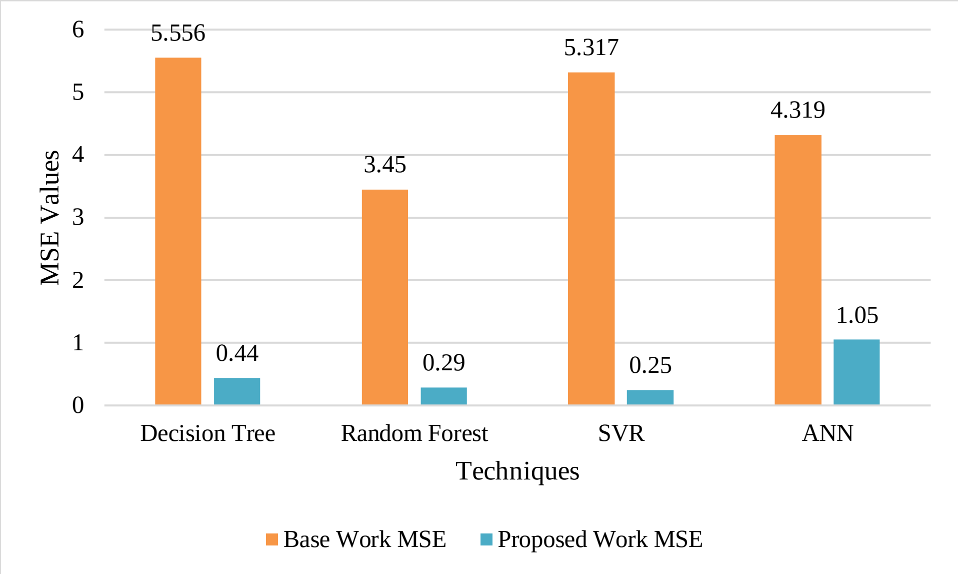

Figure 11 Comparative analysis of MSE Values

The Mean Squared Error (MSE) for different predictive algorithms under specified conditions, comparing base and proposed configurations. For the Decision Tree algorithm, optimizing the random state from 50 results in a significant drop in MSE from 5.556 to 0.44. Similarly, for the Random Forest with 15 estimators, the MSE improves from 3.45 to 0.29 with the proposed changes. Support Vector Regression (SVR) benefits from reduced gamma and epsilon values, decreasing MSE from 5.317 to 0.25. Conversely, in the case of Artificial Neural Networks (ANN) with 4 hidden layers and the ADAM optimizer, the proposed adjustments lead to an increase in MSE from 4.319 to 1.05, suggesting a potential deterioration in model performance with the new settings.

VI. CONCLUSION

Machine learning techniques have significantly advanced the optimization of antenna design. The study assessed Decision Tree, Random Forest, Support Vector Regression, and Artificial Neural Networks for their predictive accuracy. Results show that Random Forest and SVR models deliver the best performance, with lower Mean Squared Error (MSE) values compared to Decision Tree and ANN models. The ANN model exhibited reduced performance with specific optimizations, indicating areas for further improvement. Overall, machine learning provides a valuable tool for refining antenna design, and ongoing research may enhance these models for even greater precision and efficiency.

REFERENCES

[1] Rana, M. S., Hossain, S., Rana, S. B., & Rahman, M. M. (2023). Microstrip patch antennas for various applications: a review. Indonesian journal of Electrical Engineering and computer science, 29(3), 1511-1519.

[2] Khan, M. M., Hossain, S., Mozumdar, P., Akter, S., & Ashique, R. H. (2022). A review on machine learning and deep learning for various antenna design applications. Heliyon, 8(4).

[3] C. Mendes and C. Peixeiro, “A dual-mode single-band wearable microstrip antenna for body area networks,” IEEE Antennas and Wireless Propagation Letters, vol. 16, pp. 3055–3058, 2017.

[4] A. Alieldin et al., “A triple-band dual-polarized indoor base station antenna for 2G, 3G, 4G and sub-6 ghz 5G applications,” IEEE Access, vol. 6, pp. 49 209–49 216, 2018.

[5] Kurniawati, N., Arif, F., & Alam, S. (2021). Predicting Rectangular Patch Microstrip Antenna Dimension Using Machine Learning. J. Commun., 16(9), 394-399.

[6] Sharma, K., & Pandey, G. P. (2020). Designing a compact microstrip antenna using the machine learning approach. Journal of Telecommunications and Information Technology, (4), 44-52.

[7] Y. Sharma, X. Chen, J. Wu, Q. Zhou, H. H. Zhang and H. Xin, "Machine Learning Methods-Based Modeling and Optimization of 3-D-Printed Dielectrics Around Monopole Antenna," in *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 7, pp. 4997-5006, July 2022, doi: 10.1109/TAP.2022.3153688.

[8] Nakmouche, M. F., Allam, A. M., Fawzy, D. E., & Lin, D. B. (2021). Development of a high gain FSS reflector backed monopole antenna using machine learning for 5G applications. Progress In Electromagnetics Research M.

[9] Gupta, S. H., Goel, S., Kumar, M., Rajawat, A., & Singh, B. (2022). Design of terahertz antenna to detect lung cancer and classify its stages using machine learning. Optik, 249, 168271. https://doi.org/10.1016/j.ijleo.2021.168271

[10] L. P. Shi, Q. H. Zhang, S. H. Zhang, C. Yi and G. X. Liu, "Efficient Graphene Reconfigurable Reflectarray Antenna Electromagnetic Response Prediction Using Deep Learning," in IEEE Access, vol. 9, pp. 22671-22678, 2021, doi: 10.1109/ACCESS.2021.3054944.

[11] Pramudita, A. A., Kurniawan, A., Suksmono, A. B., & Lestari, A. A. (2019). Effect of antenna dimensions on the antenna footprint in ground penetrating radar applications. *IET microwaves, antennas & propagation*, *3*(8), 1271-1278.

[12] D. Mathur, S. K. Bhatnagar and V. Sahula, "Quick Estimation of Rectangular Patch Antenna Dimensions Based on Equivalent Design Concept," in *IEEE Antennas and Wireless Propagation Letters*, vol. 13, pp. 1469-1472, 2014, doi: 10.1109/LAWP.2014.2334362.

[13] M. Fallahpour and R. Zoughi, "Antenna Miniaturization Techniques: A Review of Topology- and Material-Based Methods," in *IEEE Antennas and Propagation Magazine*, vol. 60, no. 1, pp. 38-50, Feb. 2018, doi: 10.1109/MAP.2017.2774138.

[14] Alluri, S., & Rangaswamy, N. (2020). Compact high bandwidth dimension ratio steering-shaped super wideband antenna for future wireless communication applications. *Microwave and Optical Technology Letters*, *62*(12), 3985-3991.