Progress In Electromagnetics Research C, Vol. 130, 31–42, 2023

Modeling and Optimization of CPW-Fed E-Textile Antenna Using Machine Learning Algorithms

Arpan H. Shah[1, *], Kalyanbrata Ghosh[1], and Piyush N. Patel[2]

Abstract—In this paper, an electronic textile (E-textile) antenna design using machine learning (ML) algorithms such as polynomial regression, k-nearest neighbor (kNN), random forest regression, and deep neural network (DNN) is proposed for achieving the optimized solution. These ML techniques, including DNN, have been implemented on a python framework and support in selecting efficient optimum design parameters for a co-planar waveguide fed textile antenna to attain the maximum impedance bandwidth performance in 3–24 GHz band, respectively. Moreover, the accuracy of the predicted response values obtained by these ML methods has also been validated by verifying with the CST simulation software tool.

1. INTRODUCTION

The internet of things (IoT) based framework has facilitated the apparent demand for electronic textile (E-textile) antennas in the field of defense and healthcare industry. Smart electronic clothing paves the way to design a textile antenna for various applications such as health monitoring, tracking patient movement in rehabilitation therapy, sports, and navigation [1]. Hence, considering medical healthcare and security defense as the top priority sectors, designing an efficient antenna has become ineludible. The current trends in antenna design are primarily based on traditional methods such as analytical and numerical simulations approach which are inefficient, rigorous time occupying processes, and making it ineffective for precise optimization of antenna design parameters. To meet these challenges for designing complex E-textile antenna structures, machine learning (ML) techniques may be favorable. ML is the subset of artificial intelligence (AI) intended for data analysis and obtaining optimized, efficient results in various applications covering from medical diagnosis to autonomous vehicles. Different evolutionary or heuristic techniques such as fruit fly optimization [2], firefly optimization [3], particle swarm optimization [4], ant colony optimization [5], and genetic algorithms [6] have been explored by the researchers for optimizing antenna structures. All these techniques are based upon population fitness evaluation, searching for the optimum design solution by selecting the best individual until the objective function of global minima or maxima is attained, while perhaps the machine learning optimization algorithms construct a logical-mathematical model according to the input training data that assist in making predictions or obtaining an optimized, efficient solution. Using machine learning techniques, the multiobjective output results can be predicted for any data sample and eliminates the complex process of iteratively finding global or local minimum cost function in heuristic search algorithm-based techniques. The initial work on analyzing and synthesizing printed circuit board (PCB) based microstrip antennas using machine learning techniques has been introduced [7]. The artificial neural network (ANN) model is also a type of ML technique that has been implemented in the field of microstrip antennas [8].

Received 25 December 2022, Accepted 3 February 2023, Scheduled 16 February 2023
* Corresponding author: Arpan H. Shah (shah.arpan@vit.ac.in).
[1] School of Electronics Engineering, Vellore Institute of Technology (VIT), Vellore, Tamil Nadu 632014, India. [2] Electronics Engineering Department, Sardar Vallabhbhai National Institute of Technology (SVNIT), Surat, Gujarat 395007, India.

In [9], an artificial neural network model has been proposed for determining the resonant side length of rectangular microstrip antenna (RMSA) with suspended configuration.

Further, S-shaped RMSA [10] has been investigated for obtaining wideband response using multilayer perceptron ANN. The automated optimization of a double T-shaped monopole antenna using various ML techniques and ANN is also presented [11]. Apart from single element antenna, ML-based algorithms such as support vector machines (SVMs) and clustering methods have also been used to design microstrip antenna arrays [12]. From the literature studies, it has been noticed that ML techniques have been implemented for the optimization of PCB-based microstrip antennas. On the other hand, it is untouched in the field of E-textile antennas. At present, flexible textile technology is leading in all market sectors. Hence, the efficient optimization of E-textile devices using various ML algorithms needs to be primarily addressed, which will play a vital role in performance enhancement and accuracy. The ML-based optimization solutions are computationally effective, best for accurate data prediction, and practically capable of handling complex antenna design structures with more input datasets. This paper has applied various machine learning techniques like deep neural network (DNN), k-nearest neighbor (kNN), Polynomial Regression, and Random Forest Regression for achieving the optimum antenna design. Also, the performance of the antenna based on the predicted optimized values obtained by these techniques is compared with the numerical EM simulation method.

This paper is arranged as follows. Section 2 presents the antenna design geometry and specifies the details of various ML algorithms implemented. In contrast, Section 3 provides the performance results of different ML models that are evaluated based on the objective function. In the end, Section 4 includes the conclusion of this work.

2. MACHINE LEARNING ALGORITHMS

The geometry of the co-planar waveguide fed triangular shape E-textile antenna with comb formation slots inserted on the ground plane is depicted in Figure 1. The proposed antenna design contains cordura ($\varepsilon_r = 1.66$, $\tan\delta = 0.0098$, $h = 0.546\,\mathrm{mm}$) and denim ($\varepsilon_r = 1.89$, $\tan\delta = 0.031$, $h = 1.292\,\mathrm{mm}$) fabric as double-layer substrate materials. The electrical properties of these materials are measured using the resonance method [13] and open stub resonator technique [14]. The radiating patch layer of the antenna consists of electronic textile of copper polyester taffeta fabric from LessEMF [15]. The maximum impedance bandwidth value obtained for each sample data point is evaluated in terms of

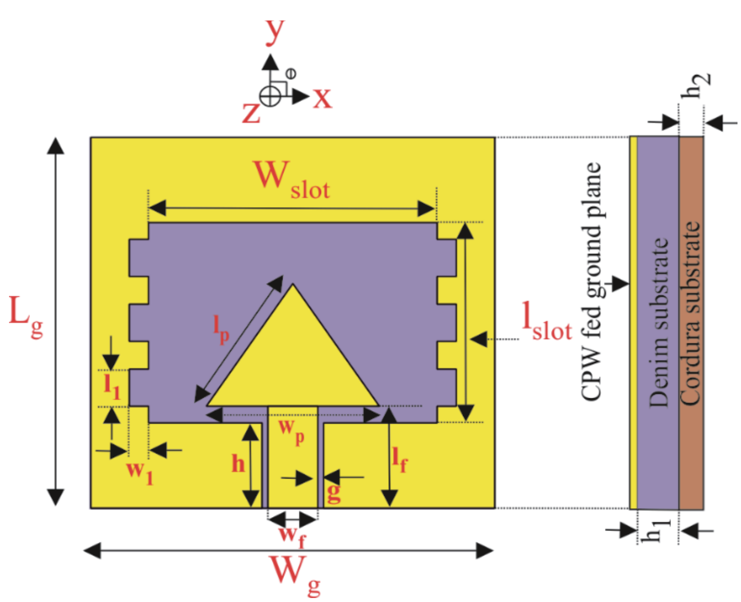


Figure 1. The geometry of the co-planar waveguide fed triangular shape E-textile antenna with comb formation slots inserted on the ground plane.

performance factor (PF), which is given by the mathematical expression as:

$$\mathrm{PF}\,(f_H - f_L) = 1 \leq \mathrm{VSWR} < 2 \tag{1}$$

where f_H and f_L indicate the higher and lower frequencies, and PF is defined as the frequency bandwidth over which the antenna satisfies the 2 : 1 voltage standing wave ratio (VSWR) parameter. The magnitude of return loss (RL) value closely in relation to VSWR is denoted by:

$$\mathrm{RL}\,(\mathrm{dB}) = -20\mathrm{log}\left(\frac{\mathrm{VSWR}-1}{\mathrm{VSWR}+1}\right) \tag{2}$$

The return loss value in the frequency bandwidth region for 2 : 1 VSWR is below -10 dB. The values assigned to the design data points are within the limits of sample space as: $l_{slot} \in [11.2, 10]$, $l_p \in [8.32, 6.82]$, $w_p \in [9, 7.8]$, $l_1 \in [2, 1.4]$ in which the initial three design variables are altered with a step of 0.3 mm, and l_1 is varied by 0.2 mm. In the proposed antenna design, these four design data points are the input predictor variables, and PF is the response variable. The sample data points have four input predictor variables $X = (l_{slot},\, l_p,\, w_p,\, l_1)$ used for training the ML models, and the following predicted response value Y is the performance factor (PF) expressed as:

$$\widehat{\mathrm{PF}} = f(l_{slot}, l_p, w_p, l_1) + \varepsilon \tag{3}$$

where f denotes the mapping function, and ε is the prediction error. The four main design parameters l_{slot}, l_p, w_p, l_1 constitute the impedance bandwidth performance of this proposed antenna device. Hence, these parameters are used as input data for training the ML models. The input data samples are generated by changing the values of these four design parameters, while the remaining parameters are kept at constant values. Here, 409 input data sample points are created that provide the highest R-squared score of 0.90, showing that sufficient input data points are used to train the ML models for obtaining the accurate output prediction value. However, the data set of independent variables (l_{slot}, l_p, w_p, l_1) and dependent or target variable (impedance bandwidth) for 409 sample points are formed. The impedance bandwidth of the antenna is evaluated using Computer Simulation Technology (CST) simulator [16] for all the input data points. The implemented ML models are trained with this prepared data set, and the performance of each model is evaluated based on the mean square error (MSE) values. The designed ML models are now validated with a new set of unseen data input consisting of 20,384 sample points. The main objective of the suggested ML algorithms is basically to predict the impedance bandwidth (performance factor) of the antenna for a given set of input design values. Hence, the response variable is then computed at all the sample points, and the optimal design parameter values are identified, which provides the maximal impedance bandwidth value. This predicted value of maximum bandwidth is verified by the actual value obtained from the CST simulation for the selected optimum design parameter values. The details of the various ML techniques that have been applied to obtain better accuracy prediction models are explained as follows.

2.1. Polynomial Regression

A statistical ML model determines the linear relationship between one or more predictor variables and the response variable. In our proposed design, four predictor variables are used to obtain a single response, which is evaluated using multiple and polynomial regression techniques by implementing them in the Python [17] programming language. In multiple regression, the different predictor variables denoted by x_p, where $p = 1, 2, 3, \ldots N$, and y_p are their respective responses which results in a training data set given by $\{(x_p, y_p),\ p = 1, 2, \ldots N\}$, where $x_p = \{x_{p1}, x_{p2}, \ldots x_{pk}\}$ are k-vector predictor variables. The multiple linear regression model with N input predictor variables generates predicted responses as $\hat{y_p} = \alpha_0 + \beta_1 x_{p1} + \beta_2 x_{p2} + \ldots \beta_p x_{pk}$ where α_0 is the intercept, and $(\beta_1, \beta_2, \ldots \beta_p)$ are the regression coefficients. The general matrix representation of predicted responses is given by:

$$\begin{pmatrix} \hat{y_1} \\ \hat{y_2} \\ \vdots \\ \hat{y_p} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{p1} & x_{p2} & \ldots & x_{pk} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \tag{4}$$

The ordinary least squared method is used to best fit the regression curve by minimizing the sum of squared residuals (SSR) function, which is used to measure the residual errors, $\varepsilon_p = y_p - \hat{y_p}$ that is the difference between the actual value output y_p and the predicted response $\hat{y_p}$ as given by:

$$\begin{aligned} \text{SSR} &= \sum_{p=1}^{N} (\varepsilon_p)^2 \\ &= \sum_{p=1}^{N} (y_p - (\alpha_0 + \beta_1 x_{p1} + \beta_2 x_{p2} + \ldots \beta_p x_{pk}))^2 \end{aligned} \tag{5}$$

We have also applied the polynomial regression method to enhance the prediction accuracy of a model by covering the nonlinear data points between the independent and dependent features. The polynomial regression is a prime class of linear regression in which the polynomial features tool is used from the scikit-learn library to transform the predictor features into a higher degree n^{th} order polynomial. In our case, the quadratic polynomial expression provides a suitable fit regression model with a better R-squared value of 0.87. The predicted response model obtained using polynomial regression with a set of predictor variables $(l_{slot}, l_p, w_p, l_1)$ is given by:

$$\begin{aligned} \widehat{\text{PF}} \;=\; & -20.50 l_{slot}^2 + 52.24 l_p^2 - 0.61 w_p^2 - 9.58 l_1^2 + 3.23 l_{slot} l_p - 5.67 l_{slot} w_p - 0.06 l_{slot} l_1 + 0.59 l_p w_p \\ & +0.01 l_p l_1 + 0.59 w_p l_1 + 0.63 l_{slot} - 0.16 l_p - 0.39 w_p + 0.46 l_1 - 90.94 \end{aligned} \tag{6}$$

2.2. K-nearest Neighbor (kNN)

The kNN regression model [18] predicts the continuous value of new data samples based on the nearest similar data features that exist in the training set. Initially, the distance between the new data point and all the present training data points is measured by the Minkowski, Manhattan, or Euclidean distance methods. By measuring the distance with any one of these methods, the k-nearest data points are selected. After that, the mean value is calculated by picking the values of all these k-nearest neighbor distances, and that value is now the predicted response of the new data point. The nearer neighbor data points favour a more similar response value to the new data point. The optimum value of $k = 3$ has been selected using the GridSerachCV function used for hyperparameter tuning with 5-fold cross-validation imported from the sklearn library and implemented in Python. The best fit value of $k = 3$ is picked such that the mean square error (MSE) of the kNN model is positioned to a minimum value. The kNN model has been implemented using the kNeighborsRegressor ( ) function present in the class of sklearn.neighbors and passing the parameters such as k value and the weighted distance.

2.3. Random Forest Regression

The Random Forest Regression ML technique follows the ensemble method that generates the prediction response value from the multiple decision trees. The Random Forest model is constructed by building various decision trees in which each tree consists of a different random set of data samples. Next, the predicted response data point from each decision tree is averaged to obtain the final response value of the overall Random Forest model. A Random Forest Regression technique operates on both the regression and classification datasets and is also termed as the bagging method. The model is trained by importing the function RandomForestRegressor ( ) from the sklearn package and passing the parameters n_estimators, where n indicates the number of trees generated in the random forest model. We have used $n = 100$ value for which the best fit trained model has been obtained in our case with better accuracy.

2.4. Deep Neural Network (DNN)

A neural network represents the artificial neurons that function in a very similar way related to human brain neurons. DNN (Deep Neural Network) is a branch of artificial neural network (ANN) that comprises multiple hidden layers that interconnect the neurons of an input layer to the output layer. The DNN model can compute complex data and provide better-predicted accuracy results than the

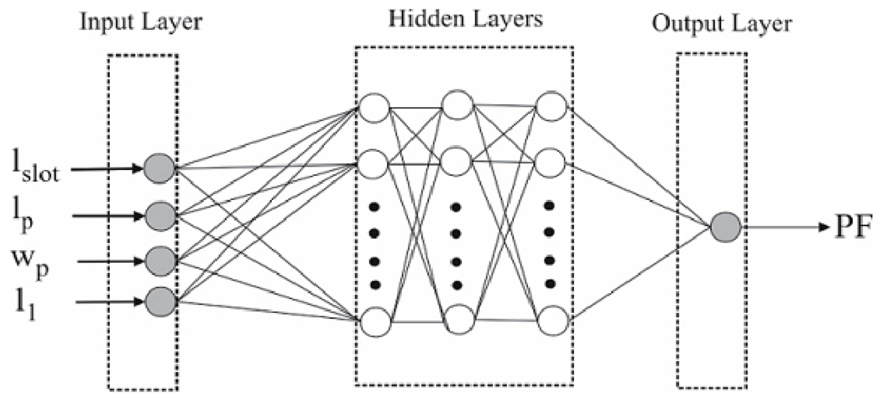


Figure 2. The architecture of deep neural network (DNN).

multilayer perceptron (MLP) model with a single hidden layer reported in the literature [19]. In our case, the DNN model consists of five layers in which it has one input layer, one output layer, and three hidden layers. The predictor variables are applied to the four neurons present in the input layer, which passes the input values x_i to each of the neurons in the first hidden layer, as shown in Figure 2. The input values received at each node j of the hidden layer are multiplied by the corresponding weights w_{ij} followed by adding the bias value b_j with it, and then its output y_j is evaluated by using Rectified Linear Units (ReLU) activation function f as shown in equation.

$$F(k) = \max(0, k)$$
$$F(k) = 0;\ \text{if } k < 0 \quad \text{and} \quad F(k) = k;\ \text{if } k \geq 0 \tag{7}$$

The successive hidden layers repeat the same process to obtain the final response value from the output layer. The DNN model is trained using a back propagation algorithm method [20] with Adam optimizer function, which iteratively updates the weights of a neural network for reducing the error margin between the predicted response value and the target value. Initially, the dataset is split into training and test data samples using Python scikit-learn library. After that, the DNN model is developed in Python using the keras framework [21] and is trained for 5000 iterations or epoch with a learning rate of 0.001. It is observed that using three hidden layers and for 5000 epochs, the minimum value of mean square error (MSE) has been achieved in the proposed DNN model.

3. RESULTS AND DISCUSSIONS

The ML algorithms such as polynomial regression, k-nearest neighbor, random forest regression, and DNN are implemented to select the optimum design data points that provide the maximum impedance bandwidth value. Using ML models, we have obtained the predicted values of impedance bandwidth by considering all the possible sets of design data points for four input predictor variables. The design predictor variables are incremented by 0.1 mm step as $l_{slot} \in [11.2, 10]$, $l_p \in [8.32, 6.82]$, $w_p \in [9, 7.8]$, $l_1 \in [2, 1.4]$ generating the unseen new dataset which has been used in the trained ML models for obtaining the optimal design data points. The maximum impedance bandwidth values predicted by these proposed ML models corresponding to the optimum design data points are compared with the CST simulation solver and listed in Table 1. The design data points calculated by the numerical method are computed in the ML models, and the corresponding impedance bandwidth results predicted by these models are also indicated in Table 1.

The results obtained by various ML models are listed and discussed further in the subsections as mentioned below.

3.1. Polynomial Regression Results

The multiple linear regression model is trained using the train_test_split ( ) function with 327 sample points, and the remaining 82 sample points are used for test data. These account into the input

Table 1. PF values predicted by various ML techniques and comparison with simulation tool.

Optimum Design data points ↓ / PF predicted by →	**Polynomial (GHz)**	**DNN (GHz)**	**Random Forest (GHz)**	**kNN (GHz)**	**CST (GHz)**
Polynomial: l_{slot} = 11.2, l_p = 6.82 , w_p = 7.8, l_1 = 1.4	19.30	20.91	18.31	18.3	20.11
DNN: l_{slot} = 11.2, l_p = 6.82 , w_p = 7.8, l_1 = 1.4	19. 30	20.91	18.31	18.3	20.11
Random Forest: l_{slot} = 11, l_p = 6.92, w_p = 7.9, l_1 = 2	16.16	16.34	18.59	18.53	16.29
kNN: l_{slot} = 10.9, l_p = 6.82 , w_p = 7.8, l_1 = 2	15.38	15.14	18.59	18.63	15.20
By numerical technique: l_{slot} = 10.9, l_p = 8.02 , w_p = 9, l_1 = 1.8	10.87	9.89	10.92	10.84	10.16
Computational time: (seconds)	102	210	133	89	603

dataset of overall 409 sample data points. The accuracy of the model is evaluated by the R-squared score parameter using the test dataset. With multiple linear regression, the R-squared score of 0.72 is obtained between the actual and predicted response values. Further, the prediction accuracy of the model is improved by implementing the polynomial regression technique, which is a subset of the linear regression method. In this technique, the input predictor variables are transformed to the second-order quadratic polynomial by using the fit-transform function. With the polynomial regression model, the R-squared score of 0.87 is observed, and the MSE is reduced from 2.81 to 1.31. The maximum impedance bandwidth predicted by this trained polynomial regression model is equal to 19.30 GHz, which has been accomplished by selecting the input values of predictor variables as: $l_{slot} = 11.2\,\text{mm}$, $l_p = 6.82\,\text{mm}$, $w_p = 7.8\,\text{mm}$, $l_1 = 1.4\,\text{mm}$. The same corresponding values of predictor variables are then applied to the CST simulation tool, and the results are verified. By CST simulation software, the value of impedance bandwidth (performance factor) obtained is 20.11 GHz.

3.2. K-Nearest Neighbor Results

In the k-Nearest Neighbor (kNN) model, initially the best fit k value is selected by the hyperparameter tuning method. The search of k value is done using GridSerachCV ( ) function along with passing the 5-fold cross-validation parameter. It is analyzed that for $k = 3$, the minimum value of MSE of 0.65 has been observed between the actual and predicted values. The maximum impedance bandwidth predicted by the trained kNN model is equal to 18.63 GHz, which has been obtained by analyzing input with unseen sample data points.

Based on the Euclidean distance method, the input predictor values that gives the maximum impedance bandwidth are identified as $l_{slot} = 10.9\,\text{mm}$, $l_p = 6.82\,\text{mm}$, $w_p = 7.8\,\text{mm}$, $l_1 = 2\,\text{mm}$. For these same design parameters, the CST simulation tool provides performance factor $= 15.20\,\text{GHz}$.

3.3. Random Forest Regression Results

As indicated, in the Random Forest Regression model using $n = 100$ multiple decision trees, better prediction accuracy results have been obtained. The proposed model is trained with 80% of data points from the overall sample space, and the remaining 20% is used for the test dataset. With the Random Forest technique, the prediction accuracy of the model is improved, and the minimum value of MSE of 0.33 is observed. The maximum impedance bandwidth result predicted by the Random Forest Regression model is 18.59 GHz based on averaging value, which occurs at the corresponding design parameters $l_{slot} = 11\,\mathrm{mm}$, $l_p = 6.92\,\mathrm{mm}$, $w_p = 7.9\,\mathrm{mm}$, $l_1 = 2\,\mathrm{mm}$. The predicted response value is also verified through CST simulation for the same set of design parameters. The CST simulation result shows an impedance bandwidth of 16.29 GHz, which can be stated as a close match with the predicted response value by the Random Forest model.

3.4. Deep Neural Network Results

The proposed deep neural network model is developed with three hidden layers in which each hidden layer contains 64 neurons, followed by single input and output layers. A maximum of 80% of the data set is used for training the neural model from the overall sample space. To improve the performance of the DNN, the mean and standard deviation scaling method has been applied to the input predictor variables. The mathematical expression representing the scaling function is given by:

$$\text{Scaled input} = \frac{\text{Input-mean}}{\text{standard deviation}} \tag{8}$$

After feature scaling, the neural model is built using a keras sequential framework consisting of hidden layers with Rectified Linear Unit (ReLU) activation function and the output layer with a linear activation function. The remaining 20% of the data is used for testing the model, in which we have observed that the predicted response value is closer to the target value and can be represented by the regression line as shown in Figure 3. The maximum number of data points from the test data set has an error margin difference below one between the predicted response values and target values as depicted by the bar count graph indicated in Figure 4. The minimum value of MSE of 0.03 has reached 5000 epoch

Figure 3. Regression line curve of DNN model.

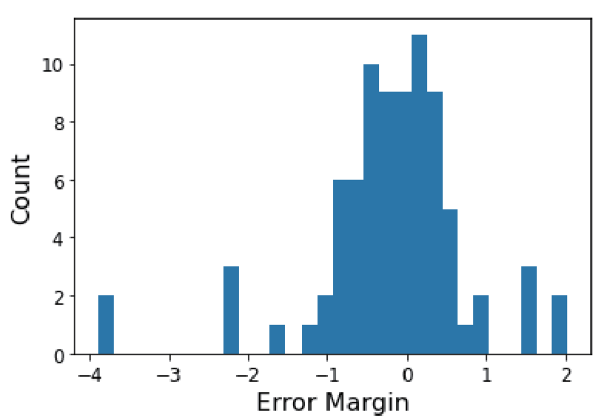


Figure 4. Error margin graph between the predicted response values and target values of the DNN model.

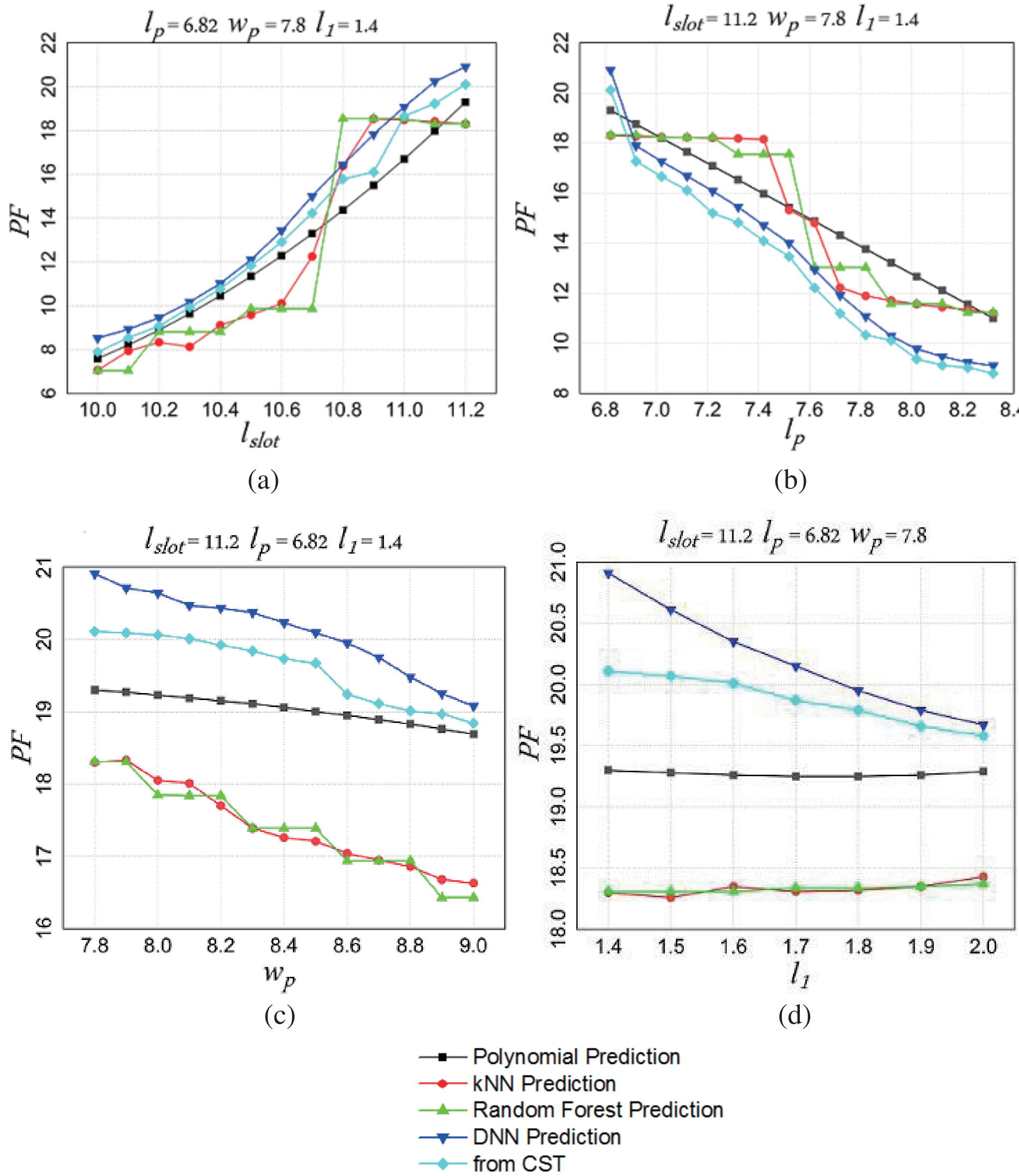
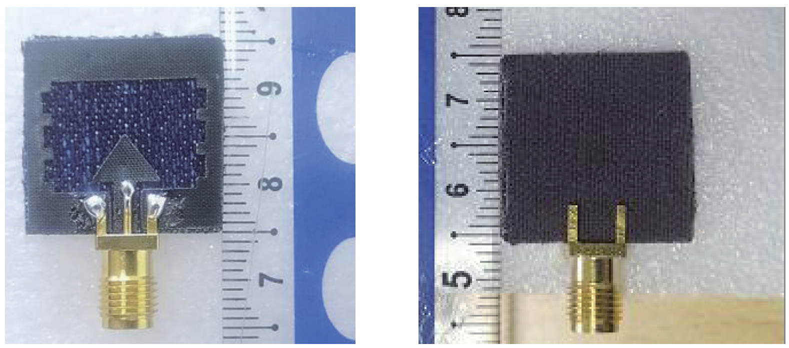


Figure 5. Performance factor values predicted by these proposed ML techniques with respect to variation in each of four design parameters (a) l_{slot}, (b) l_p (c) w_p, and (d) l_1 respectively whereas the other three parameters are kept constants as shown on the above upper line of each plot.

point. The model is then validated with the unseen data set consisting of 20384 design points. The selected design point $l_{slot} = 11.2\,\text{mm}$, $l_p = 6.82\,\text{mm}$, $w_p = 7.8\,\text{mm}$, $l_1 = 1.4\,\text{mm}$ provides the maximum impedance bandwidth value of 20.91 GHz, as predicted by the proposed DNN model. For the same set of design point, the CST result gives the impedance bandwidth equal to 20.11 GHz.

The impedance bandwidth value predicted by these proposed ML techniques by varying each design parameter one at a time is shown in Figure 5. It can be noticed from the graph plots that the predicted response values from DNN and polynomial regression are close to the target value obtained from CST simulation. Therefore, the maximum response values predicted by both DNN and polynomial regression techniques are for the same set of sample data point $l_{slot} = 11.2\,\text{mm}$, $l_p = 6.82\,\text{mm}$, $w_p = 7.8\,\text{mm}$, $l_1 = 1.4\,\text{mm}$. With this data point, the design of CPW-fed patch along with the ground plane is accurately constructed by using a fabric laser cutting machine as depicted in Figure 6, and the

Figure 6. Fabric laser cutting machine is used for proposed antenna design fabrication.



(a) (b)

Figure 7. Fabricated prototype of proposed antenna (a) front view and (b) back view.

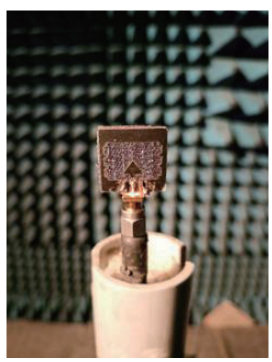


Figure 8. Reflection coefficient measurement of antenna in an anechoic chamber.

corresponding fabricated prototype of optimized antenna design is shown in Figure 7. The reflection coefficient plot is then measured in an anechoic chamber as depicted in Figure 8, as well as observed in CST simulation software, and thereafter it has been compared with the results obtained from the numerical method, as shown in Figure 9.

The data point values predicted by DNN and polynomial regression provide additional impedance bandwidth against the data point values calculated by numerical technique. This shows that better prediction accuracy results are observed by using these ML methods. With DNN and polynomial

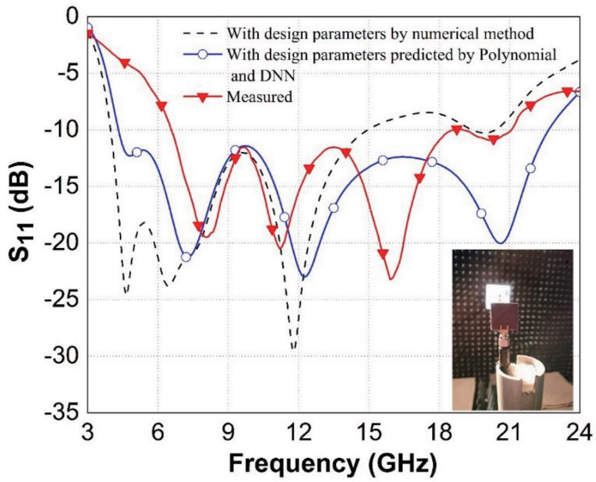


Figure 9. Simulated reflection coefficient plot for the design parameters values by numerical method and simulated as well as measured reflection coefficient plot for the design parameters values predicted by polynomial and DNN.

regression algorithms, better optimization results can be achieved than EM computational solvers. Also, the computational time required by the CST simulation solvers to perform optimization for a single set of data point values is quite more than the time taken by ML algorithms. The main benefit of implementing these ML techniques is that they can easily compute a huge amount of sample data points within a fraction of seconds, and also it supports in selecting the best optimum solution in a faster way. Hence, we can say that this work gives affirmation regarding implementing DNN technique on python platform, provides better prediction accuracy, and can compute complex structures with further more design parameters for obtaining an optimized solution. The proposed textile antenna design using ML is compared with the reported literature works, as shown in Table 2.

Table 2. Comparison of a proposed textile antenna with the other reported literature work.

Ref.	**Machine Learning Algorithms used**	**Predicted response variable**	**Material used for antenna fabrication**
[8]	Support Vector Machine	$\lvert S_{11}\rvert \leq -10\,\mathrm{dB}$	RT/duroid 5880
[9]	Multi-level perceptron	$\mathrm{L_{patch}}$ (RMSA)	Glass Epoxy substrate
[10]	Artificial Neural Network (ANN)	Wideband frequency (2.354–2.894 GHz)	Foam substrate
[11]	Lasso, ANN, and kNN	Frequency bands (2.4–3.0 GHz) and (5.15–5.6 GHz)	FR4 substrate
[12]	Support Vector Machine	Gain = 7.0 dBi and VSWR = 1.433	Printed Circuit Board ($\varepsilon_r = 4.7$)
Proposed Work	Polynomial, kNN, Random Forest, and DNN	Ultra Wideband frequency (3–24 GHz)	Fully textile material

4. CONCLUSION

This work shows that the optimized values of antenna design parameters can be determined by implementing different ML techniques such as polynomial regression, k-nearest neighbor (kNN), random forest regression, and deep neural network (DNN) for obtaining the efficient results of impedance bandwidth of antenna. A detailed explanation about using different ML models in the design of E-textile antenna and deploying DNN on the python platform has been presented in this paper. It has been observed that these ML models trained for the proposed textile antenna design provide the best-fit response values with less computational time for 20384 sample data points than the EM simulation technique. Both the polynomial regression and DNN provide better prediction accuracy than kNN and random forest models. In brief, this work shows that implementing ML techniques is a gateway for obtaining the efficient design solution in the E-textile antenna domain, and it tends to be favourable in the field of IoT and medical applications.

5. DISCLOSURE STATEMENT

No potential conflict of interest was reported by the author(s).

ACKNOWLEDGMENT

The authors thank to School of Electronics Engineering (SENSE), Vellore Institute of Technology (VIT), Vellore, and Sensor laboratory, Sardar Vallabhbhai National Institute of Technology, Surat, for all the assistance provided by the department and administration to carry out this research work is highly appreciated.

REFERENCES

1. Gao, G. P., B. Hu, S. F. Wang, and C. Yang, "Wearable circular ring slot antenna with EBG structure for wireless body area network," *IEEE Antennas Wirel. Propag. Lett.*, Vol. 17, No. 3, 434–437, 2018.
2. Polo-López, L., J. Córcoles, and J. A. Ruiz-Cruz, "Antenna design by means of the fruit fly optimization algorithm," *Electron.*, Vol. 7, No. 1, 2018.
3. Yoshimoto, E. and M. V. T. Heckler, "Optimization of planar antenna arrays using the firefly algorithm," *J. Microwaves, Optoelectron. Electromagn. Appl.*, Vol. 18, No. 1, 126–140, 2019.
4. Jin, N. and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations," *IEEE Trans. Antennas Propag.*, Vol. 55, No. 3 I, 556–567, 2007.
5. Zhu, D. Z., P. L. Werner, and D. H. Werner, "Design and optimization of 3-D frequency-selective surfaces based on a multiobjective lazy ant colony optimization algorithm," *IEEE Trans. Antennas Propag.*, Vol. 65, No. 12, 7137–7149, 2017.
6. Kiehbadroudinezhad, S., N. K. Noordin, A. Sali, and Z. Z. Abidin, "Optimization of an antenna array using genetic algorithms," *Astron. J.*, Vol. 147, No. 6, 2014.
7. Silveira, D., et al., "Improvements and analysis of nonlinear parallel behavioral models," *Int. J. RF Microw. Comput. Eng.*, Vol. 19, No. 5, 615–626, 2009.
8. Wu, Z., Y. Yang, and Z. Yao, "Multi-parameter modeling with ANN for antenna design," *IEEE Antennas Propag. Soc. Int. Symp. Usn. Natl. Radio Sci. Meet. APSURSI 2018 — Proc.*, Vol. 66, No. 7, 2381–2382, 2018.
9. Deshmukh, A. A., S. D. Kulkarni, A. P. C. Venkata, and N. V. Phatak, "Artificial neural network model for suspended rectangular microstrip antennas," *Procedia Comput. Sci.*, Vol. 49, No. 1, 332–339, 2015.

10. Aneesh, M., A. Singh, K. Kamakshi, and J. A. Ansari, "Performance investigations of S-shaped RMSA using multilayer perceptron neural network for S-band applications," *Radioelectron. Commun. Syst.*, Vol. 62, No. 8, 400–408, 2019.
11. Sharma, Y., H. H. Zhang, and H. Xin, "Machine learning techniques for optimizing design of double T-shaped monopole antenna," *IEEE Trans. Antennas Propag.*, Vol. 68, No. 7, 5658–5663, 2020.
12. Zheng, Z., X. Chen, and K. Huang, "Application of support vector machines to the antenna design," *Int. J. RF Microw. Comput. Eng.*, Vol. 21, No. 1, 85–90, 2011.
13. Sankaralingam, S. and B. Gupta, "Determination of dielectric constant of fabric materials and their use as substrates for design and development of antennas for wearable applications," *IEEE Trans. Instrum. Meas.*, Vol. 59, No. 12, 3122–3130, 2010.
14. Lätti, K. P., J. M. Heinola, M. Kettunen, J. P. Ström, and P. Silventoinen, "A review of microstrip T-resonator method in determination of dielectric properties of printed circuit board materials," *IEEE Instrum. Meas. Technol. Conf.*, Vol. 1, No. 5, 62–66, 2005.
15. The EMF Safety shop, [Online] Available:http://www.lessemf.com.
16. Computer Simulation Technology, 2016, http://www.cst.com.
17. Faouzi, J. and H. Janati, "Pyts: A python package for time series classification," *J. Mach. Learn. Res.*, Vol. 21, 1–6, 2020.
18. Cui, L., Y. Zhang, R. Zhang, and Q. H. Liu, "A modified efficient KNN method for antenna optimization and design," *IEEE Trans. Antennas Propag.*, Vol. 68, No. 10, 6858–6866, 2020.
19. Kuri-Morales, A., "Closed determination of the number of neurons in the hidden layer of a multi-layered perceptron network," *Soft Comput.*, Vol. 21, No. 3, 597–609, 2017.
20. Kumar, R., P. Kumar, S. Singh, and R. Vijay, "Fast and accurate synthesis of frequency reconfigurable slot antenna using back propagation network," *AEU — Int. J. Electron. Commun.*, Vol. 112, 152962, 2019.
21. Haghighat, E. and R. Juanes, "A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks," *Comput. Methods Appl. Mech. Eng.*, Vol. 373, 113552, 2021.