Received: 23 March 2019 | Revised: 5 July 2019 | Accepted: 29 July 2019

DOI: 10.1002/jnm.2682

SPECIAL ISSUE ON ADVANCED SOLUTION METHODS FOR MODELING COMPLEX ELECTROMAGNETIC PROBLEMS

WILEY

Deep learning base modified MLP model for precise scattering parameter prediction of capacitive feed antenna

Nurullah Calik[1] | **Mehmet Ali Belen**[2] | **Peyman Mahouti**[3]

[1] Electronics and Communications Engineering, Yildiz Technical University, Istanbul, Turkey

[2] Electrical and Electronics Engineering, Artvin Coruh University, Artvin, Turkey

[3] Electrical and Electronics Engineering, Istanbul Arel University, Istanbul, Turkey

Correspondence
Nurullah Calik, Electronics and Communications Engineering, Yildiz Technical University, Istanbul, Turkey.
Email: ncalik@yildiz.edu.tr; nurullah.calik@gmail.com

Funding information
Yildiz Teknik Üniversitesi, Grant/Award Number: FAP-2018-3427

Abstract

The relations between the antennas' geometrical parameters and design specifications usually consist of linear and nonlinear components. Especially with the increase of the requested performance measures, the design procedure becomes much more complex due to the conflicting performance criteria or design limitations. To achieve a design with high performance with feasible design parameters, a fast, accurate, and reliable design optimization process is required. Herein, to have a fast, accurate, and high-performance capacitive-feed antenna model to be used in design optimization problems, a modified multi-layer perceptron (M2LP) model has been proposed. The M2LP is an equivalent convolutional neural network (CNN) model of a standard multilayer perceptron (MLP), where instead of traditional training parameters of MLP, more advanced training parameters of CNN models such as batch-norm layer, leaky-rectified linear unit (ReLU) layer, and Adam training algorithm had been used. Furthermore, the M2LP model had been used in a design optimization process and the obtained optimal antenna had been prototyped using 3D printing technology for justification of the proposed M2LP model with experimental results. As can be seen from the results, the proposed M2LP model is a fast, accurate, and reliable regression model for design optimization of microwave antennas.

KEYWORDS

convolutional neural network, deep learning, modified multi-layer, microwave antenna design, microstrip antenna, regression, perceptron

1 | INTRODUCTION

In most of the wireless communication systems, there are requirements such as low profile designs with performance criteria such as high gain, specific radiation pattern, or wide operation band. One of the most commonly used low profile antenna design is microstrip antenna (MSA). MSA design has been extensively being used in many applications such as high-performance aircraft, spacecraft, satellites, and missiles communication stages where size, weight, cost, performance, and ease of installation are at most importance.[1-3] The main design elements in the modeling of an MSA are radiation element, dielectric substrate, feed network, or feed element. The radiation element in MSA design can take many forms such as rectangular, circular, triangular, and any other configuration. As for the feed network, There are many types of methods for feed network for patch antenna designs such as pin feed, line feed, or aperture coupling. Another method for feeding of radiation element is to use of a probe-fed capacitor patch element next to the radiation element.[4]

https://doi.org/10.1002/jnm.2682

The advantage of using capacitive feed designs is that the inductive impedance of the probe is effectively cancelled. Thus, the designed antenna can achieve a wide operation band and improvement on the radiation efficiency of the design.

The relations between the antennas' geometrical and design specifications usually consist of linear and nonlinear components such as the relation of antenna size to operation frequency, radiation pattern, and return loss.[5,6] Especially with the increase of the requested antenna performance measures such as operation bandwidth, gain, and specified radiation pattern, the design procedure of the antenna becomes much more complex due to the complex inner relationships between the geometrical design specifications and the requested performance criteria. For creating a high-performance antenna design within the feasible design parameters an optimization process based on an accurate and reliable model is needed. In such a case, the designer should either use a low accurate coarse model for computationally efficient optimization process or a highly accurate fine design model with the low computational efficient optimization process. For the last decades, many studies have been done on creating numerical or analytical methods for accurate models for the design of microwave stages, one of the most commonly used numerical methods is artificial neural network (ANN) models.[7-10] Commonly, either measured or simulated results of microwave designs are being used for creating ANN-based circuit models. Even though the gathering of training and test data might become a considerable effort, once the ANN model is created the overall computation duration of estimation can be overlooked, most especially during a design optimization process. Thus, by using ANN models, it is possible to achieve a computationally efficient solution for design optimization of microwave stages based on either simulation or measurement results. However, in case of a design problem with large scale data set with complex inner relationship between variables and outputs, ANN models fail to exploit spatially local correlation due to their nature that tries to include all the possible relations between input and output layers in a single layer.

With the recent development in high-end hardware systems deep learning (DL) algorithms have become a new solution method for many challenging complex and enormous-sized problems such as human activity recognition using mobile sensors,[11] approach for estimation of remaining useful life of a subsystem or a component using sensor data,[12] for financial time series,[13] wind power or wind speed prediction,[14,15] electrical impedance tomography (EIT) imaging,[16] and full-wave nonlinear inverse scattering problems.[17] convolutional neural network (CNN) is a class of deep-neural networks inspired by the biological process of animals visual cortex.[18-21]

The CNN architecture is similar to multilayer perceptron (MLP) models, where both models are trained with a version of the back-propagation algorithm, and both of them consist of input, output, and hidden layers. However, in CNN, the hidden layer consists of multiple convolutional layers, rectified linear unit (RELU) layers, pooling-layers, fully connected layers, and normalization layers. The main difference of CNN form the MLP is that, in case of modelling problem with high number of input variables, it is improbable to connect neurons to all neurons in the previous layer because not only an architecture like that might not take the spatial structure of the data set into account, but also it would extremely increase the duration of training process. CNN exploit spatially local correlation by enforcing a sparse local connectivity pattern, similar to the biological process of animals visual cortex, between neurons of neighbouring layers: each neuron is only connected to a limited region of the previous layer. In Figure 1, an example schematic of a CNN network has been presented. However, CNN models have a disadvantage of exorbitant training process compared with counterpart numerical models, due to their nature, which usually requires a high-end performance hardware setup such as GPU.

In this work, to achieve highly accurate and computationally efficient antenna model to be used in a design optimization problem, a modified multi-layer perceptron (M2LP) model has been proposed. The proposed M2LP is an equivalent CNN model of a standard multi-layer percptron (MLP). Where instead of traditional parameters of MLP such as "sigmoid"

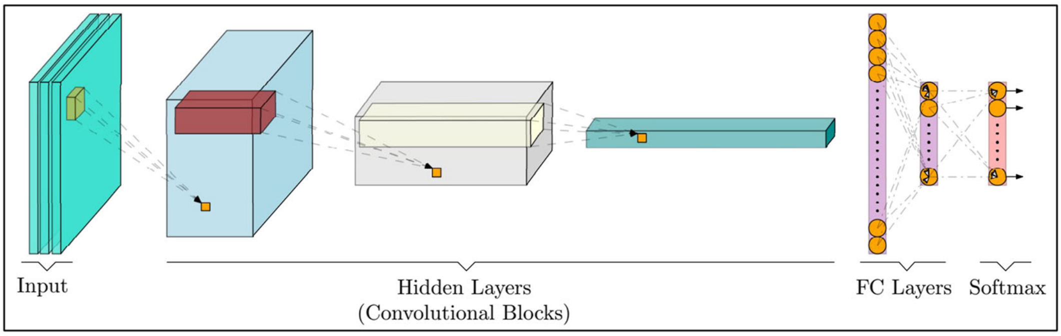


FIGURE 1 Basic block structure of convolutional neural network (CNN) for image processing applications

and "tanh" as for activation functions, Levenberq Marquart, and Bayesian regression methods as for training the models, more advanced design parameters of DL algorithms such as batch norm layer, leaky ReLU, parametric ReLU, and Adam training algorithm specifically developed for training of accurate models had been used. In section 2, design of a capacitive feed antenna (CFA) model had been proposed alongside its design variables and their upper-lower constraint for creating a 3D EM-based data set to be used in training and test process of the proposed M2LP model. In section 3, the methodology of the proposed M2LP model had been explained in details alongside its design parameters. In section 4, a study case for performance evaluation of proposed M2LP with its traditionally counterpart model based on different design parameters had been studied. Furthermore, the proposed M2LP model has been used in a design optimization process and the obtained optimal geometrical design parameters of the antenna have been used for prototyping and justification of the proposed methodology via experimental results. Finally, the work ends with a brief conclusion section in section 5.

2 | CAPACITIVE FEED ANTENNA DESIGN

As is mentioned before, one of the methods for feeding of a radiation element is to use of a probe-fed capacitor patch element next to the radiation element to achieve a wide operation band and improvement on the radiation efficiency of the design. In this section, an example design of a CFA had been presented in Figure 2. Two dielectric layers had separated the top and ground layer of the design. The top layer consists of a rectangular radiation element and a circular pin feed element.

In the studied CFA design, some important geometrical design parameters have a specific effect on the design performance measures. As an example, the increase in the length of the radiation element would decrease the operating frequency band and vice versa. Increase in the substrate height would increase the bandwidth. Decreasing the feed spacing would increase the real part of input impedance, and the increase in the feed diameter would increase the imaginary part of input impedance. Herein for the sake of simplicity on study case, the shape of the radiation element had been taken as a square patch. The data sets for training and test process of M2LP models have been given in Table 1, which are generated by using 3D Electromagnetic Simulation Tool CST Microwave Studio.

In this work, it is aimed to create a black box model of the CFA in Figure 2 by using the data sets given in Table 1. The proposed black box model can be seen in Figure 3, where the selected geometrical design parameters are taken as input variables and the real and imaginary part of the return loss characteristics are the requested outputs. Additionally,

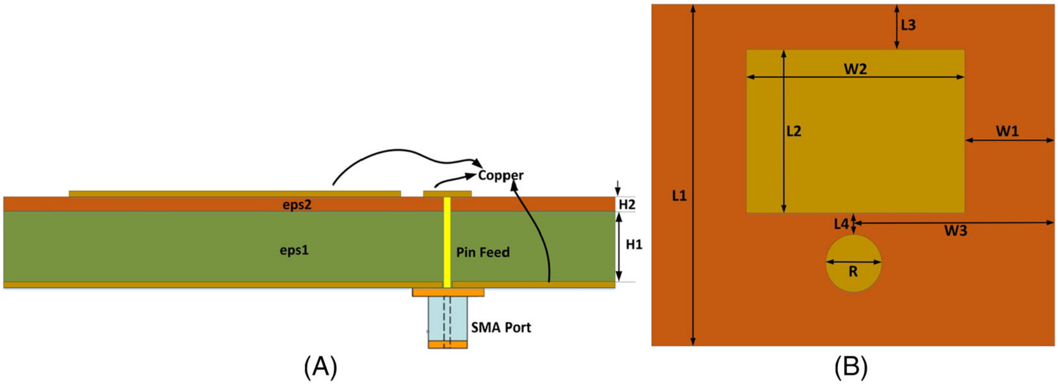


FIGURE 2 A, Side and B, top schematic view of the capacitive feed antenna (CFA) design

TABLE 1 Training and test sets specifications (mm)

	Training		Test	
Parameter	**Range**	**Step Size**	**Range**	**Step Size**
R	{4 , 6 , 8}	2	{5 , 7}	2
L4	[0.5 , 2.5]	0.5	[0.5 , 2.5]	0.5
L2	[20 , 60]	5	[20 , 60]	5
H1	[10 , 20]	5	[10 , 20]	5
H2	[0.25 , 1]	0.25	[0.25 , 1]	0.25
f: Frequency (GHz)	[1,10]	0.5	[1,10]	0.5
Total sample	30 780		20 520	

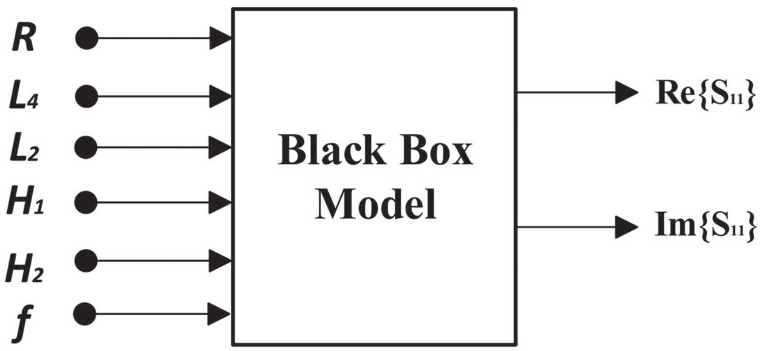


FIGURE 3 Black box representation of the capacitive feed antenna (CFA)

TABLE 2 Some samples from the training and test data sets

	Input						Output	
Dataset	R	L4	L2	H1	H2	f	Real	Imaginer
Training	4	2.5	60	20	1	10	0.717	-0.029
	6	2	50	20	0.75	3	0.775	0.196
	8	0.5	20	15	0.5	1.5	0.281	-0.920
Test	5	1	30	10	1	7.5	0.797	-0.168
	7	1.5	40	20	0.25	6	0.740	-0.483

in Table 2, some samples from the training and test data sets have been given. In the next section, the design methodology of the proposed CNN based MLP model is given in details.

3 | MODIFIED MULTI LAYER PERCEPTRON

DL has emerged in the field of image processing and achieved a high superiority compared with handcrafted feature extraction and classification algorithms. In particular, the successes of classification problems have been overcome even at human level.[22,23] DL has a wide range application area such as segmentation,[24,25] multi-object tracking,[26,27] biomedical applications,[28,29] and even lip reading.[30]

CNN is a subfield of DL. CNN, based on tensor convolution, unlike MLP, uses filter kernels with $h \times w$ (height, width) dimensions where local information is processed instead of fully-connected layers. As an example, in Figure 4, the input tensor image is in $H \times W \times C$ dimensions and the filter dimensions are given as $3 \times 3 \times C$, where C is channel size. When each filter is multiplied and summed by the values in the tensor, it produces only one value and therefore the depth of the output tensor resulting from the convolution of the filter tensor with the input tensor is 1. In the example in Figure 4, the output matrices of $H' \times W' \times 1$ dimensions for each filter kernel are concatenated in the third dimension to form the input tensor belonging to the next layer. The output of $H' \times W' \times C'$ dimensions occurs after C' filters are processed. The tensor convolution for the corresponding sample is given in Equation (1).

$$X^{\ell+1}(i,j,k) = \sum_{m=-1}^{1}\sum_{n=-1}^{1}\sum_{u=1}^{C} X^{\ell}(i-m, j-n, u) \times h_k^{\ell}(m,n,u). \tag{1}$$

With the development of CNN, there has been essential development in training algorithms, weighted initialization methods,[31] and suggestion of new layers in artificial intelligence.[32-35] To update the weights, algorithms such as SGD, AdaGrad,[36] AdaDelta,[37] and Adam,[38] which use stochastic processing, were released, and batch normalization (BN) layer[32] was used to eliminate saturation and vanishing gradient problems in a network. In addition to these, it has been proposed to use new activation functions such as ReLU, leaky-ReLU, and exponential linear unit (ELU) instead of tanh and sigmoid, which are frequently used in MLP. Also, due to the improved graphics processing unit (GPU) parallel processing capabilities, a new framework has been eventuated to handle large-scale data and to tackle the difficult problems.

CNN-based regression is a relatively new issue.[39-41] The use of DL in the regression field is not so rich compared with its use in areas such as classification, detection, and segmentation. For regression, it is necessary to design specific network models for the nature of each problem and determine the appropriate loss function. In this context, by adapting the innovations in the CNN field to MLP, we recommend that a novel M2LP model be used to estimate antenna parameters. M2LP

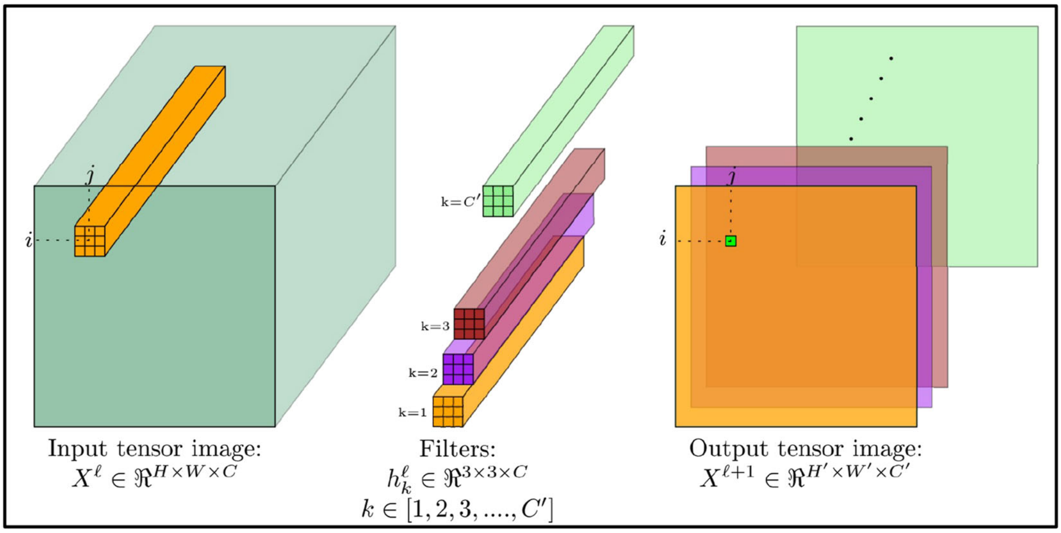

j
i
k=C′
k=3
k=2
k=1
j
i
Input tensor image:
$X^\ell \in \Re^{H\times W\times C}$
Filters:
$h_k^\ell \in \Re^{3\times 3\times C}$
$k \in [1, 2, 3, ..., C']$
Output tensor image:
$X^{\ell+1} \in \Re^{H'\times W'\times C'}$

FIGURE 4 Structure of tensor convolution operation: when a filter is convoluted with the input tensor, an output of $H' \times W' \times 1$ is produced. The size of the output tensor is $H' \times W' \times C'$ by filtering the input tensor of the C' pieces filter. As an example, h_k^ℓ filters are selected $3 \times 3 \times C$

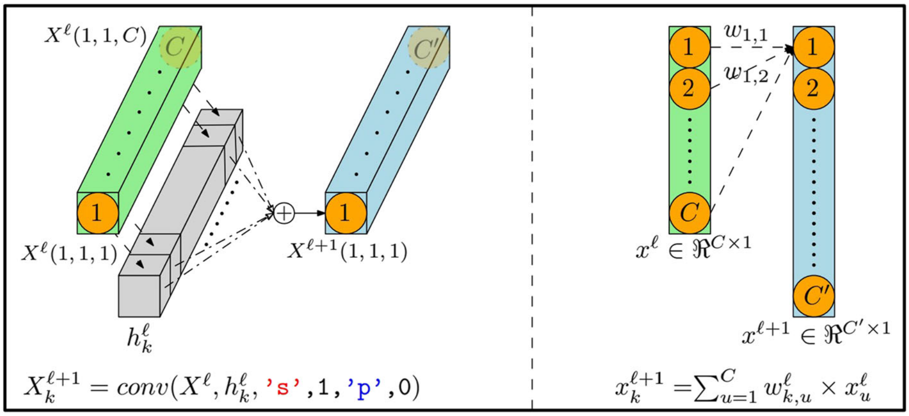

$X^\ell(1,1,C)$
C
C′
1
$X^\ell(1,1,1)$
1
$X^{\ell+1}(1,1,1)$
h_k^ℓ
$X_k^{\ell+1} = conv(X^\ell, h_k^\ell, \texttt{'s'}, 1, \texttt{'p'}, 0)$
1
2
C
$w_{1,1}$
$w_{1,2}$
1
2
C′
$x^\ell \in \Re^{C\times 1}$
$x^{\ell+1} \in \Re^{C'\times 1}$
$x_k^{\ell+1} = \sum_{u=1}^{C} w_{k,u}^\ell \times x_u^\ell$

FIGURE 5 Equivalent structure of multilayer perceptron (MLP): $\mathbf{x}^\ell$ is in vector form in MLP, while in convolutional neural network (CNN), it is in tensor form as $\mathbf{X}^\ell$. In the case where the data tensor is $\mathbf{X} \in \Re^{1\times 1\times C}$, the convolution operator produces the same results as MLP under the parameters stride $s = 1$, padding $p = 0$, and number of filter $@ = C'$. Therefore, this structure of convolution is equivalent to MLP

is created by using new layers and functions on MLP. The basis of this lies in the fact that the equation in Equation (1) can be reduced to MLP operations.

$$X^{\ell+1}(1,1,k) = \sum_{u=1}^{C} X^\ell(1,1,u) \times h_k^\ell(1,1,u). \tag{2}$$

A convolution process needs two important parameters in addition to the filter dimensions: 1-stride, 2-padding. The stride (s) represents the step size of the filter kernel, while padding (p) determines the zero thickness to be added around the image tensor. Let the image tensor has a $1 \times 1 \times D$ size. If Equation (1) is rewritten for s = 1 and p = 0, we get Equation (2). Equation (2) reflects the equivalence of the calculation for MLP, as shown in Figure 5. In the CNN, the neurons are arranged in the 3D instead of vertically aligned. The advantage of this representation is that these layers and functions developed on any DL platform can be used to construct innovative MLP networks. Some of these layers are BN and as an activation function ReLU.

The ReLU is one of the significant layers offered instead of sigmoid and tanh functions which have vanishing gradient problem in negative and positive regions. The derivable region of sigmoid and tanh is very limited. On the other hand, ReLU (Equation (3)) can be differentiated in the whole positive region. In this way, the activation function remains operable. The vanishing gradient problem may also occur in the negative region for ReLU. To prevent this, *leaky ReLU* Equation (4) is proposed in Xu et al.[42] Leaky ReLU permits negative values to be passed by using an α factor selected from (0, 1), unlike in ReLU. Another advantage of ReLU-type activation functions is that they do not contain exponential expressions, and the cost of forward and backward computational complexity is low.

$$ReLU(x) = \begin{cases} \text{x} & x \geq 0 \\ 0 & otherwise \end{cases} \tag{3}$$

$$LeakyReLU(x) = \begin{cases} \text{x} & x \geq 0 \\ \alpha x & otherwise \end{cases} \tag{4}$$

The BN layer given in Algorithm 1 (in Table 3) is an important layer developed for DL, which is not used in standard MLP designs. This layer is usually used between *conv.* and *ReLU* layers for whitening the mini-batch statistics. After whitening, mini-batch is multiplied with γ and summed with β to avoid the saturation region of the incoming ReLU layer. γ and β are learnable parameters. Hence, mini-batches are specifically redesigned to reduce loss function for every activation function. BN also helps to create more stable results by eliminating the sensitive dependence of networks on initialization values. Furthermore, there is the ability to minimize the vanishing gradient problem.

Algorithm 1 Algorithm-1: Batch Normalization

```
Input: $\mathbb{B} = \{X_{1\cdots m}\}, \gamma, \beta$        ▷ Mini Batch Set and Learnable Parameters
Output: $Y_i = BN_{\gamma,\beta}[X_i]$
    $\mu_{\mathcal{B}} \leftarrow \mathbb{E}[X_i]$
    $\sigma^2_{\mathcal{B}} \leftarrow \mathbb{E}\left[(X_i - \mu_{\mathcal{B}})^2\right]$        ▷ zero meaning
    $\hat{X}^i \leftarrow \frac{X^i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}}$        ▷ normalization
    $Y_i \leftarrow \gamma \hat{X}^i + \beta$        ▷ scaling
    return $y_i$
```

The algorithm that generates the M2LP network according to these layers is given in Algorithm 3. In terms of having a fair comparison, the same algorithm is used in the construction of MLP networks (Algorithm 2). The difference between MLP and M2LP is the use of BN and leaky ReLU layers in M2LP. Algorithms accept $\mathbf{x}_i \in \Re^{1\times1\times6}$ as the input, since the length of the data is 6. They also work with two input parameters: block depth (BD) and first neuron size (FNS). BD specifies the hierarchical increased architecture of the model. Neurons of generated twin blocks are designed to be the twice neuron of the previous block. Each block includes two layers which have the same neuron size. In the last part of the network, there are regression neurons with Œ2 expansion layer. The FNS designates neuron distribution of the network. Figure 6 is given as an example for constructing a model with BD:{1,2} FNS: 16 parameters. In the algorithms, a function called *conv(·)* works with three parameters: 1-input tensor, 2-filter depth, and 3-filter number.

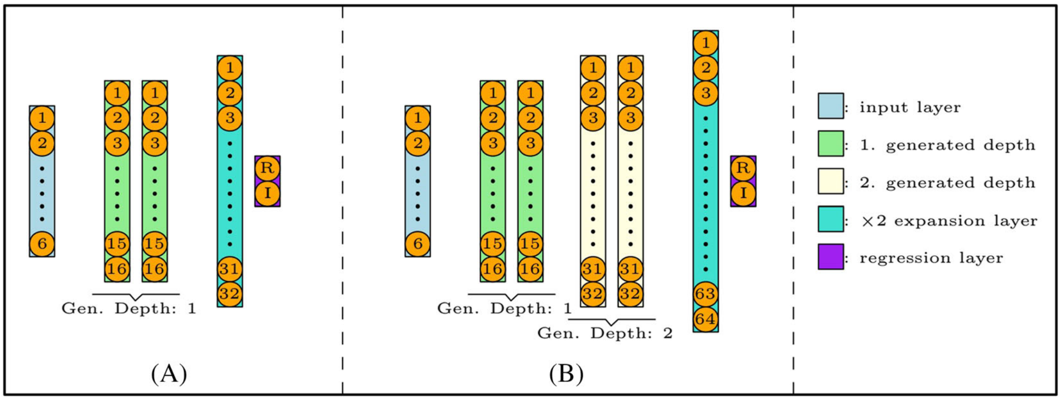


FIGURE 6 Outputs of the network generating algorithm: all networks have one input layer which has 6 neurons, and ×2 expansion + regression layers. Therefore, if the block depth and first neuron size are 1 and 16, the algorithm produces A,. Another case, if the parameters are 2 and 16, the algorithm produces B,. This procedure is same both multilayer perceptron (MLP) and modified multi-layer perceptron (M2LP) for a fair comparison. The differences are the inner structure of layers

TABLE 3 Neural network generating algorithms

Algorithm-2: MLP Generating Pseudo-code

```
Input: $\mathbf{x}_i \in \Re^{1\times1\times6}$,          ▷ Input tensor
$G \in \{16, 32\}$,                                       ▷ Generated depth neuron size
$\#_G \in \{1, 2\}$,                                      ▷ Depth indicies
Output: $\mathbf{r} \in \Re^{1\times1\times2}$            ▷ Regression layer (Real, Imag)
Parameters: $N_i, N_o$                                    ▷ Input and output neuron size of layer

$\mathbf{x}_m = Conv(\mathbf{x}_i, 6, G\{1\})$
  for $k \in \#_G$ do
    $\mathbf{x}_m \leftarrow Conv(\mathbf{x}_m, G\{k\}, G\{k\})$
    if $k \neq \#_G(end)$ then
      $\mathbf{x}_m \leftarrow Conv(\mathbf{x}_m, G\{k\}, G\{k+1\})$
    end if
  end for

  $\mathbf{x}_m = Conv(\mathbf{x}_m, G\{end\}, 2 \times G\{end\})$
  $\mathbf{r} = conv(\mathbf{x}_i, 2 \times G\{end\}, 2)$
function: Conv($\mathbf{x}$ , $N_i$ , $N_o$)
$\mathbf{x} \leftarrow conv(\mathbf{x}, N_i, N_o)$;
$\mathbf{y} \leftarrow tanh(\mathbf{x})$;
end function
```

Algorithm-3: M2LP Generating Pseudo-code

```
$\mathbf{x}_i \in \Re^{1\times6}$,                        ▷ Input tensor
$G \in \{16, 32\}$,                                       ▷ Generated depth neuron size
$\#_G \in \{1, 2\}$,                                      ▷ Depth indicies
Output: $\mathbf{r} \in \Re^{1\times1\times2}$            ▷ Regression layer (Real, Imag)
Parameters: $N_i, N_o$                                    ▷ Input and output neuron size of layer

$\mathbf{x}_m = Conv{+}{+}(\mathbf{x}_i, 6, G\{1\})$
  for $k \in \#_G$ do
    $\mathbf{x}_m \leftarrow Conv{+}{+}(\mathbf{x}_m, G\{k\}, G\{k\})$
    if $k \neq \#_G(end)$ then
      $\mathbf{x}_m \leftarrow Conv{+}{+}(\mathbf{x}_m, G\{k\}, G\{k+1\})$
    end if
  end for

  $\mathbf{x}_m = Conv{+}{+}(\mathbf{x}_m, G\{end\}, 2 \times G\{end\})$
  $\mathbf{r} = conv(\mathbf{x}_i, 2 \times G\{end\}, 2)$
function: Conv++($\mathbf{x}$ , $N_i$ , $N_o$)
$\mathbf{x} \leftarrow conv(\mathbf{x}, N_i, N_o)$;
$\mathbf{x} \leftarrow BN(\mathbf{x})$;
$\mathbf{y} \leftarrow ReLU(\mathbf{x},' leak', 0.01)$;
end function
```

Note. These are the pseudo-code of the network shown in Figure 6B. BD and FNS 2 and 16 are selected, respectively. In both cases, pseudo-codes generate the same architecture. The difference is between *Conv* function in MLP, and *Conv++* function in M2LP. Batch normalization and leak ReLU activation are the powerful sides of M2LP Abbreviations: BD, block depth; FNS, first neuron size; MLP, multilayer perceptron; M2LP, modified multi-layer perceptron; ReLU, rectified linear unit.

4 | STUDY CASE

As is mentioned before, in addition to the groundbreaking performance of DL in areas such as classification, detection, and segmentation, the regression area remains a fairly new and worthwhile field. In particular, regression of antenna parameters using CNN is a relatively new application. The use of CNN-based models instead of MLP networks currently used as state-of-the-art enables the use of innovations developed in the CNN field in regression. Herein, a novel CNN-based MLP model had been proposed for fast, accurate, and high-performance modelling of a CFA. In this section, first, design parameters of the proposed model had been presented. After that, the performance comparison of the proposed M2LP model with its traditionally counterpart algorithm has been given. Finally, the obtained M2LP model had been used with a meta-heuristic optimization algorithm to obtain an antenna design for industrial, scientific, and medical (ISM) band application. The obtained optimal antenna design than had been prototyped by using 3D printer technology for justification of the proposed modelling method with experimental results.

4.1 | Test workflow

For MLP and M2LP, network generation according to Algorithms 2 and 3 is performed for $BD \in \{1, 2, 3\}$ and $FNS \in \{8, 16, 32, 64\}$ parameters. The networks produced for each case is run 10 times due to the stochastic results caused by the random weight assignment. The mean absolute error (MAE) results of *real*, *imaginary*(*imag*) and $0.5 \times (real + imag)$ are recorded as the output of an experiment. These three cases are recorded separately for 10 trials. The results of the experiments are given in Table 5 as the average, minimum, and maximum values of the 10 trials for *real*, *imaginary*(*imag*), and $0.5 \times (real + imag)$ outputs.

The hyperparameter list used in the training of networks is given in Table 4. For the learning rate, an exponentially decreasing method is used between [10^{-2}, 10^{-5}]. As the batch size, 200 is selected since the data set is large scale. In weight initialization, Xiavier[31] is used and Adam[38] is chosen as the optimizer. MAE is used as a loss function for the training of networks instead of root-mean-square error (RMSE). Nets are trained along 200 epoch. The training of the networks is based on the 8-core 3 Ghz processor with 16 GB ram and 1080TI graphics card. MatConvNet[43] is preferred for the training platform.

4.2 | Performance comparison of the proposed M2LP

In Table 5 and Figure 7, the performance comparison of the traditional MLP model with the proposed M2LP model for regression of real and imaginary parts of the antennas S_{11} has been presented for 10 different runs using the data given in Table 1. For each of the models, a different combination of network design parameters had been used to find the optimal black box model. As can be seen from the table, both of the models have similar performance results in the models with 1-layer depth. However, when the depth and number of neurons increase, the performance of the M2LP model starts to outperform its traditionally counterpart algorithm MLP. However, it should be noted that ever increasing the design parameters would not improve the performance such as for MLP with depth 3 and neuron 64, where the model starts to lose performance compared with simpler models. Herein, the M2LP model with depth 2 and 64 neurons found to be the best model for our regression problem, where not only the model has a better performance than its counterpart model, but also has a narrower standard variation for 10 runs. Furthermore, in Figure 8, the learning graph of the networks for 300 max epoch had been presented. This process is also repeated 10 times due to the random assignment of the network weights. Red and blue shaded areas are confidence intervals of training and test random processes, respectively. As seen

TABLE 4 Parameters used in training phase of MLP and M2LP

Hyperparameter List	
Learning Rate	[10^{-2}, 10^{-5}] exp decay
Batch size	500
Weight init.	Xiavier improved
Optimizer	Adam
Weight decay	10^{-5}
Loss function	MAE
Epoch	200

Abbreviations: MAE, mean absolute error; MLP, multi-layer perceptron; M2LP, modified multi-layer perceptron.

TABLE 5 Real, Imaginary (Imag) and 0:5 × (Real+Imag) MAE (×10^{-2}) results of MLPand M2LP: Each *min max* and *avg*. value is derived from 10 trials

		MLP									M2LP								
Depth	**First Layer**	**Real**			**Imag**			**0:5 ×(Real + Imag)**			**Real**			**Imag**			**0:5 (Real + Imag)**		
Parameter	**Neuron Size**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**
1	8	4.68	7.20	5.20	4.61	8.34	5.30	4.69	7.77	5.25	4.87	6.54	5.50	4.66	6.57	5.37	4.77	6.56	5.44
	16	3.08	4.24	3.64	3.01	4.15	3.59	3.04	4.20	3.62	3.55	4.56	3.92	3.47	4.26	3.85	3.51	4.41	3.88
	32	2.37	4.47	2.79	2.33	4.31	2.79	2.37	4.39	2.79	2.61	2.87	2.74	2.48	2.81	2.66	2.60	2.84	2.70
	64	1.32	4.03	2.12	1.28	4.67	2.18	1.30	4.35	2.15	1.81	2.26	2.02	1.72	2.21	1.95	1.77	2.21	1.99
2	8	3.33	6.47	4.28	3.41	6.23	4.33	3.37	6.35	4.31	3.13	4.37	3.55	3.03	4.52	3.59	3.14	4.44	3.57
	16	1.85	2.62	2.28	1.88	2.68	2.31	1.87	2.65	2.29	2.00	2.59	2.27	2.08	2.56	2.28	2.04	2.57	2.27
	32	1.14	5.07	2.13	1.08	4.77	2.11	1.11	4.92	2.12	1.50	1.84	1.68	1.48	1.80	1.61	1.51	1.82	1.65
	64	1.00	3.88	1.89	0.92	3.82	1.81	0.96	3.85	1.85	1.18	1.43	1.27	1.16	1.32	1.22	**1.17**	**1.37**	**1.24**
3	8	2.74	4.37	3.36	2.73	4.32	3.36	2.77	4.34	3.36	2.53	3.08	2.71	2.49	3.13	2.71	2.51	3.10	2.71
	16	1.76	2.29	1.94	1.74	2.22	1.94	1.75	2.25	1.94	1.75	2.01	1.87	1.74	1.92	1.82	1.75	1.95	1.85
	32	1.39	2.50	1.79	1.46	2.67	1.88	1.44	2.59	1.91	1.32	1.85	1.47	1.32	1.70	1.43	1.32	1.77	1.45
	64	2.62	8.79	5.88	2.69	8.84	5.98	2.65	8.31	5.80	1.11	1.59	1.32	1.07	1.62	1.26	1.09	1.60	1.29

Abbreviations: MLP, multilayer perceptron; M2LP, modified multi-layer perceptron.

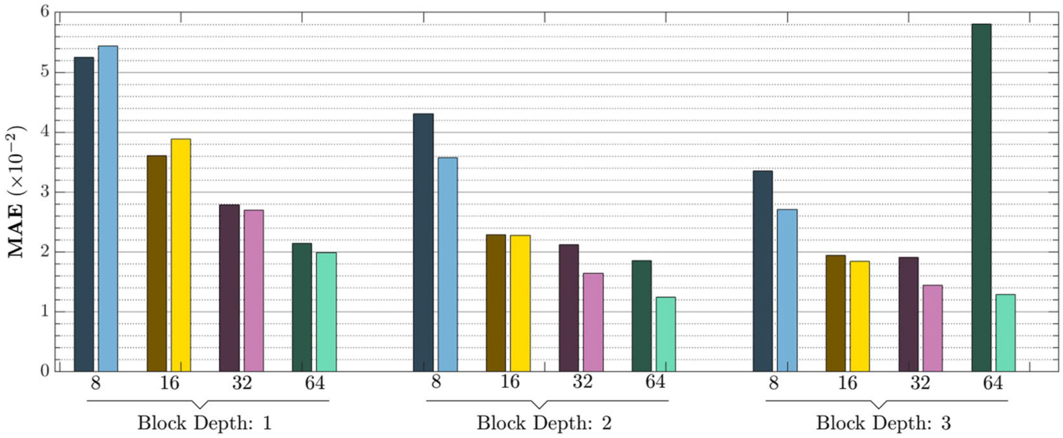


FIGURE 7 The average mean absolute error (MAE) values of $0.5 \times (Real + Imag)$ after 10 trials: each twin bars include multilayer perceptron (MLP) (dark tones) and modified multi-layer perceptron (M2LP) (light tones) results. first neuron size (FNS) $\in \{8, 16, 32, 64\}$ are related with first neuron size corresponding block depth

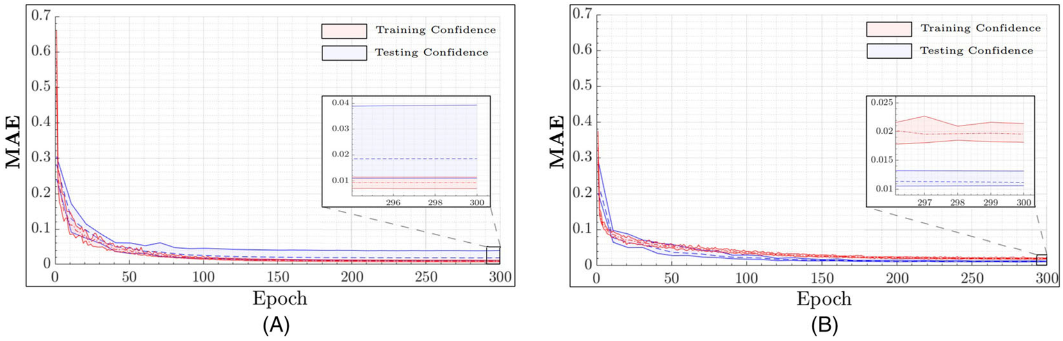


FIGURE 8 Training and test process of multilayer perceptron (MLP) and modified multi-layer perceptron (M2LP): confidence intervals of minimum, maximum, and average values are presented for the training processes obtained from 10 trials. To monitor the test performance of networks, they are tested with all test data in each 10 epoch. Although MLP appears to be better than M2LP in the training process, M2LP in the test phase produces narrower and lower mean absolute error (MAE) bu using batch normalization (BN). M2LP has a more reliable use

in Figure 8, the variance of M2LP is narrower than MLP by using batch normalization, which keeps the network in an efficient learning regime and achieves internal covariate shift problem.[32] Another important issue is the occurrence of vanishing gradient problem in some experiments of MLP network due to using the $\tanh(\cdot)$ activation. Hence, it is observed from the high upper boundary exit where the MLP can be overfitting. Thus it can be said that ReLU activation can be preferred in M2LP to prevent overfitting.

M2LP is also compared with support-vector machine (SVM) and ensemble-based regression methods which are widely used in literature. Parameter estimation of these methods has critical importance to minimize the regression cost function. The selected parameters play a key role in ensuring the compatibility between the regression model and the data. Because a comparison to be made using only a few values is not be fair, we used Bayesian optimization (BAO), which is frequently preferred in machine learning[44-46] for defining parameters of SVM and ensemble methods. The BAO handles the objective function as a Gaussian Process, and performs the selection stochastically.[47] *Box Value (C)*, *kernel scale* and ϵ for SVM type models, and *learning cycles*, *minimum leaf size*, and *maximum splits* parameters for ensemble-type models are given to BAO for optimization. While radial basis function (RBF) and polynomial kernels are chosen for SVM, bagged tree, and LSboost are preferred in Ensemble methods. The models generated by the parameters obtained after 30 iterations are used for the test. Since models such as ensemble and SVM can be trained on a single output, individual models are developed for real and imag. The parameter values obtained after the optimization are given in the *Model Parameters* column. Information on MLP and M2LP is taken from Table 5. MAE error metrics are averaged over training and test errors and given in the *Error* column.

As can be seen in Table 6, although SVM^{rbf} gives good result for training error , it remains at 0.22 MAE for test data. This shows that SVM^{rbf} overfits the training data. For SVM^{ploy}, after selection of degree parameter as 2, the optimization has

TABLE 6 Comparison of SVM vs ensemble-based regression methods results with neural networks: BAO is used for parameter estimation for SVM and ensemble methods

Methods	**Model Parameters**						**Error:** $0.5 \times (\textbf{\textit{Real}} + \textbf{\textit{Imag}})$	
	Real			**Imag**			**Training**	**Test**
SVM-Based	**Box Value**	**Kernel Scale**	ϵ	**Box Value**	**Kernel Scale**	ϵ	**SVM-Based**	
RBF	212.63	0.74028	3.4×10^{-4}	454.69	0.76717	6.8×10^{-4}	8.02×10^{-4}	0.2194
Poly. (Order = 2)	6.7287	34.24	8.9×10^{-4}	0.0010	0.15723	0.39761	0.2493	0.2556
Poly. (Order = 3)	988.8	9.9956	4.3×10^{-4}	0.0021	0.4776	0.1246	0.1979	0.2054
Ensemble-Based	**Learning Cycles**	**Min Leaf Size**	**Max Splits**	**Learning Cycles**	**Min Leaf Size**	**Max Splits**	**Ensemble-Based**	
Bag	46	1	8201	96	2	4221	0.0137	0.0507
LSBoots	496	2	180	459	110	156	0.0046	0.0542
Neural Networks	**Block Depth**		**First Neuron Size**		**Hyperparameters**		**Neural Networks**	
MLP	2		64		Table 5		0.0098	0.0185
M2LP	2		64		Table 5		0.0196	**0.0124**

Note. Model Parameters are given to BAO for optimization. Abbreviations: BAO, Bayesian optimization; SVM, support-vector machine.

been done. Therefore, it obtains 0.2493 and 0.2446 training and test MAE performance. Then, when the degree parameter is also given to the BAO method, it is determined as 3, and SVM^{ploy} achieves 0.1979 and 0.2054 training and test MAE results. Similar training and test errors show that SVM^{poly} tries to learn the training data instead of memorizing it. $Ensemble^{bag}$ method is much better than SVM in terms of modelling capability. The $Ensemble^{lsboost}$ version obtained a test error of 0.052, while it obtained 0.051 test error. MLP could not reach the success it achieved in training to the test samples. M2LP shows that it is the best network in terms of modelling capability and test error. 1.24×10^{-3} error has shown an efficient performance.

The M2LP is a capable method for modelling a CFA. Another proof of this is given in Figure 9. Herein, there are 3 simulation case results. The case parameters are presented in Table 7. Each row in Figure 9 are related to *dB*, *real*, and *imaginary* comparison, respectively. Despite difficult situations such as indented values in the 2nd case, the M2LP achieved values close to the target results. MAE performances are given in Table 7 for each case.

4.3 | Experimental results

Design optimization of the CFA has been achieved by using the proposed M2LP model and a meta-heuristic optimization algorithm differential evolutionary algorithm (DEA).[48] In this optimization problem, S_{11} (outputs of M2LP) of the design

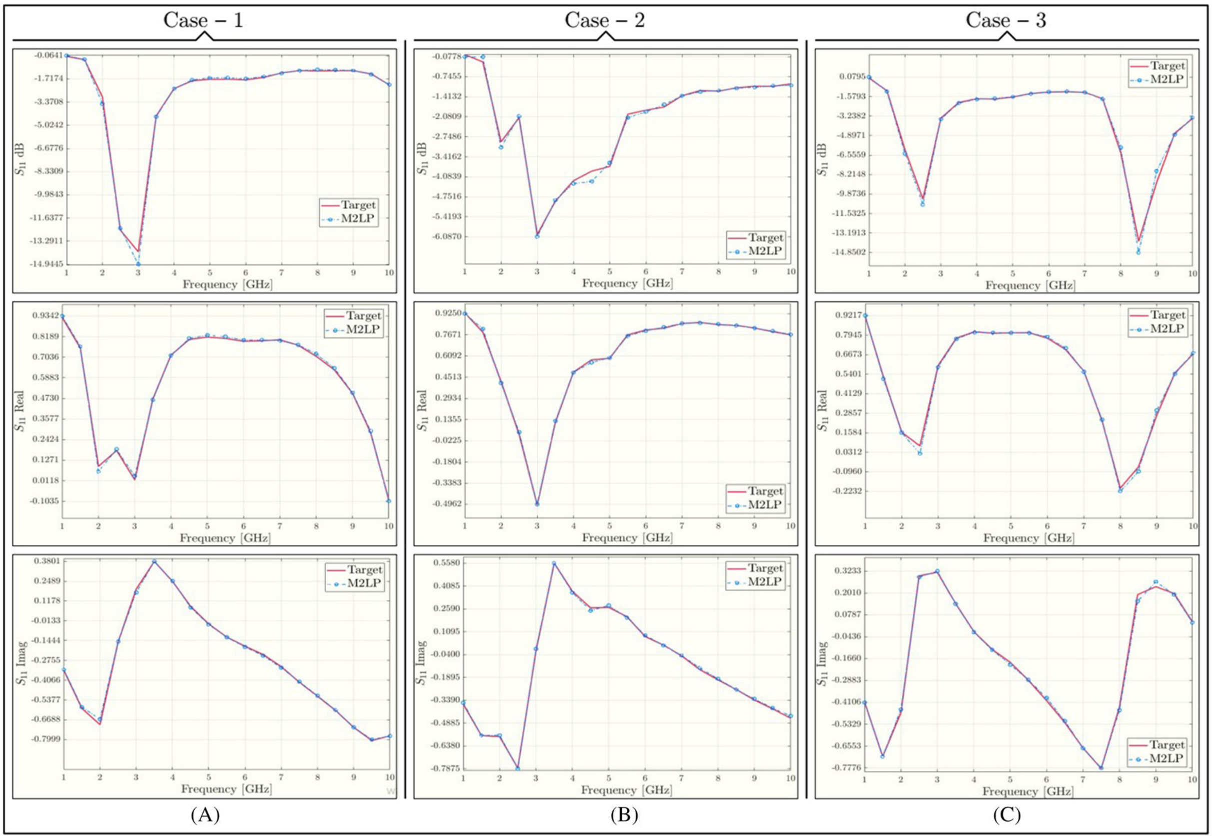


FIGURE 9 Simulation comparison of target and modified multi-layer perceptron (M2LP): column figures are related to the selected case, and rows 1,2, and 3 are *dB real* and *imag*. results, respectively. As seen from figures, M2LP predicts similar results even if indented hard situation like case 2. Hence, M2LP can be used for estimating of S_{11} parameters of new geometric cap feed designs

TABLE 7 The parameters of selected; Figure 9: First five parameters are constant, and $f \in \{0, 0.5, 1, \ldots, 10\}$

	Input Parameters						Average Error (MAE)	
Case Number	R	L4	L2	H1	H2	f	Real	Imag
1	5	1.5	40	15	0.25	[1,10]	0.0088	0.0067
2	7	2.5	55	10	0.25	[1,10]	0.0071	0.0073
3	5	2.5	55	20	0.25	[1,10]	0.0116	0.0095

Note. Average Error is calculated over different *f* for M2LP. Abbreviations: MAE, mean absolute error; M2LP, modified multi-layer perceptron.

TABLE 8 Optimally selected antenna parameters in (mm)

L1	112	W1	32
L2	40.3	W2	L2
L3	35.3	W3	51.3
L4	2.0	H1	0.7
R	6.35	H2	10.25

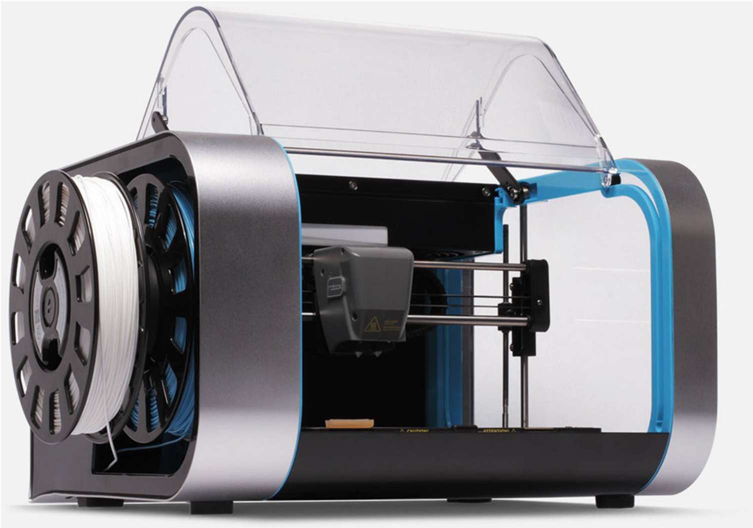
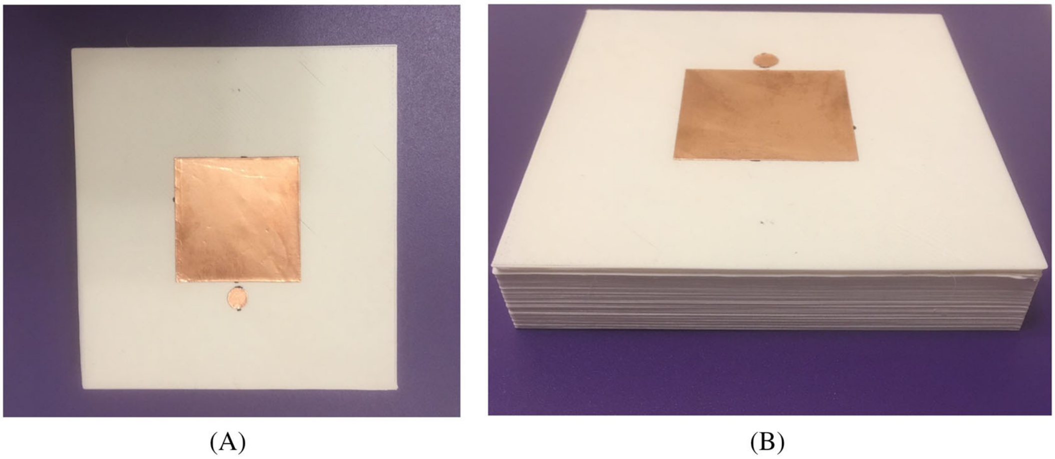


FIGURE 10 CEL Robox Micro Manufacturing Platform[49]



(A)

(B)

FIGURE 11 A, Top and B, perspective view of the 3D-printed antenna

would change concerning the geometrical design parameters (input variables of M2LP with respect to constraints in Table 1) using the cost function given in Equation (5) to obtain an antenna design operates at 2.4 GHz.

$$Cost_i(R, L_2, L_4, H_1, H_2, fr) = min[|(S_{11})(R, L_2, L_4, H_1, H_2, fr)|]. \tag{5}$$

Here, for normalization of optimization output value, S_{11} is not converted into decibels, so the absolute value of S_{11} is less than one. The user-defined parameters of the DEA had been taken as: crossover constant = 0.5, mutation scale factor F = 0.25, maximum iteration = 20, and population = 20.

The optimally selected design values from DEA had been given in Table 8. These values are then had been used for prototyping of the antenna by using 3D printing technology (Figure 10). The 3D printable material "PLA" had been used as a dielectric substrate. In Figures 11 and 12 and Table 9, the 3D-printed antenna and its measured S_{11} and gain performance had been given. As it can be seen from the measurement and M2LP-based simulation results, the proposed M2LP is a fast, accurate, and reliable model which can be used in the design optimization process.

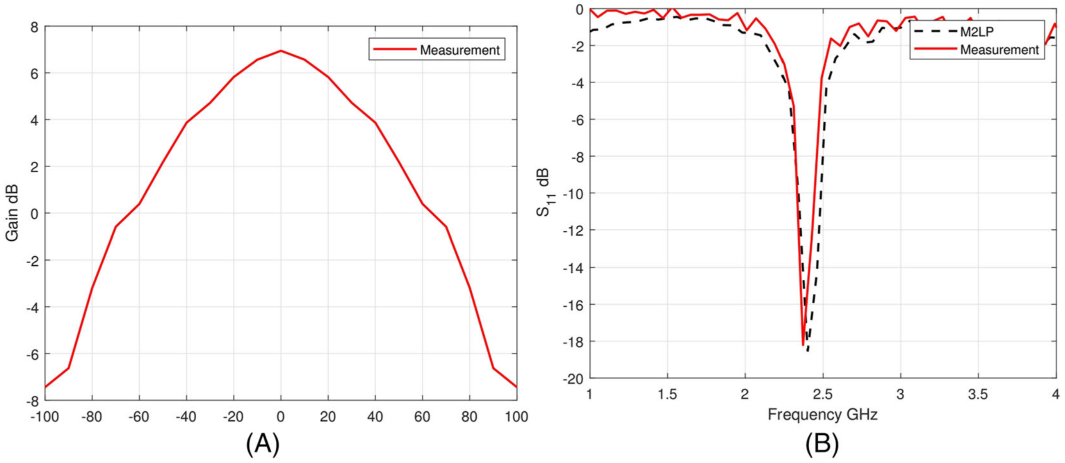


FIGURE 12 Simulated and measured A, gain and B, S_{11} of the the 3D-printed antenna

TABLE 9 Comparison of simulation and measurement results

	S_{11} dB		Gain dB
Frequency (GHz)	Simulated	Measured	Measured
2.4	-18.5	-15.5	6.9

5 | CONCLUSIONS

As mentioned before, the MSA design had been extensively being used in many wireless communication systems due to their unique features. (MSAs). However, with the increase of the requested antenna performance measures such as operation bandwidth, gain, and specified radiation pattern, the design procedure of the antenna had become a design optimization problem with complex inner relationships between the geometrical design parameters and the requested performance criteria. In this work for achieving this problem, first, a hybrid network, M2LP, based on MLP and CNN architecture had been proposed. Then the proposed M2LP model had been trained and tested with a 3D EM-based data belong to a CFA for creating a fast, accurate, and reliable regression model to be used in design optimization process. In this study, the performance of the proposed M2LP model had been compared with its traditional counterpart regression model MLP. Furthermore, the reliability of the proposed M2LP antenna model had been justified with experimental results obtained from a 3D-printed antenna. As a result from both simulation and measurements, it can be seen that the proposed M2LP algorithm is not only an effective, highly accurate, and stable algorithm compared with its counterpart method, but also has a potential to be used in modelling of other type of microwave circuits with higher complexity.

ACKNOWLEDGEMENTS

We would like to express our special thanks of gratitude to the Research Fund of the Yildiz Technical University for founding our research under project number of FAP-2018-3427, to Microwave and Antenna Laboratories of Yildiz Technical University for measurement setup, and to Aktif Neser Elektronik for providing researcher licenses.

CONFLICT OF INTERESTS

The authors declare no potential conflict of interests.

ORCID

Nurullah Calik https://orcid.org/0000-0002-7351-4980
Peyman Mahouti https://orcid.org/0000-0002-3351-4433

REFERENCES

1. Rop KV, Konditi DBO, Ouma HA, Musyoki SM. Parameter optimization in design of a rectangular microstrip patch antenna using adaptive neuro-fuzzy inference system technique. *IJTPE J ISSN*. 2012:2077-3528.

2. Kasabegoudar VG, Upadhyay DS, Vinoy KJ. Design studies of ultra-wideband microstrip antennas with a small capacitive feed. *Int J Antennas Propag*. 2007.
3. Singh I, Tripathi VS. Micro strip patch antenna and its applications: a survey. *Int J Comp Tech Appl*. 2011;2(5):1595-1599.
4. Mayhew-Ridgers G, Odendaal JW, Joubert J. Efficient full-wave modeling of patch antenna arrays with new single-layer capacitive feed probes. *IEEE Trans Antennas Propag*. 2005;53(10):3219-3228.
5. Watson PM, Gupta KC, Mahajan RL. Development of knowledge based artificial neural network models for microwave components. In: 1998 IEEE MTT-S International Microwave Symposium Digest, Baltimore, MD, USA (Cat. No. 98CH36192), Vol. 1 IEEE; 1998:9-12.
6. Suntives A, Hossain MS, Ma J, Mittra R, Veremey V. Application of artificial neural network models to linear and nonlinear RF circuit modeling. *Int J RF and Microwave Comput Aided Eng Co sponsored Cent Adv Manuf Packag Microw Opt Digit Electron (CAMPmode) Univ Colorado Boulder*. 2001;11(4):231-247.
7. Zhang QJ, Gupta KC. Neural networks for rf and microwave design (Book+ Neuromodeler Disk). *Artech House, Inc.*, 2000.
8. Simsek M, Aoad A. Efficient reconfigurable microstrip patch antenna modeling exploiting knowledge based artificial neural networks. *Simulation-driven modeling and optimization*; 2016:185-206.
9. Li X, Gao J, Boeck G. Printed dipole antenna design using artificial neural network modeling for RFID application. *Int J RF and Microwave Comput Aided Eng Co sponsored Cent Adv Manuf Packag Microw Opt Digit Electron (CAMPmode) Univ Colorado Boulder International Journal of RF and Microwave Computer-Aided*. 2006;16(6):607-611.
10. Mahouti P. Design optimization of a pattern reconfigurable microstrip antenna using differential evolution and 3d EM simulation-based neural network model. *Int J RF Microwave Comput Aided Eng*; 0(0):e21796.
11. Zeng M, Nguyen LT, Yu B, Mengshoel OJ, Zhu J, Wu P, Zhang J. Convolutional neural networks for human activity recognition using mobile sensorsIEEE; 2014:197-205.
12. Babu GS, Zhao P, Li XL. Deep convolutional neural network based regression approach for estimation of remaining useful life. In: International conference on database systems for advanced applications Springer; 2016:214-228.
13. Bao W, Yue J, Rao Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS One*. 2017;12(7):e0180944.
14. Qureshi AS, Khan A, Zameer A, Usman A. Wind power prediction using deep neural network based meta regression and transfer learning. *Appl Soft Comput*. 2017;58:742-755.
15. Li Y, Shi H, Han F, Duan Z, Liu H. Smart wind speed forecasting approach using various boosting algorithms, big multi-step forecasting strategy. *Renew Energy*. 2019;135:540-553.
16. Wei Z, Liu D, Chen X. Dominant-current deep learning scheme for electrical impedance tomography. *IEEE Trans Bio Eng*. 2019;66(9):2546-2555.
17. Wei Z, Chen X. Deep-learning schemes for full-wave nonlinear inverse scattering problems. *IEEE Trans Geosci Remote Sens*. 2019;57(4):1849-1860.
18. Bramerdorfer G, Amrhein W, Winkler SM, Affenzeller M. Identification of a nonlinear PMSM model using symbolic regression and its application to current optimization scenarios. In: IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society IEEE; 2014:628-633.
19. Hubel DH, Wiesel TN. Receptive fields of single neurones in the Cat's striate cortex. *J Physiol*. 1959;148(3):574-591.
20. Hubel DH, Wiesel TN. Receptive fields and functional architecture of monkey striate cortex. *J Physiol*. 1968;195(1):215-243.
21. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern*. 1980;36(4):193-202.
22. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the ieee international conference on computer vision; 2015:1026-1034.
23. Geirhos R, Janssen DHJ, Schütt HH, Rauber J, Bethge M, Wichmann FA. Comparing deep neural networks against humans: object recognition when the signal gets weaker. arXiv preprint arXiv:1706.06969; 2017.
24. Zhu W, Huang Y, Zeng L, et al. Anatomynet: deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy. *Medical Phys*. 2019;46(2):576-589.
25. Liu F, Zhou Z, Jang H, Samsonov A, Zhao G, Kijowski R. Deep convolutional neural network and 3d deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging. *Magn Reson Med*. 2018;79(4):2379-2391.
26. Zhang B, Li S, Huang Z, Rahi BH, Wang Q, Li M. Transfer learning-based online multiperson tracking with Gaussian process regression. *Concurrency Comput Pract Experience*. 2018;30(23):e4917.
27. Son J, Baek M, Cho M, Han B. Multi-object tracking with quadruplet convolutional neural networks. In: Proceedings of the ieee conference on computer vision and pattern recognition; 2017:5620-5629.
28. Moazzen Y, Çapar A, Albayrak A, Çalık N, Töreyin BU. Metaphase finding with deep convolutional neural networks. *Biomed Signal Process Control*. 2019;52:353-361.
29. Albayrak A, Unlu A, Calik N, et al. A whole slide image grading benchmark and tissue classification for cervical cancer precursor lesions with inter-observer variability. arXiv preprint arXiv:1812.10256; 2018.
30. Chung JS, Zisserman A. Lip reading in the wild. In: Asian conference on computer vision Springer; 2016:87-103.
31. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics; 2010:249-256.
32. Ioffe Sy, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167; 2015.

33. Ioffe S. Batch renormalization: towards reducing minibatch dependence in batch-normalized models. In: Advances in neural information processing systems; 2017:1945-1953.
34. Wu Y, He K. Group normalization. In: Proceedings of the european conference on computer vision (eccv); 2018:3-19.
35. Wang T, Qin Z, Zhu M. An ELU network with total variation for image denoising. In: International conference on neural information processing Springer; 2017:227-237.
36. Mukkamala MC, Hein M. Variants of rmsprop and adagrad with logarithmic regret bounds. In: Proceedings of the 34th international conference on machine learning-volume 70 JMLR org; 2017:2545-2553.
37. Zeiler MD. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701; 2012.
38. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980; 2014.
39. Babu GS, Zhao P, Li XL. Deep convolutional neural network based regression approach for estimation of remaining useful life. In: International conference on database systems for advanced applications Springer; 2016:214-228.
40. Held D, Thrun S, Savarese S. Learning to track at 100 fps with deep regression networks. In: European conference on computer vision Springer; 2016:749-765.
41. Ross ZE, Yue Y, Meier MA, Hauksson E, Heaton TH. Phaselink: A deep learning approach to seismic phase association. *Journal of Geophysical Research: Solid Earth*. 2018;144(1):856-869.
42. Xu B, Wang N, Chen T, Li M. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853; 2015.
43. Vedaldi A, Lenc K. Matconvnet: convolutional neural networks for matlab. In: Proceedings of the 23rd acm international conference on multimedia ACM; 2015:689-692.
44. Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems; 2012:2951-2959.
45. Pelikan M, Goldberg DE, Cantú-Paz E. Boa: The Bayesian optimization algorithm. In: Proceedings of the 1st annual conference on genetic and evolutionary computation-volume 1 Morgan Kaufmann Publishers Inc.; 1999:525-532.
46. Frazier PI. A tutorial on Bayesian optimization; 2018.
47. Gelbart MA, Snoek J, Adams RP. Bayesian optimization with unknown constraints. arXiv preprint arXiv:1403.5607; 2014.
48. Gunes F, Belen MA, Mahouti P. Competitive evolutionary algorithms for building performance database of a microwave transistor. *Int J Circ Theor Appl*; 46(2):244-258. https://doi.org/10.1002/cta.2386
49. Robox C. Cel robox micro manufacturing platform; 2012.

How to cite this article: Calik N, Belen MA, Mahouti P. Deep learning base modified MLP model for precise scattering parameter prediction of capacitive feed antenna. *Int J Numer Model*. 2020;33:e2682. https://doi.org/10.1002/jnm.2682