Machine Learning Methods-based Modeling and Optimization of 3-D-Printed Dielectrics around Monopole Antenna

Yashika Sharma, Xi Chen, Junqiang Wu, Qiang Zhou, Hao Helen Zhang, Hao Xin, *Fellow, IEEE*

***Abstract*— In this paper, we propose using new Machine Learning (ML)-based optimization methods, as an alternative to traditional optimization methods, for complex antenna designs. This is an efficient methodology to tackle computational challenges, as it is capable of handling a large number of design parameters and is more efficient as well as informative. The proposed technique is applied for modeling gain performance in the principal plane of a monopole antenna when its radiation properties are modified by placing spatially dependent dielectric material around it. Using the proposed methodology, the dielectric constant values are mapped to the gain pattern of the design. We use two ML techniques for this purpose, namely, Gaussian Process (GP) regression and Artificial Neural Network (ANN). Once each of these models is obtained, they are further used for estimating the dielectric constant values that can suggest optimal directions to modify gain patterns for single-beam and multiple-beam patterns rather than the conventional omnidirectional pattern of a monopole antenna. The performance of this technique is compared with heuristic optimization techniques such as genetic algorithms. The proposed method proves to be quite accurate in spite of being a high-dimensional non-linear problem. A prototype of a monopole design with three-beam gain pattern is fabricated and tested. The measurement results agree well with the simulation results. The proposed methodology can provide useful and scalable optimization tools for computationally intensive antenna design problems.**

***Index Terms*— Antenna radiation patterns, Gaussian Process, Optimization, 3D printing.**

I. Introduction

Wireless communication and sensing have become an integral part of modern infrastructure and will be more ubiquitous in the near future, especially in the emerging era of internet of things (IoT). Antenna, which is an important interface between propagating electromagnetic (EM) waves and electrical signals processed or generated by integrated circuits, is one of the most challenging components for future wireless communication and sensing. Antenna designing and optimization has become even more challenging due to the requirements of IoT, like ever-shrinking physical size and weight for practical IoT applications, adequate efficiency for power and thermal considerations, sufficient bandwidth for appropriate data rate and sensing resolution, etc. Generally, the antenna design mainly relies on the designer's empirical experiences and EM simulation tools, but these methods can be time-consuming, computation-intensive and often sub-optimal. Hence, techniques to design and optimize antennas adaptively for each and every application in a smarter and more efficient way are required in the current scenario. Researchers have explored optimization of antenna structures by applying population-based meta-heuristic optimization techniques [1] such as, genetic algorithms [2] [3] and particle swarm optimization [4, 5, 6]. These algorithms search for optimal solutions by analyzing the output on individual data points and generating new and possibly better search directions until a global maxima or minima is identified. Also, these algorithms require large amount of computational time and resources. This challenge of high computation-cost of EM-driven metaheuristic design optimization can be alleviated with surrogate-based optimization (SBO) techniques [7], which replaces direct optimization by iterative correction and re-optimization with less accurate but computationally cheap surrogate model. These methods provide satisfactory results by utilizing techniques like space mapping space mapping [8], response correction [9], feature-based optimization [10], etc. In recent years, machine-learning (ML) [11] [12] assisted optimization has gained attention of researchers because of their ability to learn from measured or simulated training data and accelerate the entire antenna design optimization. Few popular ML techniques like Artificial Neural Networks (ANNs) [13], k-Nearest Neighbors (kNNs) [14] [15], Support Vector Machines (SVMs) [16], etc. have been applied for antenna optimization. In most of these research works, optimization has been done for

Manuscript was first submitted on August 31, 2021 and revised on Nov 21, 2021; accepted January 6, 2022; date of current version February 14, 2022. (Corresponding author: Hao Xin)

Y. Sharma was a doctoral student with the Department of Electrical and Computer Engineering, University of Arizona and is currently working as RF Hardware Engineer at Apple Inc, Cupertino, California, USA. (e-mail: yashikasharma@email.arizona.edu)

X. Chen is Ph.D. student at the Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, USA. (e-mail: xic@email.arizona.edu)

J. Wu received his Ph.D. degree electrical and computer engineering at the University of Arizona, Tucson, AZ, USA, in 2017. He is currently working with Skyworks Inc., USA. (email: jqwu@email.arizona.edu)

Qiang Zhou is an Associate Professor at the Department of Systems and Industrial Engineering, and a faculty member of the Statistics Graduate Interdisciplinary Programs at University of Arizona, and he is a member of INFORMS and IISE. (e-mail: zhouq@arizona.edu)

H. H. Zhang is with the Department of Mathematics, University of Arizona, Tucson, Arizona 85719 USA (e-mail: hzhang@math.arizona.edu).

H. Xin is IEEE fellow and is with the Department of Electrical and Computer Engineering and Department of Physics, University of Arizona, Tucson, AZ 85721 USA. (e-mail: hxin@ece.arizona.edu)

some pre-defined goal. However, for applications where we want to modify our design goal dynamically according to the requirement, we need to have an efficient general-purpose modeling tool that can directly relate the performance parameters to the input design parameters. Towards this goal, using ML techniques first for training a model, and then further use this model for optimization, is a promising choice. The main advantage of using ML techniques is once the relational models have been "learned", we can use it to predict the output for any input, rather than just aiming for global optimal and minima points only; this property comes handy when we want to use the same data set for multiple different goals.

To demonstrate these advantages in this work, we fit two ML-based models for a quarter-wavelength monopole antenna surrounded by an optimized dielectric constant distribution on a finite ground plane. The antenna operates at 15 GHz. The dielectric profile is implemented as a 6 × 6 grid of 3-D printed dielectric blocks. The ANSYS High Frequency Structure Simulator (HFSS) [17] model for this design is as shown in Fig. 1. By changing the filling ratio of each of these dielectric grids, effective dielectric distribution is controlled, as a result, the radiation properties of the monopole antenna can be changed. ML models are used in this work to directly relate the dielectric distribution with the radiation pattern in the $\theta = 60^{o}$ plane of the monopole antenna. From the relational model perspective, the problem referred here is highly nonlinear and involves a large-dimensional design space as it relates 18 input parameters (each representing the dielectric constant of blocks surrounding the monopole antenna) and 181 output parameters (each representing gain value on $\theta = 60^{o}$-plane for φ varying from $0^{o}$to 180^{o} with 1^{o}step), and these output parameters are dependent non-linearly to the input parameters. To address this issue, we apply two ML techniques, namely Gaussian Process (GP) regression and ANNs. These two ML models are used to optimize the dielectric distribution profile to achieve various desired antenna radiation patterns including single-beam and multiple-beam cases. A genetic algorithm-based optimization has been studied previously [3] for the same purpose but it is a goal-oriented optimization and becomes computationally intensive when multiple goals are to be addressed. In this paper, we evaluate the performance of the proposed ML-based optimization and compare its efficacy with that of the previously studied genetic algorithm under a variety of design specifications.

The layout of this paper is as follows. In Section II, a theoretical background to the two ML techniques used in this work, the GP method and the ANN method are provided. Next, the approach and optimization methodology is explained. Section III describes the details of the design procedure including the simulation setup used for collecting the training data and creation of the ML models. Section IV provides the comparative analysis between the two ML techniques and the genetic algorithm-based optimization. Section V describes in detail the fabrication and measurement of a prototype designed by the proposed method. Finally, conclusions and potential future work are discussed in Section VI.

II. Theoretical Background

The general background of the two ML techniques and the box-constrained optimization method used for this work are provided in the following subsections.

A. Gaussian Process (GP) Modeling Method

The problem referred in this work is a highly nonlinear problem and involves a large design space. To address this issue, GP model is adopted as a surrogate of the simulation model. Among other modeling methods, the GP model is one of the most widely used due to its unique advantages. First, it can be built with a relatively small number of sample data points, which is critical in scenarios where large data collection is time consuming. Second, GP offers not only a single predicted value, but also quantification of the prediction uncertainty, which is extremely useful for post-modeling analysis [18]. To describe GP mathematically, consider a deterministic computer simulator with d- dimensional input vector $\boldsymbol{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$ and a scalar output $y(\boldsymbol{x})$. For m training pairs of $(\boldsymbol{x}_i, y(\boldsymbol{x}_i))$, the standard GP model is defined as:

$$\mathrm{y}(\mathbf{x}) = \sum_{\mathrm{j}=1}^{\mathrm{p}} \mathrm{f_j}(\mathbf{x})\beta_{\mathrm{j}} + \mathrm{z}(\mathbf{x}) = \mathbf{f}^{\mathrm{t}}(\mathbf{x})\boldsymbol{\beta} + \mathrm{z}(\mathbf{x}), \quad (1)$$

where $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \dots, f_p(\boldsymbol{x})]^t$ is a set of user-specified regression functions, $\boldsymbol{\beta} = \left[\beta_1, \beta_2, \dots, \beta_p\right]^t$ is a vector of unknown regression coefficients, and $\boldsymbol{f}^t(\boldsymbol{x})\boldsymbol{\beta}$ describes the global trend of $y(\boldsymbol{x})$. It is widely practiced to adopt a constant regression term, i.e., $\boldsymbol{f}^t(\boldsymbol{x})\boldsymbol{\beta} = \beta$ [19]. The error term $z(\boldsymbol{x})$ is a stationary zero-mean GP characterized by its variance σ^2, and some correlation function $K(\boldsymbol{x}_1, \boldsymbol{x}_2)$. One commonly used correlation function is the squared exponential correlation function:

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\left(-\sum_{i=1}^{d} \theta_i (x_{1i} - x_{2i})^2\right). \quad (2)$$

Parameters $\Theta = \{\beta, \sigma^2, \boldsymbol{\theta}\}$ are unknown and usually estimated by maximizing the log-likelihood function [19].

For any non-simulated input $\boldsymbol{x}_0$, model (1) implies that $y_0 = y(\boldsymbol{x}_0)$ and $\boldsymbol{y}^m$ should follow a multivariate normal distribution and $y(\boldsymbol{x}_0)$ has the predictive distribution with mean and variance given by:

$$\mu_{x_0} = \hat{\beta} + \boldsymbol{k}_0^{\mathrm{t}} \boldsymbol{K}^{-1}\left(\boldsymbol{y}^m - \mathbf{1}\hat{\beta}\right), \quad (3)$$

$$\sigma_{x_0}^2 = \hat{\sigma}^2\{1 - \boldsymbol{k}_0^{\mathrm{t}} \boldsymbol{K}^{-1}\boldsymbol{k}_0 + \boldsymbol{h}^t(\mathbf{1}^t \boldsymbol{K}^{-1}\mathbf{1})^{-1}\boldsymbol{h}\}, \quad (4)$$

where $\boldsymbol{k}_0 = (K(\boldsymbol{x}_0 - \boldsymbol{x}_1), \dots, K(\boldsymbol{x}_0 - \boldsymbol{x}_m))^t$ is the $m \times 1$ vector of correlations between $\boldsymbol{y}^m$ and $y(\boldsymbol{x}_0)$ *and* $\boldsymbol{h} = 1 - \mathbf{1}^t \boldsymbol{K}^{-1}\boldsymbol{k}_0$.

In this work, the simulation output from HFSS simulation is a curve (the gain pattern), instead of a scaler output. Assume the curve is sampled at *l* discrete locations, then it can be viewed as a vector of length *l*, denoted as $\boldsymbol{y}^l(\boldsymbol{x})$.

To apply the GP method, we reduce the dimension *l* using flexible B-spline [20]

$$\boldsymbol{y}^l(\boldsymbol{x}) = \sum_{j=1}^{l'} B_{jk} a_j, \quad (5)$$

where B_{jk} is the j^{th} B-spline of order *k*, and given the order and knot sequence, B_{jk} is completely determined, l' is significantly smaller than *l* and is the number of coefficients. We then separately fit l' GP models and predict each coefficient. The predicted $\boldsymbol{\mu}_{\boldsymbol{x}_0} = [\mu_{1x_0}, \mu_{2x_0}, \dots, \mu_{l'x_0}]^t$ can be easily converted back to the original l-dimensional $\boldsymbol{y}^l(\boldsymbol{x})$ by (5). For the prediction uncertainty $\boldsymbol{\sigma}_{\boldsymbol{x}_0}^2 \in \mathbb{R}^{l'}$, by an independent

assumption of the coefficients, it can be obtained by $\sum_{j=1}^{l'} B_{jk}^2 \, \sigma_{jx_0}^2$.

B. Artificial Neural Network Modeling Method

The other ML technique used in this work is ANN, which is a computational model motivated by the function of biological neural networks. It consists of a group of artificial neurons that process and distribute information over interconnection. The Multilayer Perceptrons (MLPs) [21] structure of ANNs is used in this work because of their ability to learn and model complex relationship. This MLP structure generally comprises of three layers, namely input, output, and hidden layer. Each parameter inside each of these layers is referred as "neuron". The neurons in the input layer distributes the input signal $u_i$to the neurons in the hidden layer. Each neuron j in the hidden layer adds up its input signals u_i after multiplying weights to each term depending on the respective connections w_{ij} from the input layer and computes its output v_j as a function g of the sum and a bias value, b_j added to it, i.e.,

$$v_j = g\left(\sum_i w_{ij} u_i + b_j\right), \tag{6}$$

where, $g(.)$ is a simple threshold function, which can be a sigmoid, or a radial basis function, etc. The output of neurons in the output layer is computed similarly. There are several algorithms to train the MLP network such as Levenberg-Marquardt (LM) [22], back-propagation, delta bar-delta, and so on. In this work, we train the MLPs using the LM algorithm because of its ability of fast learning and good convergence.

In this work, we use the neural network toolbox of Matlab [23]. In our analysis, the training data is extracted from the realized gain plots for different dielectric loading profiles around a monopole antenna. This data is divided into three parts: 70% of data is used for training and 15% each used for testing and validation respectively. The input layer consists of 18 design parameters (each corresponding input design parameter for the referred antenna design), the hidden layer of 10 hidden nodes, and the output layer of 181 nodes (each representing the output performance parameter for the referred antenna design). The details regarding the input and output nodes are explained in the design setup described in Section III.

C. Box-Constrained Optimization

The optimization method used after obtaining the behavioral model from the two ML techniques is the box-constrained optimization method. A box-constrained minimization finds a vector *x*, that is a minima point to a scalar function $f(\boldsymbol{x})$ subject to constraints on the allowable $\boldsymbol{x}$, i.e.

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \tag{7}$$

subject to the following conditions: $\boldsymbol{l} \leqslant \boldsymbol{x} \leqslant \boldsymbol{u}$, where $\boldsymbol{l}$ and $\boldsymbol{u}$ are lower and upper limits of the vector $\boldsymbol{x}$. To implement this, the *fmincon* function [24] of Matlab is used.

III. Data Collection And Modeling Setup Details

A. Antenna Design Simulation Setup for Collecting Training Data

A monopole antenna (quarter-wavelength at the design frequency of 15 GHz) is placed on a finite ground plane with a size of 40 × 40 mm^2, as shown in Fig. 1. The monopole has a 0.5 mm diameter and a 4.8 mm height. It is surrounded by 36 (6 × 6) dielectric unit cells, each colored separately around the monopole and placed symmetrically across left and right half of monopole as shown in Fig. 1. Here, 36-unit cells are chosen corresponding to the design space and frequency of operation, so that enough training data can be collected comfortably that can adequately capture the output performance variation and, the dielectric profile can significantly variate gain pattern of the antenna. Each unit cell has a size of 6.67 mm × 6.67 mm × 6.67 mm, corresponding to 0.33 λ_0 at 15 GHz. The dielectric constant ε_r of each dielectric block varies from 1.1 to 2.3 with a discretization step of 0.4. The entire antenna configuration has a symmetry plane splitting the monopole as shown in Fig. 1. The unit cell analysis of the dielectric blocks is conducted to realize the desired effective dielectric constant, which is explained further in Section III (C).

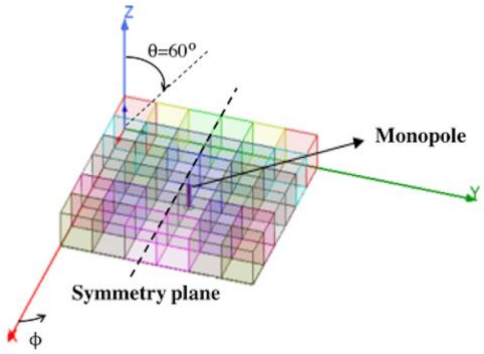


Fig. 1. HFSS Model of a monopole antenna surrounded by 6 x 6 grid of dielectric blocks with symmetry across left and right-half around the monopole.

B. ML Model Setup

The dielectric distribution can modify the radiation pattern of the monopole antenna shown in Fig. 1. Fig. 2 illustrates the top view of the HFSS model with the dielectric distribution labelled. The 18 parameters (ϵ_1 to ϵ_{18}) are considered as the input parameters while the realized gain values (in linear scale) at each of the φ angle ranging from 0° to 180° correspond to the output function. Since the structure has symmetry in the XY-plane, the gain values only in the range 0° - 180° are considered. Hence the mathematical representation of any ML model in this work can be represented as follows:

$$Gain\left(\varphi_j\right) = f(\epsilon_1, \epsilon_2 \ldots, \epsilon_{18}), \tag{8}$$

where, j = 0°, 1°, 2°…, 180°. For training and testing the proposed model, 1050 and 110 sample points are collected respectively, by running the HFSS simulations using the model explained in the previous sub-section.

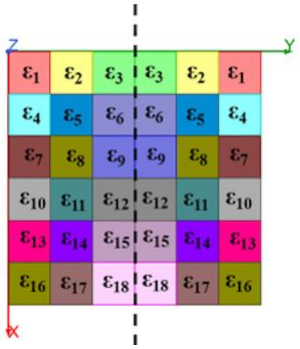


Fig. 2. Top view of the layout with dielectric distribution parameter description.

The representation of the proposed optimization methodology is shown in the form of flowchart in Fig. 3. We demonstrate the effectiveness of this methodology by achieving three different goals that are single-beam, two-beam and three-beam gain patterns in the principal radiation plane of the antenna. The fitness function for each goal is defined as follows:

1. For goal 1: Single-beam pattern:

fitness1= G(θ, φ_1) − max G(θ, φ_2), where θ = 60°, φ_1= 0°, $\varphi_2 \in$ {60°–180°}, (9)

2. For goal 2: Two-beam pattern:

*fitness*2= G(θ, φ_1) − max G(θ, φ_2), where θ = 60°, φ_1= 60°, $\varphi_2 \in$ {0, 90°–180°}, (10)

3. For goal 3: Three-beam pattern:

*fitness*3= min G(θ, φ_1) − max G(θ, φ_2), where θ = 60°, φ_1= {30°, 180°}, $\varphi_2 \in$ {0°, 90°–140°}, (11)

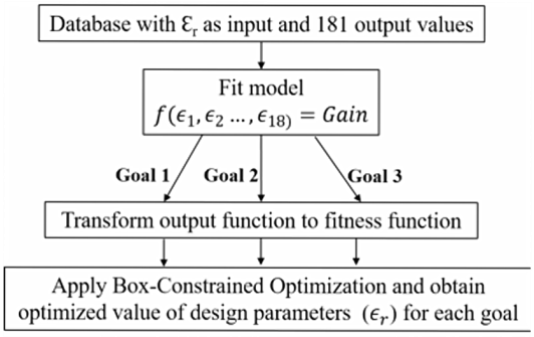


Fig. 3. Flowchart representing the optimization methodology used in this work.

Using (9), we aim to obtain the single-beam pattern that has maximum gain in φ_1 = 0° direction and the back-lobes in φ_2 = 60°–180° are suppressed. Similarly, by using (10), we aim for gain maxima directing towards φ_1 = 60° while suppressing gain in φ_2 = 0° and 90°–180° range. With (11), we aim to achieve three-beam pattern by maximizing gain in φ_1 = 30° and 180° direction and suppressing the gain in φ_2 = 0° and 90°-140°.

C. Unit cell setup to realize the estimated model

Each variation of dielectric constants can be physically realized with corresponding 3-D printed dielectric block by changing its size (filling ratio). To find the relation between dielectric constant and the corresponding size, a unit cell analysis is done as shown in Fig. 4. For this analysis, a polymer cube with its supporting rods is placed in a waveguide. PEC and PMC boundaries are used to define the periodic structure. The unit cell has a size of 6.67 mm × 6.67 mm × 6.67 mm. The gray part is the 3D-printed polymer material, which has ε_r = 2.7 and $\tan\delta = 0.02$. This polymer part consists of a dielectric cube with a variable dimension *b*. By changing *b* effectively, dielectric constants can be varied between 1 to 2.7. From the S-parameter simulation of the model and applying the standard-retrieval method [25], the effective permittivity ε_{eff} of the unit cell can be extracted for different polymer cube sizes. The size of each block is mapped to a dielectric constant value. The extracted results obtained are shown as the black marker-line curve in Fig. 5.

Alternatively, the effective permittivity can also be estimated by the polymer filling ratio f using the following equation [26]:

$$\varepsilon_r = \varepsilon_p \cdot f + 1 \cdot (1 - f), \quad (12)$$

where ε_r is the effective relative permittivity of the unit cell, and ε_p is the relative permittivity of the polymer material. The effective permittivity values obtained using this method is graphically represented as the blue dashed-line curve in Fig. 5.

It can be observed from Fig. 5 that the results approximated using filling-ratio method are quite similar to the HFSS extracted values, but this approximation becomes inaccurate when the cube size becomes larger.

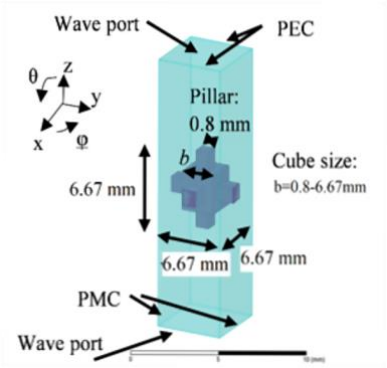


Fig. 4. Unit cell analysis setup to map dielectric block size to dielectric constant value.

Hence, to obtain an equation that can be directly used to accurately map polymer cube size with any desired permittivity value, HFSS-extracted results are used. The exponential curve-fitting tool of Matlab [27] is used to obtained the following equation (red solid line in Fig. 5):

$$b = 4.13e^{0.18\varepsilon_r} - 118.6e^{-3.57\varepsilon_r} 118.6e^{-3.57\varepsilon_r} \quad (13)$$

where b is the polymer cube size and ε_r is the desired permittivity.

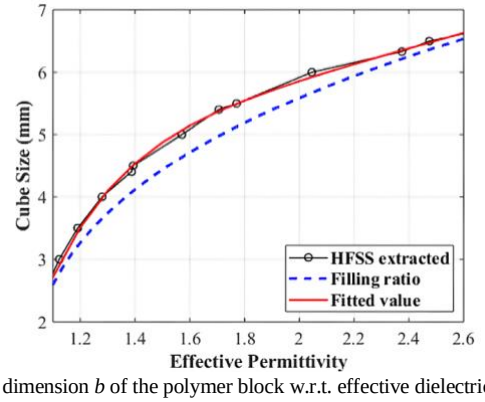


Fig. 5. Cube dimension *b* of the polymer block w.r.t. effective dielectric constant of unit cells. The black marked-line extracted from the S-parameters in the HFSS simulation; the blue dashed-line is computed using filling ratio approximation, and the red solid-line is the fitted curve corresponding to HFSS extracted values.

IV. Evaluation

A. GP and ANN-based ML Models' Performance Comparison

As mentioned earlier, two ML techniques are used in this work to model the gain performance in terms of dielectric distribution. In this section we evaluate and compare the performance of these two models. For comparison, the prediction accuracy of the two techniques is evaluated on the testing data that consists of 110 design samples. Both of the

machine learning models are trained and tested using the same testing and training datasets, which have 18 input parameters (each representing the dielectric constant values ϵ_1 to ϵ_{18}) and 181 output parameters (each representing gain value on $\theta = 60^o$-plane for φ varying from 0^o to 180^o with 1^o step). Initially the ANN model is trained using 10-hidden layer nodes but to improve the prediction accuracy, the hidden layer nodes are increased gradually in order to match its accuracy with GP. But this makes the ANN model more and more complex and increases its computation time quite significantly compared to GP. However, in GP, we reduce the output dimension by using B-spine expansion for gain patterns, where we use 25 uniformly located knots in the interval (0,180) to interpolate the 181 output values by splines of order 4. This helps in reducing the output dimension from 181 to 21 and improves the computation time for GP significantly. The results of the root-mean-square-error (RMSE) of the predicted gain by each of the ML technique are compared with the true value obtained from testing data in Table I. The total computation time for each of the ML technique are also listed in Table I. It can be clearly observed that the GP-based model is more accurate and efficient since a lower RMSE is achieved with considerably less computation time compared to the ANN-based model.

TABLE I
RMSE AND COMPUTATION TIME COMPARISON OF GP AND ANN-BASED MODELING

	GP model	ANN model with number of hidden layer nodes as			
		10	15	20	40
Average RMSE	0.19	0.27	0.25	0.22	0.18
Computation time (seconds)	140	1140	1620	5880	30360

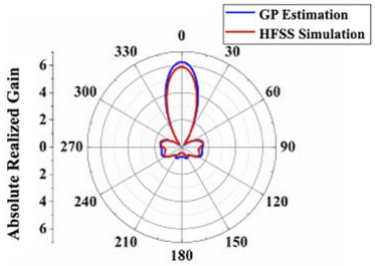
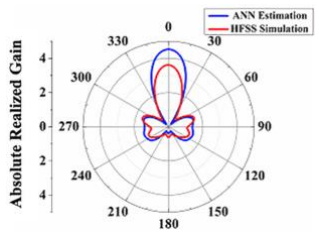
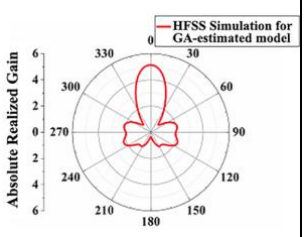
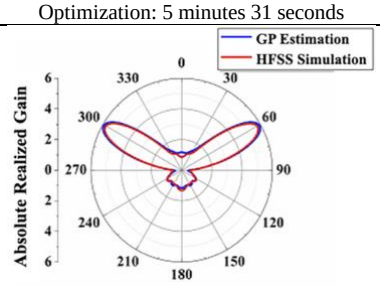
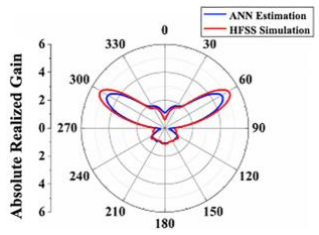
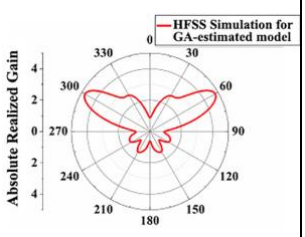
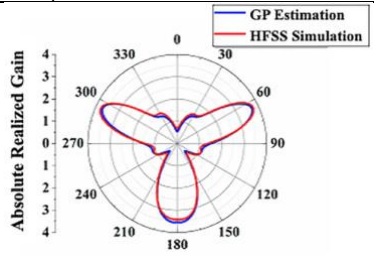
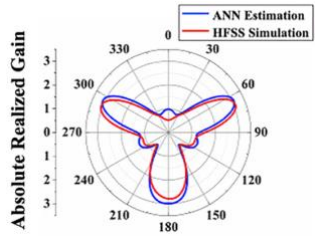
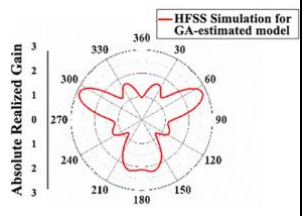
B. ML model-based optimization and Heuristic Optimization Comparison

Once the complete models are obtained using ML techniques, we can further use them for optimizing the design according to various requirements. In this case, we use the parameterized model for finding the dielectric distribution values in the input space that achieve goals such as desired radiation patterns. For this optimization, we use the constrained box optimization technique mentioned previously in Section II(c). This optimization method is applied to both the GP-based model and the ANN-based model (10-hidden layer nodes). The proposed optimization methodology is also compared with a classic heuristic optimization technique, genetic algorithm (GA). The methodology followed for GA optimization is similar to that described in [3]. This is done to have a fair comparison between optimization conducted using all the three optimization methods, namely, GP model-based optimization, ANN model-based optimization and GA-based optimization. It should be noted that in this work while running the GA model-based optimization, we save the simulations results in the memory so that data is re-used incase previously simulated datapoint gets regenerated rather than re-running the same simulations. However, such cases are rare as the optimization convergence varies with the optimization goal. GA optimization is only able to predict the optimal fitness function value and the corresponding design parameters that can potentially generate these values, while the ML-based methods can predict the gain performance from the computed model in addition to optimal fitness function value and the design parameters that can potentially provide this solution. The results for each of these are listed in Table II. The first three rows of Table II are divided into three sections, each section giving the results corresponding the three design goals, namely single-beam, two-beam, and three-beam gain patterns. The results for each of the three optimization methods are tabulated in the last three columns of the table. The first comparative property is the gain pattern plot predicted by the respective optimization method (in blue) together with that simulated by HFSS (in red). The second comparative property is the predicted fitness function value estimated using the respective optimization technique together with that predicted by HFSS. The GP-based optimization is also able to predict uncertainty value corresponding to the prediction value, as shown in the Table II in the row mentioned as 'Predicted Fitness function Value'. In addition, the total computation time involved in the respective analysis, which for the ML-based optimization methods includes both the model computation time and the optimization time are compared. Since GA-based optimization is a search algorithm and only searches for optimal design point that can potentially give optimal results and runs HFSS simulation after each search, it does not have any prediction curve. Hence, only the HFSS results (in red) corresponding to GA optimization for the gain-pattern and the fitness function value are shown.

It can be observed from Table II that the predicted fitness function values of GP-based optimization are 4.42, 5.06, and 2.25 respectively for each of the three goals, and they are close to the corresponding actual fitness values obtained by HFSS simulation, which are 4.06, 4.40 and 2.16 respectively. Moreover, these values of GP-based optimization are higher compared to those obtained using ANN-based optimization and GA-based optimization, indicating better optimization results. For the two ML-based optimization methods, i.e., the GP and ANN-based optimization, we use the same sets of training and testing data, so the total number of sample points and training-data collection time are the same. It takes 1.89×10^5 seconds for collecting the training data for both GP and ANN-based methods for achieving the three target goals. For optimization using these trained models, it takes only additional 138 seconds to complete model simulation in the GP case and then further 331, 351, and 396 seconds respectively for optimization in the one-beam, two-beam and three-beam case.

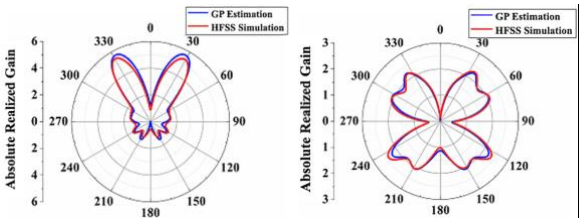
While for ANN, it takes only additional 1140 seconds to complete model simulation and then further 546, 562, and 579 seconds respectively for optimization for the one-beam, two-beam and three-beam cases. The total computation time for both the ML-based optimization methods for all three design goals together is only 22% less than that of the GA-based optimization (~53 hours vs. ~68 hours) and is mainly contributed by the time spent in obtaining the training data. To run a single HFSS simulation (considering simple dielectric blocks) takes 3 minutes 18 seconds, while once a model is trained, we can predict performance for given set of input

TABLE II
COMPARISON OF GP, ANN, AND GA-BASED OPTIMIZATION METHODS

	Properties	GP-based optimization	ANN-based optimization	GA optimization
One-beam	Gain Pattern			
	Predicted Fitness function Value	4.42 ± 0.24	3.04	3.72
	Actual Fitness function Value	4.06	2.04	
	Computation Time	Training data collection: 52 hours 30 minutes Model Computation: 2 minutes 18 seconds Optimization: 5 minutes 31 seconds	Model Computation: 19 minutes Optimization: 9 minutes 6 seconds	18 hours 53 minutes
Two-beam	Gain Pattern			
	Predicted Fitness function Value	5.06 ± 0.17	3.59	3.44
	Actual Fitness function Value	4.40	4.07	
	Total Computation Time	Training data collection: 52 hours 30 minutes Model Computation: 2 minutes 18 seconds Optimization: 5 minutes 51 seconds	Model Computation: 19 minutes Optimization: 5 minutes 22 seconds	29 hours 43 minutes
Three-beam	Gain Pattern			
	Predicted Fitness function Value	2.25 ± 0.11	1.67	1.02
	Actual Fitness function Value	2.16	1.23	
	Total Computation Time	Training data collection: 52 hours 30 minutes Model Computation: 2 minutes 18 seconds Optimization: 6 minutes 36 seconds	Model Computation: 19 minutes Optimization: 9 minutes 39 seconds	21 hours 8 minutes
Total no. of samples for all three goals		Training: 1050 Testing: 110		One-beam: 378 Two-beam: 594 Three-beam: 423 Total: 1395
Total time for all three goals		52 hours 50 minutes 16 seconds	53 hours 13 minutes 4 seconds	68 hours 1 minutes 44 seconds

design parameters within fractions of a second and the optimization takes a couple of minutes. It must be noted here that above computation time is based on the machine we used, i.e., Intel(R) Core (TM) i7-5820K CPU @ 3.30GHz processor

64 GB RAM, and 64-bit operating system. Once the training data has been collected, ML models can be easily reused for more design goals (i.e., modified fitness functions) at minimal extra computational cost, while GA will take much more time by calling for more simulations with each design goal. Based on the above data, the proposed ML-based optimization proves to be helpful in adaptive scenarios that could possibly have variable design goals to make quicker predictions. The dominant computational effort is spent in collecting data and training a relational model. This computational effort is done before, in a safe and secure environment, with no limitation of time and computation resources. The prediction based on the trained model could be calculated in few seconds, and this ultimately helps in adaptive scenarios to obtain real time response.



(a) (b)

Fig. 6. Absolute Realized Gain (on linear scale) for two other goals achieved with the proposed ML technique using the same training dataset (a) When beam separation of 60° is required for two-beam pattern; (b) when four-beam pattern is required.

Furthermore, the two ML-based optimization methods used in this work provide better results in terms of higher fitness function value search, where the GP-based method gives highest and more accurate fitness function value than the ANN method. In summary, it is demonstrated here that ML methods for antenna design are much more robust and scalable compared to the GA technique. Utilizing the GP-based ML method, we find the designs that satisfy other goals, for example, a design with two-beam radiation pattern separated by 60° instead of 120°, a design with 4-beam radiation pattern, etc. As shown in Fig. 6 (a) and (b), quite good results are achieved for both cases.

V. Experimental Realization and Validation

C. Designs Corresponding to GP-based optimization

To validate the GP-based design optimization, the obtained antenna designs are simulated in HFSS. For this purpose, the dielectric parameter values (ϵ_1 to ϵ_{18}) obtained from GP-based optimization are used to find each dielectric unit cell size using equation (13). The dielectric loading map for values predicted by the GP-based optimization for each of the three goals are shown in Fig. 7(a), (b) and (c). The corresponding realistic models are shown Fig. 7(d), (e) and (f). The resulting 3-D gain patterns from HFSS simulations are shown in Fig. 7(g), (h) and (i), which clearly represents the successful results obtained using the proposed GP-based modeling and optimization for

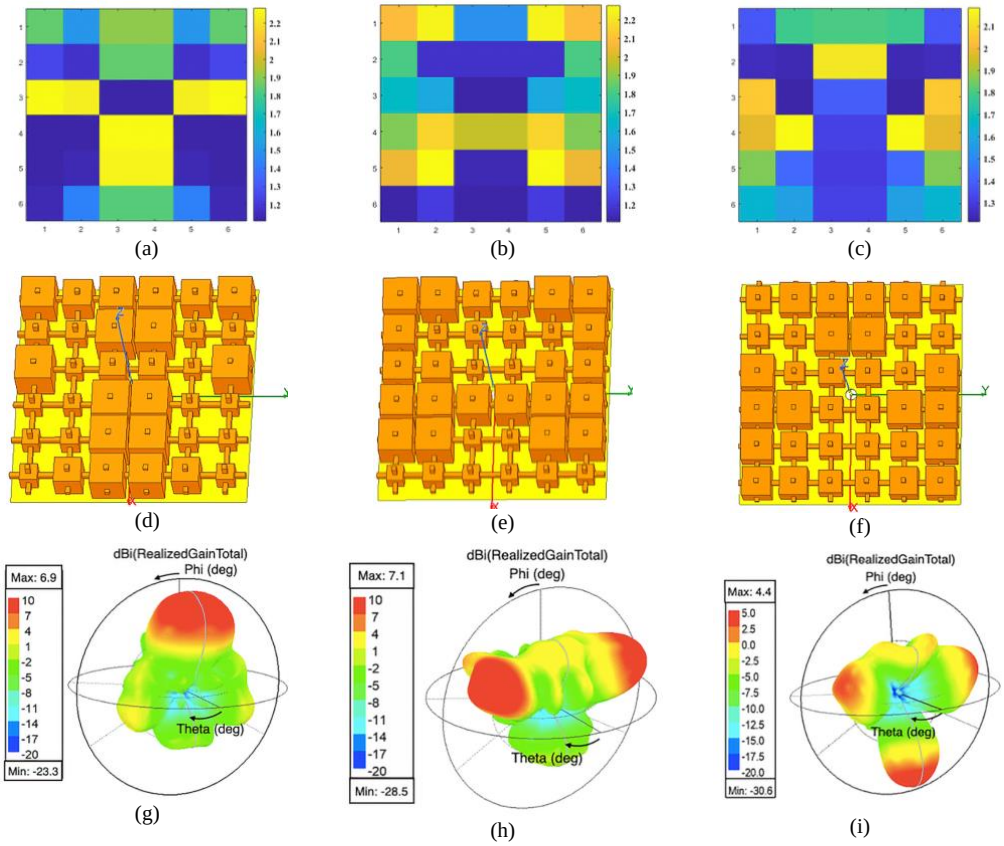


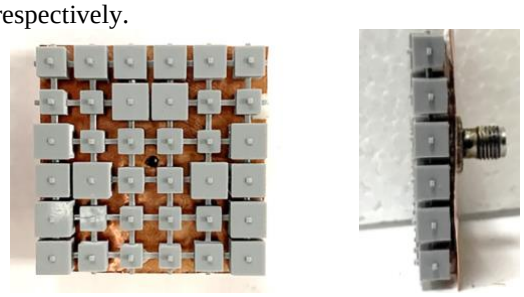
Fig. 7. Dielectric constant distributions around the monopole, predicted by GP-based optimization for (a) one-beam antenna, (b) two-beam antenna, and (c) three-beam antenna; HFSS simulation models of corresponding physical antenna realizations for (d) one-beam antenna, (e) two-beam antenna, and (f) three-beam antenna; and the corresponding 3-D (dBi) radiation patterns at 15 GHz for (g) one-beam antenna, (h) two-beam antenna, and (i) three-beam antenna, respectively.

each of the goals, i.e., one-beam, two-beam and three-beam radiation, respectively.

D. Fabrication and Measurement

As proof-of-concept, one of the obtained designs (the three-beam antenna design optimized using GP-based method) is fabricated and characterized to evaluate its performance. The dielectric part of this design (as shown in Fig. 7(f)) is 3D printed using a commercial Stereolithographic Apparatus (SLA)-based 3D printer, Asiga Max [28]. This printer uses Digital Light Processing (DLP) technology, which is easy to use and allows for fast, precise and efficient operation.

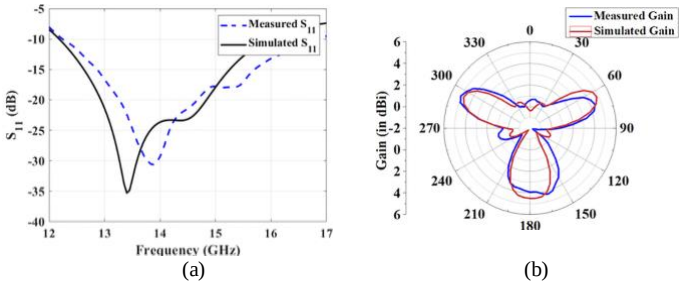
The size of the 3D-printed dielectric structure in grey color is 40 mm × 40 mm × 6.67 mm. After 3-D printing the dielectric part, a hole of 2 mm diameter is then drilled. A piece of steel wire of diameter 0.5 mm inserted into the SMA connector socket works as the monopole antenna. The dielectric structure is secured on the ground plane via thin double-sided tape. The top and side view of the final structure are shown in Fig. 8 (a) and (b) respectively.



(a) (b)

Fig. 8. Fabricated three-beam antenna prototype optimized using proposed GP-based optimization (a) Top view; (b) Side view.

The fabricated antenna prototype is characterized. The simulation and measurement results for reflection coefficient performance and 2-D realized gain pattern in the $\theta = 60°$ plane are shown in Fig. 9(a) and (b), respectively. The blue dashed curve in Fig. 9(a) is the measurement result for S_{11} with respect to frequency and the black solid curve is the simulated S_{11}. The solid blue curve in Fig. 9(b) is the measured gain in $\theta = 60°$ plane at 15 GHz while the solid red curve is the corresponding simulated gain.



(a) (b)

Fig. 9 Measured and simulated (a) S-parameter vs frequency performance and, (b) radiation pattern at $\theta = 60°$ plane and at 15 GHz.

As observed from Fig. 9(a), there is a slight shift in the S_{11} response which can possibly occur due to the fabrication tolerance of the monopole antenna with precise height. Also, in Fig. 9(b), there is some deviation between simulated and measured antenna gain pattern. Nevertheless, the overall agreement between simulation and measurement is reasonably well, demonstrating the soundness of the design and fabrication process.

VI. Conclusion

In this paper, ML-based modeling and optimization is proposed as a tool for designing complex antenna structures. As a proof-of-concept, an antenna design configuration that modifies monopole antenna's radiation pattern using dielectric loading is studied. By changing the dielectric constant distribution around the monopole, one-beam and multiple-beam radiation patterns can be obtained. In this work, two ML-based optimization techniques are studied, including Gaussian-Process (GP) and Artificial Neural Network (ANN). The performance of these two ML techniques-based optimization is compared with each other and also with a popular heuristic optimization technique, Genetic Algorithm (GA). The ML methodology provides a complete relational model between design parameters and performance specification, which can be further used to achieve designs for different optimization goals using the same set of training data. Therefore, unlike heuristic techniques such as GA, which searches for the optimal solution only by analyzing the output on individual data points until a global maxima or minima is identified, ML techniques can provide the complete relational model and hence be used as an optimization tool for computationally intensive antenna design problems involving a larger design space. The performance of the GP-based optimization method proves to be more accurate, faster and more informative (in terms of uncertainty in prediction). A three-beam monopole antenna prototype, designed using GP-based optimization, is successfully demonstrated.

This work suggests that ML-based optimization is an efficient method for the scenarios where we need to achieve multiple targets, as it can use the same set of training data to achieve different goals without requiring any new simulations. This methodology provides information regarding the performance of antenna structure with respect to the design and further uses this information for optimization, rather than just searching for the optimal design points. Hence, the ML-based optimization methodology can be a robust optimization tool for computationally intensive antenna design problems.

References

[1] A. Massa, G. Oliveri, M. Salucci, N. Anselmi and P. Rocca, "Learning by-examples techniques as applied to electromagnetics," *Journal of Electromagnetic Waves and Applications*, pp. 1-6, 2017.

[2] G. Godi, R. Sauleau, L. L. Coq and D. Thouroude, "Design and optimization of three-dimensional integrated lens antennas with genetic algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 770-775.

[3] J. Wu, A. H. Abdelrahman, M. Liang, X. Yu and H. Xin, "Monopole Antenna Radiation Pattern Control via 3D Printed Dielectrics," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 8, August 2017.

[4] Y. Rahmat-Samii and N. Jin, "Advances in Particle Swarm Optimization for Antenna Designs: Real-Number, Binary, Single-Objective and Multiobjective Implementations," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 556-567, 2007.

[5] S. Chamaani, M. S. Abrishamian and S. A. Mirtaheri, "Time-Domain Design of UWB Vivaldi Antenna Array Using Multiobjective Particle

Swarm Optimization," *IEEE Antennas and Wireless Propagation Letters*, vol. 9, pp. 666-669, 2010.

[6] T. S. Bird and A. A. Minasian, "Particle Swarm Optimization of Microstrip Antennas for Wireless Communication Systems," *IEEE Transactions on Antennas and Propagation,*, vol. 61, no. 12, pp. 6214-6217, 2013.

[7] S. Koziel and S. Ogurtsov, "Antenna design by simulation-driven optimization," in *Surrogate-Based Approach*, New York, Springer, 2014.

[8] J. Zhu, J. W. Bandler, N. K. Nikolova and S. Koziel, "Antenna optimization through space mapping," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 651-658, 2007.

[9] S. Koziel, S. Ogurtsov and S. Szczepanski, "Rapid antenna design optimization using shape-preserving response prediction," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 60, no. 1, pp. 143-149, 2012.

[10] S. Koziel, "Fast simulation-driven antenna design using response-feature surrogates," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 25, no. 5, pp. 394-402, 2015.

[11] Q. Wu, Y. Cao, H. Wang and et al., "Machine-learning-assisted optimization and its application to antenna designs: Opportunities and challenges," *China Communications*, vol. 17, no. 4, pp. 152-164, 2020.

[12] B. Lui, H. Aliakbarian, Z. Ma and et al., "An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 1, pp. 7-18, 2013.

[13] J. Tak, A. Kantemur, Y. Sharma and et al., "A 3-D-printed W-band slotted waveguide array antenna optimized using machine learning," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 11, pp. 2008-2012, 2018.

[14] L. Cui, Y. Zhang, R. Zhang and et al., "A modified efficient KNN method for antenna optimization and design," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 10, pp. 6858-6866, 2020.

[15] Y. Sharma, H. Zhang and H. Xin, "Machine learning techniques for optimizing design of double T-shaped monopole antenna," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 7, pp. 5658-5663, 2020.

[16] Z. Zheng, X. Chen and K. Huang, "Application of support vector machines to the antenna design," *International Journal on RF and Microwave Computer-Aided Engineering*, vol. 21, no. 1, pp. 85-90, 2010.

[17] "ANSYS Electromagnetics Suite 15.0, ANSYS, Inc.," [Online].

[18] C. E. Rasmussen, "Gaussian processes in machine learning," in *In Summer School on Machine Learning*, Berlin, Heidelberg, Springer, 2003, pp. 63-71.

[19] T. J. Santner, B. J. Williams, W. Notz and B. J. Williams, "The design and analysis of computer experiments," *Springer*, vol. 1, 2003.

[20] H. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 26, no. 6, pp. 508-517, 1978.

[21] I. Develi, "Application of multilayer perceptron networks to laser diode nonlinearity determination for radio-over-fibre mobile communicat-ions," *Microwave and Optical Technology Letters*, vol. 42, no. 5, p. 425–427, Sept 2004.

[22] J.-M. Wu, "Multilayer Perceptrons with Levenberg–Marquardt Learning," *IEEE Trans. Neural Network*, vol. 19, no. 12, pp. 2032-2043, Dec 2008.

[23] H. Demuth, M. Bele and M. Hagan, "Neural Network Toolbox".

[24] Matlab, "Non-linear Optimization Matlab," [Online]. Available: https://www.mathworks.com/help/optim/ug/fmincon.html.

[25] A. M. Nicholas and G. F. Ross, "Measurement of the Intrinsiv Properties of materials by Time-Domain Techniques," *IEEE Tans. Instrum. Meas.*, vol. 19, no. 4, pp. 377-382, Nov 1970.

[26] M. Liang, W. Ng, K. Chang, L. Gbele, M. Gehm and H. Xin, "A 3Luneburg lens antenna fabricated by polymer jetting rapid prototyping," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 4, pp. 1799-1807, 2014.

[27] Mathworks (2018a), "Curve Fitting Toolbox Release Notes (R2018a)," [Online]. Available: https://www.mathworks.com/help/curvefit/release-notes.html.

[28] Asiga [Online] Available: https://www.asiga.com/products/printers/max_series/max/.

[29] L. Wang, Y. Zhang and J. Feng , "On the Euclidean distance of images.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, p. 1334–1339, 2005.

Yashika Sharma received B.E. degree from Panjab University, India and M.Tech. in Electrical Engineering from IIT Kanpur, India, in 2014 and 2016, respectively and the Ph.D. degree in Electrical and Computer Engineering at the University of Arizona, Tucson, AZ, USA, in 2020. Dr. Sharma was a Chapter Chair of the Tucson Microwave Theory and Techniques Society Student Branch Chapter from 2017 to 2019. She is currently working with Apple Inc. Her research interests include RF, Microwave, Antenna design & optimization using ML techniques.

Xi Chen received B.S. and M.S. degrees in quality and reliability engineering from Beihang University, Beijing, China, in 2011 and 2015, respectively. She is currently pursuing the Ph.D. degree at Systems and industrial engineering department, University of Arizona, Tucson, AZ, USA. Her research interests include statistical modeling, industrial data analytics, and machine learning techniques on engineering applications.

Junqiang Wu received the B.S. and M. Eng. degrees in electrical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2010 and 2013, respectively, and the Ph.D. degree electrical and computer engineering at the University of Arizona, Tucson, AZ, USA, in 2017. He is currently working with Skyworks Inc., USA. Dr. Wu was a Chapter Chair of the Tucson Microwave Theory and Techniques Society Student Branch Chapter from 2015 to 2016.

Qiang Zhou is an Associate Professor at the Department of Systems and Industrial Engineering, and a faculty member of the Statistics Graduate Interdisciplinary Programs at University of Arizona. He was an Assistant Professor at the Department of Systems Engineering and Engineering Management, City University of Hong Kong during 2012~2016. His research focuses on advanced industrial data analytics, using statistics and machine learning methods, for engineering decision making and system performance improvement. He is a member of INFORMS and IISE.

Hao Helen Zhang is a Professor of Department of Mathematics at the University of Arizona, and also a faculty member of Statistics Graduate Interdisciplinary Program (GIDP). She obtained PhD degree in Statistics from University of Wisconsin at Madison, and was a faculty member of Department of Statistics at North Carolina State University between 2002-2011. Dr. Zhang's research areas include statistical machine learning, high dimensional data analysis, nonparametric smoothing, and biomedical data analysis, and her research has been funded by NSF, NIH, NSA, including an NSF CAREER award. Dr. Zhang has been serving on the editorial board of JASA, JCGS, and Statistical Analysis and Data Mining, and the current Editor-in-chief for the ISI journal Stat. Dr. Zhang is an elected member of the International Statistical Institute, Fellow of the American Statistical Association, and Fellow of the Institute of Mathematical Statistics.

Hao Xin (SM'10) received the Ph.D. degree in physics from the Massachusetts Institute of Technology (MIT), Cambridge, in 2001. He performed research studies for five years in MIT's Physics Department and at Lincoln Laboratory, where he investigated power dependence of the surface impedance of high-Tc superconducting films and Josephson junction properties at microwave frequencies. From November 2000 to November 2003, he was a Research Scientist with the Rock-well Scientific Company, where he conducted research as Principal Manager/Principal Investigator in the area of electromagnetic band-gap surfaces, quasi-optical amplifiers, electronically scanned antenna arrays, MMIC designs using various III-V semiconductor compound devices, and random power harvesting. From 2003 to 2005, he was a Sr. Principal Multidisciplinary Engineer at Raytheon Missile Systems, Tucson, AZ. He is now a professor and Director of the Cognitive Sensing Center in the Electrical and Computer Engineering Department and Physics Department at the University of Arizona, Tucson. His current research focus is in the area of microwave, millimeter wave, and THz technologies, including solid state devices and circuits, antennas, passive circuits, design and ML based optimization of 3D printed EM components and systems, and applications of new materials such as metamaterials and carbon nanotubes. Dr. Xin is an associate editor for IEEE Antennas and Wireless Propagation Magazine and the IEEE Journal of Radio Frequency Identification. He currently serves as the chair of the Young Professionals Committee of the IEEE Antennas and Propagation Society. He is also the Tucson joint chapter chair of the IEEE Microwave Theory and Techniques Society and IEEE Antennas and Propagation Society. He is also a fellow of IEEE.