

Accurate Modeling by Convolutional Neural-Network Regression of Resonant Frequencies of Dual-Band Pixelated Microstrip Antenna

Jan Pieter Jacobs , Senior Member, IEEE

Abstract—The modeling of the performance characteristics of pixelated microstrip antennas poses special difficulties as their geometries cannot be readily parametrized compared with the geometries of antennas with standard shapes. A methodology is presented for accurately modeling the two resonant frequencies of a dual-band pixelated microstrip antenna based on convolutional-neural-network (CNN) regression that takes a representation of the entire pixelated surface of the antenna as input. The predictive performance of the CNN and shallow and deep conventional feedforward neural-net architectures were compared; architectural and learning algorithm hyperparameters were determined by means of Bayesian optimization. The CNN achieved the best predictive results of all the networks with mean relative errors on test predictions of 0.13% and 0.22% for the two resonant frequencies, respectively.

Index Terms—Antennas, modeling, neural networks.

I. INTRODUCTION

MICROSTRIP antennas are a principal building block of microwave antenna engineering. Crucial for the design of single and multiband microstrip antennas is an understanding of their resonant frequency behavior. While previous studies concerned with antenna resonant frequency modeling have mostly focused on microstrip antennas intended for single-frequency operation, e.g., [1]–[5], resonant frequency modeling of dual-band antennas has also more recently been undertaken, e.g., involving patch antennas with cross slots [5], and patch antennas with U-shaped or center square slots [6]. Modeling methodologies that were used included standard feedforward neural networks, [1], [2], neural networks combined with fuzzy systems [3], [4], knowledge-based neural networks [5], support vector machines [7], and Gaussian process (GP) regression [6]. In all cases, the antennas were of standard shape (e.g., rectangular [1], [2], [4], circular or triangular [2], [4], or F-shaped or L-shaped [7]). If they contained slots, these were also of standard shape, e.g., U-shaped or square slots [6].

In view of the standard shapes of the above antennas, for modeling purposes, their geometries could be readily and sufficiently represented by a handful of geometry dimensions. For example,

a neural-network model of the resonant frequency of a probe-fed microstrip patch antenna could take the patch length and width, as well as the position coordinates of the probe feed as input variables and the corresponding resonant frequency as output.

The geometries of pixelated antennas, on the other hand, can generally not be readily represented by only a few geometry dimensions. Geometry instances of pixelated antennas are differentiated by their pixel configurations. These configurations can, for example, be obtained by the addition and/or removal of pixels with respect to some initial configuration of pixels, yielding overall antenna shapes that might seem to be irregular (e.g., with jagged outlines) while also possibly containing apertures that are irregularly shaped. It may be noted that the design of pixelated antennas usually involve evolutionary optimization algorithms that require a great many direct full-wave simulations of different geometry instances of the pixelated structure as it evolves toward a geometry that would satisfy design specifications [8], [9]. This entails significant computational expense and suggests a general need for inexpensive-to-evaluate machine-learning-based models that can be used as surrogates for the full-wave simulations.

The present letter demonstrates that the input impedance properties—in particular, resonant frequencies—of pixelated antennas can be accurately modeled using a neural-network-based strategy if the model input is a literal representation of the two-dimensional (2-D) geometry layout of the antenna. (This stands in contrast to resonant frequency modeling of antennas with standard shapes, where antenna geometries could be represented by a handful of dimensions [1]–[7]). For pixelated antennas, such a model input entails a mapping of the positions of metalized and empty pixels to a 2-D matrix of ones and zeros.

The neural-network architecture that seems most naturally suited to the above task is the convolutional neural net (CNN). These networks have, in recent years, been shown to achieve impressive results in tasks that involve classification and regression of images; the surface of a pixelated antenna expressed as a 2-D matrix containing ones and zeros is similar to black and white image data.

The predictive performance of a CNN is compared with the results obtained with various shallow and deep standard feedforward neural networks that took as vector input a flattened version of the above 2-D matrix. For all networks, important hyperparameters (architectural and learning algorithm parameters) were optimized by means of Bayesian optimization [10], [11],

Manuscript received August 16, 2021; accepted September 12, 2021. Date of publication September 20, 2021; date of current version December 20, 2021.

The author is with the Centre for Electromagnetism, Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa (e-mail: jppjacob@up.ac.za).

Digital Object Identifier 10.1109/LAWP.2021.3113389

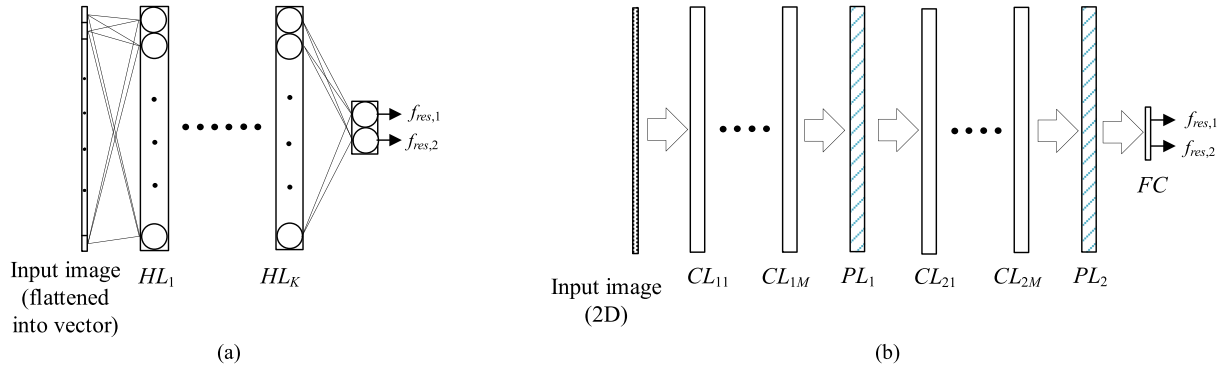


Fig. 1. Schematic representations of the neural networks implemented in the present study. Each network outputs the two resonant frequencies $f_{\text{res},1}$ and $f_{\text{res},2}$. (a) Standard feedforward neural-network architecture (HL_i is the i th hidden layer). (b) Convolutional neural network architecture. CL_{1j} , $j = 1, \dots, M$ and CL_{2k} , $k = 1, \dots, M$ are two blocks of convolutional layers that are each followed by a pooling layer, namely PL_1 and PL_2 . FC is a fully connected layer. The input image is a matrix of ones and zeros representing the pixelated region (footprint $P_x \times P_y$) of the pixelated dual-band antenna (cf. Fig. 2). All convolutional layer feature maps and pooling layers are 2-D layers.

which has very recently been demonstrated to be effective for model selection in electromagnetic modeling [12]. (It is noted that the present letter focuses exclusively on modeling input characteristics of dual-band pixelated antennas, in particular the resonant frequencies.)

The letter is organized as follows. Section II briefly gives theoretical background on neural networks, including CNNs, and Bayesian optimization. Section III presents implementation details of the proposed methodologies and numerical results. Finally, Section IV concludes this letter.

II. BACKGROUND TO NEURAL NETWORKS, INCLUDING CNNs, AND BAYESIAN OPTIMIZATION

A. Neural Networks

Fig. 1(a) shows a schematic representation of a standard fully connected feedforward neural network with K hidden layers. In general, the output vector \mathbf{y} of such a network, given an input vector \mathbf{x} , can be expressed as

$$\mathbf{y} = \mathbf{f}_{K+1}(\mathbf{f}_K(\dots \mathbf{f}_1(\mathbf{x}))) \quad (1)$$

where \mathbf{f}_i , $i = 1, \dots, K+1$ are vector functions of the form $\mathbf{f}_i(\mathbf{u}) = \mathbf{g}_i(\mathbf{W}_i \mathbf{u} + \mathbf{b}_i)$.

In the above, i is the layer index ($i = 1$ corresponds to the hidden layer adjacent to the input layer and layer K is the output layer), and \mathbf{g}_i is the activation function vector; in the present work, the familiar hyperbolic tangent (tanh) activation function was used. The parameter matrices \mathbf{W}_i (weights) and \mathbf{b}_i (offsets) are learned by using gradient descent to optimize a mean-square-error cost function [14]. When a network, such as that of Fig. 1(a), contains more than two hidden layers, it is referred to as a deep network.

A CNN [cf. Fig. 1(b)] is a specialized deep feedforward neural network that significantly reduces the number of weights to be learned, compared with the standard feedforward configuration, by focusing on local information in the input. Frequently used for image processing [13], [14], CNNs typically consist of a 2-D input layer, convolutional layers, some of which may be followed by pooling layers, and a fully connected layer or layers at the

end. Each convolutional layer is comprised of multiple feature maps, i.e., sublayers. Pooling layers have the same number of sublayers as their immediately preceding convolutional layers.

The convolutional nature of the network's operation may be understood as follows. Each feature map is associated with a convolution filter with a $p \times q$ receptive field. For the first convolutional layer (adjacent to the input layer), each filter slides across the input image using a preselected stride (one pixel in the present work), covering the full image. At each step, the convolution between the filter and the submatrix of the image included in its receptive field is calculated as the sum over elements of the elementwise product between the $p \times q$ matrix of trainable filter weights and the $p \times q$ image submatrix [13]. The sum of each convolution and a bias term gives the activation of the corresponding node in the feature map. Adjacent receptive field positions in the image are mapped to adjacent nodes in the relevant feature map. An activation function is applied to all nodes in the feature maps; in the present work, the rectified linear (ReLU) activation function was used, according to which the output $f(a)$ of a node with activation a is given by $f(a) = a$, $a \geq 0$ or $f(a) = 0$, $a < 0$.

Pooling is carried out separately for each feature map in the preceding convolutional layer; there is a pooling sublayer for each feature map. A sliding filter with max pooling was used, i.e., the output of the pooling operation gave the maximum value in the submatrix of the feature map included in the filter receptive field.

The final layer of the CNN is a fully connected layer, i.e., a 1-D layer, which is also the output layer of the network; linear activation functions were applied to nodes in this layer.

B. Bayesian Optimization

Bayesian optimization can be used to minimize an objective function $f(\mathbf{x})$ on a bounded set X [10]. In the present context, \mathbf{x} is the vector of neural-network hyperparameters to be optimized, and $f(\mathbf{x})$ is the predictive error on a validation set that is calculated after training the neural network using a particular instance of \mathbf{x} . It is assumed that $f(\mathbf{x})$ is drawn from a GP prior (a summary of GP

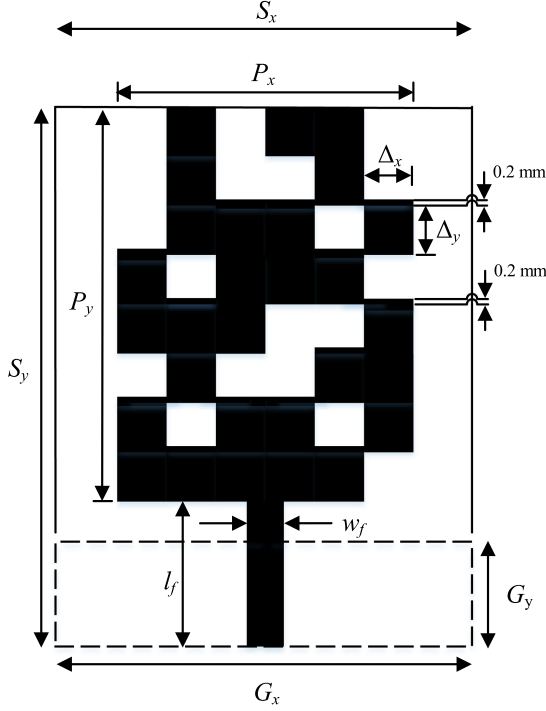


Fig. 2. Top view of dual-band pixelated antenna. Metalized pixels are indicated in black. The ground plane (on the opposite side of the substrate) is shown as a rectangle with a dashed line border.

regression is given in [6]). The strategy is to choose a new value of \mathbf{x} from a number of candidates such that the probability of improvement over the current best value of \mathbf{x} (referred to as \mathbf{x}_{best}) is maximized. The expected improvement can be expressed as [10]

$$a_{\text{EI}}(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) = \sigma(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta) \times (\gamma(\mathbf{x}) \Phi(\gamma(\mathbf{x})) + N(\gamma(\mathbf{x}) | 0, 1)) \quad (2)$$

where

$$\gamma(\mathbf{x}) = \frac{f(\mathbf{x}_{\text{best}}) - \mu(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)}{\sigma(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)}.$$

In (2), a_{EI} depends on previous observations of the objective function $\{\mathbf{x}_n, y_n\}$ as well as the GP hyperparameters θ ; $\mu(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)$ and $\sigma(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta)$ are, respectively, the predictive mean and square root of the variance function of the GP model; and $\Phi(\cdot)$ is the cumulative distribution function of the standard normal density, which is denoted by $N(\cdot; 0, 1)$. The MATLAB implementation used here included a mechanism for correcting overexploitation of subareas of the search space where applicable [11].

III. METHOD AND RESULTS

Fig. 2 shows the base geometry of the dual-band pixelated microstrip antenna considered in the present letter (derived from the article presented in [8]). Earlier experiments indicated that antenna geometries with a range of dual resonant frequencies can be obtained by filling in (with metalized pixels) two, three, or four of the empty pixels in the base geometry of Fig. 2 (all

metalized and empty pixels were confined to the rectangular area $P_x \times P_y$). Hence, geometry instances of this antenna were differentiated by the configurations of two, three, or four pixels, which are used to reduce the number of empty pixels in the base structure. Delimiting the variations in the pixelated antenna geometry in this manner rendered the modeling problem feasible. The number of geometries that could be achieved by using two or three pixels was 1140, and the number of possible geometries using four pixels was 3876; hence, the total number of possible geometries was 5016.

The base geometry had the following properties. The size of the finite substrate was $S_x \times S_y$ with $S_x = 23.85$ mm and $S_y = 34.45$ mm; the substrate had dielectric constant $\epsilon_r = 2.65$ and height $h = 0.795$ mm. The pixelated region had a footprint of size $P_x \times P_y$ with $P_x = 14.31$ mm and $P_y = 21.2$ mm, and the dimensions of an individual pixel were $\Delta_x = 2.385$ mm and $\Delta_y = 2.65$ mm. A further 0.2 mm of metalization was added to the top end of pixels bordered from above by empty pixels in order to avoid single-point connections (cf. Fig. 2). The antenna was fed by a microstrip feedline on the same side of the substrate with length $l_f = 13.25$ mm and width $w_f = 2.12$ mm. The microstrip ground plane (on the opposite side of the substrate) had dimensions $G_x = S_x$ and $G_y = 12.72$ mm.

In order to obtain the training and test data, 1940 different antenna geometries were randomly generated (1140 geometries where two/three pixels were added to the base geometry and 800 geometries with four pixels added). From these data, training, validation, and test geometries were compiled as follows: 578 geometries for training (228 two/three-pixel geometries and 350 four-pixel geometries); 378 geometries for validation (228 two/three-pixel geometries and 150 four-pixel geometries), and the remainder of the data for testing (684 two/three-pixel geometries and 300 four-pixel geometries).

The antenna geometries were simulated using the time-domain solver in CST Microwave Studio [15] over the frequency band from 1 to 11 GHz. First and second resonant frequencies $f_{\text{res},1}$ and $f_{\text{res},2}$ were determined from the minima of the $|S_{11}|$ -against-frequency responses produced by the simulations. The following ranges of $f_{\text{res},1}$ and $f_{\text{res},2}$ were represented: $3.2 \leq f_{\text{res},1} < 3.6$ GHz and $9.2 \leq f_{\text{res},2} < 11$ GHz.

Inputs to the CNN were 8×6 matrices corresponding to the pixelated part of the antenna geometries with ones/zeros used to indicate the presence/absence of a metalized pixel; these matrices were flattened into 48×1 vectors for presentation as input to the standard neural networks. Outputs of all networks were 1×2 vectors containing the two resonant frequencies (hence, the two resonant frequencies were accounted by a single model).

The CNN architecture, as shown in Fig. 1(b), used padding for all convolutional and pooling layers that maintained the 2-D dimensions of the immediately preceding layers/sublayers and filter sizes $p \times q$ of 3×3 and 2×2 for the convolutional and pooling layers, respectively. Bayesian optimization was carried out with respect to the number of convolutional layers per grouping of such layers [i.e., M in Fig. 1(b)] and the initial learning rate α . The number of filters in each convolutional layer was set equal to $16/\sqrt{M}$, rounded to the nearest integer [11]. Gradient descent via backpropagation was used to train all weights and biases [14],

TABLE I
MRE OF MODEL PREDICTIONS ON THE TRAINING AND TEST DATA

	Training geometries		Test geometries	
Model	$f_{res,1}$	$f_{res,2}$	$f_{res,1}$	$f_{res,2}$
CNN	0.05%	0.08%	0.13%	0.22%
NN ^a , 1 HL ^b	0.14%	0.28%	0.14%	0.29%
NN, 2 HLs	0.16%	0.25%	0.17%	0.28%
NN, 3 HLs	0.15%	0.26%	0.17%	0.29%
NN, 5 HLs	0.17%	0.31%	0.19%	0.32%

^aConventional feedforward neural network [cf. Fig. 1(a)]

^bHidden layer.

[16]. Xavier initialization was used to initialize the weights of the convolutional layers [17]. The Adam optimizer was used to train the network [14], [16], using a gradient decay factor of 0.9, a squared gradient decay factor of 0.999, and an epsilon value of 1×10^{-8} . The network was allowed to train for 2000 epochs, and batch normalization was used. The optimal hyperparameter values were $M = 3$ and $\alpha = 0.00101$.

Standard feedforward neural networks using, respectively, one, two, three, and five hidden layers were considered (i.e., two shallows and two deep nets). Bayesian optimization was carried out with respect to the initial learning rate α and the number of nodes per hidden layer (all hidden layers had the same number of nodes). The Adam optimizer used to train the networks employed the same parameters as the CNN. The optimal numbers of nodes per hidden layer were 25, 34, 25, and 24 for the networks with one, two, three, and five hidden layers, respectively, and the optimal values of α were 0.00262, 0.00200, 0.00224, and 0.00246.

The trained CNN was used to predict the resonant frequencies of the training and test antenna geometries. Table I gives predictive results in terms of the mean relative error (MRE), expressed as a percentage. The MRE was calculated separately for predictions on the training and test data (and separately for $f_{res,1}$ and $f_{res,2}$) using the following:

$$\text{MRE} = \frac{1}{N} \sum_{m=1}^N \frac{|f_{res,target}^{(m)} - f_{res,predicted}^{(m)}|}{f_{res,target}^{(m)}}. \quad (3)$$

In (3), m is an index over geometries, N is the number of geometries (it equals either the number of training or test geometries depending on whether the MRE is calculated for predictions on the training or test data), and $f_{res,target}^{(m)}$ and $f_{res,predicted}^{(m)}$ are the target and predicted resonant frequency values for the m th geometry, referring to either $f_{res,1}$ or $f_{res,2}$. Table I indicates that the CNN outperformed the standard feedforward neural networks, achieving highly accurate predictive results with $\text{MREs} \leq 0.22\%$ for predictions on the test data and even lower for predictions on the training data, as would be expected. For the feedforward networks, it did not seem as if adding more hidden layers had beneficial effects on the overall performance. (It may be noted

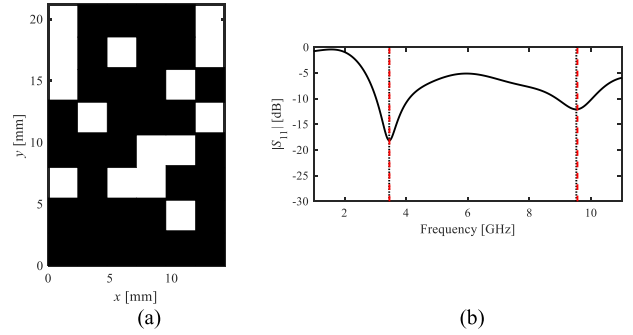


Fig. 3. CNN predictive results for a test geometry with four metalized pixels added to the base geometry. (a) Pixelated surface. (b) Corresponding predicted and target resonant frequency locations $f_{res,1}$ and $f_{res,2}$, superimposed on the $|S_{11}|$ -against-frequency response. The vertical dotted and dashed lines give the positions of, respectively, the target and predicted resonant frequencies (left: $f_{res,1}$; right: $f_{res,2}$).

that all neural nets used the same training, validation, and test data.)

If a number of independent neural-net models, such as those of Table I, were developed in practice, the selection of the best model would need to be performed without the benefit of access to test data. A possible strategy would be to use the validation error for model selection, and it indeed turned out that the average MRE for predictions of $f_{res,1}$ and $f_{res,2}$ for the validation geometries was the lowest for the CNN of all the networks.

Fig. 3 shows the pixelated region of a test geometry and the locations of its target and CNN-predicted dual resonant frequencies (these are superimposed on the simulated $|S_{11}|$ -against-frequency response); very good agreement is observed between the target and predicted values. The shown pixelated region differs from the base geometry in Fig. 2 in which four of the empty pixels of the base geometry have been filled with metalized pixels.

IV. CONCLUSION

Predicting the resonant frequencies of pixelated antennas is a novel modeling problem. The very good predictive results presented here indicate that delimiting the architecture of a CNN as described in Section II, and then using Bayesian optimization to determine the hyperparameters is an effective strategy for addressing this problem. It may be emphasized that CNNs have many parameters that can be tuned; optimization algorithms may take a long time to converge or have difficulty to converge if the search space contains many variables—hence, the above constraints on the architecture. The present approach relies on a model input representation that accounts for the entire 2-D layout of the pixelated surfaces of the geometries (unlike previous work on standard shape antennas where the geometries could be represented by a handful of geometry dimensions).

REFERENCES

- [1] D. Karaboga, K. Guney, S. Sagioglu, and M. Erler, "Neural computation of resonant frequency of electrically thin and thick rectangular microstrip antennas," *IEE Proc. Microw. Antennas Propag.*, vol. 146, no. 2, pp. 155–159, 1999.

- [2] K. Guney, S. Sagioglu, and M. Erler, "Generalized neural method to determine resonant frequencies of various microstrip antennas," *Int. J. RF Microw. Comput. Aided Eng.*, vol. 12, pp. 131–139, 2002.
- [3] G. Angiulli and M. Versaci, "Resonant frequency evaluation of microstrip antennas using a neural-fuzzy approach," *IEEE Trans. Magn.*, vol. 39, no. 3, pp. 1333–1336, May 2003.
- [4] K. Guney and N. Sarikaya, "A hybrid method based on combining artificial neural network and fuzzy inference system for simultaneous computation of resonant frequencies of rectangular, circular, and triangular microstrip antennas," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 659–668, Mar. 2007.
- [5] T. Khan and A. De, "Estimation of radiation characteristics of different slotted microstrip antennas using a knowledge-based neural networks model," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 24, no. 6, pp. 673–680, 2014.
- [6] J. P. Jacobs, "Efficient resonant frequency modeling for dual-band microstrip antennas by Gaussian process regression," *IEEE Antennas Wireless Propag. Lett.*, vol. 14, pp. 337–341, 2015.
- [7] S. Fei-Yan, T. Yu-Bo, and R. Zuo-Lin, "Modeling the resonant frequency of compact microstrip antenna by the PSO-based SVM with the hybrid kernel function," *Int. J. Numer. Model., Electron. Netw., Devices Fields*, vol. 29, no. 6, pp. 1129–1139, 2016.
- [8] M. Ding, R. Jin, and J. Geng, "Optimal design of ultra wideband antennas using a mixed model of 2-D genetic algorithm and finite-difference time-domain," *Microw., Opt. Technol. Lett.*, vol. 49, no. 12, pp. 3177–3180, 2007.
- [9] L.A. Griffiths, C. Furse, and Y. C. Chung, "Broadband and multiband antenna design using the genetic algorithm to create amorphous shapes using ellipses," *IEEE Trans. Antennas Propag.*, vol. 54, no. 10, pp. 2776–2782, Oct. 2006.
- [10] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," 2012. [Online]. Available: <https://arxiv.org/abs/1206.2944>
- [11] MATLAB and Statistics Toolbox Release 2019b, MathWorks, Inc., Natick, MA, USA, 2019.
- [12] N. Calik, M. A. Belen, P. Mahouti, and S. Koziel, "Accurate modeling of frequency selective surfaces using fully-connected regression model with automated architecture determination and parameter selection based on Bayesian optimization," *IEEE Access*, vol. 9, pp. 38396–38410, 2021.
- [13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: <http://www.deeplearningbook.org>
- [14] A. Geron, *Hands-on Machine Learning With Scikit-Learn, Keras, and Tensorflow*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.
- [15] CST Microwave Studio, version. 2020, Dassault Systemes, Vélizy-Villacoublay, France, 2020.
- [16] MATLAB and Deep Learning Toolbox Release 2019b, MathWorks, Natick, MA, USA, 2019.
- [17] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.