IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 16, 2021, accepted February 28, 2021, date of publication March 2, 2021, date of current version March 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063523

Accurate Modeling of Frequency Selective Surfaces Using Fully-Connected Regression Model With Automated Architecture Determination and Parameter Selection Based on Bayesian Optimization

NURULLAH CALIK[1], MEHMET ALI BELEN[2], PEYMAN MAHOUTI[3], AND SLAWOMIR KOZIEL[4,5], (Senior Member, IEEE)

[1]Department of Biomedical Engineering, Istanbul Medeniyet University, 34500 Istanbul, Turkey
[2]Department of Electrical and Electronic Engineering, İskenderun Technical University, 31200 Hatay, Turkey
[3]Department of Electronic and Communication, Vocational School of Technical Sciences, Istanbul University - Cerrahpasa, 34500 Istanbul, Turkey
[4]Engineering Optimization and Modeling Center, Department of Technology, Reykjavik University, 101 Reykjavik, Iceland
[5]Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, 80-233 Gdansk, Poland

Corresponding author: Slawomir Koziel (koziel@ru.is)

This work was supported in part by the Icelandic Centre for Research (RANNIS) under Grant 217771051, and in part by the National Science Centre of Poland Grant 2020/37/B/ST7/01448.

ABSTRACT Surrogate modeling has become an important tool in the design of high-frequency structures. Although full-wave electromagnetic (EM) simulation tools provide an accurate account for the circuit characteristics and performance, they entail considerable computational expenditures. Replacing EM analysis by fast surrogates provides a way to accelerate the design procedures. Unfortunately, modeling of microwave passives is a challenging task due to their highly-nonlinear outputs. Frequency selective surfaces (FSSs) constitute a representative example with their multi-resonant reflection and transmission responses that need to be represented over broad frequency ranges. Deep neural networks (DNNs) seem to be the promising techniques for handling such cases. However, a serious practical issue associated with their employment is an appropriate selection of the model parameters, including its architecture. A common practice is experience-driven setup, heavily based on trial and error, which does not guarantee the optimum model determination and may lead to multiple problems such as poor generalization or high variance of the model predictive power with respect to the training data set selection. This paper proposes a novel modeling framework, referred to as a fully-connected regression model (FCRM), where the crucial role is played by Bayesian Optimization (BO), incorporated to determine the DNN-based model setup, including both its architecture and the hyperparameter values, in a fully automated manner. For validation, FCRM is applied to construct the model of a Minkowski Fractal-Based FSS. The efficacy of the methodology is demonstrated through comparisons with several benchmark techniques, including the DNN surrogates established using the traditional methods as well as conventional regression models. The numerical results indicate that FCRM exhibits considerably improved prediction power and reduced sensitivity to the training sample assignment.

INDEX TERMS Surrogate modeling, microwave modeling, deep regression model, Bayesian optimization, metamaterials, frequency selective surfaces.

I. INTRODUCTION

Increasing performance demands imposed on contemporary communication systems, pertinent to various application areas (internet of things (IoT) [1], [2], wearable devices [3], or telemedicine appliances [4]), create strong demands for high quality microwave circuitry. Modern high-frequency components and systems are often expected to provide multi-band operation [5], [6], polarization/pattern diversity [7], [8], or to cover large portions of the frequency

The associate editor coordinating the review of this manuscript and approving it for publication was Weiren Zhu.

spectrum (e.g., [9]). Practical implementation of these features is only possible with the structures of high-level of complexity, for which full-wave electromagnetic (EM) analysis has become a mandatory design tool [10], [11], [12], [13]. EM simulation is versatile and offers high evaluation reliability. Yet, when used within the design procedures that require massive analyses, such as optimization [14], [15], or tolerance-aware design [16]–[18], the incurred computational expenses become an important consideration. Many recent studies focused on improving the efficiency of EM-based design procedures, leading to incorporation of adjoints sensitivities into gradient-based optimization algorithms [19], accelerating local procedures via sparse sensitivity updates [20], [21], the employment of machine learning [22], and surrogate-assisted procedures with both data-driven [23], [24], and physics-based surrogates [25].

Metamaterials are representative examples of passive structures, where the use of EM simulation models for design purposes is imperative. Metamaterials are created using series of unit cell elements placed in a specific pattern with respect to the wavelength pertinent to a given application. Appropriate adjustment of the unit cell material and geometry allows for realizing properties that cannot be found in nature (e.g., negative refraction index) [26]. Metamaterials may exhibit extraordinary capabilities in terms of electromagnetic wave manipulation (absorbing, bending, or enhancing), which enables designs with performance measures beyond those using conventional materials [27]–[35]. However, widespread use of metamaterials is hindered by their bulky configurations and complex fabrication processes. A possible workaround is offered by metasurfaces [36]. Similarly to metamaterials, metasufaces are two-dimensional designs consisting of unit cell elements [37]. Metasurfaces have the ability to extend the π phase span of a purely electric resonant plane, and to achieve a phase modulation of almost 2π, which permits the control of the wavefront and to excite both the electric and magnetic surface currents (e.g., to allow unidirectional scattering). In particular, the metasurfaces enable realization of transmission characteristics of maximum efficiency (i.e., with no reflection loss) [38], [39]. One of the commonly used metasurface structures are Frequency Selective Surfaces (FSSs) that merely exhibit an electric response [40], [41].

FSSs have been widely used in the design of shielding structures, absorbers, antenna radomes, reflectors [42]–[47], pre-filtering stages of filtenna designs [48], [49], or for performance enhancement of microwave antennas [50], [51]. As mentioned before, the FSS properties depend on the geometrical design parameters of the unit cell and the spatial arrangement of the cells. The relations between the design variables and the FSS performance measures are often highly nonlinear. Furthermore, the demands for low cost, limited size and broadband operation make the design process of FSS a complex endeavor, which, from numerical perspective becomes a multi-dimensional and multi-objective optimization task. The challenges are aggravated by the high cost of EM simulation, otherwise necessary to ensure FSS evaluation reliability. Using coarse (e.g., equivalent network) models compromises the accuracy, whereas attempting EM-driven design at the high-fidelity level of description may turn infeasible in computational terms.

A possible solution is the employment of fast surrogate models [52]–[54]. Surrogate modeling techniques have been widely used in high-frequency electronics, to provide low-cost representations of the various electrical and field responses such as scattering parameters [55]–[59], reflection phase of reflect-array antennas [60]–[62], characteristic impedance [63], or estimation of a microstrip patch antenna resonant frequency [64], [65]. Popular methods include polynomial regression [53], kriging [66], [67], support vector regression (SVR) [68], Gaussian process regression (GPR) [69] [70], but also artificial intelligence algorithms such as Artificial Neural Networks (ANNs) [71], [72], Machine Learning [73], [74], and Symbolic Regression [34], [35]. Among these, the Deep Learning (DL) algorithms have been recently gaining considerable attention [75]. DL has been widely used for solving estimation problems [76]–[84]. Due to their flexibility and universal approximation capability, ANN models in general, and DL surrogates in particular, seem to be adequate choices for representing the FSS electrical characteristics, which is the main point of interest of this work. Unfortunately, appropriate setup of the DL surrogate (selection of the architecture, hyperparameter adjustment) is not a trivial task [85]. The particular setups, often obtained by trial and error [86]–[89], are usually only appropriate for specific types of microwave components. The construction of models for structures such as FSS is particularly challenging: the training data sets are highly nonhomogenous, exhibit complex and nonlinear inner relationships, and rendering reliable models requires utilization of spatially local correlations between the geometry parameters and the system outputs.

This paper proposes a Deep Learning (DL)-based surrogate for accurate modeling of frequency selective surfaces, referred to as a fully-connected regression model (FCRM). One of its critical components is Bayesian Optimization (BO) [90], [91], incorporated to determine the DNN-based model setup, including its architecture (the number of layers, neurons, and batch normalization usage) and activation selection, in an automated manner. The network optimization is carried out to eliminate the need for engaging the expert knowledge in the process of model establishment, and to render a nearly-optimum surrogate for any particular microwave component. Our methodology is validated using a Minkowski Fractal-Based FSS and benchmarked against DNN surrogates obtained using the traditional methods but also kernel-based surrogates (e.g., kriging, GPR). The numerical results demonstrate that the FCRM framework exhibits considerably improved prediction power over the benchmark, and reduced sensitivity to the training sample assignment. The proposed approach can be considered a step towards making DL surrogates accessible to microwave engineers but also a technique

that constitutes a viable alternative to the standard surrogate modeling methods, in terms of being capable to adequately represent complex characteristics of structures such as FSSs.

II. MODELING METHODOLOGY

This section introduces the DL-based modelling framework considered in the paper. We start by recalling some background information about deep learning and surrogate modelling using deep neural networks (DNNs). The proposed fully-connected regression model (FCRM) is discussed in Section II. *B*, whereas the details concerning architecture and hyper-parameter selection via Bayesian Optimization (BO) are provided in Section II. *C*. The operational flow of the modelling procedure is summarized in Section II. *D*. The numerical verification of the framework can be found in Section III.

A. SURROGATE MODELING USING DEEP NEURAL NETWORKS.DEEP LEARNING AND ITS CHALLENGES

Surrogate modelling plays an important role in accelerating the simulation-driven design procedures of microwave components and devices [53]. Ultimately, this allows for rendering high quality designs in short periods of time, as compared to traditional methods. For the last decades, various techniques have been employed for modelling of microwave stages, including artificial neural networks (ANN) [71], [72], support vector machines (SVM) [68], Kriging [66], [67], or polynomial-based regression models [53]. Neural networks belong to commonly used methods. However, in the majority of applications in the field of microwave design, the basic versions of ANN surrogates are utilized, typically based on shallow neural networks, i.e., the models with no more than two hidden layers [54]. On the other hand, many of real-world microwave designs exhibit highly nonlinear dependence between the inputs (e.g., geometry parameters) and the outputs (typically, electrical characteristics) [7], [92]. Adequate representation of such dependencies by shallow ANNs is often hindered. This difficulty can be alleviated by the use of deep neural networks (DNN) [54], [75].

A distinguishing feature of DNNs is the increased number of hidden layers compared with traditional or shallow neural networks (SNN). This is, in a way, similar to the case linear perceptron: the property of a multi class classifier was ensured by increasing the complexity of model and the development of multi-layer perceptron, MLP [93]. Based on this approach, in order to overcome the limited flexibility of SNN, DNNs with considerably larger number of hidden layers and neurons have been proposed [54], [75]. The conceptual difference between SNN and DNN has been illustrated in Fig. 1.

The training process of DNN is referred to as Deep Learning (DL). Contrary to the training of SNN, DL is considerably more involved. The selection of hyperparameters (the number of layers, the number of hidden neurons, the activation functions, etc.) directly affects the overall performance of model in a significant manner. Unfortunately, no universal

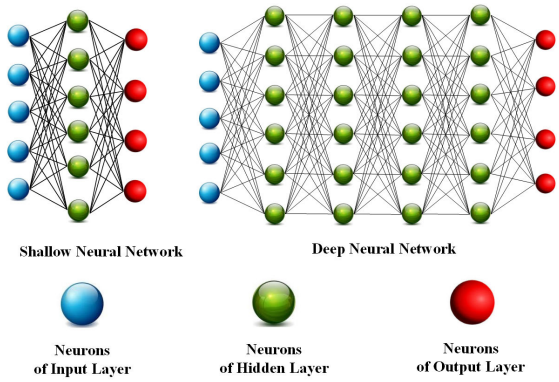

FIGURE 1. Architectural differences between SNN and DNN.

procedures for hyperparameter determination of DNN are available. Instead, the designers tend to use traditional methods such as greedy layer-by-layer [94], [95] or simply trial-and-error approaches.

Needless to say, practically useful DL procedures should ensure optimum training-validation performance, be accomplishable at a reasonable time, and avoid overfitting. To address these issues, this paper proposes a novel and fully automated training procedure for DNN-based surrogate models. It is formulated and explained in the remaining parts of this section, and, subsequently, applied to modelling of frequency selective surfaces in Section III.

B. PROPOSED TECHNIQUE: FULLY-CONNECTED REGRESSION MODEL(FCRM)

As explained before, efficient DL procedures are crucial for a successful employment of DNN surrogates. This work proposes a Fully-Connected Regression Model (FCRM), which integrates an enhanced DNN model architecture determination through Bayesian Optimization (BO) [96].

The block diagram of the FCRM architecture has been shown in Fig. 2. The model contains three types of functional units. The unit hierarchy, from the top to bottom is the following:

- *Blocks*. Blocks (BC) are the highest-level units encapsulating the detailed structure of the FCRM surrogate. The blocks may consist of different number of neurons, which make them act as inter-space transformers. Varying the number of neurons permits flexible data processing through the implementation of nonlinear mappings between the spaces of different dimensionalities. The number of blocks m is decided by the user, which is the only parameter not adjusted automatically within FCRM (cf. Section II. *C*).
- *Sub-blocks*. The sub-blocks (SB) determine the internal structure of the blocks. The number of sub-blocks in each block will be automatically determined during model identification. The sub-block layers may contain the same or different number of neurons, which affects the way of processing the data therein.

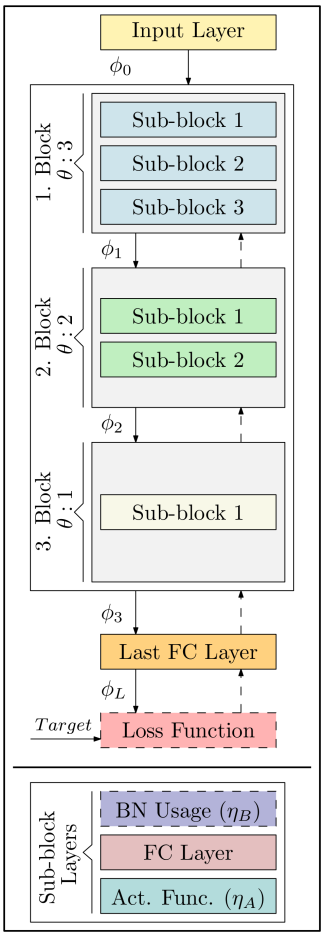

FIGURE 2. The generic architecture of the proposed Fully-Connected Regression Model (FCRM). The model contains m blocks, with $m = 3$ shown in the diagram for the sake of example. The specific architecture of the model is obtained through Bayesian Optimization (BO) (Section II. C), which yields the number θ of sub-blocks and the number ϕ of neurons for each block individually, except ϕ_0 and ϕ_L, which refer to the input and target sizes, respectively. In addition, BO also decides whether to use batch normalization (BN) ($\eta_B \epsilon 0, 1$), and which activation function (ReLU, Leak ReLU, or Tanh) will be employed ($\eta_A \epsilon 0, 1, 2$ respectively). In general, the number of sub-blocks and neurons within each block may be different. The number of neurons identified for any given block is the same in all sub-blocks therein. Using this information, the FCRM model is generated over the parameter space $S = \Theta \in \mathbb{Z}^{(mx1)}, \Phi \in \mathbb{Z}^{(mx1)}, \eta_A, \eta_B$. The Mean Absolute Error (MAE) is employed as the loss function during the model training. The data flow during the training process (gradient values φ_m) is marked using the dashed vectors. The right-hand-side panel of the picture illustrates the internal structure of the block.

- *Layers*. The layers are the internal building blocks of neural networks, here, the constituting elements of the sub-blocks. Each layer has an internal structure, which consists of the three components, referred to as batch normalization (BN), fully-connected layer (FC), and an activation unit. In FCRM, the activation unit is selected to be ReLU, Leak ReLU, or Tanh (Characterization of the various activation functions is provided below).

In a convolutional neural network (CNN) models, the input data is mapped into different spaces of the network in the form of a $N \times 1$ feature vector, which is transmitted to the FC layer for processing (classification, regression, ect.). FC layers act as MLP, where each neuron receives data from all neurons of the preceding layer [97]. This process corresponds to an affine transformation implemented by multiplying the data received from the previous layer by a matrix and adding biases. Nevertheless, a linear transformation is insufficient for representing nonlinear data distributions. In particular, appropriate nonlinear activation functions are needed to enable modelling of more generic input-output relationships. Sigmoid and *tanh* functions are the most preferred activation functions in conventional ANN structures. However, due to saturation, the gradients become close to zero on the asymptotic regions of sigmoid and *tanh* function. This problem, referred to as vanishing gradient, poses a practical challenge for DNN training.

To overcome the aforementioned issues, the Rectified Linear Unit (ReLU) activation functions have been proposed [98], [99]. ReLU has a non-vanishing gradient in the entire positive region. This way, the activation function remains operable for any positive values of the input argument. On the other hand, the vanishing gradient problem may occur in the negative region. To prevent this, a *leaky* ReLU, defined as

$$\mathrm{ReLU}_\alpha(x) = \begin{cases} x & x \geq 0 \\ \alpha x & \text{otherwise} \end{cases} \tag{1}$$

was proposed in [95]. Unlike standard ReLU, a leaky version permits negative values to be passed by means of the factor α, with $0 < \alpha < 1$.

Another advantage of ReLU-type activation functions are that they do not contain exponential expressions, so that the cost of forward and backward computations is low. Notwithstanding, although ReLU alleviates the vanishing gradient problem, the mere combination of FC and ReLU is insufficient for training large DNNs.

Training of DNNs is typically carried out using the gradient-based descent algorithms, which spread the lastest error to all trainable parameters of the model. In the meantime, the weights within the internal layers are adjusted based on the input data received. However, because the input data is generated by the preceding layers, it is altered upon completing a training phase. Consequently, the layers need to be re-trained by taking into account these changes. This problem is referred to as the *internal covariate shift*, and it has a detrimental effect on the stability of the DNN training process. The issue can be alleviated by Batch Normalization (BN) introduced [100]. BN is generally used in between the Convolution/FC and the Activation layers. It starts with a normalization of the layer means and variances. The usage of the entire dataset for this process is usually is impractical. Instead, the data is divided into smaller sets referred to as mini-batches, with normalization restrained to mini-batch members in the training process.

Based on the statistics of the mini-batch, a tensor data $\mathbf{X} \epsilon \mathbb{R}^{(NxMxCxB)}$ is first made zero-mean, then divided into channel dimension variances. Herein, N, M, C, and B are height, width, channel, and mini-batch size, respectively

where they represent the dimension parameters of 3D input data. This way, the mini-batch distribution is reshaped to have a unit variance along each axis. This process is called whitening [100]. Upon whitening, the data stack is affinely transformed using the multiplication factor γ and the shift β (both learnable). Both γ and β are adjusted in the training process so that the gradient of the activation function does not vanish after applying BN. During the training, BN is applied to each internal mini-batch. BN also improves the training process by reducing the network sensitivity to weight initialization.

The FCRM model constructed as described above is fully adjustable. In particular, the number of layers, layer neurons, activation type, and the usage of BN can be determined using Bayesian Optimization by taking into account the available training data, which will be explained in Section II. C. One of the important benefits of FCRM is that its architecture can be adaptively altered given specific input data and the model can evolve in an automated manner without involving user expertise and repetitive trial-and-error-based modifications.

C. FCRM SETUP VIA BAYESIAN OPTIMIZATION

Appropriate selection of internal parameters (also referred to as hyper-parameters) is critical for the models designed for any classification or regression tasks. In the case of ANN-based models, hyperparameters not only include the continuous variables (such as weights) but architectural parameters such as the number of layers and the number of neurons, both essentially contributing to the model performance. Although parameter identification can be carried out using population-based metaheuristic algorithms (e.g., Genetic Algorithm), this entails considerable computational expenses. In this work, the model training is executed by means of Bayesian Optimization (BO). BO utilizes probabilistic modelling techniques to enable global optimisation of complex and expensive functions [101].

BO is a population-based iterative algorithm using a surrogate model and an acquisition function as its key components [102], [103]. The surrogate model constructed from the initially acquired points is used to create a prior distribution of the probabilistic model. Various models can be used to represent the prior distribution [91], however Gaussian process (GP) priors for Bayesian optimization date back at least to the late 1970s [104], [105]. [106] explicitly set the framework for the Gaussian Process and showed that GP is well suited for the task. Thus, Bayesian optimisation uses GP as a surrogate model to generate a prior for the objective function. GP defines a function surface for the objective function based on the kernel functions it uses, and the minimum point on this surface is determined by the acquisition function. The subsequent GP model is evaluated again by incorporating the newly found data.

The acquisition function determines the new (infill) points by balancing the trade-off between exploration and exploitation [103]. During exploration, BO attempts to generate samples from the regions of the search space that were explored before. In the exploitation stage, the samples are generated based on the posterior distribution within the regions near the already explored parts of the space, which are more likely to contain the global optimum [103], [107]. BO is more efficient in computational terms than many derivative-free algorithms such as pattern search or random search methods [103]. A practical disadvantage is that BO is not suitable for parallelization. Yet, it, in many cases, the satisfactory results can be rendered in few iterations [108].

The basic steps of BO are the following. For a given objective function $f: X \rightarrow R$, BO attempts to determine the global minimum $x^* \in \text{argmin}_{x \in X} f(x)$. BO requires an initial knowledge of a prior distribution $p(f)$ over the objective function f and an acquisition function $a_{p(f)}: X \rightarrow R$. The prior distribution yields information about the input space X locations expected to contain improved objective function values, whereas the acquisition function allows us to guide the search for the optimum. Typically, the acquisition functions are defined such that high acquisition corresponds to potentially high values of the objective function, either because the prediction is high, the uncertainty is great, or both.

These components are used in an iterative process (see also Table 2): (i) find most suitable $x_i \in \text{argmax}\, a_{p(f)}(x)$ via optimization; (ii) evaluate $y_i = f(x_i)$, and add the resulting pair (x_i, y_i) to the observation set $D_i = \{x_i, f(x_i)\}_{i=1,\ldots,n}$; (iii) update the posterior distribution $p(f|D_i)$ and $a_{p(f|D_i)}$.

As mentioned before, Gaussian process (GP) is a statistical model of choice for Bayesian Optimization [109]. GP focuses on the values $f(x_j)$ of the function f on a observable set $\{x_j\}_{j=1,\ldots,n}$. It is assumed that the vector $[f(x_1)\ldots f(x_n)]$ is drawn from a prior probability distribution assumed to be multivariate normal with a particular mean and covariance matrix K obtained by evaluating the assumed covariance function (kernel) $k(x, x')$. Given the observation set $D_i = \{x_i, f(x_i)\}_{i=1,\ldots,n}$, a prior distribution $p(f)$ is combined with the likelihood function $P(D_i|f)$, the posterior distribution $p(f|D_i)$ follows another GP with mean and covariance functions [110]. The covariance function determines how observations influence the prediction. In this work, a Matérn 5/2 function [111], [112] has been used as kernel function

$$k_{5/2}(x, x') = \zeta(1+\sqrt{5}\, d_\lambda(x, x') + 5/3 d_\lambda^2(x, x'))e^{-\sqrt{5}d_\lambda(x,x')} \tag{2}$$

where $d_\lambda(x, x') = (x, x')\text{diag}(\lambda)(x, x')$ is the Mahalanobis distance. Here, ζ and λ are the free parameters of GP surrogate model. The algorithm in Table 1 also includes the noise parameter. This is because the pairs of samples in the data set can be noisy. Consequently, the surrogate model surfaces produced by GP do not fit the input and output data pairs $\{\boldsymbol{x}, f(\boldsymbol{x})\}$. This effect is added to the update phase as the noise.

The acquisition function is a key component of BO, which is used to control the balance between the parameter space exploration and exploitation. Some of the commonly used functions include Expected Improvement (EI) [113],

TABLE 1. Operation of the BN algorithm [#].

Algorithm 1 Batch Normalisation Transform	
Input 1: $\boldsymbol{B} = \{x_1 \ldots x_m\}$	Mini Batch set
Input 2: γ, β	Learnable parameters
Output: $y_i = B_{N,\gamma,\beta}[x_i]$	
Procedure	
$\mu_B \leftarrow E[x_i]$	
$\sigma_B^2 \leftarrow E[(x_i - \mu_B)^2]$	Zero meaning
$\hat{x}^i \leftarrow \frac{x^i - \mu_B}{\sqrt{\sigma_B^2 + c}}$	Normalization
$y_i \leftarrow \gamma \hat{x}^i + \beta$	Scaling
return y_i	

[#] Here, $\boldsymbol{B}$ represents the mini-batch consisting of the data points x_i, $i = 1, \ldots, m$. First, the mean vector μ_B is calculated, and the variance σ_B is found for each channel axis. Using these, the set $\boldsymbol{B}$ is rearranged and affinely transformed using the adjustable parameters γ (scaling) and β (shift) to a point where the gradients are operable for the relevant layer.

Upper Confidence Bound (UCB) [114], Entropy Search (ES) [115], and Predictive Entropy Search (PES) [116].

In this work, EI is taken as acquisition function. Its analytical form is

$$\alpha_{EI}(x|D_n) = E_p\{\max(f_{\min} - f(x), 0)\} \tag{3}$$

where $f_{\min}$ is the known best value of the function. E_p is the expectation operator over distribution p.

In the following, the application of BO for FCRM model training is outlined. It is assumed that the number m of blocks is fixed. As mentioned before, each block works as an intra-space transformer, whereas the transition between the blocks corresponds to the inter-space transformation. BO estimates the required number θ of sub-blocks and the neurons φ for each block (the number of neurons is the same for each sub-block). These are the design parameters of FCRM. In addition, FCRM decides whether to use B_N (variable η_B) within the sub-block, and which activation function should be used from the set {ReLU, Leak ReLU, Tanh} (variable η_A).

Thus, the FCRM design parameters are $S = \{\Theta \in \mathbb{Z}^{(m\times 1)}, \Phi \in \mathbb{Z}^{(m\times 1)}, \eta_A, \eta_B\}$. Therein, Θ and Φ, are $m \times 1$ column vectors. Since sub-block and neuron sizes are different for each block, these parameters are stacked in the vectors Θ and Φ, respectively. The activation function and the usage of BN are the same for all network, so these are indicated using single variables η_A, η_B, respectively.

The sub-blocks (S_B) determine the internal structure of the blocks. The sub-block function defining an S_B can be given as:

$$S_B(\cdot) = A_{\eta_A}(F_\phi(B_{N\cdot\eta_B}(\cdot))) \tag{4}$$

The function S_B is a composition of three mappings of previous S_B's outputs. As mentioned before, the control parameters of these layers are determined by BO, that allows neural architecture search. Blocks (B_C) are the highest-level units encapsulating the detailed structure of the FCRM surrogate. Using this definition, the entire intra-space transformation block (B_C) function can be defined as

$$B_C(\cdot) = S_B^{\theta}(\ \ldots\ S_B^2(S_B^1(\cdot))\ \ldots\ ) \tag{5}$$

where each S_B and B_C accept the outputs of previous S_B and B_C as input. The operation of the BO search algorithm in the context of FCRM has been summarized in Table 3.

TABLE 2. Pseudocode of bayesian optimization.

Bayesian Optimization[#] [109]
For $t = 1, 2, 3, \ldots..$ **do**
Find x_t via optimization of acquisition function over Gaussian Process
Sample the objective function $y_t = f(x_t) + N(0, \sigma^2_{noise})$,
Augment the data and update Gaussian Process
End for

[#] Here, x_i is the ith sample, y_t is the noisy observation of the objective function at x_i.

TABLE 3. Bayesian optimization algorithm.

Algorithm 2: Bayesian Optimization Search Algorithm	
Input of $s(\cdot)$: $\mathbf{D}_{tr}$, m, k, $iter$	Data matrix, block and folding params, maximum iteration index
Output $\{\phi, \theta, \eta_A, \eta_B\} = s(\mathbf{D}_{tr}, m, k, iter)$	
Procedure	
1. Set $t = 0$; Initialize $S^t = \{\Phi, \Theta, \eta_A, \eta_B\}$	
2. Generate *FCRM.net* model by using S^t	
3. Evaluate K-fold average score $Lavg$ for the model	
4. Update S using BO, store $\{S^t, L_{avg}\}$, and set $t = t + 1$	
5. **If** $t = iter$, **then** terminate the process, **else** go to 3	
6. **Return** S^{best} corresponding to minimum L_{avg}	

The input parameters of the procedure are the training data matrix ($\boldsymbol{D}_{tr}$), the number of blocks m, the fold number k, and the iteration step size $iter$. The output of the algorithm is the set of the best neural architecture parameters which lead to the minimum average k-fold loss value. BO first selects a random sample from the given parameter space. The k-fold error is evaluated for the model created using these parameters. Subsequently, the parameter set S and the average loss value $L_{(avg)}$) are stored, and S is updated. After the optimization process is completed, the parameter set that gives the minimum loss value is assigned as final model for the given training data.

In this study, the parameter ranges are set as follows: fixed (user-defined) block and sub-block size parameter: $m\epsilon\{1, 2, 3\}$, and $\theta\epsilon\{1, 2, 3\}$. The number of sub-block neurons is $\varphi\epsilon 2^a$, and BO tries to estimate best a value in the range of $a\epsilon\{1, \ldots, 10\}$. These parameters are individual for each block in the model. The parameter $\eta_A = \{0, 1, 2\}$ determines the type of the activation function to be employed: ReLU (0), Leak ReLU (1), or *tanh* (2). $\eta_B = \{0, 1\}$ indicates whether BN will be used or not, respectively. For Leak ReLU, the leakage value is set to $\alpha = 10^{-3}$. The performance of the optimized

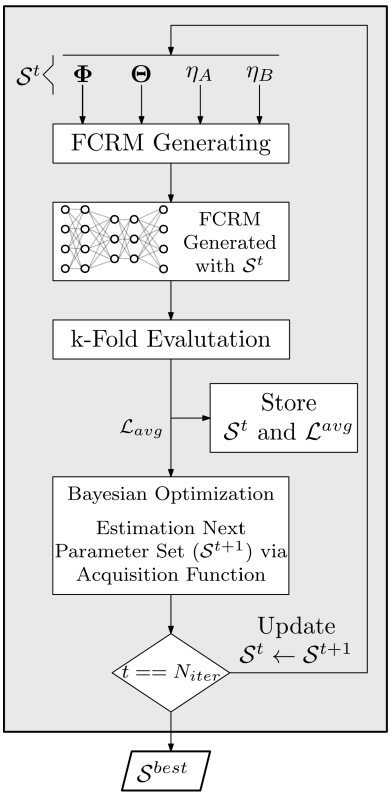

FIGURE 3. Block diagram of neural architecture search using bayesian optimization.

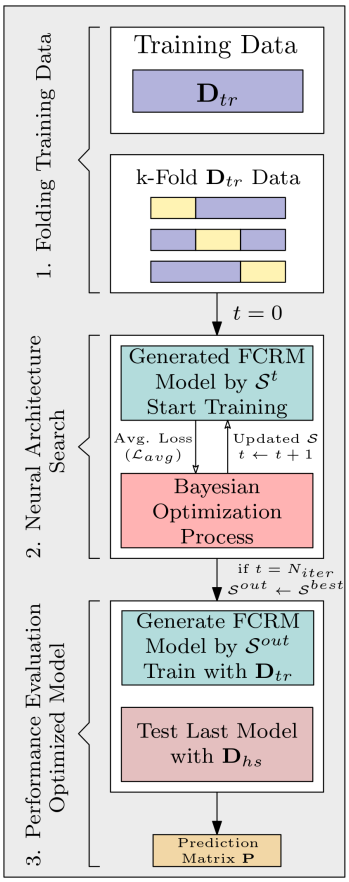

FIGURE 4. Block diagram of the modelling process using the proposed FCRM surrogate. The first step is the acquisition of the training and testing (hold out) data, D_{tr} and D_{hs}, respectively. The average loss (i.e., the modelling error, cf. Section II. *B*) of FCRM with *K*-fold (herein, $K = 3$) over the training data set is calculated at each iteration of model optimization. BO updates the FCRM architecture in order to reduce the loss as much as possible (cf. Section II. *C*). When the optimization is concluded ($t = N_{iter}$), the hyperparameter set S featuring the lowest k-fold error is assigned as S^{best}. Subsequently, the final FCRM surrogate is constructed using S^{best}, and trained over the entire D_{tr}.

architecture trained with $\mathbf{D}_{tr}$ is tested using the holdout data set ($\mathbf{D}_{hs}$). Once $S^{(best)}$ is obtained, a new model is trained with all $\mathbf{D}_{tr}$, then tested with $\mathbf{D}_{hs}$.

For the sake of additional clarification, Fig. 3 explains the process of neural architecture search using a flow diagram. As elaborated on above, the hyper-parameter determination of the FCRM surrogate is carried out through Bayesian optimization. In the first step, BO randomly generates an initial set for the FCRM architecture search S^t ($t = 0$). The Mean Absolute Error (MAE) is employed as the error metric for k-fold average score (L_{avg}) which also is the cost function of the BO process. Based on the L_{avg} performance of previously observed t parameter sets, BO tries to estimate an improved hyper parameter selection for the FCRM model in subsequent iterations using the acquisition function (Eq. 3) to reduce the k-fold average score L_{avg}. The process is repeated until the maximum iteration is achieved. In this work, the maximum iteration count is set to 30. It should be mentioned that BO can be used for hyper-parameter optimization of other types of regression models as well. Examples include GRNN and SVRM.

D. MODELING FRAMEWORK

This section summarizes the overall modelling procedure using the FCRM surrogate. It is shown in Fig. 4 in the form of a flow diagram. In the first step, the training and hold out datasets are acquired for a selected case study. In the second step, the model is constructed according to the number of m-blocks decided upon by the user. BO defines a random hyperparameter set S at iteration $t = 0$, and the model is initialized. Subsequently, the training data $\mathbf{D}_{tr}$ is partioned into k parts, and the average loss value resulting is transferred to the BO block. BO updates the hyperparameter set S, and the iteration counter is incremented. The process continues until $t = N_{iter}$.

When the BO process is completed, the hyperparameter set that results in the lowest cross-validation loss found on the training data is identified and marked as S^{best}. The FCRM surrogate is then is recreated according to S^{best} and trained using the entire $\mathbf{D}_{tr}$. The actual validation of the final model is carried out using the hold out data set $\mathbf{D}_{hs}$, which has never entered the training process.

III. VERIFICATION AND BENCHMARKING

This section provides numerical verification and benchmarking of the proposed surrogate modelling approach. The benchmark set includes ANN-based models (Radial Basis

Functions, Generalized Regression Neural Networks, and Deep Residual Neural Network), with the architectures assigned using conventional methods, as well as kernel-based surrogates such as Support Vector Regression Machines (SVRM), Gaussian Process Regression (GPR), and Gradient Boosted Greedy Trees Regressor. The verification is carried out using a specific example, which is a Minkowski fractal-based FSS. This type of structure is of sufficient level of complexity, which includes the necessity of modelling the highly-nonlinear response over broad frequency range.

A. CASE STUDY: MINKOWSKI FRACTAL-BASED FS

Here, we introduce a case study for performance evaluation of the proposed surrogate modelling technique. In particular, we consider a unit element based on Minkowski fractals to be used as a building block of a frequency selective surface (FSS). Minkowski fractals, also known as Minkowski Sausage, named after a German mathematician Hermann Minkowski in 1907. The initial geometry of the fractal, called the initiator, is a square, where each of the four straight segments of the initial structure was replaced with a generator [117].

The topology and parameterization of the FSS has been shown in Fig. 5, where C, I_{ta}, H_s, are geometrical design variables of the proposed FSS that are in millimeter; M is a multiplication coefficient determining the long-side unit cell size with respect to the parameter C. Table 4 provides the ranges of the geometry parameters and the considered frequency range. The considered FSS has been formed using a Minkowski shaped cavity in the base material, here, copper.

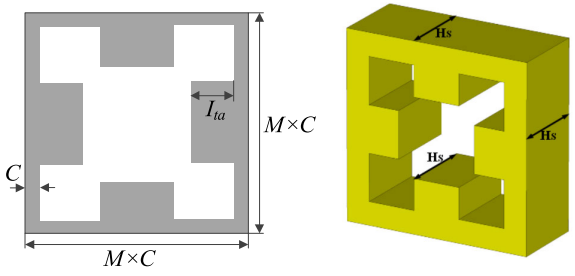

FIGURE 5. Geometry of the FSS unit cell.

TABLE 4. The ranges of geometry parameters of the FSS unit cell of Fig. 4, considered in verification experiments.

Parameter	Range
C (mm)	[0.6 , 1.0]
H_s (mm)	[0.5 , 5.0]
I_{ta} (mm)	[0.1 , 0.9]
M	[5.0 , 15.0]
Frequency (GHz)	[6.0 , 10.0]

B. EXPERIMENTAL SETUP

For the sake of validating the proposed methodology, the surrogate models have been constructed using four data sets obtained using Latin Hypercube Sampling (LHS) design of experiments strategy [118]–[121], as indicated in Table 5. The Datasets 1 through 3 are used for determination of the optimal required data for training the FSS unit element, whereas Dataset 4, which is generated via random sampling in the parameter space determined in Table 4, will be used as hold-out data for evaluating the over-fitting of each model.

TABLE 5. Generated datasets for surrogate model construction.

Data Set	Sampling Method	Number of Simulation Samples
1	Latin hypercube	100
2	Latin hypercube	300
3	Latin hypercube	500
4	Random	300

The aforementioned datasets are used to train the proposed FCRM model as well as the benchmark surrogates. The results reported in the following parts of this section concern both their training and holdout performances.

The Mean Absolute Error (MAE) and Symmetric Mean Absolute Percentage Error (SMAPE) metrics had been used for performance evaluation of all models:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|T_i - P_i| \tag{6}$$

$$SMAPE = \frac{100\%}{N}\sum_{i=1}^{N}\frac{|T_i - P_i|}{(|T_i| + |P_i|)/2} \tag{7}$$

where T_i and P_i are the target and the predicted values of the i^{th} sample; N is the total number of samples.

The proposed model has been compared with both state-of-the-art and conventional regression methods listed in Table 6. There are twelve different regression models, each trained using Datasets 1, 2, and 3. For the training process a K-fold

TABLE 6. Modelling techniques utilized in numerical experiments: Models 1 through 9 are benchmark techniques; Models 10 through 12 refer to variations of the proposed FCRM framework.

Model	Model Name	Model specifications
1	Keras Deep Residual Neural Network Regressor [122]	2 Layers: 512, 512 Units
2	Gradient Boosted Greedy Trees Regressor [125]	Early Stopping (Least-Squares Loss)
3	Nystroem Kernel SVM Regressor [128]	Gamma=0.25; number of components 500
4	Gaussian Process Regression with Block Coordinate Descent [131]	Kernel Function: matern32
5		Kernel Function: matern52
6		Kernel Function: ardmatern32
7		Kernel Function: ardmatern52
8	RBF [133]	Spread 0.65
9	GRNN [134]	Spread 0.13
10	FCRM	Block Size (m) : 1
11		Block Size (m) : 2
12		Block Size (m) : 3

validation with $K = 3$ alongside of a hold-out validation test with Dataset 4 has been carried out, in order to obtain a conclusive account for the performance, including over fitting of the models.

The hardware setup of the used computer system is Intel®Core™i7-67000K CPU @ 4.0 GHz with 16 GB installed memory, alongside of GTX 1080TI on a 64-bit operating system. BO runs for 30 iterations. The total computational time of model optimization depends on the number of blocks. For $m = 1$, it is 37, 49, and 54 minutes for Data sets 1, 2, and 3, respectively. For $m = 2$, it is 46, 55, and 63 minutes for Data sets 1, 2, and 3, respectively. Finally, for $m = 3$, we have 51, 59, and 74 minutes for Data sets 1, 2, and 3, respectively.

The data in Table 6 specifies the type of the benchmark technique as well as the setup details. Model 1 is the Keras Deep Residual Neural Network Regressor [122] with two hidden layer consisting of 512 hidden neuron each trained using the Adam's optimizer. Model 2 is the Gradient Boosted Greedy Trees Regressor with 1500 estimators, max leaf node of 8 with learning srate of 0.05. Gradient Boosting Machines (GBM), also known as Generalized Boosted Models, belong to the state-of-the-art algorithms known as highly accurate regression models [123]–[125]. A Support Vector Machine (SVM) Regressor model had been taken into consideration as Model 3. The SVM algorithm is a nonlinear generalization of the Generalized Portrait algorithm [126].

Recently several fast kernel approximation methods for having computationally efficient models have been developed such as Nystroem Kernel [127], [128]. Here, the SVM model with user defined parameters of (i) number of component = 500, (ii) Gamma = 0.25 and (iii) kernel function of Nystroem, has been used. Models 4 through 6 are models based on Gaussian Process Regression (GPR) with Block Coordinate Descent (BCD). GPR model (kriging) are nonparametric kernel-based probabilistic model [129]. For a large number of observations, using the exact method for parameter estimation and making predictions on new data might become computationally inefficient. One of the solution methods for having computationally efficient GPR model is BCD method [130], [131]. Here, four different GPR models with BCD with respect to the variant kernel functions have been used in Models 4 through 6.

To enable comparison with traditionally used methods, radial basis function Neural Network, and General Regression Neural (GRNN) have been used in Models 8 and 9 [132]–[134].

Finally, as mentioned before, the proposed FCRM framework (Models 10 through 12) is applied in three different variations, constructed assuming different block sizes $m = 1$, 2, and 3.

C. RESULTS, BENCHMARKING AND DISCUSSION

The modeling results obtained using the Datasets 1 and 4 have been gathered in Table 7. The table provides the MAE and SMAPE error values for all considered models, including the proposed FCRM framework, and the benchmark techniques. Note that in this case only 100 training data samples were used to set up the surrogates. It can be observed that most of the methods exhibit a good K-fold training error, whereas the holdout test clearly indicates that the models are over-fitted. Among the considered benchmark approaches, Gaussian

TABLE 7. Performance comparison of models using dataset 1 and 4.

Model	Error Metric	K-fold	Holdout
1	SMAPE	15.1	60.3
	MAE x 10^{-3}	17.8	132.5
2	SMAPE	16.2	30.1
	MAE x 10^{-3}	20.8	52.5
3	SMAPE	28.1	40.8
	MAE x 10^{-3}	28.6	67.25
4	SMAPE	1.99	27.4
	MAE x 10^{-3}	2.0	57.3
5	SMAPE	1.78	28.2
	MAE x 10^{-3}	1.86	59.64
6	SMAPE	1.76	17.1
	MAE x 10^{-3}	1.63	25.1
7	SMAPE	1.66	17.3
	MAE x 10^{-3}	1.67	25.1
8	SMAPE	47.9	50.2
	MAE x 10^{-3}	98.9	103.5
9	SMAPE	17.4	37.6
	MAE x 10^{-3}	22.4	66.6
10	SMAPE	3.68	5.91
	MAE x 10^{-3}	3.13	7.67
11	SMAPE	3.49	6.7
	MAE x 10^{-3}	3.11	9.73
12	SMAPE	2.74	5.9
	MAE x 10^{-3}	2.31	8.16

TABLE 8. Performance comparison of models using dataset 2 and 4.

Model	Error Metric	K-fold	Holdout
1	SMAPE	29.1	56.3
	MAE x 10^{-3}	65.9	125.2
2	SMAPE	15.7	22.2
	MAE x 10^{-3}	26.3	35
3	SMAPE	26	36.9
	MAE x 10^{-3}	44.3	55
4	SMAPE	2.66	20.4
	MAE x 10^{-3}	3.19	41.6
5	SMAPE	2.71	21.2
	MAE x 10^{-3}	3.15	43.8
6	SMAPE	3.20	9.24
	MAE x 10^{-3}	3.09	11.9
7	SMAPE	3.65	9.50
	MAE x 10^{-3}	3.68	12.2
8	SMAPE	45.1	46.5
	MAE x 10^{-3}	91.4	94.3
9	SMAPE	16.5	32.1
	MAE x 10^{-3}	21.1	53.6
10	SMAPE	2.93	3.49
	MAE x 10^{-3}	2.75	3.59
11	SMAPE	3.03	3.46
	MAE x 10^{-3}	2.78	3.34
12	SMAPE	2.8	3.41
	MAE x 10^{-3}	2.71	3.64

TABLE 9. Performance comparison of models using dataset 3 and 4.

Model	Error Metric	*K*-fold	Holdout
1	SMAPE	11.1	61.1
	MAE x 10^{-3}	10.8	136
2	SMAPE	15.9	19.7
	MAE x 10^{-3}	16.4	29
3	SMAPE	28.5	32.5
	MAE x 10^{-3}	41.9	46.7
4	SMAPE	4.33	17.1
	MAE x 10^{-3}	5.6	33.3
5	SMAPE	4.72	17.8
	MAE x 10^{-3}	6.2	35.2
6	SMAPE	3.84	6.86
	MAE x 10^{-3}	4.0	8.6
7	SMAPE	4.23	6.93
	MAE x 10^{-3}	4.7	8.7
8	SMAPE	42.2	43.2
	MAE x 10^{-3}	83.8	85.2
9	SMAPE	16.6	28.6
	MAE x 10^{-3}	21.2	46.8
10	SMAPE	3.6	3.44
	MAE x 10^{-3}	3.45	3.26
11	SMAPE	2.3	2.62
	MAE x 10^{-3}	2.35	2.66
12	SMAPE	2.53	2.52
	MAE x 10^{-3}	2.4	2.57

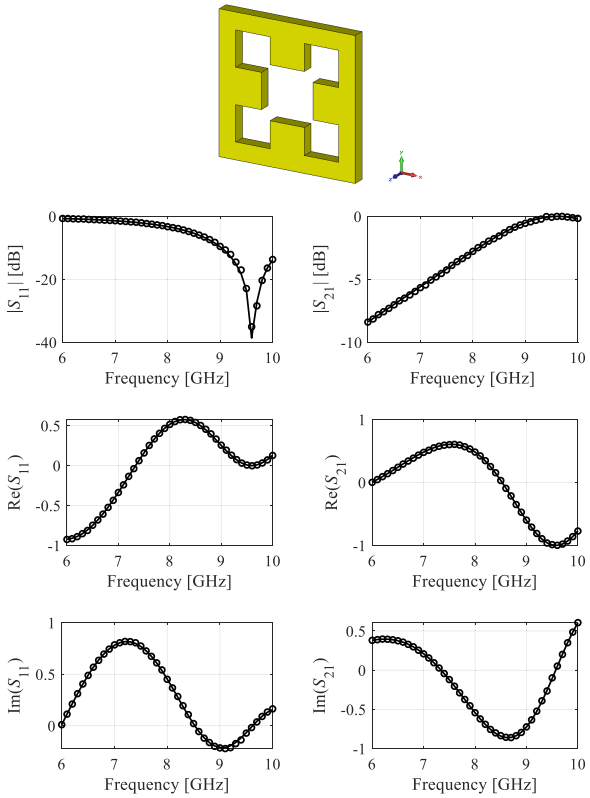

FIGURE 6. Visualization of the FCRM model performance for a selected FSS cell geometry (case 1): $C = 0.82$, $H_S = 1.4$, $l_{ta} = 0.75$, and $M = 12.5$. Surrogate constructed for $m = 3$ using Dataset 3; EM-simulation data (–), and FCRM surrogate (o).

Process Regression with Block Coordinate Descent is by far the best. Still, its overall performance is not even close to that of the proposed method due to its over-fitting performance.

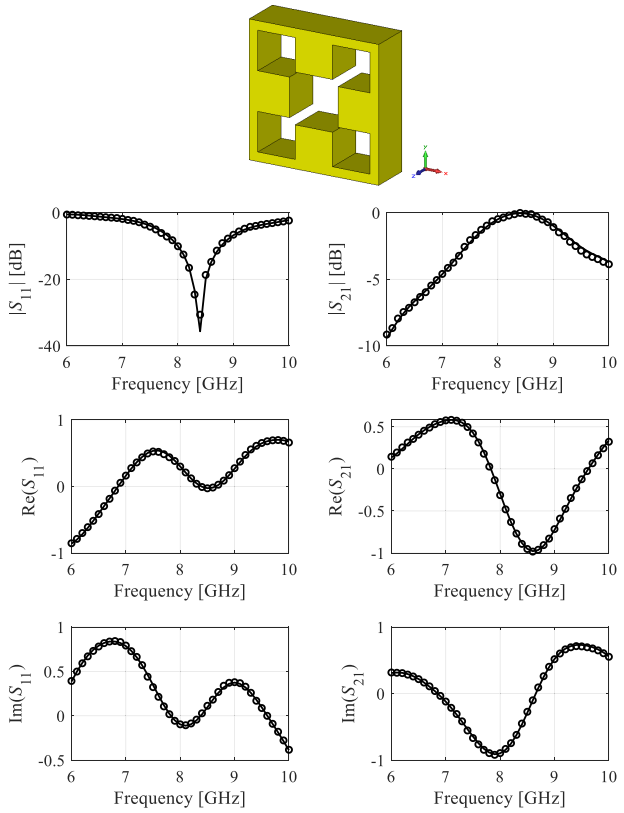

FIGURE 7. Visualization of the FCRM model performance for a selected FSS cell geometry (case 2): $C = 0.66$, $H_S = 4.43$, $l_{ta} = 0.85$, and $M = 12.4$. Surrogate constructed for $m = 3$ using Dataset 3; EM-simulation data (–), and FCRM surrogate (o).

Given the sparsity of the training data set, the accuracy of the FCRM surrogate can be considered excellent.

In the next training case, cf. Table 8, a dataset with 300 samples generated via LHS has been used for training the surrogates. The results are consistent with those presented in Table 7 with the FCRM model performing by far the best, both in terms of approximation and over-fitting indicators.

As a final verification case, the models were rendered using the largest dataset consisting of 500 samples. The results have been gathered in Table 8. Althought noticeable improvement can be observed for the majority of considered techniques, the over-fitting issue is still pronounced for the benchmark methods. This is not the case for the FCRM surrogate, which is the overall winner also in this setup. However, in this case, the performance of GPR is relatively close to that of FCRM.

The results of Tables 7 through 9 can be summarized as follows: the FCRM surrogate allows for the most efficient utilization of information contained in the training data samples, and its advantages are particularly visible for small data sets. Having in mind that verification experiments have been carried out in a comprehensive manner (including nine benchmark methods), the results can be considered conclusive.

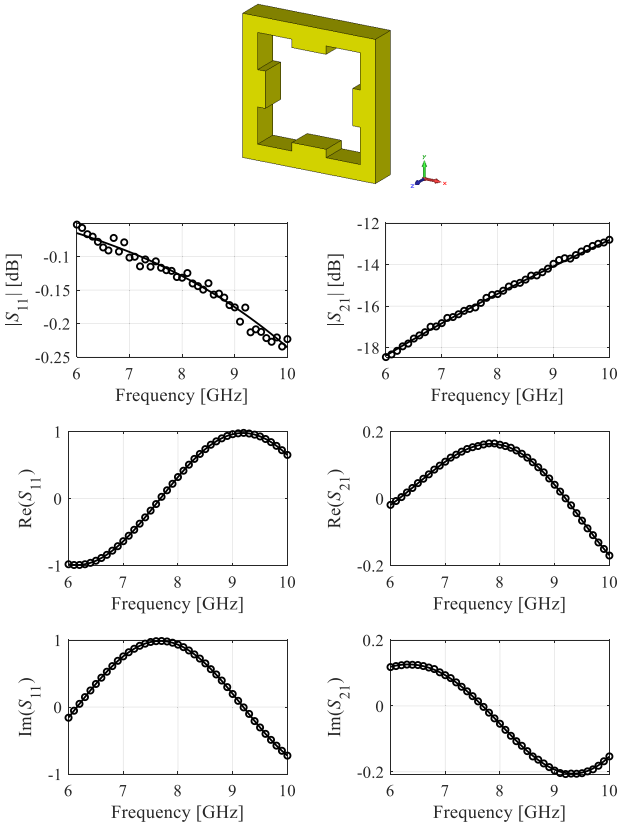

FIGURE 8. Visualization of the FCRM model performance for a selected FSS cell geometry (case 3): C = 0.79, H_s = 1.94, I_{ta} = 0.23, and M = 7.7. Surrogate constructed for $m = 3$ using Dataset 3; EM-simulation data (–), and FCRM surrogate (*o*).

For the sake of supplemental validation, visualization of the FCRM surrogate constructed with $m = 3$ versus EM-simulated FSS characteristics has been provided for the three selected designs, shown in Figs. 6 through 8, respectively. The obtained optimal hyper parameters using BO are as follows: SB = Fully Connect + Leak ReLU, $\Theta = \{2, 2, 1\}$, $\Phi = \{768, 512, 512\}$) The characteristics are obtained for random test designs (different from the training locations).

In Fig. 6, the FSS model with design parameters $C = 0.82$, $H_s = 1.4$, $I_{ta} = 0.75$, and $M = 12.5$, has been presented. The figure shows the Real and Imaginary parts of both S_{11} and S_{21}. Additionally, the moduli of S_{11} and S_{21} in dB are also presented to provide a better indication of the predictive power of the proposed surrogate. Two other design cases are presented in Figs. 7 and 8, respectively, corresponding to the following parameter sets: $C = 0.66$, $H_s = 4.43$, $I_{ta} = 0.85$, and $M = 12.4$ (Fig. 7), and $C = 0.79$, $H_s = 1.94$, $I_{ta} = 0.23$, and $M = 7.7$ (Fig. 8). Overall, a visual agreement between EM simulated responses and the surrogate-predicted data is excellent. The apparently visible differences in dB charateristcs of for the third case study originate from small absolute values the S-parameters; the error values are practically insignificant.

IV. CONCLUSION

The paper proposed a novel approach to surrogate modelling of high-frequency structures, referred to as a fully-connected regression model, or FCRM. The proposed methodology employes Bayesian Optimization to determine the architecture and the hyperparameters of the underlying deep neural network surrogate. Both are made dependent on available training data structure in pursuit of avoiding the over-fitting issues. The model setup is determined automatically, which eliminates the need for engaging engineering insight as well as trial-and-error approaches, both being a commonplace when setting up neural-networ-type of metamodels.

The FCRM framework has been comprehensively validated using a challenging problem of fractal-based frequency selective surface. The results obtained for several cardinalities of the training data sets and nine benchmark methods demonstrate the reliability of the presented modelling methodology. In particular, automatic determination of the model architecture allows—to a great extent—for mitigating the over-fitting problem, which directly translates into excellent predictive power of the model even for small training data sets. At the same time, the sensitivity of the model performance to training data allocation has been reduced.

Given the large scale of comparatory experiments, these advantages of FCRM have been demonstrated conclusively. The proposed approach can be considered a viable alternative to existing surrogate modelling approaches, especially when handing difficult cases, characterized by sparse allocation of the training samples and highly-nonlinear system outputs.

It should also be mentioned that because the FCRM surrogate is purely data-driven, it is possible to extend the modeled FSS output to represent its S-parameters at various oblique incidence angles. To this end the EM-simulated system output has to be generalized from vectors (S-parameters versus frequency) to tensors (two-dimensional tables containing S-parameters evaluated at various frequencies and incidence angles). Other than that, the model can be applied in a straightforward manner

ACKNOWLEDGMENT

The authors would like to thank Dassault Systemes, France, for making CST Microwave Studio available and Signal Processing for Computational Intelligence Group in Informatics Institute of Istanbul Technical University for providing computational resources.

REFERENCES

[1] T. Q. K. Nguyen, L. Lizzi, and F. Ferrero, "Dual-matching for single resonance miniaturized antenna for IoT applications," in *Proc. IEEE Int. Symp. Antennas Propag. USNC/URSI Nat. Radio Sci. Meeting*, Jul. 2018, pp. 793–794, doi: 10.1109/apusncursinrsm.2018.8609224.

[2] W. Hong, S. Lim, S. Ko, and Y. G. Kim, "Optically invisible antenna integrated within an OLED touch display panel for IoT applications," *IEEE Trans. Antennas Propag.*, vol. 65, no. 7, pp. 3750–3755, Jul. 2017, doi: 10.1109/tap.2017.2705127.

[3] S. M. H. Varkiani and M. Afsahi, "Compact and ultra-wideband CPW-fed square slot antenna for wearable applications," *AEU-Int. J. Electron. Commun.*, vol. 106, pp. 108–115, Jul. 2019, doi: 10.1016/j.aeue.2019.04.024.

[4] H. Wong, W. Lin, L. Huitema, and E. Arnaud, "Multi-polarization reconfigurable antenna for wireless biomedical system," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 3, pp. 652–660, Jun. 2017, doi: 10.1109/tbcas.2016.2636872.

[5] C. Goswami, R. Ghatak, and D. R. Poddar, "Multi-band bisected Hilbert monopole antenna loaded with multiple subwavelength split-ring resonators," *IET Microw., Antennas Propag.*, vol. 12, no. 10, pp. 1719–1727, May 2018, doi: 10.1049/iet-map.2017.1215.

[6] J. Singh, R. Stephan, and M. A. Hein, "Low-profile penta-band automotive patch antenna using horizontal stacking and corner feeding," *IEEE Access*, vol. 7, pp. 74198–74205, 2019, doi: 10.1109/ACCESS.2019.2919730.

[7] P. Mahouti, "Design optimization of a pattern reconfigurable microstrip antenna using differential evolution and 3D EM simulation-based neural network model," *Int. J. RF Microw Comput. Aided Eng.*, vol. 29, no. 8, p. e21796, Apr. 2019, doi: 10.1002/mmce.21796.

[8] G. Wolosinski, V. Fusco, U. Naeem, and P. Rulikowski, "Pre-matched eigenmode antenna with polarization and pattern diversity," *IEEE Trans. Antennas Propag.*, vol. 67, no. 8, pp. 5145–5153, Aug. 2019, doi: 10.1109/TAP.2019.2912464.

[9] G. Kumar and R. Kumar, "A survey on planar ultra-wideband antennas with band notch characteristics: Principle, design, and applications," *AEU-Int. J. Electron. Commun.*, vol. 109, pp. 76–98, Sep. 2019, doi: 10.1016/j.aeue.2019.07.004.

[10] J. Ossorio, J. Vague, V. E. Boria, and M. Guglielmi, "Exploring the tuning range of channel filters for satellite applications using electromagnetic-based computer aided design tools," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 2, pp. 717–725, Feb. 2018, doi: 10.1109/TMTT.2017.2769083.

[11] P. Chen, B. M. Merrick, and T. J. Brazil, "Bayesian optimization for broadband high-efficiency power amplifier designs," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 12, pp. 4263–4272, Dec. 2015, doi: 10.1109/TMTT.2015.2495360.

[12] J. Zhang, F. Feng, W. Na, S. Yan, and Q. Zhang, "Parallel space-mapping based yield-driven EM optimization incorporating trust region algorithm and polynomial chaos expansion," *IEEE Access*, vol. 7, pp. 143673–143683, 2019, doi: 10.1109/ACCESS.2019.2944415.

[13] M. U. Memon, A. Salim, H. Jeong, and S. Lim, "Metamaterial inspired radio frequency-based touchpad sensor system," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1344–1352, Apr. 2020, doi: 10.1109/TIM.2019.2908507.

[14] S. Koziel and A. Pietrenko-Dabrowska, "Reduced-cost electromagnetic-driven optimisation of antenna structures by means oftrust-region gradient-search with sparse jacobian updates," *IET Microw., Antennas Propag.*, vol. 13, no. 10, pp. 1646–1652, May 2019, doi: 10.1049/iet-map.2018.5879.

[15] A. Bautista, A.-L. Franc, and P. Ferrari, "Accurate parametric electrical model for slow-wave CPW and application to circuits design," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 12, pp. 4225–4235, Dec. 2015, doi: 10.1109/TMTT.2015.2495242.

[16] A. Ciccazzo, G. Di Pillo, and V. Latorre, "A SVM surrogate model-based method for parametric yield optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 7, pp. 1224–1228, Jul. 2016, doi: 10.1109/TCAD.2015.2501307.

[17] S. Koziel and J. W. Bandler, "Rapid yield estimation and optimization of microwave structures exploiting feature-based statistical analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 1, pp. 107–114, Jan. 2015.

[18] A. S. A. El-Hameed, M. G. Wahab, and A. Elboushi, "Elpeltagy MS. miniaturized triple band-notched quasi self-complementary fractal antenna with improved characteristics for UWB applications," *AEU-Int. J. Electr. Commun.*, vol. 108, pp. 163–171, Aug. 2019, doi: 10.1016/j.aeue.2019.06.016.

[19] J. Wang, X. Yang, and B. Wang, "Efficient gradient-based optimisation of pixel antenna with large-scale connections," *IET Microw., Antennas Propag.*, vol. 12, no. 3, pp. 385–389, Feb. 2018, doi: 10.1049/iet-map.2017.0719.

[20] A. Pietrenko-Dabrowska and S. Koziel, "Computationally-efficient design optimisation of antennas by accelerated gradient search with sensitivity and design change monitoring," *IET Microw., Antennas Propag.*, vol. 14, no. 2, pp. 165–170, Feb. 2020, doi: 10.1049/iet-map.2019.0358.

[21] S. Koziel and A. Pietrenko-Dabrowska, "Reduced-cost design closure of antennas by means of gradient search with restricted sensitivity update," *Metrol. Meas. Syst.*, vol. 26, no. 4, pp. 595–605, 2019, doi: 410.24425/mms.2019.130561.

[22] A. Darvish and A. Ebrahimzadeh, "Improved fruit-fly optimization algorithm and its applications in antenna arrays synthesis," *IEEE Trans. Antennas Propag.*, vol. 66, no. 4, pp. 1756–1766, Apr. 2018, doi: 10.1109/TAP.2018.2800695.

[23] J. Joung, "Machine learning-based antenna selection in wireless communications," *IEEE Commun. Lett.*, vol. 20, no. 11, pp. 2241–2244, Nov. 2016.

[24] S. Mishra, R. N. Yadav, and R. P. Singh, "Directivity estimations for short dipole antenna arrays using radial basis function neural networks," *IEEE Antennas Wireless Propag. Lett.*, vol. 14, pp. 1219–1222, 2015, doi: 10.1109/LAWP.2015.2399453.

[25] I. A. Baratta, C. B. de Andrade, R. R. de Assis, and E. J. Silva, "Infinitesimal dipole model using space mapping optimization for antenna placement," *IEEE Antennas Wireless Propag. Lett.*, vol. 17, no. 1, pp. 17–20, Jan. 2018, doi: 10.1109/LAWP.2017.2771721.

[26] N. Engheta and R. W. Ziolkowski, *Metamaterials: Physics and Engineering Explorations*. Hoboken, NJ, USA: Wiley, 2006.

[27] T. Tanaka, A. Ishikawa, and S. Kawata, "Unattenuated light transmission through the interface between two materials with different indices of refraction using magnetic metamaterials," *Phys. Rev. B, Condens. Matter*, vol. 73, no. 12, Mar. 2006, Art. no. 125423, doi: 10.1103/physrevb.73.125423.

[28] W. Cai and V. Shalaev, *Optical Metamaterials*. New York, NY, USA: Springer, 2010.

[29] J. Li and C. T. Chan, "Double-negative acoustic metamaterial," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 5, Nov. 2004, Art. no. 055602, doi: 10.1103/physreve.70.055602.

[30] N. Fang, "Sub-diffraction-limited optical imaging with a silver superlens," *Science*, vol. 308, no. 5721, pp. 534–537, Apr. 2005, doi: 10.1126/science.1108759.

[31] Z. Yang, J. Mei, M. Yang, N. H. Chan, and P. Sheng, "Membrane-type acoustic metamaterial with negative dynamic mass," *Phys. Rev. Lett.*, vol. 101, no. 20, Nov. 2008, Art. no. 204301, doi: 10.1103/PhysRevLett.101.204301.

[32] J. Mei, G. Ma, M. Yang, Z. Yang, W. Wen, and P. Sheng, "Dark acoustic metamaterials as super absorbers for low-frequency sound," *Nature Commun.*, vol. 3, no. 1, pp. 1–7, Jan. 2012, doi: 10.1038/ncomms1758.

[33] Y. Chen, G. Huang, X. Zhou, G. Hu, and C.-T. Sun, "Analytical coupled vibroacoustic modeling of membrane-type acoustic metamaterials: Membrane model," *J. Acoust. Soc. Amer.*, vol. 136, no. 3, pp. 969–979, Sep. 2014, doi: 10.1121/1.4892870.

[34] Y. Chen, G. Huang, X. Zhou, G. Hu, and C.-T. Sun, "Analytical coupled vibroacoustic modeling of membrane-type acoustic metamaterials: Plate model," *J. Acoust. Soc. Amer.*, vol. 136, no. 6, pp. 2926–2934, Dec. 2014.

[35] J. B. Pendry, "Negative refraction makes a perfect lens," *Phys. Rev. Lett.*, vol. 85, no. 18, p. 3966, 2000.

[36] N. Yu, P. Genevet, M. A. Kats, F. Aieta, J.-P. Tetienne, F. Capasso, and Z. Gaburro, "Light propagation with phase discontinuities: Generalized laws of reflection and refraction," *Science*, vol. 334, no. 6054, pp. 333–337, Oct. 2011, doi: 10.1126/science.1210713.

[37] R. A. Shelby, "Experimental verification of a negative index of refraction," *Science*, vol. 292, no. 5514, pp. 77–79, Apr. 2001, doi: 10.1126/science.1058847.

[38] C. Pfeiffer and A. Grbic, "Metamaterial Huygens' surfaces: Tailoring wave fronts with reflectionless sheets," *Phys. Rev. Lett.*, vol. 110, no. 19, May 2013, Art. no. 197401, doi: 10.1103/PhysRevLett.110.197401.

[39] O. Tsilipakos, T. Koschny, and C. M. Soukoulis, "Antimatched electromagnetic metasurfaces for broadband arbitrary phase manipulation in reflection," *ACS Photon.*, vol. 5, no. 3, pp. 1101–1107, Jan. 2018, doi: 10.1021/acsphotonics.7b01415.

[40] B. A. Munk, *Frequency Selective Surfaces: Theory and Design*. Hoboken, NJ, USA: Wiley, 2005.

[41] J. C. Vardaxoglou, *Frequency Selective Surfaces: Analysis and Design*. Research Studies Press, 1997.

[42] J. H. Kim, H. J. Chun, I. P. Hong, Y. J. Kim, and Y. B. Park, "Analysis of FSS radomes based on physical optics method and ray tracing technique," *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 868–871, 2014, doi: 10.1109/LAWP.2014.2320978.

[43] F. Costa, A. Kazemzadeh, S. Genovesi, and A. Monorchio, "Electromagnetic absorbers based on frequency selective surfaces," *Forum Electromagn. Res. Methods Appl. Technol.*, vol. 37, no. 1, pp. 1–23, 2016.

[44] C.-N. Chiu, C.-H. Kuo, and M.-S. Lin, "Bandpass shielding enclosure design using multipole-slot arrays for modern portable digital devices," *IEEE Trans. Electromagn. Compat.*, vol. 50, no. 4, pp. 895–904, Nov. 2008, doi: 10.1109/TEMC.2008.2004560.
[45] R. Sivasamy, B. Moorthy, M. Kanagasabai, V. R. Samsingh, and M. G. N. Alsath, "A wideband frequency tunable FSS for electromagnetic shielding applications," *IEEE Trans. Electromagn. Compat.*, vol. 60, no. 1, pp. 280–283, Feb. 2018, doi: 10.1109/TEMC.2017.2702572.
[46] D. Li, T.-W. Li, E.-P. Li, and Y.-J. Zhang, "A 2.5-D angularly stable frequency selective surface using via-based structure for 5G EMI shielding," *IEEE Trans. Electromagn. Compat.*, vol. 60, no. 3, pp. 768–775, Jun. 2018, doi: 10.1109/TEMC.2017.2748566.
[47] C.-W. Lin, C.-K. Shen, and T.-L. Wu, "Ultracompact via-based absorptive frequency-selective surface for 5-GHz Wi-Fi with passbands and high-performance stability," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 8, no. 1, pp. 41–49, Jan. 2018, doi: 10.1109/TCPMT.2017.2759221.
[48] P. Mahouti, F. Güneş, M. A. Belen, A. Çaışkan, S. Demirel, and Z. Sharipov, "Horn antennas with enhanced functionalities through the use of frequency selective surfaces," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 26, no. 4, pp. 287–293, Feb. 2016, doi: 10.1002/mmce.20971.
[49] F. Güneş, Z. Sharipov, M. A. Belen, and P. Mahouti, "GSM filtering of horn antennas using modified double square frequency selective surface," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 27, no. 9, Nov. 2017, Art. no. e21136, doi: 10.1002/mmce.21136.
[50] M. A. Belen, F. Güneş, P. Mahouti, and A. Belen, "UWB gain enhancement of horn antennas using miniaturized frequency selective surface," *ACES J.*, vol. 33, no. 9, pp. 997–1002, 2018.
[51] F. Güneş, M. A. Belen, and P. Mahouti, "Performance enhancement of a microstrip patch antenna using substrate integrated waveguide frequency selective surface for ISM band applications," *Microw. Opt. Technol. Lett.*, vol. 60, no. 5, pp. 1160–1164, May 2018, doi: 10.1002/mop.31124.
[52] W. Zhang, S. Yan, F. Feng, and Q. Zhang, "Fast and simple technique for computing circuit noise figure from component noise model using artificial neural network," in *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphys. Model. Optim. (NEMO)*, Ottawa, ON, Canada, Aug. 2015, pp. 1–3, doi: 10.1109/NEMO.2015.7415031.
[53] J. L. Chávez-Hurtado and J. E. Rayas-Sánchez, "Polynomial-based surrogate modeling of RF and microwave circuits in frequency domain exploiting the multinomial theorem," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 12, pp. 4371–4381, Dec. 2016, doi: 10.1109/TMTT.2016.2623902.
[54] J. Jin, C. Zhang, F. Feng, W. Na, J. Ma, and Q.-J. Zhang, "Deep neural network technique for high-dimensional microwave modeling and applications to parameter extraction of microwave filters," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 10, pp. 4140–4155, Oct. 2019, doi: 10.1109/TMTT.2019.2932738.
[55] F. Güneş, P. Mahouti, S. Demirel, M. A. Belen, and A. Uluslu, "Cost-effective GRNN-based modeling of microwave transistors with a reduced number of measurements," *Int. J. Numer. Model. Electron. Netw., Devices Fields*, vol. 30, nos. 3–4, p. e2089, May 2017.
[56] Z. D. Marinković and V. V. Marković, "Temperature-dependent models of low-noise microwave transistors based on neural networks," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 15, no. 6, pp. 567–577, Nov. 2005, doi: 10.1002/mmce.20102.
[57] F. Giannini, G. Leuzzi, G. Orengo, and M. Albertini, "Small-signal and large-signal modeling of active devices using CAD-optimized neural networks," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 12, no. 1, pp. 71–78, Dec. 2001, doi: 10.1002/mmce.10007.
[58] P. Mahouti, "Application of artificial intelligence algorithms on modeling of reflection phase characteristics of a nonuniform reflectarray element," *Int. J. Numer. Modelling: Electron. Netw., Devices Fields*, vol. 33, no. 2, Oct. 2019, doi: 10.1002/jnm.2689.
[59] J. Dong, W. Qin, and J. Mo, "Low-cost multi-objective optimization of multiparameter antenna structures based on the l1 optimization BPNN surrogate model," *Electronics*, vol. 8, no. 8, p. 839, Jul. 2019, doi: 10.3390/electronics8080839.
[60] F. Gáneş, S. Nesil, and S. Demirel, "Design and analysis of minkowski reflectarray antenna using 3-D cst microwave studio-based neural network model with particle swarm optimization," *Int. J. RF Microw. Comput. Aided Eng.*, vol. 23, no. 2, pp. 272–284, 2013.
[61] F. Gáneş, S. Demirel, and S. Nesil, "A novel design approach to X-band minkowski reflectarray antennas using the full-wave em simulation-based complete neural model with a hybridga-nm algorithm," *Radioengineering*, vol. 23, no. 1, pp. 144–153, 2014.
[62] P. Mahouti, F. Gáneş, M. A. Belen, and A. Çalışkan, "A novel design of non-uniform reflectarrays with symbolic regression and its realization using 3-D printer," *ACES J.*, vol. 34, no. 2, pp. 280–285, 2019.
[63] P. Mahouti, F. Gáneş, M. A. Belen, and S. Demirel, "Symbolic regression for derivation of an accurate analytical formulation using big data an application example," *ACES J.*, vol. 32, no. 5, pp. 372–380, 2017.
[64] D. Karaboga, K. Guüney, S. Sagiroglu, 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, p. 155, 1999, doi: 10.1049/ip-map:19990136.
[65] R. K. Mishra and A. Patnaik, "Neural network-based CAD model for the design of square-patch antennas," *IEEE Trans. Antennas Propag.*, vol. 46, no. 12, pp. 1890–1891, Dec. 1998.
[66] S. Koziel, L. Leifsson, I. Couckuyt, and T. Dhaene, "Robust variable-fidelity optimization of microwave filters using co-kriging and trust regions," *Microw. Opt. Technol. Lett.*, vol. 55, no. 4, pp. 765–769, Apr. 2013, doi: 10.1002/mop.27447.
[67] N. Leszczynska, I. Couckuyt, T. Dhaene, and M. Mrozowski, "Low-cost surrogate models for microwave filters," *IEEE Microw. Wireless Compon. Lett.*, vol. 26, no. 12, pp. 969–971, Dec. 2016, doi: 10.1109/LMWC.2016.2623248.
[68] G. Angiulli, M. Cacciola, and M. Versaci, "Microwave devices and antennas modelling by support vector regression machines," *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1589–1592, Apr. 2007, doi: 10.1109/TMAG.2007.892480.
[69] S. Koziel and J. P. Jacobs, "Gaussian process antenna modeling using neighborhood-data-expanded training sets," in *Proc. IEEE Antennas Propag. Soc. Int. Symp. (APSURSI)*, Orlando, FL, USA, Jul. 2013, pp. 1282–1283, doi: 10.1109/APS.2013.6711301.
[70] J. P. Jacobs and S. Koziel, "Reduced-cost microwave filter modeling using a two-stage Gaussian process regression approach," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 25, no. 5, pp. 453–462, Dec. 2014, doi: 10.1002/mmce.20880.
[71] F. Feng, C. Zhang, J. Ma, and Q.-J. Zhang, "Parametric modeling of EM behavior of microwave components using combined neural networks and pole-residue-based transfer functions," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 1, pp. 60–77, Jan. 2016, doi: 10.1109/TMTT.2015.2504099.
[72] H. Kabir, Y. Wang, M. Yu, and Q.-J. Zhang, "Neural network inverse modeling and applications to microwave filter design," *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 4, pp. 867–879, Apr. 2008, doi: 10.1109/TMTT.2008.919078.
[73] J. Tak, A. Kantemur, Y. Sharma, and H. Xin, "A 3-D-printed W-band slotted waveguide array antenna optimized using machine learning," *IEEE Antennas Wireless Propag. Lett.*, vol. 17, no. 11, pp. 2008–2012, Nov. 2018, doi: 10.1109/LAWP.2018.2857807.
[74] B. Liu, H. Aliakbarian, Z. Ma, G. A. E. Vandenbosch, G. Gielen, and P. Excell, "An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques," *IEEE Trans. Antennas Propag.*, vol. 62, no. 1, pp. 7–18, Jan. 2014, doi: 10.1109/TAP.2013.2283605.
[75] J. Jin, F. Feng, J. Zhang, S. Yan, W. Na, and Q. Zhang, "A novel deep neural network topology for parametric modeling of passive microwave components," *IEEE Access*, vol. 8, pp. 82273–82285, 2020, doi: 10.1109/ACCESS.2020.2991890.
[76] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Database Systems for Advanced Applications*. Springer, 2016, pp. 214–228.
[77] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Computer Vision—ECCV*. Springer, 2016, pp. 749–765.
[78] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0180944, doi: 10.1371/journal.pone.0180944.
[79] A. S. Qureshi, A. Khan, A. Zameer, and A. Usman, "Wind power prediction using deep neural network based meta regression and transfer learning," *Appl. Soft Comput.*, vol. 58, pp. 742–755, Sep. 2017, doi: 10.1016/j.asoc.2017.05.031.

[80] Y. Li, H. Shi, F. Han, Z. Duan, and H. Liu, "Smart wind speed forecasting approach using various boosting algorithms, big multi-step forecasting strategy," *Renew. Energy*, vol. 135, pp. 540–553, May 2019, doi: 10.1016/j.renene.2018.12.035.

[81] Z. Wei, D. Liu, and X. Chen, "Dominant-current deep learning scheme for electrical impedance tomography," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 9, pp. 2546–2555, Sep. 2019, doi: 10.1109/tbme.2019.2891676.

[82] M. N. Asad, İ. Cantürk, F. Z. Genç, and L. Özy-lmaz, "Investigation of bone age assessment with convolutional neural network by using DoG filtering and à trous wavelet as preprocessing techniques," in *Proc. 6th Int. Conf. Control Eng. Inf. Technol. (CEIT)*, Oct. 2018, pp. 1–7, doi: 10.1109/CEIT.2018.8751885.

[83] Z. Wei and X. Chen, "Deep-learning schemes for full-wave nonlinear inverse scattering problems," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 1849–1860, Apr. 2019, doi: 10.1109/TGRS.2018.2869221.

[84] N. Calik, M. A. Belen, and P. Mahouti, "Deep learning base modified MLP model for precise scattering parameter prediction of capacitive feed antenna," *Int. J. Numer. Model. Electron. Netw., Devices Fields*, vol. 33, no. 2, p. e2682, Sep. 2019, doi: 10.1002/jnm.2682.

[85] P. Neary, "Automatic hyperparameter tuning in deep convolutional neural networks using asynchronous reinforcement learning," in *Proc. IEEE Int. Conf. Cognit. Comput. (ICCC)*, San Francisco, CA, USA, Jul. 2018, pp. 73–77, doi: 10.1109/ICCC.2018.00017.

[86] C. P. Schwegmann, W. Kleynhans, B. P. Salmon, L. W. Mdakane, and R. G. V. Meyer, "Very deep learning for ship discrimination in synthetic aperture radar imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 104–107.

[87] X. Y. Chen, X. Y. Peng, Y. Peng, and J.-B. Li, "The classification of synthetic aperture radar image target based on deep learning," *J. Inf. Hiding Multimedia Signal Process.*, vol. 7, no. 6, pp. 1345–1353, 2016.

[88] L. Pu, X. Zhang, S. Wei, X. Fan, and Z. Xiong, "Target recognition of 3-D synthetic aperture radar images via deep belief network," in *Proc. CIE Int. Conf. Radar (RADAR)*, Guangzhou, Oct. 2016, pp. 1–5, doi: 10.1109/RADAR.2016.8059199.

[89] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 281–305, Feb. 2012.

[90] P. I. Frazier, "A tutorial on Bayesian optimization," 2018, *arXiv:1807.02811*. [Online]. Available: http://arxiv.org/abs/1807.02811

[91] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," 2010, *arXiv:1012.2599*. [Online]. Available: https://arxiv.org/abs/1012.2599

[92] J. Jin, F. Feng, W. Na, S. Yan, W. Liu, L. Zhu, and Q. Zhang, "Recent advances in neural network-based inverse modeling techniques for microwave applications," *Int. J. Numer. Model. Electron. Netw., Devices Fields*, vol. 33, no. 6, p. e2732, Feb. 2020, doi: 10.1002/jnm.2732.

[93] M. Minsky and S. Papert, *An Introduction to Computational Geometry*. Cambridge, MA, USA: Cambridge Tiass, 1969.

[94] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process Syst.*, 2007, pp. 153–160.

[95] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*. [Online]. Available: http://arxiv.org/abs/1505.00853

[96] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian optimization algorithm," in *Proc. Genetic Evol. Comput. Conf. (GECCO)*, vol. 1, 1999, pp. 525–532.

[97] C. Gershenson, "Artificial neural networks for beginners," 2003, *arXiv:cs/0308031*. [Online]. Available: https://arxiv.org/abs/cs/0308031

[98] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 807–814.

[99] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[100] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[101] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," 2010, *arXiv:1012.2599*. [Online]. Available: http://arxiv.org/abs/1012.2599

[102] M. Injadat, F. Salo, A. B. Nassif, A. Essex, and A. Shami, "Bayesian optimization with machine learning algorithms towards anomaly detection," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOM.2018.8647714.

[103] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," 2020, *arXiv:2007.15745*. [Online]. Available: http://arxiv.org/abs/2007.15745

[104] A. O'Hagan, "On curve fitting and optimal design for regression," *J. Royal Stat. Soc. B*, vol. 40, no. 1, pp. 1–42, 1978.

[105] A. Zilinskas, *On the Use of Statistical Models of Multimodal Functions for the Construction of the Optimization Algorithms* (Lecture Notes in Control and Information Sciences), vol. 23. Berlin, Germany: Springer-Verlag, 1980.

[106] J. Mockus, "Application of Bayesian approach to numerical methods of global and stochastic optimization," *J. Global Optim.*, vol. 4, no. 4, pp. 347–365, Jun. 1994.

[107] J. Snoek, H. Larochelle, and R. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Red Hook, NY, USA: Curran Associates, pp. 2951–2959.

[108] N. DeCastro-García, Á. L. M. Castañeda, D. E. García, and M. V. Carriegos, "Effect of the sampling of a dataset in the hyperparameter optimization phase over the efficiency of a machine learning algorithm," *Complexity*, vol. 2019, pp. 1–16, Feb. 2019, doi: 10.1155/2019/6278908.

[109] C. E. Rasmussen, "Gaussian processes for machine learning," in *Summer School on Machine Learning*. Berlin, Germany: Springer, 2003, pp. 63–71.

[110] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast Bayesian hyperparameter optimization on large datasets," *Electron. J. Statist.*, vol. 11, no. 2, pp. 4945–4968, 2017.

[111] B. Matérn, "Spatial variation: Stochastic models and their application to some problems in forest surveys and other sampling investigations," *Eur. J. Res. Band*, vol. 49, no. 5, p. 144, 1960.

[112] B. Minasny and A. B. McBratney, "The Matérn function as a general model for soil variograms," *Geoderma*, vol. 128, nos. 3–4, pp. 192–207, 2005.

[113] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of Bayesian methods for seeking the extremum," *Towards Global Optim.*, vol. 2, pp. 117–129, Dec. 1978.

[114] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," 2009, *arXiv:0912.3995*. [Online]. Available: http://arxiv.org/abs/0912.3995

[115] P. Hennig and C. J. Schuler, "Entropy search for information-efficient global optimization," *J. Mach. Learn. Res.*, vol. 2, pp. 1809–1837, Dec. 2012.

[116] H. Lobato, J. Miguel, M. W. Hoffman, and Z. Ghahramani, "Predictive entropy search for efficient global optimization of black-box functions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 918–926.

[117] J. P. Gianvittorio and Y. Rahmat-Samii, "Fractal antennas: A novel antenna miniaturization technique, and applications," *IEEE Antennas Propag. Mag.*, vol. 44, no. 1, pp. 20–36, Feb. 2002, doi: 10.1109/74.997888.

[118] B. Beachkofski and R. Grandhi, "Improved distributed hypercube sampling," in *Proc. 43rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn., Mater. Conf.*, Apr. 2002, p. 1274, doi: 10.2514/6.2002-1274.

[119] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, p. 239, May 1979, doi: 10.2307/1268522.

[120] S. Leary, A. Bhaskar, and A. Keane, "Optimal orthogonal-array-based Latin hypercubes," *J. Appl. Statist.*, vol. 30, no. 5, pp. 585–598, Jun. 2003.

[121] T. J. Santner, B. J. Williams, and W. I. Notz, "Space-filling designs for computer experiments," in *The Design and Analysis of Computer Experiments* (Springer Series in Statistics). New York, NY, USA: Springer, 2003, pp. 121–161.

[122] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Adv. Neural Inf. Process Syst.*, 2017, pp. 6389–6399.

[123] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/jcss.1997.1504.

[124] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers Neurorobot.*, vol. 7, pp. 1–21, Dec. 2013, Art. no. 21, doi: 10.3389/fnbot.2013.00021.
[125] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, 2001, pp. 1189–1232.
[126] V. Vapnik, "Pattern recognition using generalized portrait method," *Automat. Remote Control*, vol. 24, no. 6, pp. 774–780, Jan. 1963.
[127] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004, doi: 10.1023/b:stco.0000035301.49549.88.
[128] C. K. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Adv. Neural Inf. Process Syst.*, 2001, pp. 682–688.
[129] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[130] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss-Seifel method under convex constraints," *Oper. Res. Lett.*, vol. 26, pp. 127–136, Apr. 2000.
[131] L. Bo and C. Sminchisescu, "Greed block coordinate descent for large scale Gaussian process regression," in *Proc. 24th Conf. Uncertainty Artif. Intell. (UAI)*, 2012. [Online]. Available: http://arxiv.org/abs/1206.3238
[132] P. D. Wasserman, *Advanced Methods in Neural Computing*. New York, NY, USA: Van Nostrand, 1993, pp. 61–155.
[133] J. Park and I. W. Sandberg, "Approximation and radial-basis-function networks," *Neural Comput.*, vol. 5, no. 2, pp. 305–316, Mar. 1993, doi: 10.1162/neco.1993.5.2.305.
[134] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Netw.*, vol. 2, no. 6, pp. 568–576, Nov. 1991, doi: 10.1109/72.97934.

NURULLAH CALIK received the M.Sc. and Ph.D. degrees in electronics and communication engineering from Yıldız Technical University, Turkey, in 2013 and 2019, respectively. He was worked as a Postdoctoral Researcher with the Informatics Institute, Istanbul Technical University. He is currently an Assistant Professor with the Department of Biomedical Engineering, Istanbul Medeniyet University, Turkey. His main research interests include large-scale data analysis, signal and image processing, AI applications in engineering, deep learning regression, optimization, and surrogate modeling.

MEHMET ALI BELEN received the Ph.D. degree in electronics and communication engineering from Yıldız Technical University, in 2016. He is currently an Associate Professor with İskenderun Technical University. His current activities include teaching and researching electromagnetics and microwaves along with developing additive manufacturing 3D printed microwave components for rapid prototyping. His current research interests include multivariable network theory, device modeling, computer aided microwave circuit design, monolithic microwave integrated circuits, and antenna arrays, active/passive microwave components especially in the field of metamaterial-based antennas and microwave filters.

PEYMAN MAHOUTI received the M.Sc. and Ph.D. degrees in electronics and communication engineering from Yıldız Technical University, Turkey, in 2013 and 2016, respectively. He is currently an Associate Professor with the Department of Electronic and Communication, Istanbul University - Cerrahpasa, Turkey. His main research interests include analytical and numerical modeling of microwave devices, optimization techniques for microwave stages, application of artificial intelligence-based algorithms, analytical and numerical modeling of microwave and antenna structures, surrogate-based optimization, and application of artificial intelligence algorithms.

SLAWOMIR KOZIEL (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electronic engineering from the Gdansk University of Technology, Poland, in 1995 and 2000, respectively, the M.Sc. degrees in theoretical physics and in mathematics, in 2000 and 2002, respectively, and the Ph.D. degree in mathematics from the University of Gdansk, Poland, in 2003. He is currently a Professor with the Department of Engineering, Reykjavik University, Iceland. His research interests include CAD and modeling of microwave and antenna structures, simulation-driven design, surrogate-based optimization, space mapping, circuit theory, analog signal processing, evolutionary computation, and numerical analysis.