IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 15, 2021, accepted February 25, 2021, date of publication March 1, 2021, date of current version March 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3062940

Resonant Frequency Modeling of Microstrip Antenna Based on Deep Kernel Learning

SHUDAN HAN[1], YUBO TIAN[1,2], WEITONG DING[1], AND PENGFEI LI[1]

[1]School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China
[2]School of Information and Communication Engineering, Guangzhou Maritime University, Guangzhou 510725, China

Corresponding author: Yubo Tian (tianyubo@just.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61771225, and in part by the Qinglan Project of Jiangsu Higher Education.

ABSTRACT When modeling and optimizing electromagnetic components, it is the most time consuming for obtaining the training samples with labels from full-wave electromagnetic simulation software. The traditional machine learning (ML) model is usually effective in the training process, unfortunately, its generalization ability is limited for practical application. Inspired by the artificial neural network (ANN) and the Gaussian process (GP) kernel learning model, a deep kernel learning (DKL) model with multiple-nonlinear-mapping layers is proposed based on particle swarm optimization (PSO) algorithm. In this work, correlation characteristics of training samples are transformed by multiple layers, and then they act as inputs of the GP model. Simultaneously, we use PSO to optimize the weights, biases of each mapping layer, and GP hyperparameters, aiming to determine and optimize the DKL model structure. As a result, the proposed DKL is suitable for processing small samples, as well as has the generalization ability of a deep network, which avoids the phenomenon of gradient disappearance to some extent caused by error back propagation (BP) in the deep network and improves the modeling accuracy while effectively ensuring the modeling efficiency. In this study, the performance of the DKL model is evaluated by using the resonant frequencies of two microstrip antennas (MSAs), and the predicted results of the DKL model are compared with those via different modeling methods. The results show that prediction accuracy of the DKL is 18.906% higher than that of the GP, 38.926% higher than the ANN, and 46.660% higher than the neural network ensemble (NNE).

INDEX TERMS Gaussian process (GP), particle swarm optimization (PSO), resonance frequency.

I. INTRODUCTION

In recent years, optimization design of electromagnetic components is generally by means of numerical simulation or full wave electromagnetic simulation software, e.g., high frequency structure simulator (HFSS), computer simulation technology (CST), combined with global optimization algorithm [1], [2], which have achieved some good research results. Generally, high-fidelity results can be obtained by simulating microwave components using full wave electromagnetic simulation software, while the process along with high computational cost and time-consuming, since requiring the invocation of electromagnetic simulation software to evaluate individuals of global optimization method many times. Therefore, using a surrogate model instead of electromagnetic simulation software to evaluate the fitness of electromagnetic components can save optimization time, which is a popular topic of electromagnetic optimization design at present. Many modeling methods have been proposed by researchers, such as artificial neural network (ANN) [3]–[5], support vector machine (SVM) [6], [7], extreme learning machine (ELM) [8]–[10], Gaussian process (GP) [11]–[13].

ANN is an intelligent science and information processing system based on imitating the structure and function of the human brain. It consists of a number of very simple neurons connected to each other in a certain way, relying on the dynamic response of the system itself to external input information processing information [14], but its internal network structure is difficult to be determined, and the deep network has the phenomenon of gradient disappearance. SVM is a machine learning (ML) method based on statistical learning theory. Its obvious feature is to improve the generalization ability as much as possible, depending on the Vapnik structural risk minimization principle, i.e., small error obtained from the limited training set samples can still ensure that it has small error for the independent test set [15].

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

However, its kernel function parameter is difficult to be determined and exposed. ELM is a novel feedforward ANN with one single-layer. Based on Moore-Penrose generalized inverse matrix theory, the output weight of the network can be solved analytically through one-step calculation. More and more researchers has been attracted by this simple and effective learning algorithm [9]. Unfortunately, ELM is not suitable for dealing with complex problems due to its simple network structure.

GP is a rapid-developing ML method in recent years. As a typical model of kernel learning (KL), it has a strict statistical theoretical basis, and is suitable for dealing with complex problems such as small samples, high dimensions and nonlinearity [16]–[18]. GP is gradually developed on the basis of continuous research on Bayesian neural network (NN), which has the advantages of flexible nonparametric inference, hyperparametric adaptive acquisition and probabilistic significance of output [19]. Actually, GP is a multi-layer perceptron with infinite units in its layer, which considers sophisticated covariance functions or embeds GP in more complex probabilistic structures. Therefore, it is able to learn more from the data to own the powerful representation ability. So far, however, most GP-based approaches don't lead to a principle way of obtaining truly deep architectures [20]. Deep neural network (DNN) is strong in processing big data, however, when dealing with scarce electromagnetic data samples, the trained DNN often exists "over-fitting" phenomenon, and also the process of training model takes a long time [21].

Among all of evolution algorithms, particle swarm optimization (PSO) algorithm is very effective. Due to its faster convergence speed and lesser algorithm parameters, more attention is being paid by lots of experts and researchers in recent years. The iterative process leads to a stochastic manipulation of velocities and flying directions according to the best experiences of the swarm to search for the global optimum in solution space [22].

In this paper, on the basis of the DNN structure and the GP model, different from the method proposed in the literature [23], a deep KL (DKL) with multiple-nonlinear-mapping layers is put forward, which is suitable for processing small samples, owning the generalization ability of the deep network and avoiding the phenomenon of gradient disappearance in the error back propagation (BP) of deep network. In the process of the proposed DKL model, relevant characteristics of samples are transformed via multiple-nonlinear-mapping layers, and then as inputs to the GP model for prediction. Meanwhile, in order to determine the DKL model structure, PSO algorithm is used to optimize the weights, biases of each layer and GP hyperparameters. The resonant frequencies of two microstrip antennas (MSAs) are used to evaluate the performance of the proposed DKL.

The rest of paper is organized as follows. In Section II, we describe the fundamental principle, model structure and algorithm flowchart of the PSO-based DKL model. Then, resonance frequency modeling results of rectangular MSA (RMSA) and annular ring compact MSA (ARCMSA) employing different modeling methods are compared, which illustrate that the proposed DKL model is more effective and has certain advantages in Section III. Finally, Section IV concludes this study. Appendix is the relevant information supplement to the proposed approach.

II. DEEP KERNEL LEARNING (DKL) MODEL

In this section, we will introduce the proposed DKL model with multiple-nonlinear-mapping layers. This section includes two parts. The first part introduces training and prediction process of traditional GP. Based on traditional GP, the second part proposes the DKL model structure, and introduces specific modeling process exploiting PSO algorithm.

A. GAUSSIAN PROCESS

The properties of GP are determined by the mean value function and the covariance function [18], which can be expressed as,

$$\begin{cases} m(\mathbf{x}) = E(f(\mathbf{x})) \\ k(\mathbf{x}, \mathbf{x}') = E\left\{[f(\mathbf{x}) - m(\mathbf{x})]\left[f(\mathbf{x}') - m(\mathbf{x}')\right]\right\} \end{cases} \tag{1}$$

where vectors $\mathbf{x}, \mathbf{x}' \in R^d$, $m(\mathbf{x})$ and $k\left(\mathbf{x}, \mathbf{x}'\right)$ are mean function and covariance function. The GP can be further expressed as,

$$f(x) \sim GP\left(m(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right) \tag{2}$$

For the regression model, $y = f(x) + \varepsilon$. The observed target value y is affected by an extra Gaussian noise term ε, which is random variable subject to normal distribution, and its mean is 0 and variance is σ_n^2, denoted as,

$$\varepsilon \sim N(0, \sigma_n^2) \tag{3}$$

Further, the prior distribution of y is written as,

$$y \sim N\left(0, \mathbf{K} + \sigma_n^2\mathbf{I}\right) \tag{4}$$

where $K = K(X, X)$ is a positive definite covariance matrix of $n \times n$ order symmetry, and matrix elements are used to measure the correlation between $\mathbf{x_i}$ and $\mathbf{x_j}$. n training sample outputs y and n^* test sample outputs f^* form a joint Gaussian prior distribution, defined as,

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim N\left(0, \begin{pmatrix} K(X,X) + \sigma_n^2\mathbf{I} & K(X, X*) \\ K(X*, X) & K(X*, X*) \end{pmatrix}\right) \tag{5}$$

where $K = K(X, X^*)$ is covariance matrix of order $n \times n^*$ between n^* test input samples and n training input samples, $K = K(X^*, X^*)$ is covariance matrix of order $n^* \times n^*$ for test input samples itself.

The automatic relevance determination (ARD) squared exponential kernel is often a default choice for GP regression. In this case, sample functions are not ideally smooth for practical optimization problems, so we use the ARD Matern 5/2 kernel [24]:

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \theta_0\left(1 + \sqrt{5r^2\left(\mathbf{x}, \mathbf{x}'\right)} + \frac{5}{3}r^2\left(\mathbf{x}, \mathbf{x}'\right)\right) \times \exp\left\{-\sqrt{5r^2\left(\mathbf{x}, \mathbf{x}'\right)}\right\} \tag{6}$$

where $r^2\left(\mathbf{x}, \mathbf{x}'\right) = \sum_{j=1}^{D}\left(\mathbf{x}_j - \mathbf{x}'_j\right)^2/\theta_j^2$, D is dimension of the input variables of GP.

The properties of GP mean function and covariance function are determined by a set of hyperparameters, which can be determined by the maximum likelihood function. By establishing the logarithmic marginal likelihood function of the conditional probability of training samples, the partial derivative of the hyperparameters is obtained, and then conjugate gradient optimization method is used to find the optimal solution of the hyperparameters. The negative logarithm likelihood function can be expressed as,

$$\mathbf{l} = logp(y|\theta, X) = -\frac{1}{2}\mathrm{y}^T K^{-1} y - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \tag{7}$$

when the optimal hyperparameters are obtained, the trained GP can be used for relevant prediction.

Based on new input x^*, the input value $\mathbf{X}$ of the training sample set, and the observation target value $\mathbf{y}$, distribution of maximum possible predictive posterior y^* can be given by

$$p(y^*|x^*, \mathbf{X}, \mathbf{y}) = N(\mathbf{m}, \mathbf{\Sigma}) \tag{8}$$

where $\mathbf{m}$ and $\mathbf{\Sigma}$ is predicted mean and covariance respectively, which can be defined by

$$\mathbf{m} = K\left(X^*, X\right)K\left(X, X\right)^{-1}\mathbf{y} \tag{9}$$

$$\mathbf{\Sigma} = K\left(X^*, X^*\right) - K\left(X^*, X\right)K\left(X, X\right)^{-1}K\left(X, X^*\right) \tag{10}$$

The predicted mean and covariance describe the Gaussian distribution that the predicted output is likely to follow, where the predicted mean can be regarded as the predicted output value of the nonlinear fitting tool, and the predicted variance can be regarded as the evaluation of uncertainty of the predicted mean. The prediction variance reflects the accuracy of the model at this point. The smaller the variance is, the higher confidence in predicted results of the model will be.

B. THE PROPOSED MODEL

PSO is a random search algorithm that finds the best solution by simulating the foraging behavior of birds, which is very effective for finding the global optimal solution [21]. This paper proposes a DKL model based on PSO algorithm, owning multiple-nonlinear-mapping layers, which transform data sample correlation characteristics. PSO is used to optimize the DKL model structure, as well as the weights and biases of each layer and GP hyperparameters.

Suppose N training sample labels are obtained by using full wave electromagnetic simulation software. For the modeling of microwave components with single variable output, ith training sample set can be expressed as $\left\{\mathbf{G}_i = (\mathbf{x}_i, \mathbf{y}_i)\,|\mathbf{x}_i \in R^d, \mathbf{y}_i \in R, i = 1, \ldots, N\right\}$, where the input is $\{\mathbf{x}_i\}_{k=1}^{d} = \{x_{i1}, x_{i2}, \cdots, x_{id}\}, k = 1, 2, \cdots, d$, d is number of variables in a set of input, and $\mathbf{y_i}$ is output of corresponding inputs $\{\mathbf{x}_i\}_{k=1}^{d}$.

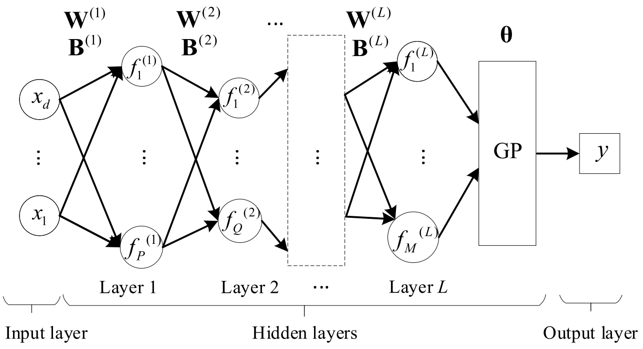


FIGURE 1. Schematic diagram of the proposed DKL model.

Suppose the number of multiple-nonlinear-mapping layers is L, then the covariance function of GP is given by

$$k(\mathbf{x}, \mathbf{x}' \mid \boldsymbol{\theta}) \rightarrow k(f^{(L)}(\ldots(f^{(1)}(\mathbf{x}))), f^{(L)}(\ldots(f^{(1)}(\mathbf{x}'))) \mid \mathbf{W}, \mathbf{B}, \boldsymbol{\theta}) \tag{11}$$

where $\mathbf{W}$ is weight matrix and $\mathbf{B}$ is bias matrix of the mapping layers, $\mathbf{W} = (\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)})$, $\mathbf{B} = (\mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(L)})$. Layer 1, layer 2, ..., and layer (L-1) adopt *Sigmoid* activation function for each unit, and the layer L adopts *Linear* activation function for each unit. They can be expressed as,

$$f^{(1)}\left(\mathbf{x}_i^{(1)}\right) = \frac{1}{1 + e^{-\mathbf{W}^{(1)}\mathbf{x}_i^{(1)} + \mathbf{B}^{(1)}}} \tag{12}$$

where, $\mathbf{x}_i^{(1)} = \mathbf{x}_i$.

$$f^{(2)}\left(\mathbf{x}_i^{(2)}\right) = \frac{1}{1 + e^{-\mathbf{W}^{(2)}\mathbf{x}_i^{(2)} + \mathbf{B}^{(2)}}} \tag{13}$$

where, $\mathbf{x}_i^{(2)} = f^{(1)}(\mathbf{x}_i^{(1)})$.

$$f^{(L-1)}\left(\mathbf{x}_i^{(L-1)}\right) = \frac{1}{1 + e^{-\mathbf{W}^{(L-1)}\mathbf{x}_i^{(L-1)} + \mathbf{B}^{(L-1)}}} \tag{14}$$

where, $\mathbf{x}_i^{(L-1)} = f^{(L-2)}(..f^{(2)}(f^{(1)}(\mathbf{x}_i^{(1)})))$, (L>3).

$$f^{(L)}\left(\mathbf{x}_i^{(L)}\right) = \mathbf{W}^{(L)}\mathbf{x}_i^{(L)} - \mathbf{B}^{(L)} \tag{15}$$

where, $\mathbf{x}_i^{(L-1)} = f^{(L-2)}(..f^{(2)}(f^{(1)}(\mathbf{x}_i^{(1)})))$, (L>2).

Suppose there are M neurons in the layer i (i >2) and Q neurons in the layer ($i - 1$), then the weight matrix $\mathbf{W}^{(i)}$ of the layer i is M rows and Q columns, and the bias matrix $\mathbf{B}^{(i)}$ is M rows and 1 column, denoted as,

$$\mathbf{W}^{(i)} = \begin{bmatrix} w_{11}^{(i)} & \ldots & w_{1Q}^{(i)} \\ \vdots & \ldots & \vdots \\ w_{M1}^{(i)} & \ldots & w_{MQ}^{(i)} \end{bmatrix}, \quad \mathbf{B}^{(i)} = \begin{bmatrix} B_1^{(i)} \\ \vdots \\ B_M^{(i)} \end{bmatrix} \tag{16}$$

Figure 1 is the schematic diagram of the proposed DKL model with multiple layers.

The specific operation process of the proposed DKL is followed.

Step 1: Set the number of multiple-nonlinear-mapping layers and the number of neurons of each mapping layer initially.

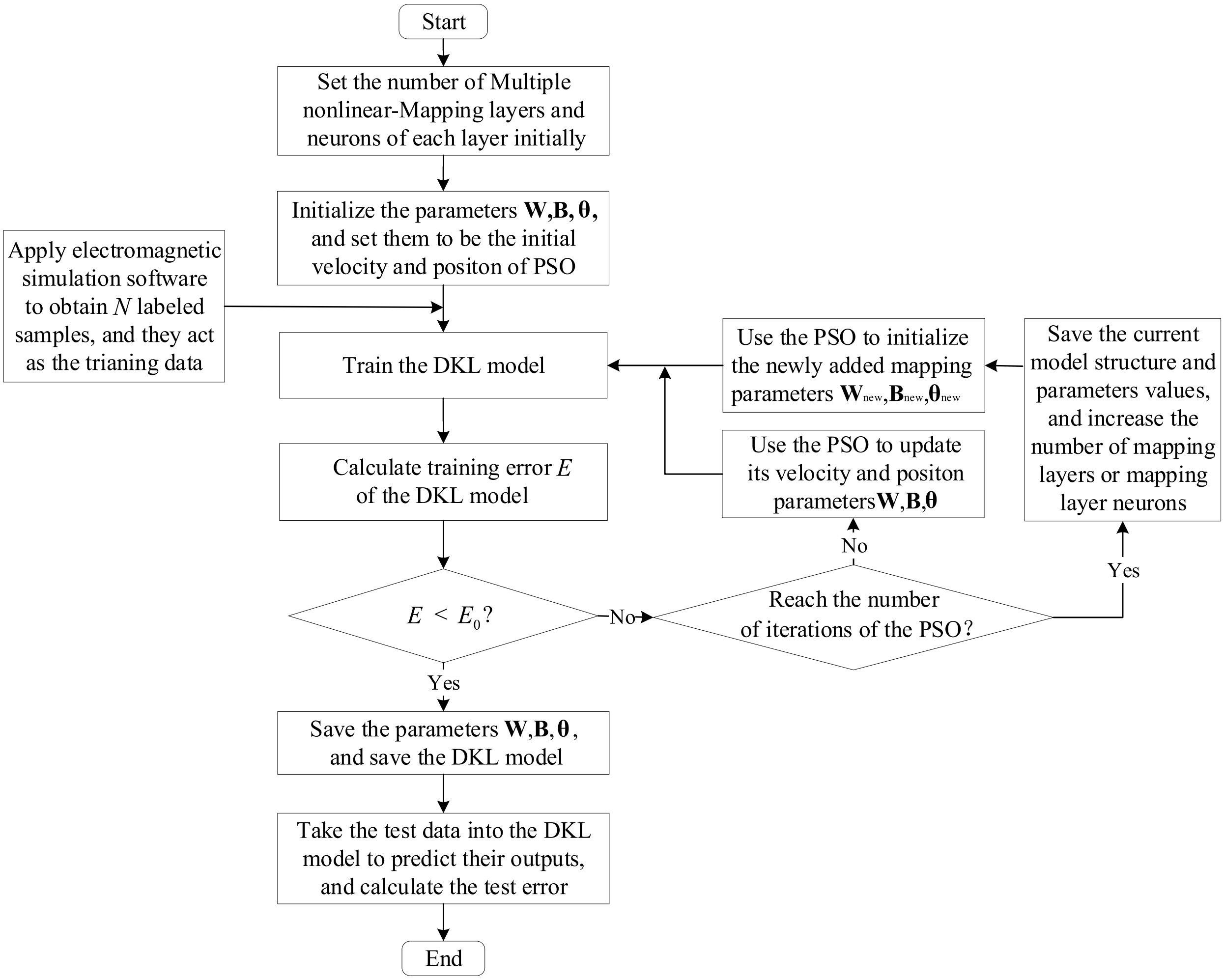

Start
Set the number of Multiple nonlinear-Mapping layers and neurons of each layer initially
Initialize the parameters W,B, θ, and set them to be the initial velocity and positon of PSO
Apply electromagnetic simulation software to obtain N labeled samples, and they act as the trianing data
Train the DKL model
Use the PSO to initialize the newly added mapping parameters Wnew,Bnew,θnew
Save the current model structure and parameters values, and increase the number of mapping layers or mapping layer neurons
Calculate training error E of the DKL model
Use the PSO to update its velocity and positon parameters W,B,θ
No
Yes
E < E0?
No
Reach the number of iterations of the PSO?
Yes
Save the parameters W,B,θ, and save the DKL model
Take the test data into the DKL model to predict their outputs, and calculate the test error
End

FIGURE 2. Flowchart of the proposed DKL model.

Step 2: Initialize the parameters $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(L)}$ of each layer and hyperparameters $\boldsymbol{\theta}$ of GP, and set them as the initial position of PSO simultaneously.

Step 3: Apply the full-wave electromagnetic simulation software to obtain N labeled samples, and they act as the training data.

Step 4: Train the DKL model.

Step 5: Calculate the training error E of the DKL model.

Step 6: Judge whether the training error E is less than threshold error E_0. If yes, go to Step 7; If not, determine whether the number of iterations of the PSO algorithm is reached. if yes, we save the current model structure and parameter values, and increase the number of mapping layers or mapping layer neurons. Then, initialize the newly added mapping parameters $\mathbf{W}_{\text{new}}, \mathbf{B}_{\text{new}}, \boldsymbol{\theta}_{\text{new}}$, and go to Step 4; If not, we update parameters $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(L)}$ of each layer and hyperparameters $\boldsymbol{\theta}$ of the GP model. Here, the $\mathbf{W}$, $\mathbf{B}$, $\boldsymbol{\theta}$ are position parameters of PSO, and their relationships are outlined in the Appendix. Then, go to Step 5.

Step 7: Save the parameters $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}, \mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(L)}$ of each layer and the hyperparameters $\boldsymbol{\theta}$ of GP at this time, and also save the DKL model structure.

Step 8: Take the test data into the trained DKL model to predict their outputs, and calculate their test error.

Figure 2 is the flowchart of the proposed DKL model.

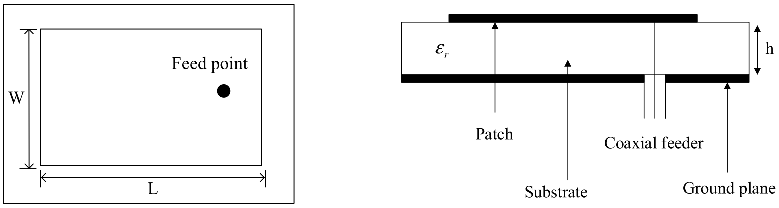
III. CASES STUDY

In this study, two MSA resonant frequency modeling examples are selected to validate the performance of the proposed DKL model. Because the inputs dimension in the selected MSA examples is relatively low (four dimensions), the layers number is set to three reasonably. In such a condition, the GP covariance function equation (11) can be updated by

$$k(\mathbf{x}, \mathbf{x}' \mid \boldsymbol{\theta}) \rightarrow k(f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x}))), f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x}'))) \mid \mathbf{W}, \mathbf{B}, \boldsymbol{\theta}) \quad \text{where,} \ \mathbf{W} = (\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}), \mathbf{B} = (\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \mathbf{B}^{(3)}). \tag{17}$$

A. RESONANT FREQUENCY OF THE RMSA

The first case is a RMSA from [25], [26], and its schematic diagram from top view is shown in Figure 3a, and from the side view in Figure 3b. The RMSA is composed of radiant element, dielectric layer and ground, with the width W,



a. Top view

b. Side view

FIGURE 3. Schematic diagram of the RMSA.

TABLE 1. Resonant frequency of the RMSA.

Number	W/cm	L/cm	h/cm	ε_r	$f_{measured}$/MHz
1★	1.256	2.756	0.952	2.55	3580
2	0.883	2.676	1.000	2.55	3980
3	2.000	2.500	0.079	2.22	3970
4	1.063	1.183	0.079	2.25	7730
5	0.910	1.000	0.127	10.20	4600
6	1.720	1.860	0.157	2.33	5060
7★	1.810	1.960	0.157	2.33	4805
8	1.270	1.350	0.163	2.55	6560
9	1.500	1.621	0.163	2.55	5600
10★	1.375	1.580	0.476	2.55	5100
11	1.120	1.200	0.242	2.55	7050
12	1.403	1.485	0.252	2.55	5800
13	1.530	1.630	0.300	2.50	5270
14	0.905	1.018	0.300	2.50	7990
15	1.170	1.280	0.300	2.50	6570
16★	1.030	3.380	1.281	2.55	3200
17	0.776	1.080	0.330	2.55	8000
18★	1.000	1.520	0.476	2.55	5820
19	0.987	1.450	0.450	2.55	6070
20★	1.337	1.412	0.200	2.55	6200
21	0.814	1.440	0.476	2.55	6380
22	0.790	1.620	0.550	2.55	5990
23	1.200	1.970	0.626	2.55	4660
24	0.783	2.300	0.854	2.55	4600
25	0.790	1.255	0.400	2.55	7134
26	0.974	2.620	0.952	2.55	3980
27	1.020	2.640	0.952	2.55	3900
28★	0.790	1.185	0.017	2.22	8450
29	0.777	2.835	1.100	2.55	3900
30	0.920	3.130	1.200	2.55	3470
31	0.850	1.290	0.017	2.22	7740
32	1.265	3.500	1.281	2.55	2980
33	1.080	3.400	1.281	2.55	3150

the length L, the dielectric layer thickness h, and relative dielectric constant ε_r.

As shown in Table 1, 33 sets of samples are measured by Kara [25], [26], in where 26 sets are selected as the training samples of the proposed DKL, and other 7 sets with the suffix ★ are as test samples.

The error threshold E_0 in this case is set as 1e-03 in advance. We set neurons of layer 1 as 6, the neurons of layer 2 as 6, and the neurons of layer 3 as 4, which means the network structure is 6-6-4 initially. At this time, there are totally 105 parameters that need to be optimized by the proposed DKL. In detail, the layer 1 has (24+6) parameters, the layer 2 has (36+6) parameters, the layer 3 has (24+4) parameters, and the GP model has 5 parameters. In the PSO algorithm, the total number of particles is 720, the maximum number of iterations is 1000, the acceleration constant is $c_1 = c_2 = 2$, and the inertia weight is $\omega = 0.9$. The range of the particle position is from -2 to 2, and the range of particle velocity is from -0.118 to 0.118. Figure 4 shows the iterations process of the RMSA. After 1000 iterations, the training error E is 0.002903, which is larger than the error threshold of 1e-03. The DKL model will be optimized further. For the case of the RMSA, it is set in the algorithm in advance that if the predicted accuracy of the model is not enough, the total number of layers will remain unchanged, and only the number of neurons in hidden layer 1 will increase. Since the error predicted by network structure 6-6-4 is larger than the threshold of 1e-03, the algorithm will continue to optimize the internal network structure. Then, the number of neurons in hidden layer 1 is updated from 6 to 7. At this time, the network structure is 7-6-4. In this case, the proposed DKL will optimize 116 parameters, among which, layer 1 has (28+7) parameters, the layer 2 has (42+6) parameters, the layer 3 has (24+4) parameters, and the GP model has 5 parameters. As shown in Figure 5, when the PSO algorithm iterates to 128th time, the training error E is 0.00098912 and less than the error threshold of 1e-03, satisfying the termination conditions. Therefore, the procedure stop.

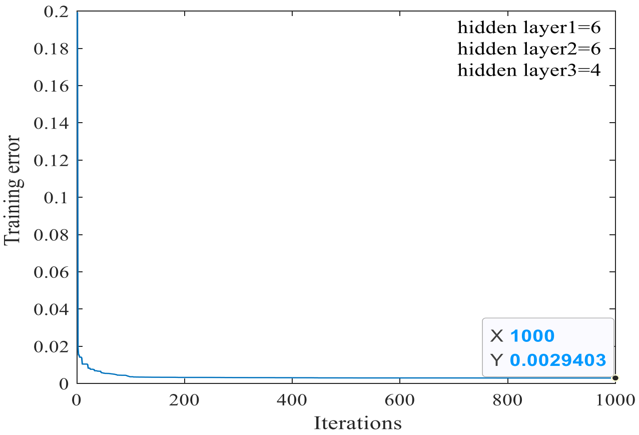


FIGURE 4. Iteration result of the RMSA with network structure 6-6-4.

TABLE 2. Test results of the RMSA.

f_{ME}/MHz	DBD [27]	BP [27]	EDBD [27]	PTS [28]	BiPSO+NNE [29]	GP52	DKL52
8450	8226	8233.1	8328.2	8148.6	8169	8251.843	8239.441
4805	4684.8	4703.3	4699.2	4879	4796	4830.272	4823.193
6200	6142.6	6147.2	6176.6	6205.7	6196	6198.311	6197.412
5100	5293.2	5291.4	5311.8	5191.4	5182	5145.394	5088.715
5820	5918	5924.5	5931	5780.3	5853	5850.37	5835.184
3580	3655.7	3644.6	3659.8	3685.2	3614	3582.141	3584.779
3200	3184.7	3178	3230.3	3167	3196	3196.581	3196.579
APE	2.020	1.939	1.894	1.663	0.973	0.640	0.519

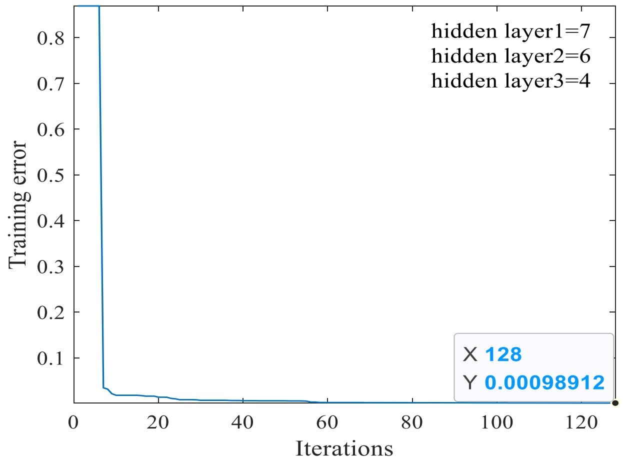


FIGURE 5. Iteration result of the RMSA with network structure 7-6-4.

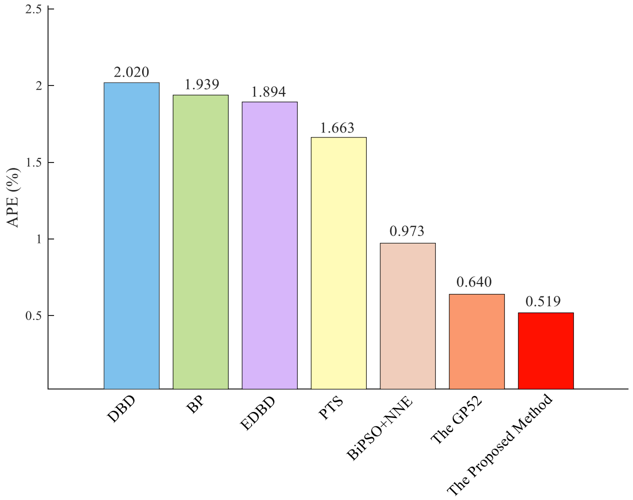


FIGURE 6. APE of the RMSA.

Unlabeled test samples can be predicted by the trained DKL model, and the test results are shown in Table 2, in which column 1 is the measured resonant frequency.

In order to validate the prediction precision of the proposed DKL, different modeling methods with the same 26 training samples are used to predict the labels of 7 sets of test samples. Average percentage error (APE) given by equation (18) is used to evaluate the performance of different models. Column 8 in Table 2 is the predicted results obtained by the proposed DKL model with ARD Matern 5/2 kernel function, named DKL52 here. Columns 2 through 5 show the predicted results by different single NN model with training algorithm of DBD (delta-bar-delta) [27], BP (back-propagation) [27], EDBD (extended delta-bar-delta) [27], and PTS (Parallel Tabu Search) [28]. Columns 6 through 7 show the results obtained by NNE (NN ensemble) based on BiPSO (binary particle swarm optimization) [29] algorithm, and GP52 model (GP with ARD Matern 5/2 kernel function), respectively.

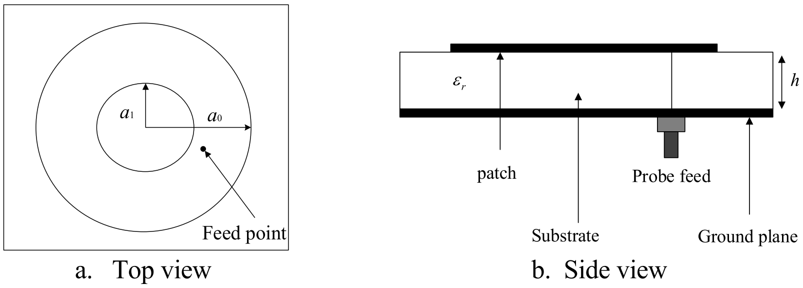
$$\mathrm{APE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_{pred_i} - y_{test_i}|}{|y_{test_i}|} \times 100 \tag{18}$$

where y_{pred} are the predicted labels, y_{test} are the true labels, and n is the number of test samples. The predicted APE results by different modeling methods are presented in Figure 6.

As can be seen from Table 2 and Figure 6, for the RMSA, the APE of the proposed DKL52 is 0.519, which is within the allowable range for antenna design. At the same time, it is least among the other 6 modeling methods. By the calculation, prediction accuracy of the DKL52 is higher 18.906% than that of the GP52, 68.791% than that of the best single NN model that is based on PTS algorithms, and 46.660% than that of the NNE based on BiPSO algorithms. We can conclude that the proposed DKL52 has strong reliability and high prediction accuracy, and can successfully model the resonant frequency of the RMSA.

B. RESONANT FREQUENCY OF ARCMSA

The second case is an ARCMSA from [30], [31], and its schematic diagram from top view is shown in Figure 7a, and from the side view in Figure 7b. The ARCMSA has an annular ring patch formed with the outer radius a_0 and inside radius a_1. The relative dielectric constant of substrate is ε_r. 70 sets



a. Top view

b. Side view

FIGURE 7. Schematic diagram of the ARCMSA.

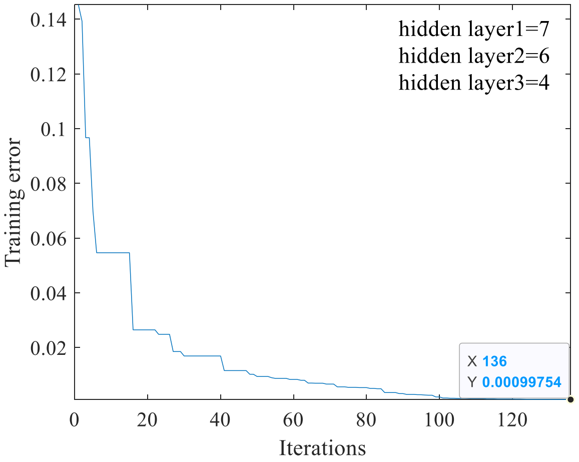


FIGURE 8. Iteration result of the ARCMSA with network structure 7-6-4.

TABLE 3. Test samples of the ARCMSA.

No.	a_0/mm	a_1/mm	h/mm	ε_r
1	15	4	2.5	9.8
2	15	6	1.57	2.33
3	15	10	3.175	2.2
4	20	3	0.64	4.5
5	20	6	1.57	2.33
6	20	15	2.5	9.8
7	25	4	3.175	2.2
8	25	12	1.57	2.33
9	30	10	0.64	4.5
10	30	20	3.175	2.2

of variables (a_0, a_1, h, ε_r) are selected as training samples of the proposed DKL.

The error threshold E_0 is also set as 1e-03 in this case in advance. We set neurons of the layer 1 as 7, neurons of layer 2 as 6, and neurons of layer 3 as 4 initially, which means the network structure is 7-6-4. There are totally 116 parameters optimized. In detail, the layer 1 has (28+7) parameters, the layer 2 has (42+6) parameters, layer 3 has (24+4) parameters, and the GP model has 5 parameters. In the PSO algorithm, particles number is 720, iterations number is 1000, the acceleration constant is $c_1 = c_2 = 2$, and the inertia weight is $\omega = 0.9$. The range of particle position is from -2 to 2, and the range of particle velocity is from -0.118 to 0.118. As shown in Figure 8, when the PSO algorithm research 136 iterations, the training error E

TABLE 4. Test results of the ARCMSA.

IE3D[32]	COIU[32]	PBCG[32]	OSS[32]	SCG[32]	PRCG[32]	FPCG[32]	LM[32]	BR[32]	GP52	DKL52
1.734	1.736	1.73	1.693	1.717	1.711	1.71	1.729	1.744	1.725	1.724
3.323	3.267	3.34	3.305	3.331	3.268	3.332	3.341	3.329	3.297	3.307
3.563	3.544	3.676	3.566	3.545	3.64	3.659	3.64	3.498	3.502	3.598
2.006	2.011	1.97	2.021	2.102	2.001	1.975	2.026	1.965	2.006	2.006
2.594	2.561	2.57	2.584	2.607	2.571	2.586	2.609	2.6	2.565	2.594
1.033	1.006	1.018	1.013	1.02	1.008	1.015	1.032	1.035	1.036	1.076
2.258	2.093	2.325	2.218	2.313	2.265	2.28	2.258	2.3	2.23	2.258
1.833	1.767	1.848	1.877	1.907	1.841	1.848	1.805	1.819	1.827	1.838
1.189	1.186	1.173	1.192	1.18	1.181	1.188	1.214	1.19	1.189	1.189
1.547	1.807	1.628	1.669	1.576	1.573	1.588	1.581	1.601	1.57	1.547
APE	3.448	1.852	1.833	1.739	1.174	1.259	1.061	1.118	0.758	0.648

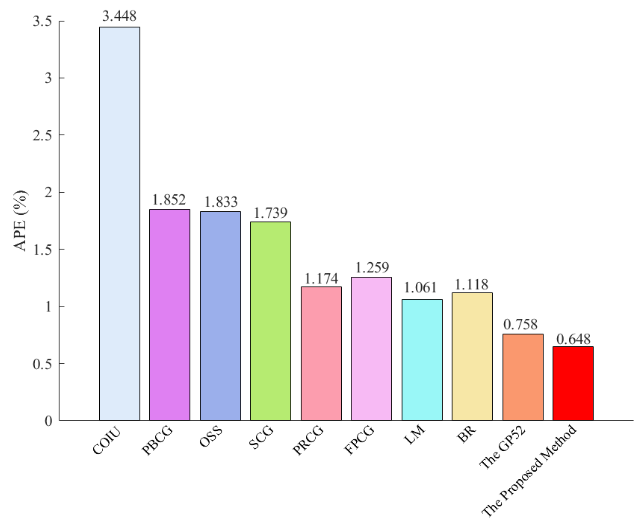


FIGURE 9. APE of the ARCMSA.

is 0.00099754 and less than the error threshold of 1e-03, satisfying the termination conditions. Then, the trained DKL model is used to predict the test samples listed in Table 3, and predicted results are shown in Table 4.

In order to validate the prediction precision of the proposed DKL, we compare its predicted result with different modeling methods in [32] that are trained by the same samples, and their predicted APE are exhibited in Table 4 and Figure 9. Column 1 in Table 4 is the measured resonant frequency, and column 11 is the predicted results by the proposed DKL52 model. Column 2 through 9 in Table 4 present the predicted results by the ANNs based on cyclical order incremental update (COIU) algorithm, Powel-Beale conjugate gradient (PBCG) algorithm, one step secant (OSS)algorithm, scaled conjugate gradient (SCG) algorithm, reduced symmetrical conjugate gradient (PRCG) algorithm, Fletcher-Powel conjugate gradient (FPCG) algorithm, LM algorithm, and Bayesian regularization (BR) algorithm, respectively. Column 10 in Table 4 is the predicted results by the GP52 model.

It can be seen from Table 4 and Figure 9 that the APE of the proposed DKL52 model is 0.648, which is least among all the other 9 modeling methods. Its prediction accuracy is higher 14.512% than that of the GP52, 81.206% than that of the ANN model based on COIU algorithm, and 38.926% than that of the ANN model based on LM algorithm. Therefore, we may conclude that the DKL52 has strong reliability and high prediction accuracy.

IV. CONCLUSION

In order to save the time of collecting labeled training samples for modeling and optimizing electromagnetic component and improve the design efficiency, a DKL model with multiple-nonlinear-mapping layers is proposed in this study. The model transforms correlation characteristics of training samples through multiple layer mappings, and takes the mapping output as input of the GP. At the same time, PSO algorithm is applied to optimize weight and bias of each layer and GP hyperparameters, also including model structure. These tricks make up for the shortcomings of the ANN and the traditional GP, and solves the over-fitting problem of the NNE. Moreover, to some extent, the proposed DKL model can avoid the error propagation caused by gradient disappear phenomenon of deep network, improve the prediction precision, and shorten modeling and optimization time. The resonant frequency of the RMSA and ARCMSA are used to evaluate the performance of the proposed model. The experiments results show that the performance of the proposed DKL is better than the ANN, GP and NNE in the terms of prediction precision. The DKL is suitable for processing small samples, and its generalization ability is great.

Although the experiments presented here considering only three layers in the hierarchy, the methodology is directly applicable to deeper architectures, and we also find that the model prediction ability is associated with number of each layer's neurons in the actual experiments, with which we intend to experiment in the future. The proposed method is not appropriated for the electromagnetic model with too high dimensions, e.g., the electromagnetic model with inputs of 30 dimensions, because it will take too much time to train the GP, leading to low-cost performance. In conclusion, the proposed DKL model with multiple-nonlinear-mapping layers promotes the research on the optimization design of electromagnetic components further.

APPENDIX

In PSO algorithm, velocity **v** and position **x** of the *j*th particle at generation *t* can be updated by the following two formulas:

$$\begin{aligned} \mathbf{v}_j(t) &= \omega \cdot \mathbf{v}_j(t-1) + c_1 \cdot random(0,1) \\ &\quad \cdot \left(\mathbf{pbest}_j - \mathbf{x}_j(t-1)\right) \\ &\quad + c_2 \cdot random(0,1) \cdot \left(\mathbf{gbest} - \mathbf{x}_j(t-1)\right) \quad \text{(A.1)} \\ \mathbf{x}_j(t) &= \mathbf{x}_j(t-1) + \mathbf{v}_j(t) \quad \text{(A.2)} \end{aligned}$$

where $\mathbf{x}_j$ is the position of the *j*th particle, $\mathbf{v}_j$ represents the velocity of particle *j*. Parameter ω is the inertia weight, and parameters c_1, c_2 are the acceleration constants. $\mathbf{Pbest}_j$ represents the personal best solution (position) found by the *j*th particle and **gbest** is the global best solution (position). *random*(0, 1) represents random numbers in the range [0,1].

According to the optimized objective matrix **W**, **B**, $\boldsymbol{\theta}$ in the study, the two formulas (A.1), (A.2) can be updated. Variable matrix **W** will be:

$$\begin{aligned} \mathbf{v}_{\mathbf{W}j}(t) &= \omega \cdot \mathbf{v}_{\mathbf{W}j}(t-1) + c_1 \cdot random(0,1) \\ &\quad \cdot \left(\mathbf{pbest}_{\mathbf{W}j} - \mathbf{W}_j(t-1)\right) \\ &\quad + c_2 \cdot random(0,1) \cdot \left(\mathbf{gbest}_{\mathbf{W}} - \mathbf{W}_j(t-1)\right) \quad \text{(A.3)} \\ \mathbf{W}_j(t) &= \mathbf{W}_j(t-1) + \mathbf{v}_{\mathbf{W}j}(t) \quad \text{(A.4)} \end{aligned}$$

The rows and columns of $\mathbf{W}_j$, $\mathbf{v}_{\mathbf{W}j}$, $\mathbf{pbest}_{\mathbf{W}j}$, $\mathbf{gbest}_{\mathbf{W}}$ are same as the input matrix $\mathbf{W}$. Variable matrix $\mathbf{B}$ will be:

$$\begin{aligned}\mathbf{v}_{\mathbf{B}j}(t) = {} & \omega \cdot \mathbf{v}_{\mathbf{B}j}(t-1) + c_1 \cdot random(0, 1) \\ & \cdot \left(\mathbf{pbest}_{\mathbf{B}j} - \mathbf{B}_j(t-1)\right) \\ & + c_2 \cdot random(0, 1) \cdot \left(\mathbf{gbest}_{\mathbf{B}} - \mathbf{B}_j(t-1)\right) \quad \text{(A.5)}\end{aligned}$$

$$\mathbf{B}_j(t) = \mathbf{B}_j(t-1) + \mathbf{v}_{\mathbf{B}j}(t) \quad \text{(A.6)}$$

The rows and columns of $\mathbf{B}_j$, $\mathbf{v}_{\mathbf{B}j}$, $\mathbf{pbest}_{\mathbf{B}j}$, $\mathbf{gbest}_{\mathbf{B}}$ are same as the input matrix $\mathbf{B}$. And variable matrix $\boldsymbol{\theta}$ will be:

$$\begin{aligned}\mathbf{v}_{\boldsymbol{\theta}j}(t) = {} & \omega \cdot \mathbf{v}_{\boldsymbol{\theta}j}(t-1) + c_1 \cdot random(0, 1) \\ & \cdot \left(\mathbf{pbest}_{\boldsymbol{\theta}j} - \boldsymbol{\theta}_j(t-1)\right) \\ & + c_2 \cdot random(0, 1) \cdot \left(\mathbf{gbest}_{\boldsymbol{\theta}} - \boldsymbol{\theta}_j(t-1)\right) \quad \text{(A.7)}\end{aligned}$$

$$\boldsymbol{\theta}_j(t) = \boldsymbol{\theta}_j(t-1) + \mathbf{v}_{\boldsymbol{\theta}j}(t) \quad \text{(A.8)}$$

The rows and columns of $\boldsymbol{\theta}_j$, $\mathbf{v}_{\boldsymbol{\theta}j}$, $\mathbf{pbest}_{\boldsymbol{\theta}j}$, $\mathbf{gbest}_{\boldsymbol{\theta}}$ are same as the input matrix $\boldsymbol{\theta}$. The specific values of parameter ω, c_1, c_2 are set in Section III for the experiments.

ACKNOWLEDGMENT

(Shudan Han and Yubo Tian are co-first authors.)

REFERENCES

[1] S. K. Behera and Y. Choukiker, "Design and optimization of dual band microstrip antenna using particle swarm optimization technique," *J. Infr., Millim., Terahertz Waves*, vol. 31, no. 11, pp. 1346–1354, Nov. 2010.

[2] V. Grout, M. O. Akinsolu, B. Liu, P. I. Lazaridis, K. K. Mistry, and Z. D. Zaharis, "Software solutions for antenna design exploration: A comparison of packages, tools, techniques, and algorithms for various design challenges," *IEEE Antennas Propag. Mag.*, vol. 61, no. 3, pp. 48–59, Jun. 2019.

[3] F. Chen and Y.-B. Tian, "Modeling resonant frequency of rectangular microstrip antenna using CUDA-based artificial neural network trained by particle swarm optimization algorithm," *Appl. Comput. Electromagn. Soc. J.*, vol. 29, no. 12, pp. 1025–1034, 2014.

[4] L.-Y. Xiao, W. Shao, X. Ding, Q. H. Liu, and W. T. Joines, "Multigrade artificial neural network for the design of finite periodic arrays," *IEEE Trans. Antennas Propag.*, vol. 67, no. 5, pp. 3109–3116, May 2019.

[5] Y. Chen, Y. Tian, Z. Qiang, and L. Xu, "Optimisation of reflection coefficient of microstrip antennas based on KBNN exploiting GPR model," *IET Microw., Antennas Propag.*, vol. 12, no. 4, pp. 602–606, Mar. 2018.

[6] Z. Pitra, L. Bajer, and M. Holena, "Comparing SVM, Gaussian process and random forest surrogate models for the CMA-ES," in *Proc. ITAT*, 2015, pp. 186–193.

[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, Nov. 2016.

[8] H. Fu, C.-M. Vong, P.-K. Wong, and Z. Yang, "Fast detection of impact location using kernel extreme learning machine," *Neural Comput. Appl.*, vol. 27, no. 1, pp. 121–130, Jan. 2016.

[9] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[10] J. Zhu, J. J. Fox, N. Yi, and H. Cheng, "Structural design for stretchable microstrip antennas," *ACS Appl. Mater. Interfaces*, vol. 11, no. 9, pp. 8867–8877, Mar. 2019.

[11] J. Gao, Y. Tian, and X. Chen, "Antenna optimization based on co-training algorithm of Gaussian process and support vector machine," *IEEE Access*, vol. 8, pp. 211380–211390, 2020, doi: 10.1109/ACCESS.2020.3039269.

[12] X. Chen, Y. Tian, T. Zhang, and J. Gao, "Differential evolution based manifold Gaussian process machine learning for microwave filter's parameter extraction," *IEEE Access*, vol. 8, pp. 146450–146462, 2020.

[13] J. Gao, Y. Tian, X. Zheng, and X. Chen, "Resonant frequency modeling of microwave antennas using Gaussian process based on semisupervised learning," *Complexity*, vol. 2020, pp. 1–12, May 2020.

[14] T.-J. Huang, "Imitating the brain with neurocomputer a 'new' way towards artificial general intelligence," *Int. J. Autom. Comput.*, vol. 14, no. 5, pp. 520–531, 2017.

[15] Z. Sun, K. Hu, T. Hu, J. Liu, and K. Zhu, "Fast multi-label low-rank linearized SVM classification algorithm based on approximate extreme points," *IEEE Access*, vol. 6, pp. 42319–42326, 2018.

[16] A. Wilson and H. Nickisch, "Kernel interpolation for scalable structured Gaussian processes (KISS-GP)," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1775–1784.

[17] L. Bajer, Z. Pitra, J. Repický, and M. Holena, "Gaussian process surrogate models for the CMA evolution strategy," *Evol. Comput.*, vol. 27, no. 4, pp. 665–697, Dec. 2019.

[18] D. R. Burt, C. E. Rasmussen, and M. van der Wilk, "Rates of convergence for sparse variational Gaussian process regression," 2019, *arXiv:1903.03571*. [Online]. Available: http://arxiv.org/abs/1903.03571

[19] C. E. Rasmussen, "Gaussian processes in machine learning," in *Proc. Summer School Mach. Learn.* Berlin, Germany: Springer, 2004, pp. 63–71.

[20] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Proc. Artif. Intell. Statist.*, 2016, pp. 370–378.

[21] M. Belkin, S. Ma, and S. Mandal, "To understand deep learning we need to understand kernel learning," 2018, *arXiv:1802.01396*. [Online]. Available: http://arxiv.org/abs/1802.01396

[22] C. Mao, R. Lin, C. Xu, and Q. He, "Towards a trust prediction framework for cloud services based on PSO-driven neural network," *IEEE Access*, vol. 5, pp. 2187–2199, 2017.

[23] A. Damianou and N. D. Lawrence, "Deep Gaussian processes," in *Proc. Artif. Intell. Statist.*, 2013, pp. 207–215.

[24] C.-M. Yan, G.-Y. Lu, Y.-T. Liu, and X.-Y. Deng, "A modified PSO algorithm with exponential decay weight," in *Proc. 13th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Jul. 2017, pp. 239–242.

[25] M. Kara, "Closed-form expressions for the resonant frequency of rectangular microstrip antenna elements with thick substrates," *Microw. Opt. Technol. Lett.*, vol. 12, no. 3, pp. 131–136, Jun. 1996.

[26] M. Kara, "The resonant frequency of rectangular microstrip antenna elements with various substrate thicknesses," *Microw. Opt. Technol. Lett.*, vol. 11, no. 2, pp. 55–59, Feb. 1996.

[27] K. Guney, S. Sagiroglu, and M. Erler, "Generalized neural method to determine resonant frequencies of various microstrip antennas," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 12, no. 1, pp. 131–139, Jan. 2002.

[28] S. Sagiroglu and A. Kalinli, "Determining resonant frequencies of various microstrip antennas within a single neural model trained using parallel tabu search algorithm," *Electromagnetics*, vol. 25, no. 6, pp. 551–565, Aug. 2005.

[29] T. Yu-Bo, Z. Su-Ling, and L. Jing-Yi, "Modeling resonant frequency of microstrip antenna based on neural network ensemble," *Int. J. Numer. Model., Electron. Netw., Devices Fields*, vol. 24, no. 1, pp. 78–88, Jan. 2011.

[30] J. Singh, A. P. Singh, and T. S. Kamal, "Design of circular microstrip antenna using artificial neural networks," in *Proc. World Congr. Eng.*, 2011, pp. 9–12.

[31] S. Bedra, "Artificial neural network to design of circular microstrip antenna," *Global J. Res. Eng.*, vol. 12, no. 10, pp. 1–7, Oct. 2012.

[32] A. Akdagli and A. Kayabasi, "An accurate computation method based on artificial neural networks with different learning algorithms for resonant frequency of annular ring microstrip antennas," *J. Comput. Electron.*, vol. 13, no. 4, pp. 1014–1019, Dec. 2014.

SHUDAN HAN was born in Yancheng, China, in 1997. She received the B.Tech. degree from the Suzhou University of Science and Technology. She is currently pursuing the master's degree with the Jiangsu University of Science and Technology. Her research interests include applications of intelligent optimization algorithms and deep learning methods to the design of electromagnetic microwave devices.

YUBO TIAN was born in Tieling, Liaoning, China, in 1971. He received the Ph.D. degree in radio physics from the Department of Electronic Science and Engineering, Nanjing University, Nanjing, China. From 1997 to 2004, he was with the Department of Information Engineering, Shenyang University, Shenyang, China. From 2005 to 2020, he was with the School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, China, where he has been a Full Professor and the Vice Dean, since 2011. He was a Visiting Scholar with the University of California at Los Angeles, in 2009, and Griffith University, in 2015. He is currently with the School of Information and Communication Engineering, Guangzhou Maritime University, Guangzhou, China. He has authored and coauthored more than 100 journal articles and three books. He also holds more than 20 filed/granted China patents. His current research interest includes machine learning methods and their applications in electronics and electromagnetics.

WEITONG DING was born in Suzhou, China, in 1997. He received the B.Tech. degree from the North China Institute of Science and Technology. He is currently pursuing the master's degree with the Jiangsu University of Science and Technology. His research interest includes machine learning and its application in antenna optimization.

PENGFEI LI was born in Taizhou, China, in 1994. He received the bachelor's degree from the Jiangsu University of Science and Technology, in 2018, where he is currently pursuing the master's degree. His research interests include deep learning and array signal processing.