Research Article

Resonant Frequency Modeling of Microwave Antennas Using Gaussian Process Based on Semisupervised Learning

Jing Gao , Yubo Tian , Xie Zheng, and Xuezhi Chen

School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, Jiangsu, China

Correspondence should be addressed to Yubo Tian; tianyubo@just.edu.cn

Received 8 January 2020; Revised 8 April 2020; Accepted 30 April 2020; Published 15 May 2020

Academic Editor: Marcin Mrugalski

Copyright © 2020 Jing Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For the optimal design of electromagnetic devices, it is the most time consuming to obtain the training samples from full wave electromagnetic simulation software, including HFSS, CST, and IE3D. Traditional machine learning methods usually use only labeled samples or unlabeled samples, but in practical problems, labeled samples and unlabeled samples coexist, and the acquisition cost of labeled samples is relatively high. This paper proposes a semisupervised learning Gaussian Process (GP), which combines unlabeled samples to improve the accuracy of the GP model and reduce the number of labeled training samples required. The proposed GP model consists two parts: initial training and self-training. In the process of initial training, a small number of labeled samples obtained by full wave electromagnetic simulation are used for training the initial GP model. Afterwards, the trained GP model is copied to another GP model in the process of self-training, and then the two GP models will update after crosstraining with different unlabeled samples. Using the same test samples for testing and updating, a model with a smaller error will replace another. Repeat the self-training process until a predefined stopping criterion is met. Four different benchmark functions and resonant frequency modeling problems of three different microstrip antennas are used to evaluate the effectiveness of the GP model. The results show that the proposed GP model has a good fitting effectiveness on benchmark functions. For microstrip antennas resonant frequency modeling problems, in the case of using the same labeled samples, its predictive ability is better than that of the traditional supervised GP model.

1. Introduction

In recent years, for the optimization design of electromagnetic devices, some excellent research results have been achieved by numerical simulation calculation or combining full wave electromagnetic simulation software such as HFSS with global optimization algorithm, such as particle swarm optimization (PSO) [1]. In general, microwave devices can be simulated by HFSS software to obtain accurate results. However, when global optimization algorithm is combined with HFSS software, each updating needs to call HFSS for evaluation, which is costly and very time consuming. Therefore, using a surrogate method instead of HFSS to evaluate the fitness of electromagnetic devices can save greatly optimization time, which is a hot topic in electromagnetic optimization design. Many researchers have proposed lots of surrogate methods, such as artificial neural

network (ANN) [2, 3], support vector machine (SVM) [4, 5], kernel extreme learning machine (KELM) [6, 7], and Gaussian process (GP) [8, 9].

GP is a machine learning method that has developed rapidly in recent years. It has a strict statistical theoretical basis and is suitable for dealing with complex problems such as high dimensions, small samples, and nonlinearity [10, 11]. GP has been developed with continuous research studies on Bayesian neural network (NN) and has the advantages such as flexible nonparametric inference, adaptive acquisition of super-parameters, and predictive outputs. In the electromagnetic field, many scholars have made some achievements in the application of GP, verifying the feasibility of GP as an alternative method of electromagnetic simulation software.

However, to our best knowledge, the most GP modeling of electromagnetic behaviors are based on supervised

learning. The labeled training samples used in GP modeling are based on HFSS. The acquisition of labeled samples by HFSS will consume a lot of time, which is also the main reason affecting the efficiency of antenna optimization. Therefore, semisupervised learning (SSL) [12, 13] method is proposed on the basis of existing research studies in this study. Traditional machine learning techniques rely on large numbers of labeled samples for training. In practical electromagnetic engineering, it is difficult to obtain labeled samples, while unlabeled samples are cheap and easy to be obtained [14]. SSL is a learning method between supervised and unsupervised learning [15], mainly considering the combination of labeled samples and unlabeled samples to improve the learning efficiency, which is suitable for regression and classification problems. Specifically, SSL methods include self-training [16], co-training [17], graph-based methods [18], EM-with generative modes [19], and transductive SVM [20]. In this paper, the self-training method is combined with GP modeling to solve the antenna optimization design.

Self-training is one of the SSL methods, which is simple and effective without specific assumptions [21, 22] and is commonly used to deal with classification problems. Based on the traditional self-training method, this paper proposes the SSL-based GP model, which is used to predict the resonant frequency of microstrip antennas (MSAs) that belongs to the regression problems. The SSL-based GP model proposed in this study includes two parts: initial training and self-training. In the initial training, a few labeled samples are used to obtain a GP model with low accuracy, and then the initial error of the GP model can be obtained. Before self-training, the trained GP model in the initial training process is copied to another. Next, different unlabeled samples are inputted, respectively, into each GP model and the corresponding outputs are achieved. The two models are cross-trained with the generated pseudolabeled samples, and then the two GP models are updated and differ from each other. The same test samples are used to verify the two updated GP models, and the model with smaller error is used to substitute another one, and then a more accurate training sample set can be produced for future training the GP model. Repeat the self-training process until a predefined error threshold is met. Four benchmark functions and resonant frequency modeling problems of three MSAs are used to evaluate the effectiveness of the proposed algorithm. Through the experiments of test functions and resonant frequency of three different MSAs, we can get the conclusion that the predictive ability of the proposed GP model in this study is better than that of the traditional supervised GP model.

2. Semisupervised Learning Model

2.1. Gaussian Process Modeling

2.1.1. Training. The properties of GP are determined by the mean function and the covariance function [23], which can be expressed as

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

where $\mathbf{x}, \mathbf{x}' \in R^d$, $m(\mathbf{x})$ is the mean function, and $k(\mathbf{x}, \mathbf{x}')$ is the covariance function. Furthermore, it can be expressed as

$$f(\mathbf{x}) \sim \text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \quad (2)$$

For the regression model $y = f(\mathbf{x}) + \varepsilon$, the observed target value y polluted by additive noise ε that is the random variable subject to normal distribution, and its mean is 0 and its variance is σ_n^2 ; therefore,

$$\varepsilon \sim N(0, \sigma_n^2). \quad (3)$$

The prior distribution of y is given by

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

where $\mathbf{K} = K(X, X)$ is the symmetric positive definite covariance matrix of order $n \times n$, k_{ij} measures the correlation $\mathbf{x}_i$ and $\mathbf{x}_j$, and n training sample outputs y , and n^* testing sample outputs f^* constitute the joint Gauss prior distribution, that is,

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

GP can choose different covariance functions [24], usually using the square exponential covariance function as follows:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left[-0.5(\mathbf{x} - \mathbf{x}')^T M(\mathbf{x} - \mathbf{x}')\right]. \quad (6)$$

The properties of the mean function and the covariance function of GP are determined by a set of super-parameters. By establishing the log-likelihood function of the conditional probability of the training samples, the partial derivative of the super-parameters is obtained. The optimal solution of the super-parameters is found by the conjugate gradient optimization method. The form of the negative log-likelihood function is

$$\mathbf{l} = \log p(y | X) = -\frac{1}{2} y^T K^{-1} y - \frac{1}{2} \log |K| - \frac{n}{2} \log 2\pi. \quad (7)$$

After obtaining the optimal super-parameters, the trained GP used to perform the relevant prediction.

2.1.2. Predicting. Given the new input x^* , the input value of the training set $\mathbf{X}$, and the observed target value $\mathbf{y}$, which are used to infer the maximum possible predicted posterior distribution of y^* , it is written as

$$y^* | x^*, X, \mathbf{y} \sim N\left(\mathbf{m}, \sum\right), \quad (8)$$

where $\mathbf{m}$ and $\sum$ are the mean and the covariance of the prediction, given by

$$\begin{aligned} \mathbf{m} &= K(X^*, X)K(X, X)^{-1}\mathbf{y}, \\ \sum &= K(X^*, X^*) - K(X^*, X)K(X, X)^{-1}K(X, X^*). \end{aligned} \quad (9)$$

The predicted mean and covariance describe the Gaussian distribution that the predicted output may follow. The predicted mean value can be regarded as the predicted output value of the nonlinear fitting tool, and the predicted variance can be regarded as uncertainty evaluation of the predicted mean value. The magnitude of the prediction variance reflects the accuracy of the model at this point. The smaller the variance, the higher the accuracy of the model.

2.2. Training Process of the Semisupervised Learning Model.

For the proposed method in the study, there are two training processes. The first part is the initial training with one GP model, the second part is the self-training with two GP models.

2.2.1. Initial Training Process. Firstly, we use HFSS to simulate a small number of labeled samples, denoted as N_0 . Then, we use these N_0 samples as the original training samples. The i th training sample is expressed as $\{\mathbf{G}_i = (\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in R^n, \mathbf{y}_i \in R^m\}$, where $\{\mathbf{x}_i\}_{i=1}^n = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ is the input variable, n is the dimension of the input variables, while $\{\mathbf{y}_i\}_{i=1}^m = \{y_{i1}, y_{i2}, \dots, y_{im}\}$ is the corresponding output, m is the dimension of the output variables. After that we train the GP model with the N_0 training samples. Figure 1 is the flow chart of the initial training, and the process can be summarized as follows:

Step 1: using HFSS software simulates a few labeled training samples, denoted as N_0 .

Step 2: applying these N_0 samples to train the GP model. After training, the GP model is with relatively low accuracy because of small number of samples.

Step 3: the initial error of the trained GP model can be obtained.

2.2.2. Self-Training Process. Before the self-training, the trained GP model in the initial training process is copied to another one, denoted as GP_1 and GP_2 . Two groups of unlabeled sample sets are denoted as $\mathbf{Self}_1.X = \{\mathbf{x}_{11}, \mathbf{x}_{12}, \dots, \mathbf{x}_{1n}\}$ and $\mathbf{Self}_2.X = \{\mathbf{x}_{21}, \mathbf{x}_{22}, \dots, \mathbf{x}_{2n}\}$. In each iteration, we select, respectively, one sample from the unlabeled sample set and input it into GP_1 and GP_2 . We can obtain the corresponding output $\mathbf{Self}_1.Y = \{y_{11}, y_{12}, \dots, y_{1n}\}$ and $\mathbf{Self}_2.Y = \{y_{21}, y_{22}, \dots, y_{2n}\}$. The two groups of pseudolabeled samples generated by GP_1 and GP_2 are denoted, respectively, as $\mathbf{Self}_1.GP_1 = (\mathbf{Self}_1.X, \mathbf{Self}_1.Y)$ and $\mathbf{Self}_2.GP_2 = (\mathbf{Self}_2.X, \mathbf{Self}_2.Y)$. In each iteration, we use the HFSS software to simulate one sample and get its label, used as the test sample, denoted as $\{\mathbf{Test.G}_i = (\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in R^n, \mathbf{y}_i \in R^m\}$. The test sample is used to evaluate the performance of the GP_1 and GP_2 , and the test errors are denoted as e_1 and e_2 , respectively.

Figure 2 is the flow chart of the self-training algorithm, where i is the number of iterations, and its process can be summarized as follows:

Step 1: copy the GP model from the initial training process to another one, denoted, respectively, as GP_1 and GP_2 .

FIGURE 1: Initial training process.

Step 2: select num samples $\mathbf{Self}_1.X = \{\mathbf{x}_{11}, \mathbf{x}_{12}, \dots, \mathbf{x}_{1n}\}$ from the unlabeled sample set and then input them into GP_1 , and we can obtain corresponded outputs $\mathbf{Self}_1.GP_1 = (\mathbf{Self}_1.X, \mathbf{Self}_1.Y)$. Similarly, we select another num samples $\mathbf{Self}_2.X = \{\mathbf{x}_{21}, \mathbf{x}_{22}, \dots, \mathbf{x}_{2n}\}$ and input them into GP_2 , then we can obtain the corresponded outputs $\mathbf{Self}_2.GP_2 = (\mathbf{Self}_2.X, \mathbf{Self}_2.Y)$. In conclusion, the two groups of pseudolabeled samples are denoted as $\mathbf{Self}_1.GP_1 = (\mathbf{Self}_1.X, \mathbf{Self}_1.Y)$ and $\mathbf{Self}_2.GP_2 = (\mathbf{Self}_2.X, \mathbf{Self}_2.Y)$.

Step 3: we use pseudolabeled $\mathbf{Self}_1.GP_1 = (\mathbf{Self}_1.X, \mathbf{Self}_1.Y)$ to future train GP_2 and use $\mathbf{Self}_2.GP_2 = (\mathbf{Self}_2.X, \mathbf{Self}_2.Y)$ to future train GP_1 . After that GP_1 and GP_2 are updated, and the two updated models differ from each other.

Step 4: the test sample in each iteration $\{\mathbf{Test.G}_i = (\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in R^n, \mathbf{y}_i \in R^m\}$ simulated by HFSS software is used to test the performance of the updated GP_1 and GP_2 . The test error of GP_1 is denoted as e_1 , and the test error of GP_2 is denoted as e_2 .

Step 5: compare the value of e_1 and e_2 . If e_1 is large than e_2 , which means the accuracy of GP_2 model is higher than that of GP_1 , we can replace GP_1 with GP_2 , denoted as GP. Otherwise, we replace GP_2 with GP_1 , denoted as GP.

Step 6: judge whether the smaller one in e_1 and e_2 meet the error threshold or not. If yes, the program will end. Otherwise, the test sample in this iteration $\{\mathbf{Test.G}_i = (\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{x}_i \in R^n, \mathbf{y}_i \in R^m\}$ is used to future train the above trained GP in step 5, and then go to step 1.

The pseudocode of the self-training process is shown in Algorithm 1. In the process of the self-training, the GP model uses the information obtained by unlabeled samples to update constantly. We add the test sample in each iteration to the training sample, set in order to improve the

FIGURE 2: Self-training process.

accuracy of the GP model. The whole training process can control the number of unlabeled samples added and prevent the model from losing precision due to too much unlabeled samples introduced.

3. Cases study

3.1. Benchmark Functions. To verify the feasibility of the proposed SSL-based GP model, we select four benchmark functions for testing. The functions are, respectively, Sphere function, Sum Squares function, Rastrigin function, and Schwefel function. They all have five dimensions, and the value interval of the independent variable is $[-30, 30]$. Their formulas are given by (10)–(13). The first two functions are unimodal functions, and the error threshold is set as $1e-06$. The last two functions are multimodal functions, and the error threshold is set as $1e-04$. We set the maximum number of iterations to 100:

$$f_1(x) = \sum_{i=1}^d x_i^2, \quad (10)$$

$$f_2(x) = \sum_{i=1}^d ix_i^2, \quad (11)$$

$$f_3(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)], \quad (12)$$

$$f_4(x) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|}). \quad (13)$$

The selected four benchmark functions are used to test the performance of the proposed SSL-based GP model. For each iteration, the Relative Error (RE) is used for evaluation, and it is given by

$$RE = \frac{|y_{\text{pred}} - y_{\text{test}}|}{y_{\text{test}}}, \quad (14)$$

where y_{pred} is the label predicted by the SSL-based GP model and y_{test} is the true label of the test sample. Table 1 records the iterative results of the four benchmark functions. We know from Table 1 that the error threshold of unimodal functions can be reached after about 25 iterations, while the error threshold of multimodal functions can be reached after about 75 iterations. All the models have achieved high accuracy within 100 iterations, which verifies the effectiveness of the proposed GP model.

```

(1) begin
(2)   $e_1 \leftarrow 1$  // assign initial value
(3)   $e_2 \leftarrow 1$ 
(4)   $i \leftarrow 0$ 
(5)  while  $\min(e_1, e_2) \geq \text{error threshold}$  do
(6)    copy the GP model from the initial training, denoted as  $GP_1$  and  $GP_2$ 
(7)     $i \leftarrow i + 1$ 
(8)    select num samples  $\text{Self}_1.X$  from the unlabeled sample set
(9)    input  $\text{Self}_1.X$  into  $GP_1$ 
(10)   output  $\text{Self}_1.Y$ 
(11)   select num samples  $\text{Self}_2.X$  from the unlabeled sample set
(12)   input  $\text{Self}_2.X$  into  $GP_2$ 
(13)   output  $\text{Self}_2.Y$ 
(14)   use  $\text{Self}_1.GP_1$  to future train  $GP_2$ 
(15)   use  $\text{Self}_2.GP_2$  to future train  $GP_1$ 
(16)   use one test sample  $\text{Test}.G$  by HFSS simulation to test the  $GP_1$  and  $GP_2$ 
(17)   output  $e_1$  and  $e_2$ 
(18)   if  $e_1 > e_2$ 
(19)     then replace  $GP_1$  with  $GP_2$ , denoted as  $GP$ 
(20)   else replace  $GP_2$  with  $GP_1$ , denoted as  $GP$ 
(21)   end
(22)   put test sample into existing training sample set to future train  $GP$ 
(23) end
(24) end

```

ALGORITHM 1: Pseudocode of the self-training process.

TABLE 1: Computing results of the four benchmark functions.

	Sphere function	Sum squares function	Rastrigin function	Schwefel function
Iteration number	27	22	75	73
Reached error	2.9879e-08	4.1165e-07	9.2784e-06	5.2916e-05

Figure 3 shows the iterative results of the four benchmark functions. The left side named (a) is the test error curves of the above four functions. From the curves, we know that the test error for each iteration is small. For the unimodal functions, the order of magnitude of the maximum error is $1e-05$, and for the multimodal function, the order of magnitude of the maximum error is $1e-02$. Meanwhile, the right side named (b) is the fitting effect diagrams of the four functions, showing the fitting effects of each function for 50 test points. As can be seen, although the fitting effects of the multimodal functions are not perfect in some points, the fitting effects of the four functions all have reached good levels.

3.2. Resonant Frequency of MSAs

3.2.1. Rectangular MSA. Figure 4 is a schematic diagram of the rectangular MSA [25] that is composed of the radiation element, the medium layer, and the reference ground. The width of the rectangular patch is W , the length is L , the thickness of the dielectric layer is h , and the relative dielectric constant is ϵ_r .

The design variables are W , L , h , and ϵ_r , and the resonant frequency points are measured by Mehmet Kara [26]. There are 33 sets of data in total shown in Table 2. When selecting

the training sample set, we should consider the information on each dimension and make sure that the samples in each dimension are uniformly dispersed. Then, 13 sets of data with suffix # are used as the initial training samples, and 10 sets of data with suffix * are used as test samples. The label information of 10 sets of data with suffix ★ are removed firstly and then used as unlabeled samples because we have no unlabeled sample in the case.

In initial training, we set the four design variables with W , L , h , and ϵ_r as the input variables of the GP model and the resonant frequency as the output of GP model; then, we establish the initial GP model by using the training samples in Table 2. The test samples are used to obtain the initial error, and the result of the mean RE is 0.0093.

In each iteration, GP_1 and GP_2 select, respectively, one different unlabeled sample for cross-training and take one test sample for verification. We set the iteration termination condition, that is, the test error has met the error threshold and the error of the latter iteration is worse than that of this iteration simultaneously. If it satisfies the condition, the program stops at the latter iteration. In the case, the error threshold is $1e-05$, and the smallest test error is $3.4683e-6$ at the 5th iteration. Therefore, we stop the iteration at the 6th iteration. Table 3 shows the test errors of the six iterations, and Figure 5 shows the error curves.

(a)

FIGURE 3: Continued.

FIGURE 3: Error and fitting curves of the four benchmark functions. (a) Sphere function. (A) Error curve. (B) Fitting figure. (b) Sum squares function. (A) Error curve. (B) Fitting figure. (c) Rastrigin function. (A) Error curve. (B) Fitting figure. (d) Schwefel function. (A) Error curve. (B) Fitting figure.

FIGURE 4: Schematic diagram of rectangular MSA.

From the above results, except for unlabeled samples for each iteration, the optimal model has been further trained with four more test samples. At the same time, the four test samples are added into the original training sample set. We train a traditional GP model with the updated training sample set. The 5th test sample is used to test the traditional GP model, and the result is 7.7814×10^{-4} , larger than 3.4683×10^{-6} . We can preliminarily consider that the SSL-based GP model has advantages over the traditional supervised GP model.

Considering the effects of different test samples on the error, we use the above trained SSL-based GP model to predict the 5th to 10th test samples, and the predicted results for these six test samples are shown in bold and underlined, in column 7 named $\hat{f}_{\text{proposed}}$ in Table 2. We also use the above traditional GP model to predict the 5th to 10th test samples for comparing. Here, we use Mean Relative Error (MRE) to evaluate the performance of the model. The MRE is given by

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_{\text{pred}} - y_{\text{test}}|}{y_{\text{test}}} \quad (15)$$

For the rectangular MSA, the MRE of the proposed SSL-based GP model is 0.0075, while the MRE of traditional supervised GP model is 0.0081. We may conclude that, with

the same training samples and the same test samples, the performance of the SSL-based GP model is better than traditional supervised GP model, and the test error is smaller and the accuracy is improved.

3.2.2. Circular MSA. As can be seen from Figure 6, replacing the rectangular patch with the circular patch is a schematic diagram of the circular MSA. The radius of the circular patch is a , the thickness of the dielectric layer is h , and the relative dielectric constant is ϵ_r . Its resonant frequency is measured by Sing [27] and Seref Sagioglu [28].

The design variables are a , h , and ϵ_r , as shown in Table 4, and totally there are 20 sets of data. As described above, considering the information on each dimension, 8 sets of data with suffix * are used as test samples, and their label information is removed firstly; then, they are used as unlabeled samples. The remaining samples are used as training samples.

In initial training, the same as above, we get the initial error is 0.0807, which is large and the GP model needs further improvement. In this case, the error threshold is 1×10^{-2} , and the smallest test error is 0.0071 at the 7th iteration. Therefore, the program stops at the 8th iteration. Table 5

TABLE 2: Resonant frequency of the rectangular MSA in TM₁₀ mode.

Number	W (cm)	L (cm)	H (cm)	ϵ_r	f_{measured}	f_{proposed}
1*	1.256	2.756	0.952	2.55	3580	3522
2*	0.883	2.676	1.000	2.55	3980	3979.6
3#	2.000	2.500	0.079	2.22	3970	3970.4
4#	1.063	1.183	0.079	2.25	7730	7730
5#	0.910	1.000	0.127	10.20	4600	4600.1
6#	1.720	1.860	0.157	2.33	5060	5059.4
7*	1.810	1.960	0.157	2.33	4805	4848.7
8#	1.270	1.350	0.163	2.55	6560	6559.7
9★	1.500	1.621	0.163	2.55	5600	5569.6
10*	1.375	1.580	0.476	2.55	5100	5114
11★	1.120	1.200	0.242	2.55	7050	7101.5
12★	1.403	1.485	0.252	2.55	5800	5799.5
13#	1.530	1.630	0.300	2.50	5270	5269.9
14★	0.905	1.018	0.300	2.50	7990	8036.1
15★	1.170	1.280	0.300	2.50	6570	6632.3
16*	1.030	3.380	1.281	2.55	3200	3200.1
17#	0.776	1.080	0.330	2.55	8000	7998.2
18*	1.000	1.520	0.476	2.55	5820	5804.1
19#	0.987	1.450	0.450	2.55	6070	6072.7
20*	1.337	1.412	0.200	2.55	6200	6196.7
21★	0.814	1.440	0.476	2.55	6380	6404.6
22★	0.790	1.620	0.550	2.55	5990	5848.2
23★	1.200	1.970	0.626	2.55	4660	4444
24#	0.783	2.300	0.854	2.55	4600	4599.3
25*	0.790	1.255	0.400	2.55	7134	7203.8
26#	0.974	2.620	0.952	2.55	3980	3979.6
27★	1.020	2.640	0.952	2.55	3900	3900.3
28*	0.790	1.185	0.017	2.22	8450	8293.9
29★	0.777	2.835	1.100	2.55	3900	3971.0
30#	0.920	3.130	1.200	2.55	3470	3472.4
31*	0.850	1.290	0.017	2.22	7740	7846
32#	1.265	3.500	1.281	2.55	2980	2981.5
33#	1.080	3.400	1.281	2.55	3150	3147.1

TABLE 3: Test error result of the rectangular MSA.

Iteration number	1	2	3	4	5	6
Test error	0.1760	0.0103	0.0096	0.0013	3.4683e-06	0.0027

FIGURE 5: Test error curve of the rectangular MSA.

FIGURE 6: Structure diagram of circular MSA.

TABLE 4: Resonant frequency of circular MSA in TM_{11} mode.

Number	a (cm)	h (cm)	ϵ_r	f_{measured}	f_{proposed}
1	6.800	0.08000	2.32	835	834.96
2	3.800	0.15240	2.49	1443	1443.7
3*	0.770	0.23500	4.55	4945	4884.2
4	1.270	0.07940	2.59	4070	4073.1
5	3.493	0.31750	2.50	1510	1509.1
6*	3.493	0.15880	2.50	1570	1441.6
7	6.800	0.31800	2.32	815	815.18
8*	4.850	0.31800	2.52	1099	1198.3
9	4.950	0.23500	4.55	825	825.11
10*	2.000	0.23500	4.55	2003	2043.9
11	0.820	0.15875	2.65	6074	6073.2
12	3.975	0.23500	4.55	1030	1030
13*	1.150	0.15875	2.65	4425	4430.3
14	2.990	0.23500	4.55	1360	1360
15*	1.070	0.15875	2.65	4723	4866.6
16	1.040	0.23500	4.55	3750	3750
17*	6.800	0.15900	2.32	829	823.076
18	0.960	0.15875	2.65	5524	5521.2
19*	0.740	0.15875	2.65	6634	6468.8
20	5.000	0.15900	2.32	1128	1128

TABLE 5: Test error result of circular MSA.

Iteration number	1	2	3	4	5	6	7	8
Test error	0.1430	0.1323	0.0971	0.0624	0.0688	0.0415	0.0071	0.0380

shows the test errors of eight iterations, and Figure 7 shows the error curve.

From the above results, the GP model has been further trained with six more test samples. Same as above, these six test samples are added to the original training sample set to train a traditional supervised GP model. The 7th test sample is used for comparison, and the result is 0.0073, which is larger than 0.0071.

We use the above trained GP model to predict the 7th to 8th test samples, and the predicted results are shown in bold and underlined, in column 7 named f_{proposed} in Table 4. The MRE of the SSL-based GP model is 0.0160, while that of the traditional GP model is 0.0240, which means the proposed GP model is better. In conclusion, for the resonant frequency of circular MSA, the performance of the proposed SSL-based GP model is better than that of the traditional GP model.

3.2.3. Triangular MSA. As can be seen from Figure 8, we replaced the rectangular patch with the triangular patch and obtained schematic diagram of the triangular MSA. The length of equilateral triangle is s , the height is h , and the relative dielectric constant is ϵ_r . The antenna has five different modes of operation. The design variables are $mode$, s , h , and ϵ_r , and its resonant frequency is measured by Chen [29] and Danele [30].

As described above, considering the information on each dimension, 5 sets of samples with suffix * are used as the test samples in Table 6. At the same time, their label information was removed because we want them to be as unlabeled samples. The remaining samples are training samples.

In the initial training process, the error is 0.1428. Due to less training samples, the initial error is large, which needs to be further improved. In this case, the error threshold is

FIGURE 7: Test error curve of circular MSA.

FIGURE 8: Structure diagram of triangular MSA.

TABLE 6: Resonant frequency of triangular MSA.

Number	Mode	S (cm)	H (cm)	ϵ_r	f_{measured}	f_{proposed}
1	TM ₁₀	4.1	0.070	10.5	1519	1519.1
2*	TM ₁₀ *	8.7	0.078	2.32	1489	1505.3
3	TM ₁₀	10	0.159	2.32	1280	1280.6
4*	TM ₁₁ *	4.1	0.070	10.5	2637	2511
5	TM ₁₁	8.7	0.078	2.32	2596	2596
6	TM ₁₁	10	0.159	2.32	2242	2241.1
7	TM ₃₀	4.1	0.070	10.5	4439	4438.8
8	TM ₃₀	8.7	0.078	2.32	4443	4442.9
9*	TM ₃₀ *	10	0.159	2.32	3824	3884.8
10	TM ₂₀	4.1	0.070	10.5	2995	2995
11*	TM ₂₀ *	8.7	0.078	2.32	2969	2949.1
12	TM ₂₀	10	0.159	2.32	2550	2550.8
13	TM ₂₁	4.1	0.070	10.5	3973	3973
14*	TM ₂₁ *	8.7	0.078	2.32	3968	3840.8
15	TM ₂₁	10	0.159	2.32	3400	3399.6

1e-02, and the smallest test error is 0.0067 at the 4th iteration. Table 7 shows the test errors, and Figure 9 is the error curve. The optimal model has been further trained with three more test samples. Same as above, these three test samples are added to the original training sample set to train a

TABLE 7: Test error result of triangular MSA.

Iteration number	1	2	3	4	5
Test error	0.2569	0.2376	0.1140	0.0067	0.0550

FIGURE 9: Test error curve of triangular MSA.

traditional supervised GP model. The 4th test sample is used to test, and the result is 0.0781, which is larger than 0.0067.

We use the above trained SSL-based GP model to predict the 4th and 5th test samples, and the predicted results are shown in bold and underlined, in column 7 named f_{proposed} in Table 6. The MRE of the SSL-based GP model is 0.0194, while that of the traditional GP model is 0.0423. From the result, we can conclude that this error is smaller than the initial training error, and the accuracy of the model can be improved by using the proposed algorithm when the accuracy of initial model is not good enough. At the same time, the MRE of the SSL-based GP model is smaller than that of the traditional supervised GP model, which means the SSL-based GP model is better. In conclusion, for the resonant frequency of the triangular MSA, the performance of the proposed SSL-based GP model is better than that of the traditional supervised GP model.

3.2.4. Comparison with Other Algorithms. The three basic geometries including rectangular MSA, circular MSA, and triangular MSA lead to the development of fractal geometries for the design of multiband antennas. Many research studies have researched on the data of these three different resonant frequency points we used. Firstly, we compare the proposed algorithm in this study with NNs including the backpropagation (BP), the delta-bar-delta (DBD), and the extended delta-bae-delta (EDBD) in reference [26]. Different models are used to predict all the samples including training samples and test samples, and the total absolute errors are obtained. The predicted results of the proposed GP model are, respectively, shown in column 7 named f_{proposed} in Tables 2, 4, and 6 for different MSA. The comparison results are showed in Table 8. From the results of Table 8, for the

TABLE 8: Total absolute errors of three types of MSAs from different methods.

Types of MSAs	f_{proposed}	f_{EDBD} [26]	f_{DBD} [26]	f_{BP} [26]	[31]	[32]	[33]	[34]	[35]	[29]
Rectangular	1176.8	2392	2427	2372	13136	8468	—	—	—	—
Circular	672.46	462	727	922	—	—	965	1047	—	—
Triangular	353.3	272	622	729	—	—	—	—	590	408

rectangular MSA, the total absolute error is obviously smaller than the other algorithms. For the circular MSA and the triangular MSA, the proposed algorithm is better than DBD and BP, but a little worse than EDBD. In a word, the proposed SSL-based GP model uses fewer training samples, but has almost same or better predication ability than the NNs in [26]. However, as we all know, how to decide the structure of NN is difficult. Usually, it depends on the researcher's experience or adopts the trial and error method. For our proposed method, it has no problem and is very easy to model.

We also cite some results from other references for comparison, and the results also shown in Table 8. From references [31, 32], for the rectangular MSA, the proposed algorithm is better than the methods in references. From references [29–33], for the circular MSA and triangular MSA, we can get the same conclusion. In a word, using a few numbers of training samples, the proposed SSL-based GP model has better predication ability than these methods in the references.

4. Conclusion

In order to improve the optimal design efficiency of electromagnetic devices and save the time for collecting the training samples simulated by full wave electromagnetic software, this study proposes a semisupervised GP model, which covers initial training process and self-training process. In the initial training process, a few labeled samples are used to train the GP model with relatively low accuracy. In each iteration of self-training process, the trained GP model is copied to another GP model firstly, and the two GP models are further updated with unlabeled samples. After using the same test sample for testing, the GP model with small error is used to replace another GP model for self-updating. Repeat the self-training process until the error threshold is met. Four benchmark functions are used to test the effectiveness of the proposed algorithm. Experimental results show that both unimodal functions and multimodal functions can reach the expected error with fewer iterations. Meanwhile, the problems of resonant frequency of three different microstrip antennas are used to verify the effectiveness of the proposed GP model. Comparing with supervised GP model, the results show that the accuracy of the proposed semisupervised GP model is improved, and the error is smaller than that of the traditional supervised GP model. Comparing with other algorithms, the proposed GP model uses the fewer labeled samples, while the prediction ability has some advantages over other methods. In a word, the proposed semisupervised GP model in this study further promotes the research on the optimal design of electromagnetic devices.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under no.61771225, Postgraduate Research and Practice Innovation Program of Jiangsu Province China under no. SJCX19-0593, and Qinglan Project of Jiangsu Higher Education.

References

- [1] Y. B. Tian, *Particle Swarm Optimization and Application on Electromagnetics*, Science Press, Beijing, China, 2014.
- [2] 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," *Applied Computational Electromagnetics Society Journal*, vol. 29, no. 12, pp. 1025–1034, 2014.
- [3] 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 Transactions on Antennas and Propagation*, vol. 67, no. 5, pp. 3109–3116, 2019.
- [4] F. Y. Sun, Y. B. Tian, and Z. L. Ren, "Modeling the resonant frequency of compact microstrip antenna by the PSO-based SVM with the hybrid kernel function," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 29, no. 6, pp. 1129–1139, 2016.
- [5] M. Smieja, J. Tabor, and P. Spurek, "SVM with a neutral class," *Pattern Analysis and Applications*, vol. 22, no. 2, pp. 573–582, 2019.
- [6] Y. Gao and Y. B. Li, "Misfire identification of automobile engines based on wavelet packet and extreme learning machine," *Testing Science and Instruments*, vol. 4, pp. 384–395, 2017.
- [7] L.-Y. Xiao, W. Shao, X. Ding, and B.-Z. Wang, "Dynamic adjustment kernel extreme learning machine for microwave component design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 10, pp. 4452–4461, 2018.
- [8] Y. Chen, Y.-B. Tian, Z. Qiang, and L. Xu, "Optimisation of reflection coefficient of microstrip antennas based on KBNN exploiting GPR model," *IET Microwaves, Antennas & Propagation*, vol. 12, no. 4, pp. 602–606, 2018.
- [9] X. H. Fan, Y. B. Tian, and Y. Zhao, "Optimal design of microwave devices by fitness-estimation-based particle swarm optimization algorithm," *Applied Computational Electromagnetics Society Journal*, vol. 33, no. 11, pp. 1259–1267, 2018.

- [10] W. G. and K. Kaew, "A Gaussian process regression model for the traveling salesman problem," *Journal of Computer Science*, vol. 8, no. 10, pp. 1749–1758, 2012.
- [11] J. P. Jacobs and S. Koziel, "Reduced-cost microwave filter modeling using A two-stage Gaussian process regression approach," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 25, no. 5, pp. 453–462, 2015.
- [12] Z.-H. Zhou and M. Li, "Semi-supervised learning by disagreement," *Knowledge and Information Systems*, vol. 24, no. 3, pp. 415–439, 2010.
- [13] R. K. Sanodiya, J. Mathew, S. Saha, and M. D. Thalakkottur, "A new transfer learning algorithm in semi-supervised setting," *IEEE Access*, vol. 7, pp. 42956–42967, 2019.
- [14] Y.-F. Li and D.-M. Liang, "Safe semi-supervised learning: a brief introduction," *Frontiers of Computer Science*, vol. 13, no. 4, pp. 669–676, 2019.
- [15] K. Georgios and S. Kotsiantis, "Semi-supervised regression: a recent review," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 2, pp. 1483–1500, 2018.
- [16] X. Zhu, "Semi-supervised learning literature survey," *Computer Science*, vol. 37, no. 1, pp. 63–77, 2005.
- [17] M. L. Zhang and Z. H. Zhou, "Co-trade: confident co-training with data editing," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 6, pp. 1612–1626, 2011.
- [18] X. Guo and K. Uehara, "Graph-based semi-supervised regression and its extensions," *International Journal Advanced Computer Science Application*, vol. 6, no. 6, 2015.
- [19] X. H. Fan, Z. Y. Guo, and H. F. Ma, "An improved EM-based semi-supervised learning method," in *Proceedings of the International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing*, pp. 529–532, Shanghai, China, August 2009.
- [20] G. Fung and O. L. Mangasarian, "Semi-supervised support vector machines for unlabeled data classification," *Optimization Methods and Software*, vol. 15, no. 1, pp. 29–44, 2001.
- [21] P. Sun, L. Xie, and J. Hui, "Spatial batch optimal design based on self-learning Gaussian process models for LPCVD processes," *Chinese Journal of Chemical Engineering*, vol. 23, pp. 1958–1964, 2015.
- [22] D. Wu, M. Shang, X. Luo et al., "Self-training semi-supervised classification based on density peaks of data," *Neurocomputing*, vol. 275, pp. 180–191, 2018.
- [23] J. Zhang, S. Zeng, Y. Jiang, and X. Li, "A Gaussian process based method for antenna design optimization," *Communications in Computer and Information Science*, vol. 575, pp. 230–240, 2016.
- [24] Y. X. Xu, Y. B. Tian, and X. P. Hu, "Design of dual-frequency microstrip antenna based on Gaussian process of a new kernel function," *Computer Application Research*, vol. 35, no. 8, pp. 2477–2479, 2018.
- [25] M. Kara, "Closed-form expressions for the resonant frequency of rectangular microstrip antenna elements with thick substrates," *Microwave and Optical Technology Letters*, vol. 12, no. 3, pp. 131–136, 1996.
- [26] K. Guney, S. Sagiroglu, and M. Erler, "Generalized neural method to determine resonant frequencies of various microstrip antennas," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 12, no. 1, pp. 131–139, 2002.
- [27] J. Singh, A. Singh, and T. Kamal, "Design of circular microstrip antenna using artificial neural networks," in *Proceedings of the World Congress on Engineering*, pp. 1013–1016, London, UK, July 2011.
- [28] S. Sagiroglu, K. Guney, and M. Erler, "Resonant frequency calculation for circular microstrip antennas using artificial neural networks," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 8, no. 3, pp. 270–277, 1998.
- [29] W. Chen, K.-F. Lee, and J. S. Dahele, "Theoretical and experimental studies of the resonant frequencies of the equilateral triangular microstrip antenna," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 10, pp. 1253–1256, 1992.
- [30] J. Dahele and K. F. Kai Lee, "On the resonant frequencies of the triangular patch antenna," *IEEE Transactions on Antennas and Propagation*, vol. 35, no. 1, pp. 100–101, 1987.
- [31] J. Q. Howell, "Microstrip antennas," *IEEE Transaction on Antennas Propagation*, vol. 23, pp. 90–93, 1975.
- [32] D. L. Sengupta, "Approximate expression for the resonant frequency of a rectangular patch antenna," *Electronics Letters*, vol. 19, no. 20, pp. 834–835, 1983.
- [33] K. R. Carver, *Practical Analytical Techniques for the Microstrip Antenna Technology*, pp. 7.1–7.20, New Mexico State University, Las Cruces, NM, USA, 1970.
- [34] K.-F. Lee and Z. Fan, "Cad formulas for resonant frequencies of tm₁₁ mode of circular patch antenna with or without superstrate," *Microwave and Optical Technology Letters*, vol. 7, no. 12, pp. 570–573, 1994.
- [35] K. Guney, "Comments on "On the resonant frequencies of microstrip antennas," *IEEE Transactions on Antennas and Propagation*, vol. 42, no. 9, pp. 1363–1365, 1994.