This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2020.3001743, IEEE Transactions on Antennas and Propagation

A Modified Efficient KNN Method for Antenna Optimization and Design

Liangze Cui, Yao Zhang, *Student Member*, *IEEE*, Runren Zhang, and Qing Huo Liu*, *Fellow*, *IEEE*

***Abstract*—Effective machine learning methods are usually trained by large datasets to guarantee accuracy and avoid overfitting. However, large datasets restrict the popularization of machine learning methods when the data acquisition is nontrivial. To solve this problem, a novel machine learning method is proposed in this paper based on the modified K-Nearest Neighbors (KNN) algorithm, which can extract more features from datasets through advanced workflow and simulation techniques. In the applications presented here, our method is 5 to 30 times faster than traditional machine learning methods such as artificial neural network (ANN) and Bayesian optimization by reducing the required size of datasets. The proposed method is then employed to optimize the antenna parameters, while an additional branch is built to run the simulation tools (e.g. HFSS) and update the dataset during the training process instead of constructing the dataset beforehand. The validity and efficiency of this proposed method are confirmed by four different antenna examples and other machine learning and gradient-based algorithms. In summary, the proposed method can obtain a satisfactory optimal antenna design at little cost.**

***Index Terms*—Machine learning, K-Nearest Neighbors (KNN), antenna optimization.**

I. Introduction

Recently, more and more engineers start to improve and optimize their design with the help of machine learning methods to meet the requirements for high-performance antennas. Although well-trained models usually perform accurately and efficiently, their training process usually involves a great number of training samples and will pose a critical problem if the training samples acquisition is very time-consuming. Sometimes the cost of the training process in machine learning methods could be larger than the cost of other traditional optimization methods. For example, an ANN model [1] utilizing 12,800,000 data samples is trained to optimize a 10-layer photonic metamaterial, which contains ten variables for the thickness of the ten layers. In [2], neural networks with 600 training data samples are used to optimize a bandstop microstrip filter of five geometrical variables. In [3], an ANN model with 100 training data samples is used to optimize a Fabry-Perot (FP) resonator antenna of three geometrical variables. All these methods require nontrivial time to prepare the datasets, restricting their applications in real life. From this point of view, it is not a good choice to use machine learning methods when the cost of the training process is beyond our tolerance. It is hignly desirable to reduce the number of training datasets. While in the example given in part IV case C, our method only needs 8 training data and testing samples to optimize a 3-parameter resonator antenna; in the example we give in part IV case D, only 30 training and testing data samples are needed for a 9-parameter antenna model by using our method.

To decrease the complexity of the neural networks system and the number of training and testing data samples, a new method based on the modified K-Nearest Neighbors (KNN) algorithm is proposed. The model is not necessarily well-trained before it is used to predict the optimal results and therefore can start with very few training and testing data samples (at least five). The model can predict the optimal results during the training process. Then the predictions as well as exact results will be recorded back to original training dataset. To obtain the exact results during the training process, we build a branch to control the simulation tool (e.g. HFSS). Therefore, this method avoids the randomness of the training data selection and instead generates more valuable training data based on the existing but not well-prepared model. The model can evolve by predicting itself in a so-called self-learning manner.

Neighbors-based regression method can be used in cases where the data labels are continuous, which are the target geometrical parameters of the antennas in this paper. The label assigned to a query point is computed based on the mean of the labels of its (default is five) nearest neighbors. As little prior knowledge of antennas is available, we choose uniform weights in this case and each point in the local neighborhood then contributes uniformly to the regression of a query point. In sum, the only prior information we need to input is the reasonable domain of the geometrical parameters and the basic target antenna model to be optimized. Therefore, the algorithm can be easily generalized to other scenarios. To verify the universality and the effectiveness of the algorithm, four quite different cases will be tested and given later in this paper.

The rest of the paper is organized as follows. Section II describes the conventional nearest neighbor method and the proposed optimization scheme. Section III demonstrates the implementation of the proposed method. Section IV gives four different cases including the design for a Bragg reflector, the industry base-station antenna model, the antenna model optimized by other machine learning methods and the basic antenna model design process to test and verify the proposed algorithm. Finally, conclusions are summarized in section V.

L. Cui, R. Zhang and Q.-H. Liu are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA.

Y. Zhang is with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, 510641, China

II. ALGORITHM

The k-nearest neighbor algorithm in this paper is a supervised neighbors-based learning regression algorithm [4]. The principle behind the nearest neighbor algorithm is to find a predefined number of training samples closest to the new point. The algorithm uses 'feature similarity' to predict new data points, which means that the new point is assigned a value based on how closely it resembles the training set [5]. When using the KNN regressor, $\boldsymbol{x}$ represents the input vector, $\boldsymbol{y}$ represents the output vector and k denotes the number of neighbors used to perform the prediction. In this paper, we choose $k=5$ to balance accuracy and speed, although other values can also be used. For the weighting scheme in this paper, we choose a uniform weight, however, a weight of $1/d$ or some other weights can also be used, where d is the distance to the neighbor. Additionally, the distance we used in this paper is Euclidean distance. The neighbors are taken from the set of object values in the input data and serve as the training set for the algorithm [6].

We can implement a KNN method by following the steps:

1. Calculate the distance between the new point and each training point.
2. The closest k points are selected based on the distance.
3. The weighted average value of these selected data points serves as the final prediction for the new point.

Mathematically, assume a value k for KNN and a prediction point x_0, then use $\boldsymbol{N_0}$ to denote the k closest training observations to the prediction point x_0. KNN returns the estimation $f\left(x_0\right)$ using the average of all the responses in $\boldsymbol{N_0}$

$$\hat{f}\left(x_0\right)=\frac{1}{k}\sum_{x_i\in N_0} y_i \tag{1}$$

In this paper, we first generate a dense mesh and use the KNN regressor to calculate the value at each mesh point. The steps of the KNN implementation are detailed as follows.

1. Distance Calculation

There are many methods to calculate the distance in KNN. The most commonly used three methods are: Euclidian, Manhattan and Hamming distance

$$Euclidean\ distance=\sqrt{\sum_{i=1}^{k}\left(x_i-y_i\right)^2} \tag{2}$$

$$Manhattan\ distance=\sum_{i=1}^{k}\left|x_i-y_i\right| \tag{3}$$

$$Hamming\ distance=\sum_{i=1}^{k}D_i,\ D_i=\begin{cases}0 & if\ x_i=y_i\\ 1 & if\ x_i\neq y_i\end{cases} \tag{4}$$

The Hamming distance is usually used for classification. Here we can just choose the Euclidean distance.

The training process of the KNN algorithm shown in Fig. 1 is to compute the distance between all pairs of points in the dataset and store them efficiently. Usually, three effective methods, namely brute force, K-D tree and ball tree are employed to compute nearest neighbors [8]. According to [9, 10], the three methods should be applied in different situations.

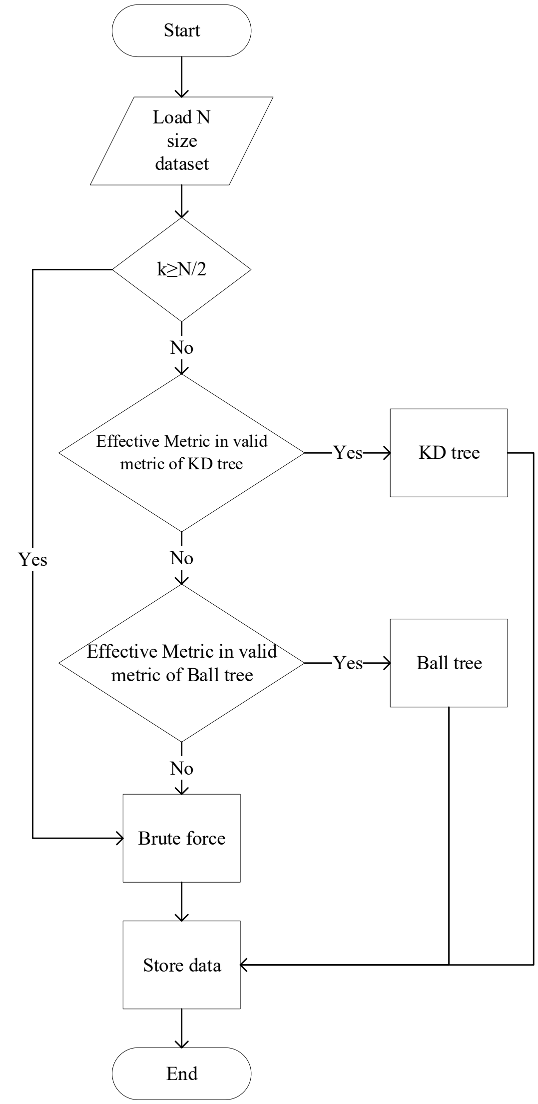


Fig. 1. KNN training process.

For the given the values of k and N (N is the number of data in the dataset), if $k \geq N/2$ or effective metric (an attribute of distance computation, which is Euclidean in our cases) is not in the valid metric list for either K-D tree or ball tree, brute force would be applied. Otherwise, the choice would base on the effective metric which belongs to the valid metric list for K-D tree or ball tree. In this paper, the effective metrics of our cases select K-D tree in the algorithm.

2. K Factor Selection

In general, the optimal value of k depends on the bias-variance tradeoff

$$\hat{f}_k\left(X\right)=\mathrm{E}_D\left[f_k\left(X\right)\right] \tag{5}$$

where X is the input data, Y is the ground truth of X, D is the dataset, $f_k\left(X\right)$ is the prediction based on the dataset and value k. $\hat{f}_k\left(X\right)$ is the expected prediction of the model.

Specifically, in our cases, the training and testing data come from analytical solution or numerical simulation tools, which are assumed as the exact solution. Therefore, the variance errors of our cases are very small. The variance is

$$Var(X) = \mathrm{E}_D\left[\left(f_k(X) - \hat{f}_k(X)\right)^2\right] \quad (6)$$

In other words, the k we use in our method does not have to be very large. And the bias is

$$bias^2(X) = \mathrm{E}_D\left[\left(Y - \hat{f}_k(X)\right)^2\right] \quad (7)$$

The expected prediction error in the system is (ignore the noise of dataset)

$$\begin{aligned} EPE &= \mathrm{E}_D\left[\left(Y - f_k(X)\right)^2\right] \\ &= Var(X) + bias^2(X) \\ &\approx bias^2(X) \\ &= \mathrm{E}_D\left[\left(Y - \hat{f}_k(X)\right)^2\right] \end{aligned} \quad (8)$$

According to [4] and [7], in most situations, we should use $k \leq 10$ to avoid overfitting. In this paper, we choose $k = 5$ to balance the accuracy and the speed from experience.

3. Weight Selection

Indeed, only the k nearest neighbors influence the prediction; however, this influence can be different for each of these neighbors. To achieve this influence, the distances have to be transformed into some measures, which can then be used as weights. The transition from distances to weights should follow some kernel functions. Typical examples for this kind of function include rectangular kernel, triangular kernel, cosine kernel, Gauss kernel and inversion kernel. Empirically, the choice of a special kernel (apart from the special case of the rectangular kernel that gives equal weights to all neighbors) is not crucial, especially for the antenna cases in this paper, since little prior information is exposed. Therefore, we chose a uniform weight (rectangular kernel) in this paper to avoid penalizing any neighbor [7].

III. Implementation of Modified Efficient KNN Method

In this paper, we used scikit-learn version 0.21.2 tool [6] to implement the modified efficient KNN method, MATLAB_R2018b to compute the analytical solution and ANSYS 18.2.0 (HFSS) to obtain the simulation for numerical solution. All computations in this paper were performed on a 2.3 GHz Intel Core i5 machine with 8 GB RAM.

The whole process of the proposed modified efficient KNN method is shown in Fig. 2. The implementation of the method is demonstrated in the following steps:

Step 1: Construct a dataset for the initial start. This is the most important step for this method. We do not need to choose too many samples as conventional machine learning methods in this step. Empirically, only 10~100 samples are required in this step for most situations. Additionally, the samples should be chosen uniformly in the range of the variables. More details of dataset are shown in the later specific examples.

Step 2: Randomly divide the original dataset into two parts: 90% for training and 10% for testing. The preprocessing StandardScaler tool is used to standardize dataset and the KNeighborsRegressor tool is employed to train and test the KNN model. In this step, we obtain the model Ω: $x \rightarrow f(x)$ for KNN.

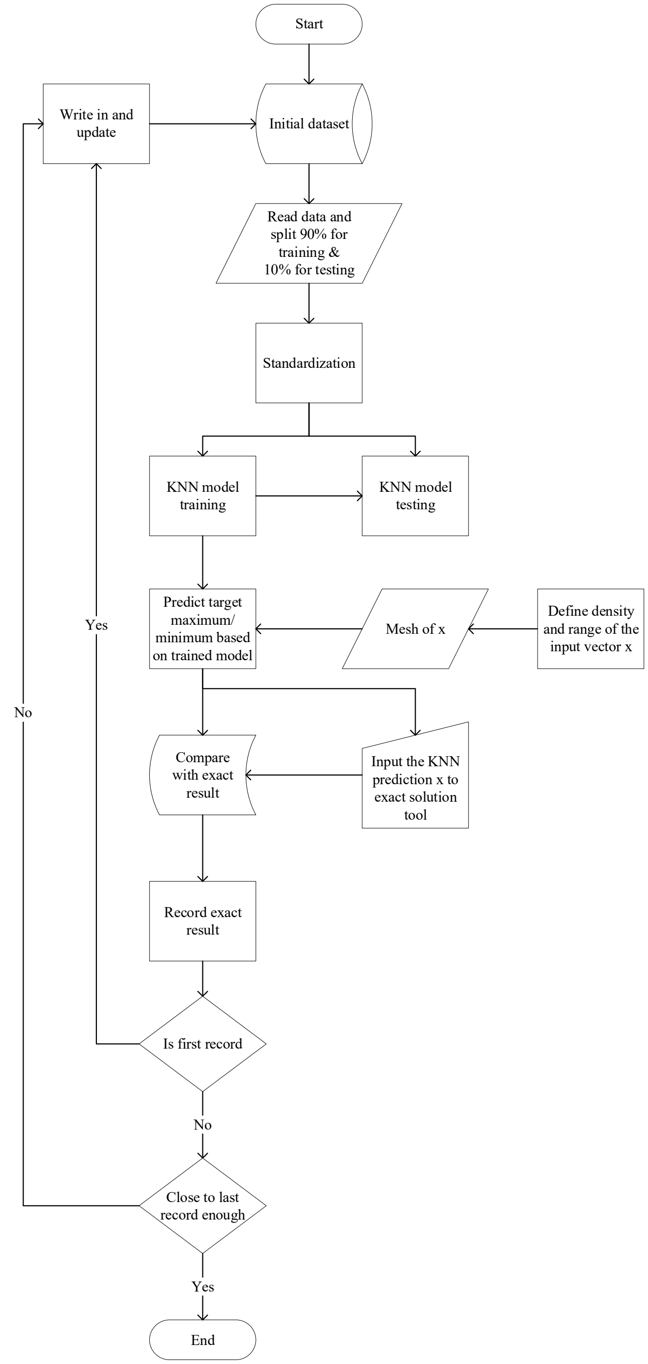


Fig. 2. Whole process of the modified efficient KNN method.

Step 3: Generate a mesh for input vector x (use all target parameters to construct the vector x). The density of the mesh should guarantee the accuracy of the variables (usually 0.01). And the mesh should not offer point groups which have the same K nearest neighbors to avoid an ambiguous output. The domain of the variables should also be set at the beginning. If the loop of the algorithm stops too fast to obtain a satisfied result, increasing the density of the mesh should help.

Step 4: Input the mesh from step 3 to the model Ω from step 2 and use the KNN regression to predict each value on the mesh. Obtain the maximum or minimum among the predictions based

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2020.3001743, IEEE Transactions on Antennas and Propagation

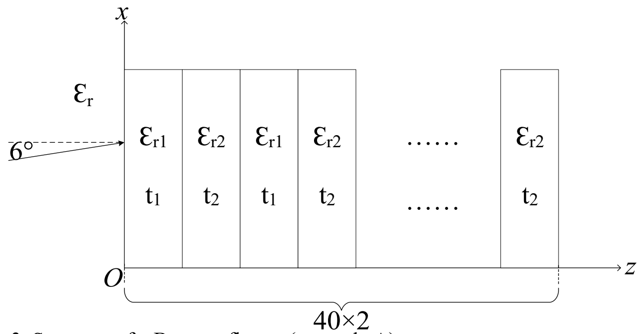


Fig. 3. Structure of a Bragg reflector (example A).

on the requirement for the extreme value. Record the corresponding value of vector $\boldsymbol{x}$ and then export its value to the solver for its exact solution (either analytically through MATLAB or numerically through solvers such as HFSS). Record the vector $\boldsymbol{x}$ and exact solution into the initial dataset to update the dataset.

Step 5: If the exact solution is equal to the last record within the allowed error in the nth iteration, as $|f(x_n)\text{-}f(x_{n-1})| < |\varepsilon f(x_n)|$, stop the loop, because the algorithm cannot predict a better value in this case. If the exact solution is not equal to the last record or if it is the first iteration, repeat steps 1 to 4. In case of an infinite loop, we set a counter to stop the loop at the 500th iteration, but the loop always stops before this maximum number in our experiments so far.

It is noteworthy that the most time-consuming step in this algorithm is step 4 because of the expensive numerical simulation process. In other words, the efficiency of the optimization algorithm usually depends on the time it needs to run the simulation tool. This is also the reason to reduce the number of samples in this work.

IV. Application Examples

In this section, four different examples will be presented including a Bragg reflector, a base-station antenna model, an antenna model optimized by other machine learning methods, and a basic antenna model design process. In case A, we choose a simple problem which has analytical solution to verify the effectiveness of the proposed method. In case B, we optimize an antenna for base-station [11] to verify our method in real industry work. In case C, we compare the efficiency of this proposed method with an ANN, a Bayesian optimization and nonlinear conjugate gradient (CG) to optimize a resonator antenna [3]. In case D, we design a completely new antenna based on some basic antenna models and knowledge. In case B, C and D, HFSS is used to obtain the simulation results.

A. Comparison with analytical solution for a Bragg reflector

In this case, a simple layered medium with analytical solutions is implemented to verify the effectiveness of our modified KNN efficient method, because we can easily obtain the optimal value of this problem and construct the data samples as much as we want.

Consider a plane wave impinging a Bragg reflector for the EUV lithography (shown in Fig. 3), where the wavelength in vacuum is λ =13.5 nm and the incident angle is 6° [12]. The Bragg reflector includes 40 periodic bilayer materials with relative permittivity ε_{r1} = 0.998-0.00363j and ε_{r2} =


Fig. 4. Photograph of the dipole element (example B).

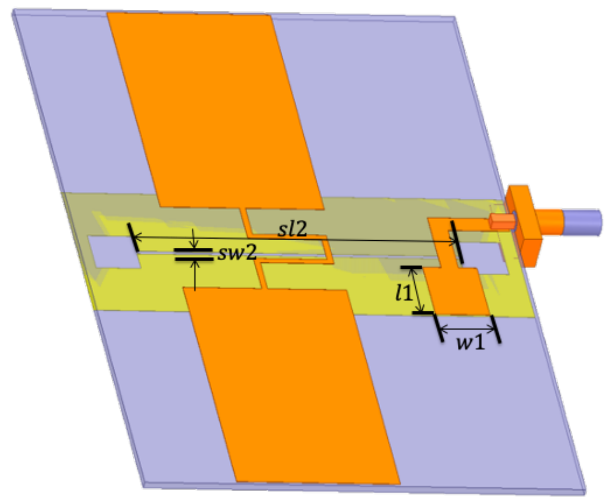


Fig. 5. Geometry of the dipole element (example B).

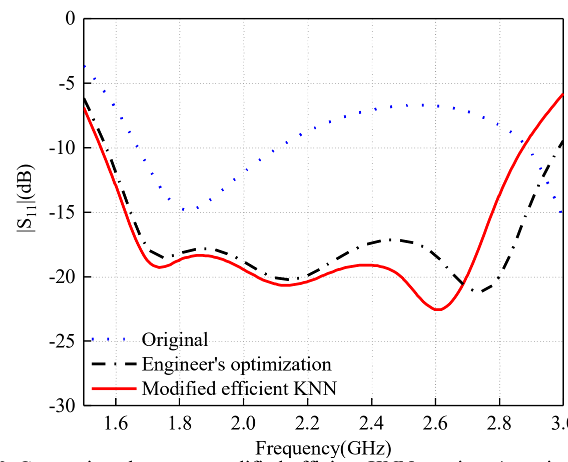


Fig. 6. Comparison between modified efficient KNN, engineer's optimization and the original model (example B).

0.854-0.0119j. The optimization target is to find the optimal thickness of material 1 and material 2 (t_1 and t_2) to achieve the maximum reflectance for both TE and TM waves, where t_1 and t_2 should be bounded between 2 nm and 5 nm.

To solve this problem, we define a total reflectance $r = r_{TE} + r_{TM}$, where r_{TE} is the reflectance of the TE wave and r_{TM} is the reflectance of the TM wave. According to [13], we calculate the analytical solutions of this problem. By using MATLAB to traverse all 372100 samples between 2 nm and 5 nm for the input vector $x = [t_1, t_2]^T$, or using a function comparison based optimization method [14], we can obtain the optimal result r = 1.4519, r_{TE} = 0.7297 and r_{TM} = 0.7222, where t_1 =4.13 nm, t_2 = 2.82 nm.

In order to use the proposed method, we first generate a small dataset with 25 samples [15]. The optimization stops at the 40th

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2020.3001743, IEEE Transactions on Antennas and Propagation

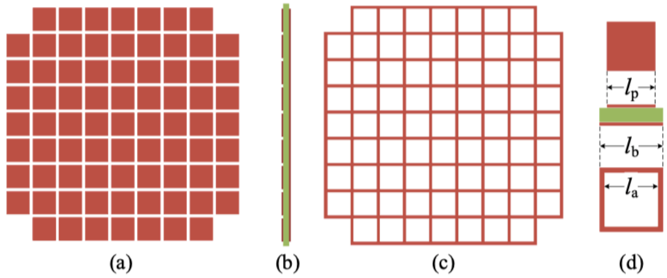


Fig. 7. EBG structure of the Fabry-Perot resonator antenna (example C): (a) top view, (b) side view, (c) bottom view, and (d) unit cell.

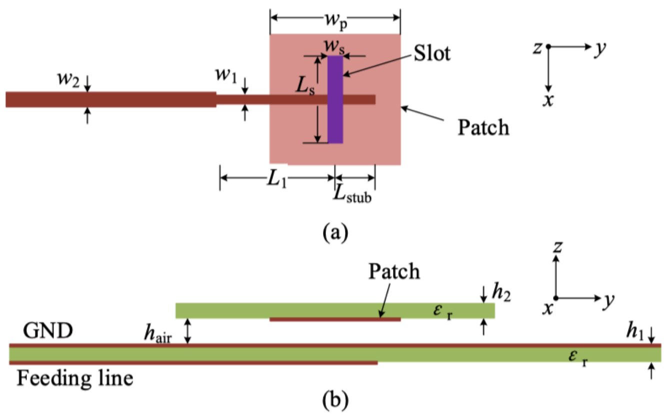


Fig. 8. Structure of the feeding antenna (example C): (a) top view and (b) side view.

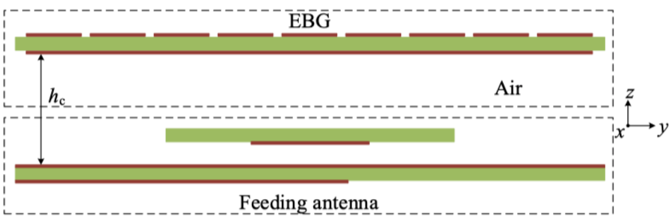


Fig. 9. Structure of the Fabry-Perot resonator antenna (example C).

TABLE I
DEFINITION OF TRAINING AND TESTING DATA FOR THE FP RESONATOR ANTENNA.

Geometrical variables		Training data (64 samples)			Testing data (36 samples)		
		Min	Max	Step	Min	Max	Step
Case 1	l_p (mm)	5.6	6.3	0.1	5.65	6.15	0.1
	l_a (mm)	5.2	5.9	0.1	5.25	5.75	0.1
	h_c (mm)	14.5	15.55	0.15	14.575	15.325	0.15
Case 2	l_p (mm)	5.4	6.45	0.15	5.475	6.225	0.15
	l_a (mm)	5	6.05	0.15	5.075	5.825	0.15
	h_c (mm)	14.25	16	0.25	14.375	15.625	0.25

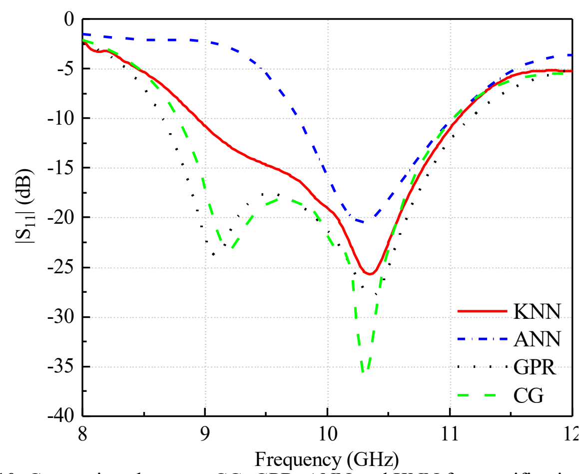


Fig. 10. Comparison between CG, GPR, ANN and KNN for specification 1 (example C).

iteration and gives the final results $r = 1.4519$, $r_{TE} = 0.7297$ and $r_{TM} = 0.7222$, where t_1 =4.14 nm, t_2 = 2.81 nm.

This case shows that the results of the proposed KNN method conform to the analytical solution.

B. Comparison with an well designed base-station antenna

In this case, we optimize an antenna for base-station to test the effectiveness of our method in a real industry case, because to help engineer optimize the real engineering design is the purpose of our method.

In this paper, we simplify a broadband horizontally polarized omnidirectional antenna array [11] to consider a dipole element among the six omnidirectional antenna units. The detail of the design parameters of the dipole element can be found in [11]. Fig. 4 and Fig. 5 show the photograph and geometry of the dipole element. We use the target variables to construct the input vector $x = [w1, l1, sw2, sl2]^T$ (unit: mm). And [11] gives the optimal value x = $[5.8, 7.0, 0.4, 34.5]^T$ optimized by experienced engineers. For the modified efficient KNN algorithm, we set the range for each parameter firstly, where $w1$: 5.5~6.5, $l1$: 6.5-7.5, $sw2$: 0.2~0.6, $sl2$: 34~35(unit: mm). By initializing the dataset with 81 samples [16], the loop would stop at the 122th iteration. The optimal vector from the modified efficient KNN method is $x' = [6.0, 7.5, 0.4, 35.0]^T$. The working frequency range of this antenna is 1.69 ~ 2.69 GHz. The objective is the minimal value of $|S_{11}|$ curve and the optimization purpose is to minimize the $|S_{11}|$ curve in the working frequency band.

Fig. 6 compares the corresponding S_{11} curve of the parameters optimized by our modified efficient KNN method ($x' = [6.0, 7.5, 0.4, 35.0]^T$) and by engineers ($x = [5.8, 7.0, 0.4, 34.5]^T$), as well as the original values before optimization.

The curve from the modified efficient KNN in the working frequency range is lower than those under the engineer's optimization and original value, indicating that our proposed algorithm can enhance and replace the human design to some extent. The effect of our method is even better than experienced engineers according to the results.

C. Comparison with other machine learning methods

In this case, we compare our method with other published machine learning methods like ANN and Bayesian optimization and conventional nonlinear CG method based on a published antenna. The results show that our method only use about 10% as many data samples as the ANN, Bayesian optimization and CG to obtain a satisfactory S_{11} curve.

According to [3], the authors applied an ANN into a Fabry-Perot resonator antenna proposed in [17]. The EBG structure of the antenna, as shown in Fig. 7, is a combination of two complementary frequency selective surface structures. At each corner of the rectangular array, one unit cell is eliminated. l_b = 8 nm. The feeding structure is an integral part of the Fabry-Perot resonator antenna, which is shown in Fig. 8. The

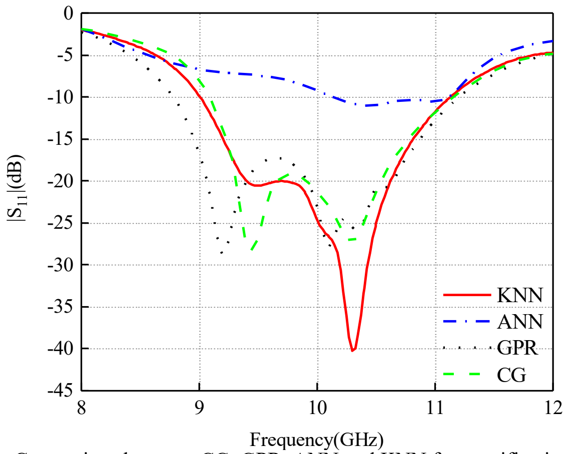


Fig. 11. Comparison between CG, GPR, ANN and KNN for specification 2 (example C).

TABLE II
FINAL OUTCOMES FOR DIFFERENT METHODS AND THEIRE ITERATION NUMBER FOR SPECIFICATION 1.

Method	l_p (mm)	l_a (mm)	h_c (mm)	Iteration
KNN	6.071	5.886	15.550	20
GPR	5.6423	6.05	16	94
ANN	5.904	5.189	14.748	100
CG	5.9177	5.7887	16.0094	658

TABLE III
FINAL OUTCOMES FOR DIFFERENT METHODS AND THEIRE ITERATION NUMBER FOR SPECIFICATION 2.

Method	l_p (mm)	l_a (mm)	h_c (mm)	Iteration
KNN	5.635	5.900	15.529	36
GPR	5.4	6.05	15.8693	316
ANN	6.401	6.011	14.733	100
CG	5.4841	5,7585	15.6338	344

parameters of the antenna are as follows: $w_p = 9.3$ mm, $w_1 = 1.2$ mm, $w_2 = 2.3$ mm, $w_s = 2.3$ mm, $L_1 = 9.5$ mm, $L_s = 8.2$ mm, $L_{stub} = 3$ mm, $h_{air} = 2.5$ mm and $h_1 = h_2 = 0.787$ mm.

The whole structure of the Fabry-Perot resonator antenna is shown in Fig. 9, where h_c is the cavity height between the EBG layer and the ground plane. We again use the target variables to construct the input vector $x = [l_p, l_a, h_c]^T$ (unit: mm). The objective is to minimize $|S_{11}|$ curve in the working frequency band, which is the same as the case in section B.

Ref. [3] applied the ANN to two different cases, a narrow parameter range case and a wide range case. There are 64 training data and 36 testing data samples needed (shown in Table I), which means that HFSS should run at least 100 times to generate the dataset in this case. For our KNN method, we only consider a wide range of parameters directly and fewer samples are needed. And the same initial dataset samples at the beginning are used to run our proposed KNN method and Bayesian optimization to present a fair comparison. Gaussian process regression (GPR) and probability of improvement are used as the surrogate model and the acquisition function for the Bayesian optimization [18], which implies that the KNN regressor was replaced by GPR in the process shown in Fig. 2.

We test the same two specifications as in [3], where specification 1 is at the frequency range of 8.75 ~ 11.25 GHz and specification 2 is at the frequency range of 10 ~ 11 GHz. To minimize the $|S_{11}|$ of this antenna, the GPR, ANN and KNN are trained to give the optimal results respectively.

For specification 1, only 15 samples are used for the modified efficient KNN method and GPR method as the training dataset [19] at the beginning. The optimal values from ANN are $x_{opt1,1}$ = $[5.904, 5.189, 14.748]^T$, while the optimal values from KNN are $x_{opt1,2}$ = $[6.071, 5.886, 15.550]^T$, the optimal values from GPR are $x_{opt1,3} = [5.6423, 6.05, 16]^T$, the optimal values from CG are $x_{opt1,4} = [5.9177, 5.7887, 16.0094]^T$. The comparison of the $|S_{11}|$ curve between them is shown in Fig. 10. Note that the optimization of our proposed method stops at the 20th iteration, which means that only 20 HFSS simulations are needed in total. The ANN needs 100 times according to [3]. The GPR stops at the 94th iteration, thus 94 HFSS simulations are needed in total. Therefore, the proposed KNN method needs about one fifth simulation times of GPR and ANN. The conventional CG needs 658 iterations which is almost 30 times as much as the modified efficient KNN method.

For specification 2, only 8 samples are used for the modified efficient KNN method and GPR method as the training dataset [20] at the beginning. The optimal values from ANN are $x_{opt2,1}$ = $[6.401, 6.011, 14.733]^T$, while the optimal values from KNN are $x_{opt2,2}$ = $[5.635, 5.900, 15.529]^T$, the optimal values from GPR are $x_{opt2,3} = [5.4, 6.05, 15.8693]^T$, the optimal values from CG are $x_{opt2,4} = [5.4841, 5.7585, 15.6338]^T$. The comparison of the $|S_{11}|$ curve between them is shown in Fig. 11. Note that the optimizations of our proposed method stops at the 36th iteration, which means that only 36 times HFSS simulations are needed in total. The ANN needs 100 times according to [3]. The GPR stops at the 316th iteration, thus 316 times HFSS simulations are needed in total. And the conventional CG needs 344 times. Therefore, the proposed KNN method needs about one tenth of simulation in GPR and CG.

These two specifications demonstrate that the modified efficient KNN method is more effective than the ANN method, because the $|S_{11}|$ curve of the KNN is much lower than the curve of the ANN. Additionally, the $|S_{11}|$ curve of the KNN is close to the GPR and CG, so the results of KNN, GPR and CG are almost the same. In Tables II&III, we indicate the final outcomes and iteration number for specifications 1&2. To present a fair comparison we use the same initial dataset samples and stop criterion for KNN, GPR and CG, so they generate similar $|S_{11}|$ curves. The results of ANN come from [3], which used 100 simulation times. Fewer HFSS simulations are needed for the KNN than those for the ANN, GPR and CG. KNN only needs 20 iterations for specification 1 and 36 for specification 2, which are the smallest compared with other methods. Therefore, the proposed modified efficient KNN method is more efficient than these other methods.

D. *Design of a new antenna*

In the final case, our method is employed to design a new antenna with nine geometric parameters. The number of parameters is large so that this case can test the efficiency and scalability of our method properly. And we only used 30 training and testing data samples to design this model.

We design a new basic dipole antenna as shown in Fig.12. A ground with the dimensions of $(2 \times Gnda) \times (2 \times Gndb)$ is placed above the substrate. A dumbbell-shaped slot is etched in the middle of the ground. On the bottom side, a racquet-shaped microstrip feed line, two triangular stepped radiating strips which are connected by a microstrip line are assigned. Note that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2020.3001743, IEEE Transactions on Antennas and Propagation

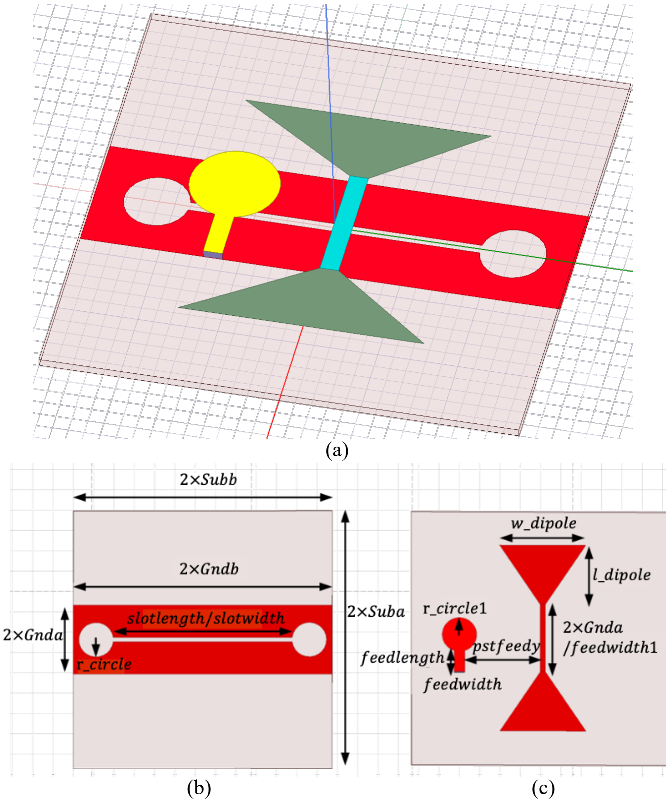


Fig. 12. Structure of the new antenna design (example D): (a) 3D model and geometry of the (b) top side and (c) bottom side.

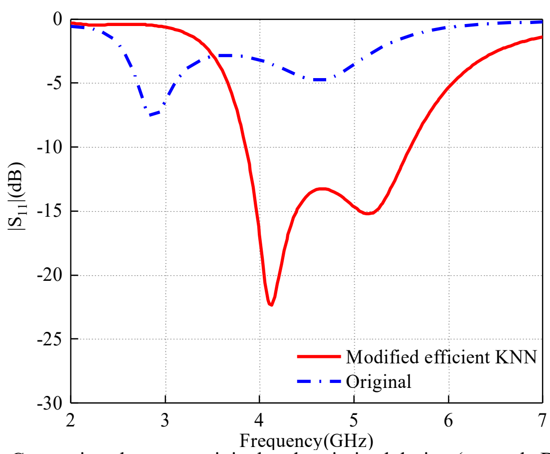


Fig. 13. Comparison between original and optimized design (example D).

the dumbbell-shaped slot is used to guide the signals from the feed line to radiators. The terminal of the racquet-shaped microstrip feed line is in the rectangular bar, and the terminals of the slotline are in the circle slot. As a result, the feed line and the slotline can be coupled from the slotline to the radiating strips. The fixed geometric parameters are: *Gnda* = 4 mm, *suba* = sub = *Gndb* = 15 mm, *feedwidth* = 1.2 mm and the thickness of the substrate *subh* = 0.508 mm.

The expected working frequency range of the antenna is 4 ~ 5 GHz with the design specification of $|S_{11}| < -10$dB. The nine target parameters for this antenna are employed to construct the input vector as x = [*slotlength*, *slotwidth*, *r_circle*, *l_dipole*, *w_dipole*, *feedwidth1*, *pstfeedy*, *r_circle1*, *feedlength*]^T (unit: mm). According to the antenna design experience, we constrain the target parameter domain as: *slotlength*: 15~25, *slotwidth*: 0.1~1.2 , *r_circle*: 1.8~2.8, *l_dipole*: 5~12, *w_dipole*: 7~20, *feedwidth1*: 0.2~1.5, *pstfeedy*: -12 ~ -5, *r_circle*: 1.8~2.8, *feedlength*: 2~4 (unit: mm). Thus, there are 30 samples [21] in the initial dataset and the optimization process would stop at the 58th iteration by using modified efficient KNN method. The best S_{11} curve from the original dataset and the optimized value from the modified efficient KNN method are shown in Fig. 13, where the best original vector is x = [21.90, 0.86, 2.49, 9.83, 15.97, 1.10, -7.17, 2.49, 3.38]^T and the optimal vector from modified efficient KNN is x' = [15.00, 0.93, 2.55, 6.17, 9.89, 1.50, -6.17, 2.30, 3.50]^T. The corresponding $|S_{11}|$ curves shown in Fig. 13 shows a very good performance.

It is thus demonstrated that the proposed modified efficient KNN method is effective and efficient in the new antenna design and optimization process.

V. Conclusion

A novel modified efficient KNN method has been proposed in this paper to help design and optimize antennas with a limited number of training and testing data samples. The model only requires very few samples (only 10~100 samples for cases below ten parameters) for the dataset and some prior information at the beginning to constrain the target domain; then it can self-learn and rapidly predict the optimal value with some simulation tools rapidly. The efficiency of the proposed method is very high because it can generate more valuable data samples during training process. Four examples are given in this paper to validate this proposed method and demonstrate its excellent efficiency compared to other machine learning methods or traditional engineering methods, with a speedup factor of 5 to 30 times. The future work could focus on the multiple target optimization, because generally the more targets and parameters to optimize, the more data samples we need. The goal is to use limited training and testing data samples to solve more antenna design problems.

Acknowledgment

The authors wish to express their gratitude to the editors and reviewers of this manuscript.

References

[1] Y. Chen, J. Zhu, Y. Xie, N. Feng, and Q. H. Liu, "Smart inverse design of graphene-based photonic metamaterials by an adaptive artificial neural network," Nanoscale, vol. 11, no. 19, pp. 9749-9755, May 16 2019.

[2] Y. Cao, G. Wang, and Q.-J. Zhang, "A new training approach for parametric modeling of microwave passive components using combined neural networks and transfer functions," *IEEE. Trans. Microw. Theory Techn.*, vol. 57, no. 11, pp. 2727-2742, Nov. 2009.

[3] L.-Y. Xiao, W. Shao, F.-L. Jin, and B.-Z. Wang, "Multiparameter modeling with ANN for antenna design," *IEEE Trans. Antennas Propag.*, vol. 66, no. 7, pp. 3718-3723, Apr. 2018.

[4] T. Srivastava. (2014, Oct.). Introduction to k-nearest neighbors: a powerful machine learning algorithm. *Analytics Vidhya*. [Online]. Available: https: //www.analyticsvidhya.com /blog/2018/03/introduction-k-neighbours-algorithm-clustering.

[5] J. Ortiz-Bejar, M. Graff, E. S. Tellez, J. Ortiz-Bejar and J. C. Jacobo, "k-nearest neighbor regressors optimized by using random search," *2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, Ixtapa, Mexico, 2018, pp. 1-5.

[6] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: machine learning in python," *Journal of machine learning research*, vol. 12, pp. 2825-2830, 2011.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAP.2020.3001743, IEEE Transactions on Antennas and Propagation

[7] K. Hechenbichler, K. Schliep. (2014). Weighted k-nearest-neighbor techniques and ordinal classification. [Online]. Available: https://doi.org/10.5282/ubm/epub.1769

[8] J. Goldberger, S. Roweis, G. Hinton and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information*, vol. 17, 2005, pp. 513-520.

[9] J. Vanderplas, F. Pedregosa and A. Gramfort. (2011, Sep.). Base and mixin classes for nearest neighbors. *GitHub*. [Online]. Available: https://github.com/scikit-learn/scikit-learn/blob/7813f7efb/sklearn/neighbors/base.py#L858

[10] J. Vanderplas, F. Pedregosa and A. Gramfort. (2017). Source code for sklearn.neighbors.base. *tslearn*. [Online]. Available:https://tslearn.readthedocs.io/en/latest/_modules/sklearn/neighbors/base.html

[11] L.-H. Ye, Y. Zhang, X.-Y. Zhang, and Q. Xue, "Broadband horizontally polarized omnidirectional antenna array for base-station applications," *IEEE Trans. Antennas Propag.*, vol. 67, no. 4, pp. 2792-2797, Apr. 2019.

[12] J. Niu, Y. Ren, Q. H. Liu, "Spectral element boundary integral method with periodic layered medium dyadic Green's function for multiscale nano-optical scattering analysis", *Opt. Express*, vol. 25, no. 20, pp. 24199-24214, Oct. 2017.

[13] C. A. Balanis, "Reflection and transmission" in *Advanced Engineering Electromagnetics*, 1989, pp. 180-253.

[14] R. Zhang, Q. Sun, M. Zhuang, W.F. Huang, Q. Zhan, D. Wang and Q.H. Liu, "Optimization of the Periodic PML for SEM," *IEEE Trans. Electromagn. Compat.* 2018

[15] L. Cui, ModifiedKNNModel, 2019 [Dataset]. Avaliable: https://github.com/cuiliangze/ModifiedKNNModel/blob/master/MultiRdata25.txt

[16] L. Cui, ModifiedKNNModel, 2019 [Dataset]. Avaliable: https://github.com/cuiliangze/ModifiedKNNModel/blob/master/S11Max81.txt

[17] N. Wang, Q. Liu, C. Wu, L. Talbi, Q. Zeng, and J. Xu, "Wideband Fabry-Perot resonator antenna with two complementary FSS layers," *IEEE Trans. Antennas Propag.*, vol. 62, no. 5, pp. 2463–2471, May. 2014.

[18] E. Brochu, V. Cora and N. Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, Dec. 2010.

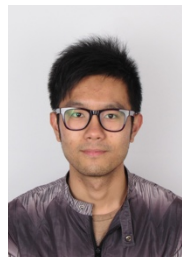
[19] L. Cui, ModifiedKNNModel, 2019 [Dataset]. Avaliable: https://github.com/cuiliangze/ModifiedKNNModel/blob/master/S11MaxModelXiao-N-15.txt

[20] L. Cui, ModifiedKNNModel, 2019 [Dataset]. Avaliable: https://github.com/cuiliangze/ModifiedKNNModel/blob/master/S11MaxModelXiao-2-8.txt

[21] L. Cui, ModifiedKNNModel, 2019 [Dataset]. Avaliable: https://github.com/cuiliangze/ModifiedKNNModel/blob/master/S11MaxModel1-N-30.txt

Liangze Cui received the joint B.S. degree in telecommunications enginnering from Beijing University of Posts and Telecommunications, Beijing, China, and Queen Mary Univeristy of London, London, UK in 2018. He is currently pursuing the master degree with the department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. His current research interests include the wireless and optical communications, the finite-element method and the inverse problem in computational electromagnetics, machine learning methods and their applications.

Yao Zhang (Student Member, IEEE) received the Ph.D. degree in electronics and information engineering from the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China, in 2019.
In 2014, he joined the City University of Hong Kong, Shenzhen Research Institute, Shenzhen, China, as a Researcher. In September 2018, he joined the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA, as Visiting Scholar, under the financial support from the China Scholarship Council. He is currently an Assistant Professor with the School of Electronic Science and Engineering, Xiamen University, Xiamen, China. He has authored or coauthored more than 20 internationally referred journal articles and Conference papers, held 5 China patents and 2 US patents. His current research interests include massive MIMO antennas, millimeter-wave/THz antenna and systems, and multi-functional microwave circuits.
Dr. Zhang was a recipient of the Best Student Paper Award at the IEEE 5th Asia–Pacific Conference on Antenna and Propagation (2016 IEEE 5th APCAP), the National Scholarship for Graduate Students in 2015, 2016, and 2017, and the Year 2019 Outstanding Graduate Students of Guangdong Province, China. He also serves as a reviewer for several international journals, such as the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION and *IET Electronics Letters*.



Runren Zhang received the B.S. degree in information and communication engineering and the M.S. degree in electronic science and technology from Zhejiang University, Hangzhou, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA.
His current research interests include the finite-element method, the domain decomposition method and their application for subsurface and oil exploration, as well as machine learning/deep learning and their application in inverse problem.

Qing Huo Liu (S'88-M'89-SM'94-F'05) received his B.S. and M.S. degrees in physics from Xiamen University, China, and Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign.
He was with the Electromagnetics Laboratory at the University of Illinois at Urbana-Champaign as a Research Assistant from September 1986 to December 1988, and as a Postdoctoral Research Associate from January 1989 to February 1990. He was a Research Scientist and Program Leader with Schlumberger-Doll Research, Ridgefield, CT from 1990 to 1995. From 1996 to May 1999 he was an Associate Professor with New Mexico State University. Since June 1999 he has been with Duke University where he is now a Professor of Electrical and Computer Engineering. He has been also the founder and chairman of Wave Computation Technologies, Inc. since 2005. His research interests include computational electromagnetics and acoustics, inverse problems, and their application in nanophotonics, geophysics, biomedical imaging, and electronic packaging. He has published widely in these areas.
Dr. Liu is a Fellow of the IEEE, the Acoustical Society of America, the Electromagnetics Academy, and the Optical Society of America. He served as the founding Editor-in-Chief for the *IEEE Journal on Multiscale and Multiphysics Computational Techniques*. He received the 1996 Presidential Early Career Award for Scientists and Engineers (PECASE) from the White House, the 1996 Early Career Research Award from the Environmental Protection Agency, and the 1997 CAREER Award from the National Science Foundation. He has served as an IEEE Antennas and Propagation Society Distinguished Lecturer. He received the 2017 Technical Achievement Award and the 2018 Computational Electromagnetics Award from the Applied Computational Electromagnetics Society, and the 2018 Harrington-Mittra Award in Computational Electromagnetics from IEEE Antennas and Propagation Society. In 2018, he also received the University of Illinois at Urbana-Champaign ECE Distinguished Alumni Award.