A general framework for building surrogate models for uncertainty quantification in computational electromagnetics

Runze Hu, Vikass Monebhurrun, *Senior Member, IEEE*, Ryutaro Himeno, Hideo Yokota, Fumie Costen *Senior Member, IEEE*,

***Abstract*—In uncertainty analysis, surrogate modelling techniques demonstrate high efficiency and reliable precision in estimating the uncertainty for the finite difference time domain (FDTD) computation. However, building an accurate surrogate model may require a considerable number of system simulations which could be computationally expensive. To reduce such computational cost to build an accurate model, a general framework to build surrogate models for the FDTD computation in the human body based on the least angle regression (LARS) method and the artificial neural network (ANN) is proposed. The LARS method is adapted to dynamically select a number of informative random parameters which are significantly relevant to system outputs. We design a series of convergence criteria for ANN and introduce the adaptive moment estimation (ADAM) optimiser to ANN in order to improve the computational efficiency and accuracy of ANN. This is the first dynamic surrogate modelling technique for the FDTD computation designed by taking both accuracy and computational cost into account.**

***Index Terms*—Bioelectromagnetics, biological tissues, least angle regression (LARS), artificial neural network (ANN), finite difference time domain (FDTD), Debye media, cross-validation, uncertainty quantification (UQ).**

I. Introduction

In computational electromagnetics (EM), the accuracy of the numerical simulation is compromised by randomness which may arise in various forms such as electrical parameters and input sources [1]. Though the EM problems are usually treated as deterministic, there exist many instances where some input parameters of EM simulators cannot be strictly determined due to limited knowledge about these parameters. The ambiguity of input parameters causes a degree of uncertainty of system responses. Therefore, we should discuss the possible uncertainty together with the system response in order to increase the reliability of numerical simulations.

The uncertainty quantification (UQ) [2] techniques can be utilised to estimate the uncertainty of the system responses. However, considerable number of system simulations may be required in order to achieve high accuracy in UQ. For some computationally-intensive systems, such as the 3-dimensional (3D) finite difference time domain (FDTD) simulations [3], [4], performing such a large number of simulations can be impractical. The surrogate modelling technique [5], [6], that has emerged within the last decades, is effective in reducing the computational cost of UQ. This paper is concerned with the construction of a surrogate model for computation in electromagnetics. A surrogate model mimics the system simulation based on the analysis of the relationship between uncertain input parameters and the system response. An accurate surrogate model is capable of emulating the response of a system simulation while reducing the required computational resources.

Many advanced surrogate modelling techniques, such as the polynomial chaos expansion based methods [7], [8] and Kriging method [9], have been proposed over recent decades, each with their relative merits. For example, the work in [10] proposes a surrogate modelling technique based on sparse polynomial chaos (PC) expansion and the least angle regression (LARS) method [11]. It adopts the *q-quasi*-norms to reduce the number of polynomials and then utilises the LARS method to select significant polynomials from the remaining polynomials. [12] carries out PC expansion and utilises the LARS method to select significant polynomials and builds a surrogate model based on the Kriging method using the selected polynomials.

Machine learning techniques, such as the artificial neural network (ANN) [13]–[16] and the support vector machine (SVM) [17], are also widely applied to build surrogate models for complex systems. For example, an ANN-based surrogate modelling method in [18] adopts the SVM for sample selection. [19] utilises a SVM to build a surrogate model of electronic circuits. Though these surrogate modelling techniques demonstrate powerful performance in solving their specific problems, some challenges still remain. In general, a surrogate model can be highly accurate when sufficient data are available to train the surrogate model. However, generating significant quantities of data requires a considerable number of system simulations which can be computationally expensive. For example, when the 3D FDTD method is applied for the simulation of the wave propagating in the human body, a single

R. Hu and F. Costen are with the School of Electrical and Electronic Engineering, The University of Manchester, U.K. (email: fumie.costen@manchester.ac.uk).

V. Monebhurrun is with the GeePs, CentraleSupelec, 11 rue Joliot Curie Plateau de Moulon, Gif sur Yvette 91192, France (e-mail: Vikass.MONEBHURRUN@centralesupelec.fr).

R. Himeno is with Head Office for Information Systems and Cybersecurity, RIKEN, Saitama, Japan.

H. Yokota and F. Costen are with the Image Processing Research Team, Centre for Advanced Photonics, RIKEN, Saitama, Japan.

Color version of the figures in this paper are available online at http://ieeexplore.ieee.org.

Additional research data supporting this publication are available from http://dx.doi.org/ repository at 10.17632/tpvrtz3w6z.1.

system simulation may take hours to complete. Constructing an accurate surrogate model while minimising the cost of data generation is one of the challenges facing the surrogate modelling techniques.

To alleviate this challenge, this paper proposes a general framework to build a surrogate model for computation in electromagnetics with the minimum cost of data generation while maintaining the accuracy of the surrogate model. The proposed technique is based on the LARS method and ANN. The LARS method is a classical model selection method. This method is employed to select significant input parameters which are highly correlated with system output. The procedure of the LARS method consists of a number of LARS iterations, each of which selects one input parameter in the order of significance. For example, the input parameter, selected at the first LARS iteration, is considered the most significant input parameter in the system. This paper aims to minimise the number of generated data. We assume that there are very limited input samples available to build the surrogate model. However, the LARS method may not be able to accurately identify the significances of input parameters due to the limited number of input samples. Therefore, we propose a dynamic LARS method to improve the accuracy of identification without increasing the cost of data generation.

This dynamic LARS method is inspired by the leave-one-out cross-validation (LOOCV) method [20]. We drop out one set of input samples from a given dataset, with the remaining sets of input samples utilised to run the LARS method to obtain the significances of input parameters. This process is repeated for a number of times by dropping out a different set of input samples at each run and produces a set of results of the significances of input parameters by the LARS method. We determine the significance of each input parameter by the majority of these results and build a surrogate model with a certain number of significant input parameters and ANN.

An ANN learns from the loss function which measures the error of the prediction of system output, and utilises an optimiser to update the weights of ANN. The most popular optimiser is the gradient descent (GD) method [21]. It iteratively updates the weights in ANN in order to minimise the loss function. This paper adopts an adaptation of the GD method, namely the adaptive moment estimation (ADAM) optimiser [22], to update the weights of ANN. Comparing with the conventional GD method, ADAM is more computationally-efficient and has the potential to further improve the predictive accuracy of the system output via ANN. The conventional GD method utilises a single learning rate (LR) to update all weights of an ANN, whereas the ADAM updates each weight individually based on the LR, the average values and variances of the gradients.

The remaining sections of this paper are as follows. Section II describes the principles of the LARS method, the ANN and the proposed method. Section III details the experiments of building a surrogate model for FDTD simulation by the proposed method. Section IV presents the results of the predictions of FDTD output and evaluates the performance of the proposed method from the viewpoints of the accuracy and computational efficiency. Section V provides a conclusion.

II. Methodology

In FDTD calculations, the Debye model [23] or the Cole-Cole model [24] is implemented to account for the frequency dependency of each human tissue. The one-pole Debye model is written as [25] $\boldsymbol{D} = \epsilon_0 \left[\epsilon_\infty + \frac{\epsilon_S - \epsilon_\infty}{1 + \jmath\omega\tau} + \frac{\sigma_S}{\jmath\omega\epsilon_0} \right] \boldsymbol{E}$, where $\boldsymbol{E}$ is the electric field, $\boldsymbol{D}$ is the electric flux density, ϵ_0 is the permittivity of vacuum, ω is the angular frequency, $\jmath$ is the imaginary unit satisfying $\jmath = \sqrt{-1}$, σ_S is the static conductivity, ϵ_∞ is the optical relative permittivity, ϵ_S is the static relative permittivity, and τ is the relaxation time. These Debye parameters $(\sigma_S, \epsilon_\infty, \epsilon_S, \tau)$ differ among people.

Such variation is taken into account by introducing uncertainty. In the FDTD simulation, a certain number of Debye parameters of interest ξ as the random input parameters used to determine the system output of

$$|\boldsymbol{E}|^2 = \sum_l \left(\left| E_x^l(\hat{\imath}, \hat{\jmath}, \hat{k}) \right|^2 + \left| E_y^l(\hat{\imath}, \hat{\jmath}, \hat{k}) \right|^2 + \left| E_z^l(\hat{\imath}, \hat{\jmath}, \hat{k}) \right|^2 \right), \tag{1}$$

at an observation location $(\hat{\imath}, \hat{\jmath}, \hat{k})$, where l is the FDTD time step. A surrogate model is effectively derived by running a number of FDTD simulations for a given set $\mathcal{M}$ of $\mathcal{N}$ Debye parameters of interest.

A. The least angle regression

The LARS method is a model-selection method [10]. It selects the Debye parameters that highly correlate with FDTD output. Normalisation of the Debye parameters is a prerequisite prior to the application of the LARS method [26]. Let the vector $\boldsymbol{\xi} = [\xi_1, \xi_2, \dots, \xi_\mathcal{N}]$ be a combination of $\mathcal{N}$ normalised Debye parameters of interest and the $\mathcal{M} \times \mathcal{N}$ matrix $\boldsymbol{\mathcal{X}}(\mathcal{M}) = \left\{ {\xi_n}^{(m)}, m = 1 \sim \mathcal{M}, n = 1 \sim \mathcal{N} \right\}$ be an input sample set consisting of $\mathcal{M}$ $\boldsymbol{\xi}^{(m)}$, where $\boldsymbol{\xi}^{(m)}$ indicates the m-th sample of $\boldsymbol{\xi}$.

We define the linear regression model as in

$$\mathcal{T}\left(\boldsymbol{\xi}^{(m)}\right) = \sum_{n=1}^{\mathcal{N}} a_n {\xi_n}^{(m)} = \boldsymbol{\xi}^{(m)} \boldsymbol{a}, \tag{2}$$

where the vector $\boldsymbol{a} = [a_1, \cdots, a_\mathcal{N}]^T$ contains $\mathcal{N}$ coefficients of ξ. $\mathcal{T}(\boldsymbol{\xi}^{(m)})$ is the predicted value of $\left| \boldsymbol{E}^{(m)} \right|^2$ and $\left| \boldsymbol{E}^{(m)} \right|^2$ is $|\boldsymbol{E}|^2$ obtained by the FDTD simulation using $\boldsymbol{\xi}^{(m)}$. The process of LARS method involves a number of LARS iterations. At each LARS iteration, $\boldsymbol{a}$ is updated resulting in the update of $\mathcal{T}(\boldsymbol{\xi}^{(m)})$ in (2). Let $\mathcal{T}^{(k)}(\boldsymbol{\xi}^{(m)})$ be $\mathcal{T}(\boldsymbol{\xi}^{(m)})$ at k-th LARS iteration and the vector $\boldsymbol{\mathcal{T}}^{(k)} = \left[\mathcal{T}^{(k)}(\boldsymbol{\xi}^{(1)}), \cdots, \mathcal{T}^{(k)}(\boldsymbol{\xi}^{(\mathcal{M})}) \right]^T$. At the start of the LARS iteration process, $\boldsymbol{a}$ is initialised to $\mathbf{0}$. Thus, $\boldsymbol{\mathcal{T}}^{(1)}(\boldsymbol{\xi}^{(m)}) = \mathbf{0}$. We define $\boldsymbol{\mathcal{C}}^{(k)} = \left[{\mathcal{C}_1}^{(k)}, {\mathcal{C}_2}^{(k)}, \cdots, {\mathcal{C}_\mathcal{N}}^{(k)} \right]^T$ as a vector of the correlation between ξ and $(\boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{T}}^{(k)})$ calculated as in

$$\boldsymbol{\mathcal{C}}^{(k)} = \boldsymbol{\mathcal{X}}(\mathcal{M})^T (\boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{T}}^{(k)}), \tag{3}$$

where $\boldsymbol{\mathcal{E}} = \left[\left|\boldsymbol{E}^{(1)}\right|^2, \cdots, \left|\boldsymbol{E}^{(\mathcal{M})}\right|^2\right]^T$ and the difference between $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{T}}^{(k)}$ is called the residual vector at k-th LARS iteration. Each element in $\boldsymbol{\mathcal{C}}^{(k)}$, such as $\mathcal{C}_n{}^{(k)}$, refers to the correlation between ξ_n and the residual vector. When $\widetilde{n}^{(k)}$ satisfies

$$|\mathcal{C}_{\widetilde{n}^{(k)}}{}^{(k)}| = \max_{1 \le n \le \mathcal{N}} |\mathcal{C}_n{}^{(k)}|, \tag{4}$$

$\xi_{\widetilde{n}^{(k)}}$ is chosen as the highest correlated Debye parameter with the residual vector. $|\mathcal{C}_{\widetilde{n}^{(k)}}{}^{(k)}|$ in (4) represents the $\widetilde{n}^{(k)}$-th element of $\mathcal{C}$ in $\boldsymbol{\mathcal{C}}$ whose absolute value is maximum among $\{\mathcal{C}_1 \ldots \mathcal{C}_\mathcal{N}\}$ at k-th LARS iteration. For example, when $\mathcal{C}_3$ is maximum at 5th LARS iteration, $\widetilde{n}^{(k)}$ can be written as $\widetilde{n}^{(5)} = 3$. At the k-th LARS iteration, a matrix $\hat{\boldsymbol{\mathcal{X}}}(\mathcal{M})$ is formed to represent the chosen ξ denoted as $\hat{\boldsymbol{\mathcal{X}}}(\mathcal{M}) = \left\{\boldsymbol{\xi}_{\widetilde{n}^{(k)}}{}^{(m)}, m = 1 \sim \mathcal{M}, \widetilde{n}^{(k)} \in [1, \mathcal{N}]\right\}$.

$a_{\widetilde{n}^{(k)}}$ of those chosen ξ at the k-th LARS iteration are calculated as in [10]

$$\hat{\boldsymbol{a}}^{(k)} = \boldsymbol{b}^{(k)} + \ \gamma_{\ \widetilde{n}^{(k)}} \left(A \left(\hat{\boldsymbol{\mathcal{X}}}(\mathcal{M})^T \hat{\boldsymbol{\mathcal{X}}}(\mathcal{M}) \right)^{-1} \boldsymbol{1} \right), \tag{5}$$

where $\hat{\boldsymbol{a}}^{(k)} = \left[a_{\widetilde{n}^{(k)}}, \widetilde{n}^{(k)} \in [1, \mathcal{N}]\right]^T$, and $\boldsymbol{b}^{(k)} = \left[b_{\widetilde{n}^{(k)}}, \widetilde{n}^{(k)} \in [1, \mathcal{N}]\right]^T$ satisfying $\boldsymbol{b}^{(1)} = \boldsymbol{0}$. When $k > 1$, $b_{\widetilde{n}^{(k^\dagger)}} = a_{\widetilde{n}^{(k^\dagger)}}$ and $b_{\widetilde{n}^{(k)}} \triangleq 0$, where $1 \le k^\dagger \le k-1$ and $a_{\widetilde{n}^{(k^\dagger)}} \in \hat{\boldsymbol{a}}^{(k-1)}$. $A = \left(\boldsymbol{1}^T \left(\hat{\boldsymbol{\mathcal{X}}}(\mathcal{M})^T \hat{\boldsymbol{\mathcal{X}}}(\mathcal{M}) \right)^{-1} \boldsymbol{1} \right)^{-\frac{1}{2}}$, and $\boldsymbol{1}$ is a $k \times 1$ vector of 1. $\gamma_{\ \widetilde{n}^{(k)}}$ is calculated as follows [10]

$$\gamma_{\ \widetilde{n}^{(k)}} = \min{}^+ \left\{ \frac{|\mathcal{C}_{\widetilde{n}^{(k)}}| - \mathcal{C}_{\widetilde{n}^{(k)}}}{A - \ \hat{b}_{\widetilde{n}^{(k)}}}, \frac{|\mathcal{C}_{\widetilde{n}^{(k)}}| + \mathcal{C}_{\widetilde{n}^{(k)}}}{A + \ \hat{b}_{\widetilde{n}^{(k)}}} \right\}, \tag{6}$$

where "$\min^+$" indicates that $\gamma_{\ \widetilde{n}^{(k)}}$ only takes the positive minimum component in (6), and $\hat{b}_{\widetilde{n}^{(k)}}$ is the $\widetilde{n}^{(k)}$-th element of $\boldsymbol{\mathcal{X}}(\mathcal{M})^T \left(A \hat{\boldsymbol{\mathcal{X}}}(\mathcal{M}) \left(\hat{\boldsymbol{\mathcal{X}}}(\mathcal{M})^T \hat{\boldsymbol{\mathcal{X}}}(\mathcal{M}) \right)^{-1} \boldsymbol{1} \right)$.

The number of elements in $\hat{\boldsymbol{a}}$ increases with the progress of the LARS iteration, and (5) practically calculates the value of the element in $\boldsymbol{a}$. For example, during the first LARS iteration, $\hat{\boldsymbol{a}}$ contains one element of $a_{\widetilde{n}^{(1)}}$, and $\boldsymbol{a}$ in (2) becomes $\boldsymbol{a} = [0, \cdots, 0, a_{\widetilde{n}^{(1)}}, 0, \cdots, 0]^T$. We use the set $\boldsymbol{\Phi}$ to store $a_{\widetilde{n}^{(k)}} \xi_{\widetilde{n}^{(k)}}$ as in $\boldsymbol{\Phi} = \left[a_{\widetilde{n}^{(k)}} \xi_{\widetilde{n}^{(k)}}, \widetilde{n}^{(k)} \in [1, \mathcal{N}]\right]$ and the set $\boldsymbol{\Lambda}$ to store $\widetilde{n}^{(k)}$ as in

$$\boldsymbol{\Lambda} = \left[\widetilde{n}^{(k)}, \widetilde{n}^{(k)} \in [1, \mathcal{N}]\right], \tag{7}$$

where the number of elements in $\boldsymbol{\Lambda}$ and $\boldsymbol{\Phi}$ increases as the number of LARS iteration increases. The next step of the LARS iteration is the update of $\boldsymbol{\mathcal{T}}^{(k)}$ given by

$$\boldsymbol{\mathcal{T}}^{(k+1)} = \boldsymbol{\mathcal{T}}^{(k)} + \ \gamma_{\ \widetilde{n}^{(k)}} \left(A \hat{\boldsymbol{\mathcal{X}}}(\mathcal{M}) \left(\hat{\boldsymbol{\mathcal{X}}}(\mathcal{M})^T \hat{\boldsymbol{\mathcal{X}}}(\mathcal{M}) \right)^{-1} \boldsymbol{1} \right).$$

Such a process of determining $a_{\widetilde{n}^{(k)}} \xi_{\widetilde{n}^{(k)}}$, adding it into $\boldsymbol{\Phi}$, and updating $\mathcal{T}^{(k)}(\boldsymbol{\xi}^{(m)})$ is called a LARS iteration. The maximum number of the LARS iteration equals $\mathcal{N}$, in which case all ξ are incorporated in $\boldsymbol{\Phi}$.

B. The artificial neural network

The traditional ANN [14], [15] is adapted to build a surrogate model for the FDTD computation, where the input and the output are the Debye parameters of interest and the prediction of $|\boldsymbol{E}|^2$, respectively. We elaborate the ANN with the $1st$-hidden layer with G_1 neurons and $2nd$-hidden layer with G_2 neurons. Three adaptations are introduced to the traditional ANN:

1) Activation function:
The activation function (AF) of

$$f_p\left(\boldsymbol{\mathcal{R}}_i\right) = \boldsymbol{\mathcal{R}}_i + \boldsymbol{\mathcal{R}}_i \odot \boldsymbol{\mathcal{R}}_i, \text{ for } i = 1, 2 \tag{8}$$

has the potential to enable the ANN to effectively learn the data from the FDTD computation [27], where $\boldsymbol{\mathcal{R}}_1 = \boldsymbol{\mathcal{X}}(\mathcal{M})\boldsymbol{\mathcal{W}}_1$ and $\boldsymbol{\mathcal{R}}_2 = f_p\left(\boldsymbol{\mathcal{X}}(\mathcal{M})\boldsymbol{\mathcal{W}}_1\right) \cdot \boldsymbol{\mathcal{W}}_2$ are the inputs of the $1st$-hidden layer and $2nd$-hidden layer, respectively, and they are converted to $f_p\left(\boldsymbol{\mathcal{R}}_i\right)$ as the output of the i-th hidden layer, for $i = 1, 2$. Note that the operation in (8) is performed in the element-wise way. For example, the calculation of $f_p\left(\boldsymbol{\mathcal{R}}_1\right)$ is written as $f_p\left(\boldsymbol{\mathcal{R}}_1\right) = \boldsymbol{\mathcal{R}}_1 + \boldsymbol{\mathcal{R}}_1 \odot \boldsymbol{\mathcal{R}}_1 = \left\{ \mathcal{R}_{1_{(mg_1)}} + \left(\mathcal{R}_{1_{(mg_1)}}\right)^2, m = 1 \sim \mathcal{M}, g_1 = 1 \sim G_1 \right\}$.

2) Weights update:
The training process of an ANN consists of a number of iterations, each of which updates the weights of ANN in order to minimise the *training* error $\mathcal{L}_{tr} = \frac{1}{\mathcal{M}} \left|\left| \widehat{\boldsymbol{E}} - \boldsymbol{\mathcal{E}} \right|\right|^2 = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \left(\widehat{E}^{(m)} - \left|\boldsymbol{E}^{(m)}\right|^2 \right)^2$, where $||\cdot||$ is the Frobenius norm [28] and the vector $\widehat{\boldsymbol{E}} = \left[\widehat{E}^{(1)}, \cdots, \widehat{E}^{(\mathcal{M})}\right]^T$ contains $\mathcal{M}$ predictions of $|\boldsymbol{E}|^2$ via ANN corresponding to the $\mathcal{M}$ $|\boldsymbol{E}|^2$ from the FDTD simulations. $\mathcal{L}_{tr}$ is calculated at the end of each ANN iteration. For this two-hidden layer ANN, $\widehat{\boldsymbol{E}}$ is calculated as in

$$\widehat{\boldsymbol{E}} = f_p\left(f_p\left(\boldsymbol{\mathcal{X}}(\mathcal{M})\boldsymbol{\mathcal{W}}_1 \right) \cdot \boldsymbol{\mathcal{W}}_2 \right) \cdot \boldsymbol{\mathcal{W}}_3, \tag{9}$$

where matrix $\boldsymbol{\mathcal{W}}_1 = \{\mathcal{W}_{1_{(ng_1)}}, n = 1 \sim \mathcal{N}, g_1 = 1 \sim G_1\}$ and matrix $\boldsymbol{\mathcal{W}}_2 = \{\mathcal{W}_{2_{(g_1 g_2)}}, g_1 = 1 \sim G_1, g_2 = 1 \sim G_2\}$ are the weights between the input layer and the $1st$-hidden layer and those between the $1st$-hidden layer and the $2nd$-hidden layer, respectively. The vector $\boldsymbol{\mathcal{W}}_3 = \left[\mathcal{W}_{3_{(1)}}, \mathcal{W}_{3_{(2)}}, \ldots, \mathcal{W}_{3_{(G_2)}}\right]^T$ consists of the weights between the $2nd$-hidden layer and the output layer. The process of weights-update is generally based on the GD method. We hereby replace the GD method with the ADAM optimiser because the weights of ANN tend to converge more quickly with the ADAM optimiser than with the GD method, and the ADAM potentially improves the accuracy of ANN [22].

3) Termination criteria of the training process of ANN:
Traditionally, the ANN iteration is terminated when $\mathcal{L}_{tr}$ becomes stable. However, this way of termination may incur overfitting [29], whereby the *trained* ANN fails to generalise the relationship between the input Debye parameters and the FDTD output. When overfitting occurs, the *trained* ANN cannot accurately provide predictions for those data which are not utilised in the training. To terminate the ANN iteration while

decreasing the risk of overfitting, the following convergence criterion is proposed:

$$\sqrt{\frac{\sum_{i=1}^{3}\left\|{\boldsymbol{\mathcal{W}}_i}^{(j+1)}-{\boldsymbol{\mathcal{W}}_i}^{(j)}\right\|^2}{\sum_{i=1}^{3}\left\|{\boldsymbol{\mathcal{W}}_i}^{(j)}\right\|^2}}\leq\rho, \tag{10}$$

where ρ is a small constant. ${\boldsymbol{\mathcal{W}}_i}^{(j)}$ for $i=1\sim 3$ are the weights of this two-hidden layer ANN at j-th ANN iteration. We apply (10) to detect the change of weights with the ANN iteration. When (10) is met, the accuracy of ANN may not be improved with the further increase of the number of ANN iterations, leading to the risk of overfitting.

C. Leave-one-out cross-validation method

When the ANN iteration is terminated, we obtain ${\boldsymbol{\mathcal{W}}_i}^{(j)}$ for $i=1\sim 3$ as the ANN model. The accuracy of the ANN model can be evaluated by the LOOCV method. Given an input dataset $\boldsymbol{\mathcal{X}}(\mathcal{M})$, we split $\boldsymbol{\mathcal{X}}(\mathcal{M})$ into *training* and *test* data. The *training* data are used to train the ANN and obtain ${\boldsymbol{\mathcal{W}}_i}^{(j)}$ for $i=1\sim 3$ as the *trained* ANN model. The *test* data are used to estimate the accuracy of the *trained* ANN model.

We choose one $\boldsymbol{\xi}^{(m)}$ from $\boldsymbol{\mathcal{X}}(\mathcal{M})$ as the *test* data denoted as $\boldsymbol{\mathcal{X}}_{ts}\triangleq\{\boldsymbol{\xi}^{(m)},\ m\in[1,\mathcal{M}]\}$, and utilise the remaining data of $\boldsymbol{\mathcal{X}}(\mathcal{M}_{tr})$ as the *training* data to train the ANN, where $\mathcal{M}_{tr}=(\mathcal{M}-1)$ and $\boldsymbol{\mathcal{X}}(\mathcal{M}_{tr})\bigcup\boldsymbol{\mathcal{X}}_{ts}=\boldsymbol{\mathcal{X}}(\mathcal{M})$. When we use $\boldsymbol{\mathcal{X}}_{ts}=\{\boldsymbol{\xi}^{(m)}\}$ as an input to the *trained* ANN model, the *test* error ${\mathcal{L}_{ts}}^{(m)}$ is calculated as

$${\mathcal{L}_{ts}}^{(m)}=\left({\widehat{E}_{ts}}^{(m)}-\left|\boldsymbol{E}^{(m)}\right|^2\right)^2, \tag{11}$$

where ${\widehat{E}_{ts}}^{(m)}$ is the output of the *trained* ANN model calculated as in ${\widehat{E}_{ts}}^{(m)}=f_p\left(f_p\left(\boldsymbol{\mathcal{X}}_{ts}{\boldsymbol{\mathcal{W}}_1}^{(j)}\right)\cdot{\boldsymbol{\mathcal{W}}_2}^{(j)}\right)\cdot{\boldsymbol{\mathcal{W}}_3}^{(j)}$ using ${\boldsymbol{\mathcal{W}}_i}^{(j)}$ for $i=1\sim 3$ obtained from the *training*. We call the process of training the ANN using $\boldsymbol{\mathcal{X}}(\mathcal{M}_{tr})$ and calculating ${\mathcal{L}_{ts}}^{(m)}$ as a leave-one-out (LOO) iteration. Scanning m from 1 to $\mathcal{M}$, there are $\mathcal{M}$ LOO iterations in total. After the $\mathcal{M}$ LOO iterations, the LOO error $\mathcal{L}_l$ is calculated as in

$$\mathcal{L}_l=\frac{1}{\mathcal{M}}\sum_{m=1}^{\mathcal{M}}{\mathcal{L}_{ts}}^{(m)}. \tag{12}$$

D. The linear leave-one-out cross-validation method

The regression analysis is a linear regression when the model can be expressed by a summation of a set of terms. For example, the LARS method is a linear regression method because the regression model $\mathcal{T}(\boldsymbol{\xi}^{(m)})$ in (2) is the summation of $\mathcal{N}$ terms of $a_n{\xi_n}^{(m)}$. When the LOOCV method is used in the linear regression analysis, the LOO error can be calculated by a more computationally-efficient procedure [10] than the conventional LOOCV method described in Section II-C. We call the computationally-efficient LOOCV method for the linear regression as *linear* LOOCV method in order to distinguish it from the conventional LOOCV method. The *linear* LOOCV method is hereby adopted to estimate the accuracy of the LARS method. The *linear* LOO error $\mathcal{L}_{ln}$ of $\mathcal{T}(\boldsymbol{\xi}^{(m)})$ in (2) is calculated as follows:

$$\mathcal{L}_{ln}=\frac{1}{\mathcal{M}}\sum_{m=1}^{\mathcal{M}}\left(\frac{\left|\boldsymbol{E}^{(m)}\right|^2-\mathcal{T}(\boldsymbol{\xi}^{(m)})}{1-h_m}\right)^2, \tag{13}$$

where h_m is the m-th diagonal element of a $\mathcal{M}\times\mathcal{M}$ matrix $\hat{\boldsymbol{\mathcal{X}}}(\mathcal{M})\left(\hat{\boldsymbol{\mathcal{X}}}(\mathcal{M})^T\hat{\boldsymbol{\mathcal{X}}}(\mathcal{M})\right)^{-1}\hat{\boldsymbol{\mathcal{X}}}(\mathcal{M})^T$ [10].

E. Modelling FDTD computation

The surrogate model proposed hereafter is based on the LARS and ANN methods. It consists of two stages of dropout and modelling.

1) Dropout:

In general, various criteria are introduced to the LARS method in order to terminate the LARS iteration. In our proposed method, we run through all $\mathcal{N}$ LARS iterations without any stopping criteria. Thus, $\boldsymbol{\Lambda}$ in (7) contains $\mathcal{N}$ elements when a LARS run completes. The indexes of elements in $\boldsymbol{\Lambda}$ indicate the significances of correlation between Debye parameters and the residual vector. For example, when the first element (Λ_1) of $\boldsymbol{\Lambda}$ is 6, ξ_6 has the highest correlation with the residual vector, and thus we regard ξ_6 as the most significant Debye parameter.

The LARS method may not be accurate in identifying the significances of correlation between the Debye parameters and residual vector when we use only one dataset. The order of significance of Debye parameters may change with a different dataset. In order to improve the accuracy of the conventional LARS method, a dynamic LARS method is implemented.

For a given number $\mathcal{M}$ of samples, the LARS algorithm is successively run using $\mathcal{M}-1$ samples by removing $\boldsymbol{\xi}^{(m)}$ from $\boldsymbol{\mathcal{X}}(\mathcal{M})$ and utilising the remaining data of $\boldsymbol{\mathcal{X}}(\mathcal{M}_{re})$ to run the LARS method, where $\mathcal{M}_{re}=\mathcal{M}-1$, $\boldsymbol{\mathcal{X}}(\mathcal{M}_{re})\bigcup\boldsymbol{\mathcal{X}}_{dp}=\boldsymbol{\mathcal{X}}(\mathcal{M})$, and $\boldsymbol{\mathcal{X}}_{dp}\triangleq\{\boldsymbol{\xi}^{(m)},\ m\in[1,\mathcal{M}]\}$. When the LARS method completes all $\mathcal{N}$ iterations, $\boldsymbol{\Lambda}$ contains $\mathcal{N}$ elements presented as in $\boldsymbol{\Lambda}=[\Lambda_1,\Lambda_2,\cdots,\Lambda_{\mathcal{N}}]$. Let $\boldsymbol{\Lambda}^{(m)}$ be $\boldsymbol{\Lambda}$ when $\boldsymbol{\xi}^{(m)}$ is treated as the dropped data $\boldsymbol{\mathcal{X}}_{dp}$, and $\boldsymbol{\Lambda}^{(0)}$ indicates that none of $\boldsymbol{\xi}^{(m)}$ is dropped from $\boldsymbol{\mathcal{X}}(\mathcal{M})$ and the entire dataset is utilised to run the LARS method. We use top $(\mathcal{N}-P)$ significant Debye parameters to build a surrogate model using an ANN as follows:

(a) Obtain $\boldsymbol{\Lambda}^{(0)}$ by the LARS method as described in Section II-A using $\boldsymbol{\mathcal{X}}(\mathcal{M})$ and calculate ${\mathcal{L}_{ln}}^{(0)}$ in (13), where ${\mathcal{L}_{ln}}^{(0)}$ is $\mathcal{L}_{ln}$ in (13) when $P=0$, which is the LOO error of $\mathcal{T}(\boldsymbol{\xi}^{(m)})$ when we do not remove any Debye parameter.

(b) Split $\boldsymbol{\mathcal{X}}(\mathcal{M})$ into two parts of $\boldsymbol{\mathcal{X}}(\mathcal{M}_{re})$ and $\boldsymbol{\mathcal{X}}_{dp}$. Scanning m from 1 to $\mathcal{M}$, obtain $\boldsymbol{\Lambda}^{(m)}$ using the LARS method as described in Section II-A by replacing $\boldsymbol{\mathcal{X}}(\mathcal{M})$ with $\boldsymbol{\mathcal{X}}(\mathcal{M}_{re})$.

(c) Select the last P elements of $\boldsymbol{\Lambda}^{(m)}$ obtained from Step(a) and Step(b) for $0\leq m\leq\mathcal{M}$. These selected elements are presented as in

$[\Lambda(\mathcal{N}-P+1,0),\ \Lambda(\mathcal{N}-P+2,0),\cdots,\Lambda(\mathcal{N},0),$ $\Lambda(\mathcal{N}-P+1,1),\cdots,\Lambda(\mathcal{N},1),\cdots,\Lambda(\mathcal{N},\mathcal{M})]$, where $\Lambda(i,m)$ is the i-th element of $\mathbf{\Lambda}^{(m)}$ for $\mathcal{N}-P+1 \le i \le \mathcal{N}$.

(d) Identify the unique elements in $[\Lambda(\mathcal{N}-P+1,0),\ \Lambda(\mathcal{N}-P+2,0),\cdots,\Lambda(\mathcal{N},0),$ $\Lambda(\mathcal{N}-P+1,1),\cdots,\Lambda(\mathcal{N},1),\cdots,\Lambda(\mathcal{N},\mathcal{M})]$ and count the number of occurrences of these unique elements.

(e) Determine P unique elements in $[\Lambda(\mathcal{N}-P+1,0),\ \Lambda(\mathcal{N}-P+2,0),\cdots,\Lambda(\mathcal{N},0),$ $\Lambda(\mathcal{N}-P+1,1),\cdots,\Lambda(\mathcal{N},1),\cdots,\Lambda(\mathcal{N},\mathcal{M})]$ which correspond to the first P largest number of occurrences.

(f) Remove P unique Debye parameters, which are determined in Step(e), from $\boldsymbol{\mathcal{X}}(\mathcal{M})$. The subset of $\boldsymbol{\mathcal{X}}(\mathcal{M})$ is denoted by

$$\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M}) = \left[\boldsymbol{\xi}^{\ddagger(1)}, \boldsymbol{\xi}^{\ddagger(2)}, \cdots, \boldsymbol{\xi}^{\ddagger(\mathcal{M})}\right]^T \tag{14}$$

where $\boldsymbol{\xi}^{\ddagger(m)} = \left[\xi_1{}^{(m)}, \xi_2{}^{(m)}, \ldots, \xi_{\mathcal{N}-P}{}^{(m)}\right]$.

(g) Run the LARS method in Section II-A to obtain the coefficients $\boldsymbol{a}$ in (2) of these remaining Debye parameters by replacing $\boldsymbol{\mathcal{X}}(\mathcal{M})$ and $\mathcal{N}$ with $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$ and $\mathcal{N}-P$, respectively, and calculate $\mathcal{L}_{ln}$ in (13).

(h) Run Step(c)-Step(g) at each value of P, varying P from 1 to $P^{\dagger}(<\mathcal{N})$ to obtain $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$ and $\mathcal{L}_{ln}{}^{(P)}$, where $\mathcal{L}_{ln}{}^{(P)}$ is $\mathcal{L}_{ln}$ in (13) when $(\mathcal{N}-P)$ Debye parameters are used to form $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$.

(i) Determine P which has the minimum $\mathcal{L}_{ln}{}^{(P)}$.

The Step(c)-Step(e) in the case of $\mathcal{N}=10$ and $P=5$ is depicted in Fig. 1.

The usage of the limited number of significant Debye parameters improves the performance of ANN from the following aspects.

- Reduce the computational cost on the training process of ANN.
  The less number of input parameters may require less number of ANN iterations in the training process of ANN, and thus reducing the computational cost.
- Reduce the risk of overfitting.
  We aim to build a surrogate model of the FDTD simulation using a low number of input samples, such as $\mathcal{M}=\mathcal{N}$, which may incur overfitting. Reducing the number of input parameters has the potential to reduce the risk of overfitting and may improve the accuracy of ANN [29].

2) Modelling:

The ANN is utilised to build a surrogate model for FDTD computation using $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$. We define $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M}_{tr})$ and the associated *test* data in the same way as we do with $\boldsymbol{\mathcal{X}}(\mathcal{M}_{tr})$ in Section II-C. When a number of Debye parameters are excluded from the regression model, the complexity of the regression model is reduced. The AF of (8) is proposed for the complex regression analysis with relatively high number of variables. When the determined P in Step(i) is relatively high, the number of variables becomes relatively low, making the regression analysis simpler. In this case, AF of (8) may not be optimal. In order to set an AF appropriate to each case, and thus maximise the modelling accuracy via ANN, we introduce another AF to ANN, namely *linear* AF, as in

$$f_{lr}\left(\boldsymbol{\mathcal{R}}_i\right) = \boldsymbol{\mathcal{R}}_i, \text{ for } i=1,2. \tag{15}$$

These two AFs are treated as hyperparameters of ANN, and we utilise the LOOCV method to determine the most appropriate AF which presents the lowest $\mathcal{L}_l$. Let $\mathcal{L}_l{}^{(p)}$ and $\mathcal{L}_l{}^{(lr)}$ be $\mathcal{L}_l$ in (12) when (8) and (15) are adopted as the AF of ANN, respectively. Procedures for modelling the FDTD computation are as follows.

(A) Initialisation
At the start of the ANN iteration, $\boldsymbol{\mathcal{W}}_i$ for $i=1\sim 3$ are initialised with random numbers.

(B) Training
Split $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$ into $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M}_{tr})$ and the *test* data. The ANN is trained in Section II-B using $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M}_{tr})$ as the input dataset which contains $\mathcal{M}_{tr}$ input samples of $\mathcal{N}-P$ Debye parameters. When (10) is met, we terminate the ANN iteration and obtain $\boldsymbol{\mathcal{W}}_i{}^{(j)}$ for $i=1\sim 3$ as the *trained* ANN model.

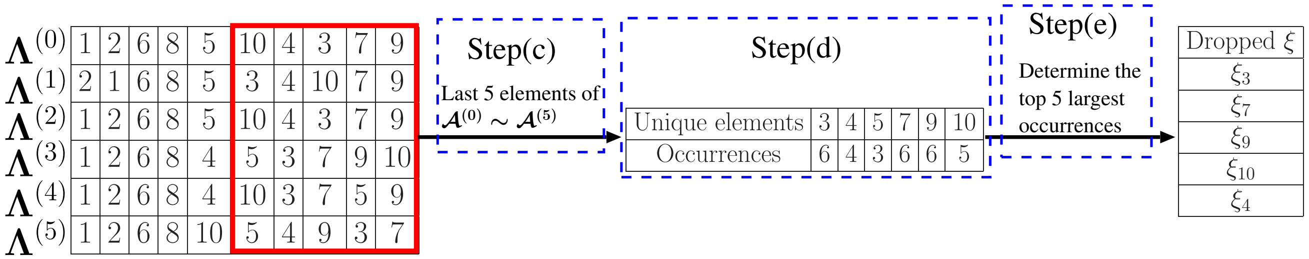
(C) Calculation of $\mathcal{L}_l$
Varying m from 1 to $\mathcal{M}$, run Step(A) and Step(B), and calculate $\mathcal{L}_{ts}{}^{(m)}$ in (11) by replacing $\boldsymbol{\mathcal{X}}_{ts}$ with $\boldsymbol{\xi}^{\ddagger(m)}$. When all $\boldsymbol{\xi}^{\ddagger(m)}$ in $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$ are used as *test* data, calculate $\mathcal{L}_l{}^{(p)}$ in (12).

(D) Determination of AF
Run Step(A), Step(B) and Step(C) to obtain $\mathcal{L}_l{}^{(lr)}$ by replacing (8) with (15). Determine AF of ANN that has the minimum $\mathcal{L}_l$.

(E) Modeling
Retrain the ANN in Section II-B using the AF determined in Step(D) and the entire dataset $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$. When (10) is met, we terminate the ANN iteration and save $\boldsymbol{\mathcal{W}}_i{}^{(j)}$ for $i=1\sim 3$ as the surrogate model of FDTD computation.

Note that Step(B) and Step(E) train the ANN using $\mathcal{M}_{tr}$ and $\mathcal{M}$ input samples, respectively, where $\mathcal{M}_{tr}=\mathcal{M}-1$. In general, when there are sufficient samples available to train the ANN, the influence of a single input sample on the accuracy of ANN is not significant. Therefore, in the case of sufficient input samples, we can skip Step(E) and consider the weights corresponding to the minimum $\mathcal{L}_l$ in Step(D) as the surrogate model of FDTD computation.

However, we hereby assume that the number of input samples is insufficient and thus each sample is important in the construction of the surrogate model. Furthermore, most of the computational resources required in the proposed method is in the process of determining the optimal AF (Step(D)). The computational cost in Step(E) is substantially smaller than that of Step(D). Thus, Step(E) is included in the proposed method to retrain the ANN using all input samples. The entire process of the proposed method is presented in Fig. 2. To assist readers to understand the proposed algorithm with ease, a simple numerical example of the proposed algorithm for the case of 2 samples and 2 random variables is presented in APPENDIX A.


Step(c)
Last 5 elements of $\mathcal{A}^{(0)} \sim \mathcal{A}^{(5)}$
Step(d)
Step(e)
Determine the top 5 largest occurrences
Dropped ξ
ξ_3
ξ_7
ξ_9
ξ_{10}
ξ_4

$\mathbf{\Lambda}^{(0)}$	1	2	6	8	5	10	4	3	7	9
$\mathbf{\Lambda}^{(1)}$	2	1	6	8	5	3	4	10	7	9
$\mathbf{\Lambda}^{(2)}$	1	2	6	8	5	10	4	3	7	9
$\mathbf{\Lambda}^{(3)}$	1	2	6	8	4	5	3	7	9	10
$\mathbf{\Lambda}^{(4)}$	1	2	6	8	4	10	3	7	5	9
$\mathbf{\Lambda}^{(5)}$	1	2	6	8	10	5	4	9	3	7

Unique elements	3	4	5	7	9	10
Occurrences	6	4	3	6	6	5

Fig. 1. Illustration of Step(c), Step(d) and Step(e) in Section II-E1 in the case of $\mathcal{N} = 10$ and $P = 5$.

III. Numerical Experiments

Section III presents the numerical experiment to build a surrogate model for the FDTD computation of the electromagnetic field inside the human body by applying the proposed method.

A. Numerical simulation setup

We create two scenarios to simulate the electromagnetic wave propagating in the human body. The digital human phantom (DHP) used in this work is provided by RIKEN (Saitama, Japan) under the non-disclosure agreement between RIKEN and the University of Manchester. The usage was approved by RIKEN ethical committee. We define some of the human tissues as influential tissues, where $|\boldsymbol{E}|^2$ in (1) changes significantly when one or more of the Debye parameters of the influential tissues are changed. The Debye parameters for all human tissues in the DHP are presented in [30]. In our scenario, the input parameters refer to ϵ_∞ and ϵ_S of the influential tissues. Descriptions of these two scenarios are as follows.

1) Scenario 1:
The simulation setup for scenario 1 is depicted in Fig. 3, where the FDTD space is $265 \times 490 \times 601$ mm with a resolution of 1 mm. The excitation is located 17 mm away from the human body and the observation is placed in the middle of the prostate tissue. 10 layers of the complex frequency shifted-perfect matched layers (CFS-PML) [31] [32] are used to terminate the FDTD space.
There are 10 influential tissues (skin, fat, muscle, bone, colon, small intestine, bladder, prostate, seminal vesicle, erectile tissue) nearby the observation point. We conduct the numerical experiments with the ten influential tissues which yield 20 input parameters in total.
2) Scenario 2:
The simulation setup for scenario 2 is depicted in Fig. 4, where the FDTD space is $265 \times 490 \times 240$ mm. The excitation is located 2 mm away from the human head and the observation is placed in the middle of the tissue of cerebrospinal fluid. CFS-PML for scenario 1 is applied.
This simulation scenario contains 11 influential tissues (skin, fat, muscle, bone, cerebral cortex, white matter, cerebellum, cerebrospinal fluid, salivary gland, middle ear, inner ear), yielding 22 input parameters in total.

B. Numerical experiments for modelling FDTD computation

Section III-B presents the numerical experiment mainly for scenario 1, where two stages of dropout and modelling are involved.

1) Dropout:

Procedures for generating samples of the input parameters are as follows.

a) Generate 2000 random ϵ_∞ and ϵ_S for each of 10 influential tissues yielding the normal distribution, where the relative permittivity and the conductivity of each influential tissue vary within ±10%. The work in [27] shows that, the accuracy of ANN saturates with 2000 input samples in the case of $\mathcal{N} = 20$. Therefore, we prepare 2000 input samples for the numerical experiments. The probability description of the 20 input parameters are presented in Table I, where $\Delta\epsilon = \epsilon_S - \epsilon_\infty$.
b) Randomly choose one sample out of the 2000 samples for each input parameter and combine these chosen samples to produce 1 combination of the 20 input parameters of $\boldsymbol{\xi}$.
c) Repeat Stepb) 2000 times to produce 2000 $\boldsymbol{\xi}$. In Stepb), we do not choose the same samples as those chosen earlier.

ξ	Meaning of ξ	Average	Standard deviation
ξ_1	ϵ_∞ of bone	7.37	0.74
ξ_2	$\Delta\epsilon$ of bone	6.80	0.68
ξ_3	ϵ_∞ of skin	29.87	2.99
ξ_4	$\Delta\epsilon$ of skin	18.07	1.81
ξ_5	ϵ_∞ of muscle	28.02	2.81
ξ_6	$\Delta\epsilon$ of muscle	28.93	2.88
ξ_7	ϵ_∞ of fat	4.01	0.40
ξ_8	$\Delta\epsilon$ of fat	1.53	0.15
ξ_9	ϵ_∞ of prostate	27.73	2.78
ξ_{10}	$\Delta\epsilon$ of prostate	32.82	3.28
ξ_{11}	ϵ_∞ of colon	34.76	3.57
ξ_{12}	$\Delta\epsilon$ of colon	26.46	2.60
ξ_{13}	ϵ_∞ of small intestine	39.26	3.79
ξ_{14}	$\Delta\epsilon$ of small intestine	25.65	2.58
ξ_{15}	ϵ_∞ of bladder	9.67	0.98
ξ_{16}	$\Delta\epsilon$ of bladder	9.66	0.97
ξ_{17}	ϵ_∞ of seminal vesicle	27.71	2.72
ξ_{18}	$\Delta\epsilon$ of seminal vesicle	32.92	3.24
ξ_{19}	ϵ_∞ of erectile tissue	30.54	3.06
ξ_{20}	$\Delta\epsilon$ of erectile tissue	32.40	3.26

TABLE I
The average values and standard deviations of the 20 Debye parameters and their corresponding notations

In the LARS method, the number of input samples should be larger than or equal to the number of input parameters

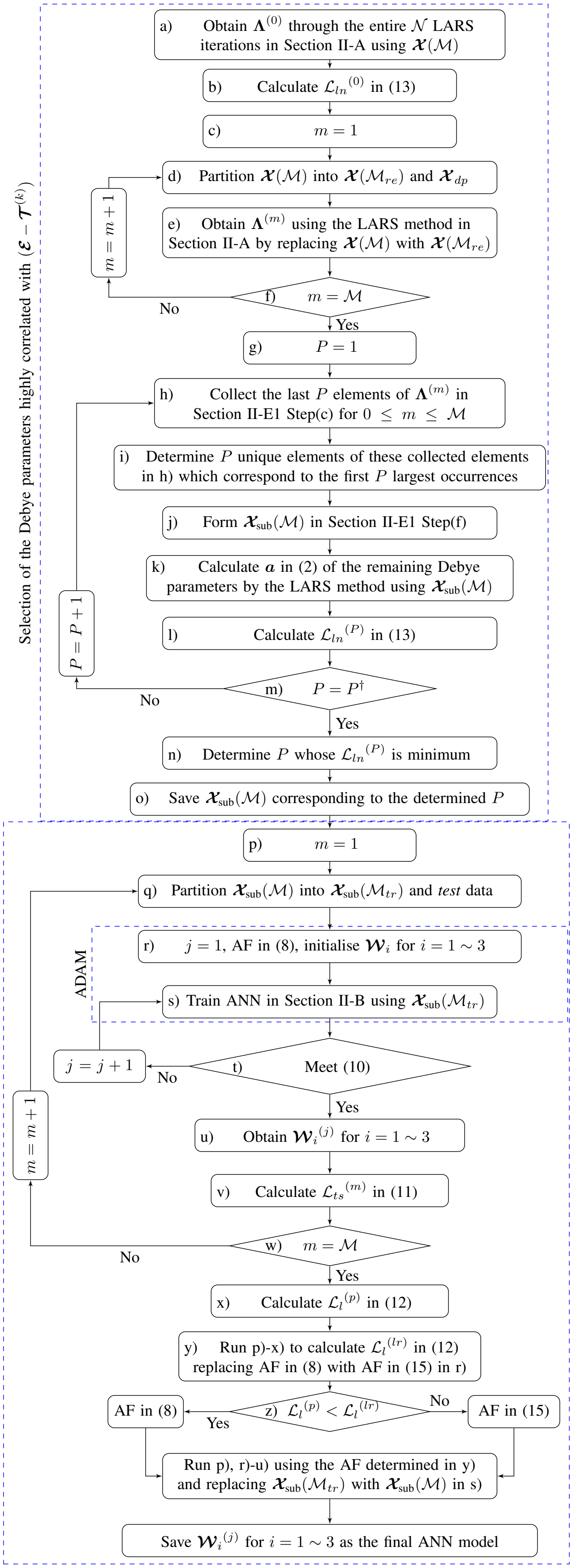

Selection of the Debye parameters highly correlated with $(\boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{T}}^{(k)})$
a) Obtain $\boldsymbol{\Lambda}^{(0)}$ through the entire $\mathcal{N}$ LARS iterations in Section II-A using $\boldsymbol{\mathcal{X}}(\mathcal{M})$
b) Calculate ${\mathcal{L}_{ln}}^{(0)}$ in (13)
c) $m = 1$
d) Partition $\boldsymbol{\mathcal{X}}(\mathcal{M})$ into $\boldsymbol{\mathcal{X}}(\mathcal{M}_{re})$ and $\boldsymbol{\mathcal{X}}_{dp}$
e) Obtain $\boldsymbol{\Lambda}^{(m)}$ using the LARS method in Section II-A by replacing $\boldsymbol{\mathcal{X}}(\mathcal{M})$ with $\boldsymbol{\mathcal{X}}(\mathcal{M}_{re})$
$m = m + 1$
f) $m = \mathcal{M}$
No
Yes
g) $P = 1$
h) Collect the last P elements of $\boldsymbol{\Lambda}^{(m)}$ in Section II-E1 Step(c) for $0 \le m \le \mathcal{M}$
i) Determine P unique elements of these collected elements in h) which correspond to the first P largest occurrences
j) Form $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$ in Section II-E1 Step(f)
k) Calculate $\boldsymbol{a}$ in (2) of the remaining Debye parameters by the LARS method using $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$
$P = P + 1$
l) Calculate ${\mathcal{L}_{ln}}^{(P)}$ in (13)
m) $P = P^{\dagger}$
No
Yes
n) Determine P whose ${\mathcal{L}_{ln}}^{(P)}$ is minimum
o) Save $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$ corresponding to the determined P
ANN
p) $m = 1$
q) Partition $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$ into $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M}_{tr})$ and *test* data
ADAM
r) $j = 1$, AF in (8), initialise $\boldsymbol{\mathcal{W}}_i$ for $i = 1 \sim 3$
s) Train ANN in Section II-B using $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M}_{tr})$
$j = j + 1$
t) Meet (10)
No
Yes
u) Obtain ${\boldsymbol{\mathcal{W}}_i}^{(j)}$ for $i = 1 \sim 3$
$m = m + 1$
v) Calculate ${\mathcal{L}_{ts}}^{(m)}$ in (11)
w) $m = \mathcal{M}$
No
Yes
x) Calculate ${\mathcal{L}_l}^{(p)}$ in (12)
y) Run p)-x) to calculate ${\mathcal{L}_l}^{(lr)}$ in (12) replacing AF in (8) with AF in (15) in r)
AF in (8)
Yes
z) ${\mathcal{L}_l}^{(p)} < {\mathcal{L}_l}^{(lr)}$
No
AF in (15)
Run p), r)-u) using the AF determined in y) and replacing $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M}_{tr})$ with $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$ in s)
Save ${\boldsymbol{\mathcal{W}}_i}^{(j)}$ for $i = 1 \sim 3$ as the final ANN model

Fig. 2. The flow chart of the proposed method to produce the final ANN model

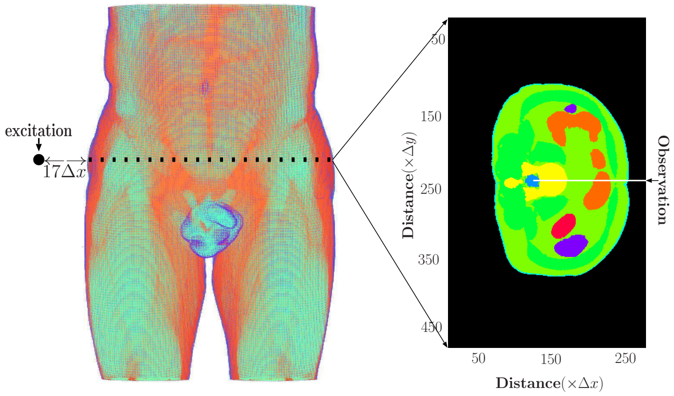


Fig. 3. Numerical simulation setup for scenario 1.

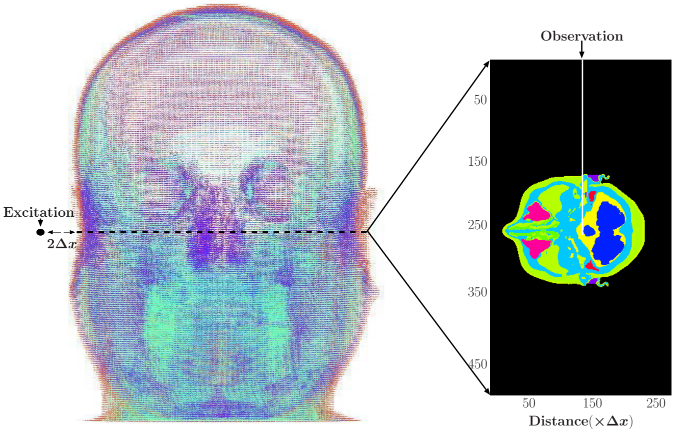


Fig. 4. Numerical simulation setup for scenario 2.

in order to obtain a unique solution for the coefficients of input parameters. In Step e) in Fig. 2, we obtain $\boldsymbol{\Lambda}^{(m)}$ using $\mathcal{M}_{re}(=\mathcal{M}-1)$ samples of $\boldsymbol{\xi}$ in the LARS method. Therefore, $\mathcal{M}-1$ should be not less than $\mathcal{N}(=20)$ as in $\mathcal{M}-1 \ge 20$. We set $\mathcal{M}$ to 21 in order to minimise the computational cost. We form $\boldsymbol{\mathcal{X}}(21)$ by randomly choosing 21 $\boldsymbol{\xi}$ out of the 2000 $\boldsymbol{\xi}$ and form $\boldsymbol{\mathcal{E}}$ by the FDTD simulations using the chosen $\boldsymbol{\xi}$.

We determine P at which ${\mathcal{L}_{ln}}^{(P)}$ reaches minimum varying P from 1 to $P^{\dagger}$. The setting of $P^{\dagger}$ is application dependent. For each value of P, one LARS run is required to obtain ${\mathcal{L}_{ln}}^{(P)}$. Therefore, a high value of $P^{\dagger}$ demands more LARS runs, which may render the computational inefficiency, whereas a small value of $P^{\dagger}$ may result in missing the global minimum of ${\mathcal{L}_{ln}}^{(P)}$. In our scenario, $P^{\dagger}$ is set to half of the number of input parameters as in $P^{\dagger} = \lfloor \frac{\mathcal{N}}{2} \rfloor = 10$, where $\lfloor \cdot \rfloor$ is the floor function that gives the largest integer which is less than or equal to $\frac{\mathcal{N}}{2}$. Varying P from 1 to 10, ${\mathcal{L}_{ln}}^{(P)}$ reaches minimum when $P = 3$. Thus, we choose 17 out of 20 input parameters to form the transformed $\boldsymbol{\mathcal{X}}_{\text{sub}}(21)$.

2) *Modelling:*

Before the ANN is trained, we set its hyperparameters. The number of hidden layers and the learning rate are set to 2 and 0.1, respectively. The number of neurons in the input layer is 17 equalling the number of chosen input parameters and there is one neuron in the output layer which is the prediction of $|\boldsymbol{E}|^2$. The number of neurons in the hidden layer is usually up to the size of input layer. We set it to half of the size of input layer as in $G_1 = G_2 = 8$. The reason for such settings is detailed in [14].

The ADAM optimiser is adopted to update the weights of ANN, where we use the recommended settings [22] for the hyperparameters associated with ADAM. The setting of ρ in (10) is application-dependent. In our scenarios, we set ρ to 0.01. When $j = 369$, condition (10) is met. Thus, we terminate the ANN iteration at the 369th ANN iteration and calculate ${\mathcal{L}_{ts}}^{(1)}$ in (11) by replacing $\boldsymbol{\mathcal{X}}_{ts}$ with ${\boldsymbol{\xi}^{\ddagger}}^{(1)}$ and using ${\boldsymbol{\mathcal{W}}_3}^{(369)}, {\boldsymbol{\mathcal{W}}_2}^{(369)}$, and ${\boldsymbol{\mathcal{W}}_1}^{(369)}$ as the *trained* ANN model of the first LOO iteration. When all 21 ${\boldsymbol{\xi}^{\ddagger}}^{(m)}$ in $\boldsymbol{\mathcal{X}}_{\text{sub}}(21)$ are used as *test* data, we calculate ${\mathcal{L}_l}^{(p)}$ in (12).

The same process is conducted to calculate ${\mathcal{L}_l}^{(lr)}$ in (12) using (15) as the AF of ANN. We obtain ${\mathcal{L}_l}^{(p)} = 1.39$ and ${\mathcal{L}_l}^{(lr)} = 1.95$. Thus, (8) is chosen as the AF of ANN. The ANN is then re-trained using $\boldsymbol{\mathcal{X}}_{\text{sub}}(21)$. Condition (10) is met at $j = 379$. ${\boldsymbol{\mathcal{W}}_3}^{(379)}, {\boldsymbol{\mathcal{W}}_2}^{(379)}$, and ${\boldsymbol{\mathcal{W}}_1}^{(379)}$ are therefore saved as the final ANN model.

In scenario 2, We conduct the same experiment as that in scenario 1 to build a surrogate model for the FDTD computation, where $\mathcal{N} = 22$ and $\mathcal{M} = 23$.

IV. Results and Discussion

In this section we evaluate the performance of the proposed method in terms of accuracy and computational efficiency.

A. Results

We feed 100 $\boldsymbol{\xi}$ which are not used for the training process of ANN into the final ANN model to obtain 100 predictions of $|\boldsymbol{E}|^2$, and assess the accuracy of the surrogate model by comparing 100 $|\boldsymbol{E}|^2$ from the FDTD simulations with the 100 predictions of $|\boldsymbol{E}|^2$ from the proposed method for scenario 1 and scenario 2. Fig. 5 shows the results for scenario 1 including the 100 $|\boldsymbol{E}|^2$ obtained from the FDTD simulations and their predictions $\widehat{E}$ in (9) from the proposed method, where the 100 $|\boldsymbol{E}|^2$ are sorted in ascending order for the sake of clarity. The accuracy of the 100 predictions of $|\boldsymbol{E}|^2$ is calculated as in $1-\frac{1}{100}\sum_{m=1}^{100}\frac{\left|\left|\boldsymbol{E}^{(m)}\right|^2-\widehat{E}^{(m)}\right|}{\left|\boldsymbol{E}^{(m)}\right|^2} = 96.42\%$. The results for Scenario 2 are presented in Fig. 6, where the accuracy of the 100 predictions of $|\boldsymbol{E}|^2$ is 95.58%.

B. Discussion

The proposed method mainly involves three techniques of the LARS method, the ANN, and the ADAM, where the ADAM is incorporated in the ANN for weights update. We discuss the influences of the LARS method and the ADAM on the accuracy of the proposed method based on scenario 1.

We use LARS+(ANN+ADAM) to indicate the proposed method, where (ANN+ADAM) refers to the ANN using the ADAM to update its weights. The traditional ANN, which utilises the GD method to update weights, is denoted as (ANN+GD).

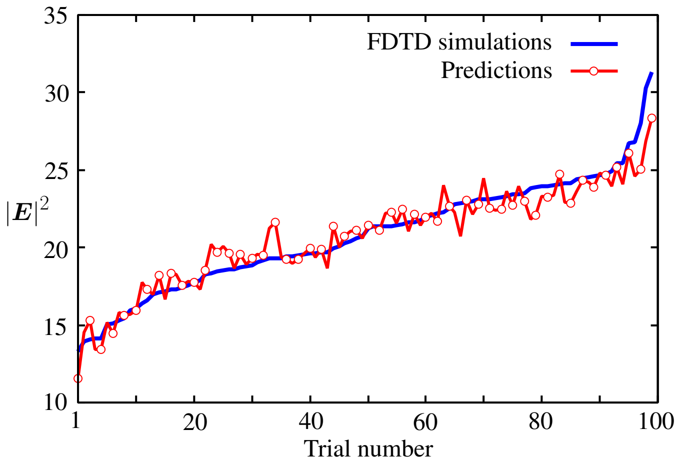


Fig. 5. Results for scenario 1, including 100 $|\boldsymbol{E}|^2$ from the FDTD simulations, sorted in ascending order and 100 predictions of $|\boldsymbol{E}|^2$ from the proposed method.

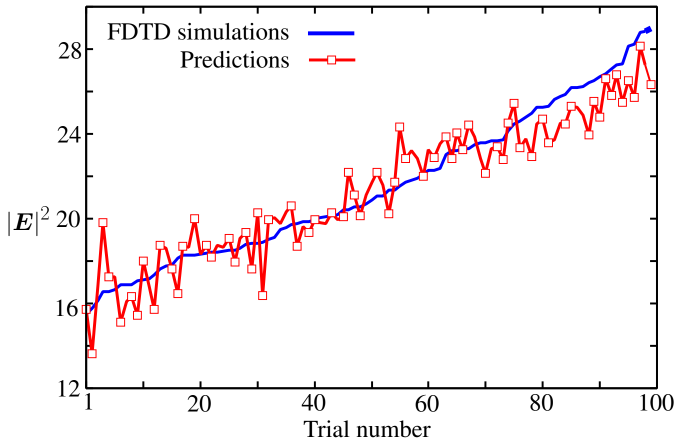
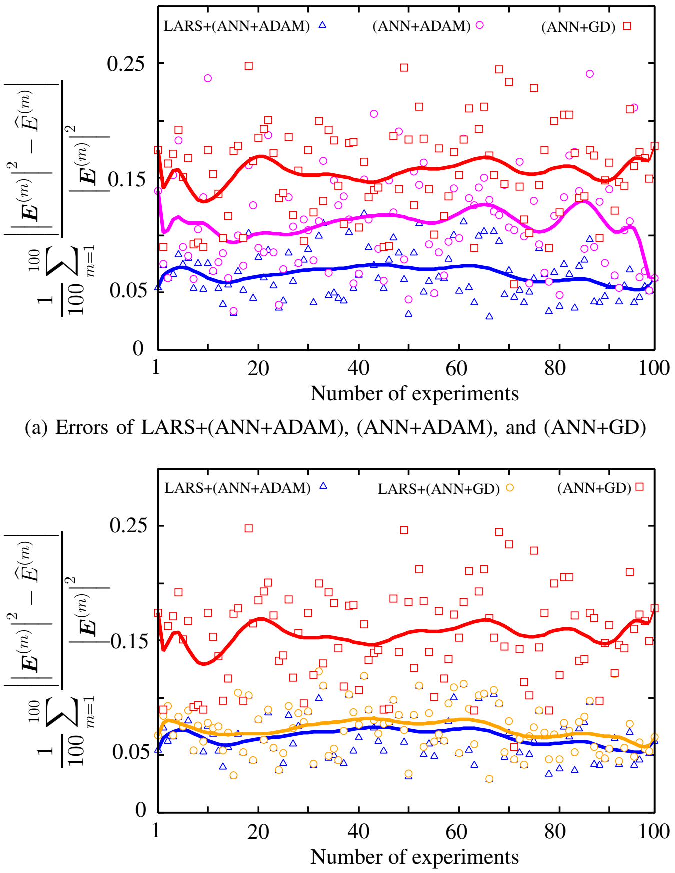


Fig. 6. Results for scenario 2, including 100 $|\boldsymbol{E}|^2$ from the FDTD simulations, sorted in ascending order and 100 predictions of $|\boldsymbol{E}|^2$ from the proposed method.

Fours methods of LARS+(ANN+ADAM), LARS+(ANN+GD), (ANN+ADAM), and (ANN+GD) are adopted to build four surrogate models for the FDTD computation. We utilise the 100 $\boldsymbol{\xi}$ in Section IV-A to evaluate their accuracy, where methods of (ANN+ADAM) and (ANN+GD) do not form $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M})$ but use $\boldsymbol{\mathcal{X}}(\mathcal{M})$ to train the ANN. Fig. 7 shows errors of $\frac{1}{100}\sum_{m=1}^{100}\frac{\left|\left|\boldsymbol{E}^{(m)}\right|^2-\widehat{E}^{(m)}\right|}{\left|\boldsymbol{E}^{(m)}\right|^2}$ for the four methods, where we conduct 100 experiments to obtain 100 errors for each method and different $\boldsymbol{\mathcal{X}}(21)$ randomly sampled from the 2000 samples in Section III-B1 is used at each experiment for the training. Table II lists the average values and the standard deviations of the 100 errors for the four methods.

From Fig. 7 and Table II, we conclude that

- both the LARS method and the ADAM optimiser con-



(a) Errors of LARS+(ANN+ADAM), (ANN+ADAM), and (ANN+GD)

(b) Errors of LARS+(ANN+ADAM), LARS+(ANN+GD), and (ANN+GD)

Fig. 7. Errors of LARS+(ANN+ADAM), LARS+(ANN+GD), (ANN+ADAM), and (ANN+GD). Solid lines are Bezier curves [33] for 100 errors.

Methods	Average value	Standard deviation
LARS+(ANN+ADAM)	0.066	0.021
LARS+(ANN+GD)	0.073	0.022
(ANN+ADAM)	0.110	0.045
(ANN+GD)	0.155	0.041

TABLE II
AVERAGE VALUES AND STANDARD DEVIATIONS OF 100 ERRORS FOR THE METHODS OF LARS+(ANN+ADAM), LARS+(ANN+GD), (ANN+ADAM) AND (ANN+GD).

tribute to the improvement of the accuracy of the proposed method and the traditional ANN.

- the LARS method demonstrates a greater significance in the proposed method and the traditional ANN than the ADAM optimiser.

Furthermore, we compare the proposed method with other state-of-the-art modelling techniques, including the works in [34] and [35], from the viewpoints of accuracy, stability and computational efficiency. The computational efficiency is assessed by the number of required FDTD simulations. Our in-house FDTD code was implemented based on the openMP and was executed on the Intel Xeon computer at 2.40 GHz with 128 GB of memory operating Red Hat Enterprise Linux 7.3 system, where the number of threads was set to 8. The computational requirements of one FDTD simulation for Scenario 1 and Scenario 2 are listed in Table III.

	Memory Usage (MB)	Real time (h)	FDTD iteration number
Scenario 1	9033.73	1.27	5000
Scenario 2	3235.84	0.42	5000

TABLE III
COMPUTATIONAL REQUIREMENTS OF ONE FDTD SIMULATION FOR SCENARIO 1 AND SCENARIO 2.

[34] builds a surrogate model based on the LARS method and the polynomial chaos expansion, where the LARS method is utilised to reduce the number of polynomials and the highest degree of Hermite polynomials is set to 2. [35] proposes a surrogate modelling technique based on the principal component analysis (PCA) and the Kriging method, where 5% information of the input dataset is dropped in the PCA. We apply the proposed method, [34] and [35] to build surrogate models for the FDTD computation. Thus, there are three surrogate models in total. 100 $\boldsymbol{\xi}$ generated in Section IV-A are fed into each surrogate model to obtain 100 predictions of $|\boldsymbol{E}|^2$. Fig. 8 shows the 100 predictions of $|\boldsymbol{E}|^2$ from these three methods for scenario 1 and scenario 2, respectively. Table IV and Table V list the accuracies of these three methods, and the average values and standard deviations of the 100 predicted $|\boldsymbol{E}|^2$ for scenario 1 and scenario 2, respectively. The proposed method demonstrates the highest accuracy in the comparison with other two techniques, and it also presents a higher efficiency because it requires the least number of FDTD simulations.

Methods	Accuracy	Average	Standard deviation	FDTD simulations
FDTD output		20.79	3.57	100
Proposed method	96.42%	20.70	3.18	21
Work in [35]	93.12%	21.01	4.26	21
Work in [34]	88.84%	21.12	5.05	41

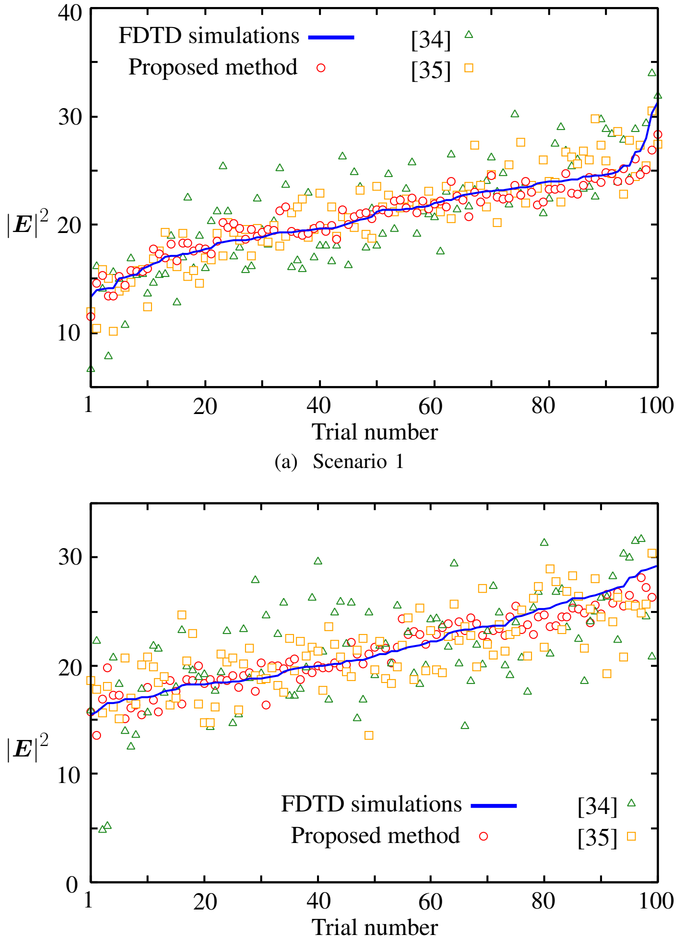
TABLE IV
ACCURACY, AVERAGE AND STANDARD DEVIATION OF 100 PREDICTED $|\boldsymbol{E}|^2$ FOR SCENARIO 1. FDTD OUTPUT REFER TO THE 100 $|\boldsymbol{E}|^2$ OBTAINED FROM 100 FDTD SIMULATIONS USING 100 $\boldsymbol{\xi}$.

Methods	Accuracy	Average	Standard deviation	FDTD simulations
FDTD output		21.51	3.54	100
Proposed method	95.58%	20.31	3.23	23
Work in [35]	89.98%	21.31	3.52	23
Work in [34]	84.69%	21.48	4.91	44

TABLE V
ACCURACY, AVERAGE AND STANDARD DEVIATION OF 100 PREDICTED $|\boldsymbol{E}|^2$ FOR SCENARIO 2. FDTD OUTPUT REFER TO THE 100 $|\boldsymbol{E}|^2$ OBTAINED FROM 100 FDTD SIMULATIONS USING 100 $\boldsymbol{\xi}$.

V. CONCLUSION

We proposed a general framework of surrogate modelling technique to minimise the computational cost associated with the number of system simulations while maintaining accuracy of the surrogate model. The proposed method is based on the LARS method and ANN, where the LARS method is adapted to improve the accuracy in selecting the informative input parameters, and thus reducing the complexity of the surrogate model. Furthermore, we introduce the ADAM optimiser and the hyperparameters of activation functions to ANN in order



(a) Scenario 1

(b) Scenario 2

Fig. 8. 100 $|\boldsymbol{E}|^2$ from the FDTD simulations and 100 predictions of $|\boldsymbol{E}|^2$ from the proposed method, works in [34] and [35]

to improve the modelling accuracy of ANN. The performance of the proposed method is evaluated in comparison to other state-of-the-art techniques. The proposed method has better performance in terms of accuracy, efficiency, and robustness, which could potentially be of significant use for uncertainty quantification, for example, in bioelectromagnetics field.

Appendix A
Basic numerical example of proposed algorithm

Assuming there are $2(=\mathcal{M})$ sets of samples for $2(=\mathcal{N})$ variables ξ_1, ξ_2, the dataset $\boldsymbol{\mathcal{X}}(2)$ is given as $\boldsymbol{\mathcal{X}}(2) = \begin{bmatrix} 2.38 & 1.93 \\ 2.41 & 1.99 \end{bmatrix}$. Procedures of *dropout* are presented as follows:

Step(a): $\boldsymbol{\Lambda}^{(0)} = [2, 1]$ is obtained by the LARS method using $\boldsymbol{\mathcal{X}}(2)$ and $\mathcal{L}_{ln}{}^{(0)} = 0.6$ is calculated in (13).

Step(b): m is varied from 1 to 2 to obtain $\boldsymbol{\Lambda}^{(1)}$ and $\boldsymbol{\Lambda}^{(2)}$, respectively.

When $m = 1$, remove $\boldsymbol{\xi}^{(1)}$ from $\boldsymbol{\mathcal{X}}(2)$, and run the LARS method for the remaining $\mathcal{M}-1 = 1$ sample as in $\boldsymbol{\mathcal{X}}(\mathcal{M}_{re}) = [2.41, 1.99]$ to obtain $\boldsymbol{\Lambda}^{(1)} = [2, 1]$.

When $m = 2$, reinsert $\boldsymbol{\xi}^{(1)}$ and remove $\boldsymbol{\xi}^{(2)}$, and run the LARS method for the remaining $\mathcal{M} - 1 = 1$ sample as in $\boldsymbol{\mathcal{X}}(\mathcal{M}_{re}) = [2.38, 1.93]$ to obtain $\boldsymbol{\Lambda}^{(2)} = [1, 2]$.

Step(c): Varying P from 1 to $\mathcal{N}-1$, in the case of $\mathcal{N} = 2$, P only has one value equalling 1. When $P = 1$, we select the last element of $\boldsymbol{\Lambda}^{(0)}, \boldsymbol{\Lambda}^{(1)}$ and $\boldsymbol{\Lambda}^{(2)}$. $\Lambda(i, m)$ in Step(c) are presented as $[\Lambda(2, 0), \Lambda(2, 1), \Lambda(2, 2)] = [1, 1, 2]$.

Step(d): The unique elements in $\Lambda(i, m)$ and their occurrences are presented in Table VI as

Unique elements	1	2
Occurrences	2	1

TABLE VI
Unique elements in $\Lambda(i, m)$ and occurrences

Step(e): Determine $1(= P)$ unique element which is ξ_1 because it has the largest number of occurrences.

Step(f): $\boldsymbol{\mathcal{X}}_{\text{sub}}(2)$ is obtained by removing the samples of ξ_1 from $\boldsymbol{\mathcal{X}}(2)$ as $\boldsymbol{\mathcal{X}}_{\text{sub}}(2) = \begin{bmatrix} 1.93 \\ 1.99 \end{bmatrix}$.

Step(g) and Step(h): $\boldsymbol{\mathcal{X}}_{\text{sub}}(2)$ is then utilised to run the LARS method in Section II-A and $\mathcal{L}_{ln}{}^{(1)} = 0.35$ is calculated in (13).

Step(i): Since $\mathcal{L}_{ln}{}^{(1)} = 0.35 < \mathcal{L}_{ln}{}^{(0)} = 0.6$, $\mathcal{L}_{ln}{}^{(P)}$ reaches the minimum at which $P = 1$. $\boldsymbol{\mathcal{X}}_{\text{sub}}(2)$ obtained in Step(f) is saved as the input data to the ANN.

In the stage of *Modelling*:

Step(A)-Step(C): Split $\boldsymbol{\mathcal{X}}_{\text{sub}}(2)$ into $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M}_{tr})$ and the *test* data varying m from 1 to $2(=\mathcal{M})$, where $\mathcal{M}_{tr} = \mathcal{M}-1 = 1$. When $m = 1$, $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M}_{tr}) = [1.99]$ and the *test* data is the first set of sample in $\boldsymbol{\mathcal{X}}_{\text{sub}}(2)$ denoted as $\boldsymbol{\xi}^{\ddagger(1)} = 1.93$. The ANN is trained in Section II-B using the AF of (8) and $\boldsymbol{\mathcal{X}}_{\text{sub}}(\mathcal{M}_{tr})$. $\mathcal{L}_{ts}{}^{(1)} = 0.4$ is then calculated in (11) by replacing $\boldsymbol{\mathcal{X}}_{ts}$ with $\boldsymbol{\xi}^{\ddagger(1)}$. Similarily, we can obtain $\mathcal{L}_{ts}{}^{(2)} = 0.5$ in the case of $m = 2$. $\mathcal{L}_{l}{}^{(p)} = 0.45$ is then calculated in (12) using $\mathcal{L}_{ts}{}^{(1)}$ and $\mathcal{L}_{ts}{}^{(2)}$.

Step(D): Utilise (15) as the AF of ANN, and run Step(A)-Step(C). $\mathcal{L}_{l}{}^{(lr)} = 0.3$ is obtained in (12). Since $\mathcal{L}_{l}{}^{(lr)} < \mathcal{L}_{l}{}^{(p)}$, (15) is determined to be the AF of ANN.

Step(E): Build the surrogate model by retraining the ANN in Section II-B using the AF of (15) and $\boldsymbol{\mathcal{X}}_{\text{sub}}(2)$.

Acknowledgement

This work was completed in part with HOKUSAI-GreatWave Computer System at RIKEN. http://i.riken.jp/download/sites/2/HOKUSAI_system_overview_en.pdf.

References

[1] K. Masumnia-Bisheh, K. Forooraghi, and M. Ghaffari-Miab, "Electromagnetic uncertainty analysis using stochastic fdfd method," *IEEE Trans. Antennas Propag.*, vol. 67, no. 5, pp. 3268–3277, 2019.

[2] B. Sudret, "Uncertainty propagation and sensitivity analysis in mechanical models–Contributions to structural reliability and stochastic spectral methods," *Habilitationa diriger des recherches, Université Blaise Pascal, Clermont-Ferrand, France*, vol. 147, 2007.

[3] A. Taflove and S. C. Hagness, *Computational electrodynamics: the finite-difference time-domain method*. Artech house, 2005.

[4] K. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Trans. Antennas Propag.*, vol. 14, no. 3, pp. 302–307, 1966.

[5] B. Sudret, S. Marelli, and J. Wiart, "Surrogate models for uncertainty quantification: An overview," in *2017 11th European conference on antennas and propagation (EUCAP)*, pp. 793–797, IEEE, 2017.

[6] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*, vol. 10. John Wiley & Sons, 2016.

[7] D. Xiu and G. E. Karniadakis, "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM journal on scientific computing*, vol. 24, no. 2, pp. 619–644, 2002.

[8] D. Xiu and G. E. Karniadakis, "Modeling uncertainty in flow simulations via generalized polynomial chaos," *Journal of computational physics*, vol. 187, no. 1, pp. 137–167, 2003.

[9] N. Cressie, "The origins of kriging," *Mathematical geology*, vol. 22, no. 3, pp. 239–252, 1990.

[10] G. Blatman and B. Sudret, "Adaptive sparse polynomial chaos expansion based on least angle regression," *Journal of Computational Physics*, vol. 230, no. 6, pp. 2345–2367, 2011.

[11] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[12] P. Kersaudy, B. Sudret, N. Varsier, O. Picon, and J. Wiart, "A new surrogate modeling technique combining kriging and polynomial chaos expansions–application to uncertainty analysis in computational dosimetry," *Journal of Computational Physics*, vol. 286, pp. 103–117, 2015.

[13] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[14] R. J. Schalkoff, *Artificial neural networks*, vol. 1. McGraw-Hill New York, 1997.

[15] Y. Kim, S. Keely, J. Ghosh, and H. Ling, "Application of artificial neural networks to broadband antenna design based on a parametric frequency model," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 669–674, 2007.

[16] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications," *IEEE access*, vol. 6, pp. 36218–36234, 2018.

[17] N. Cristianini, J. Shawe-Taylor, *et al.*, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[18] D. Ibrahim, M. Jobson, J. Li, and G. Guillén-Gosálbez, "Surrogate models combined with a support vector machine for the optimized design of a crude oil distillation unit using genetic algorithms," in *Computer Aided Chemical Engineering*, vol. 40, pp. 481–486, Elsevier, 2017.

[19] A. Ciccazzo, G. Di Pillo, and V. Latorre, "A svm surrogate model-based method for parametric yield optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 7, pp. 1224–1228, 2015.

[20] A. M. Molinaro, R. Simon, and R. M. Pfeiffer, "Prediction error estimation: a comparison of resampling methods," *Bioinformatics*, vol. 21, no. 15, pp. 3301–3307, 2005.

[21] R. Fletcher and M. J. Powell, "A rapidly convergent descent method for minimization," *The computer journal*, vol. 6, no. 2, pp. 163–168, 1963.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] K. Tekbas, F. Costen, J.-P. Bérenger, R. Himeno, and H. Yokota, "Subcell modeling of frequency-dependent thin layers in the FDTD method," *IEEE Trans. Antennas Propag.*, vol. 65, no. 1, pp. 278–286, 2017.

[24] K. S. Cole and R. H. Cole, "Dispersion and Absorption in Dielectrics I. Alternating Current Characteristics," *The Journal of Chemical Physics*, vol. 9, no. 4, pp. 341–351, 1941.

[25] K. Tekbas, F. Costen, J.-P. Bérenger, R. Himeno, and H. Yokota, "Subcell modeling of frequency-dependent thin layers in the FDTD method," *IEEE Trans. Antennas Propag.*, vol. 65, no. 1, pp. 278–286, 2016.

[26] R. Hu, V. Monebhurrun, R. Himeno, H. Yokota, and F. Costen, "A statistical parsimony method for uncertainty quantification of FDTD computation based on the PCA and ridge regression," *IEEE Trans. Antennas Propag.*, 2019.

[27] R. Hu, V. Monebhurrun, R. Himeno, H. Yokota, and F. Costen, "An adaptive least angle regression method for uncertainty quantification in FDTD computation," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 2131–2134, 2018.

[28] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[29] G. N. Karystinos and D. A. Pados, "On overfitting, generalization, and randomly expanded training sets," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1050–1057, 2000.

[30] "The RIKEN webpage, media parameters for the Debye relaxation model," http://cfd-duo.riken.jp/cbms-mp/. Accessed: Sept. 3, 2018.

[31] J.-P. Bérenger, "A perfectly matched layer for the absorption of electromagnetic waves," *Journal of computational physics*, vol. 114, no. 2, pp. 185–200, 1994.

[32] S. D. Gedney, G. Liu, J. A. Roden, and A. Zhu, "Perfectly matched layer media with CFS for an unconditionally stable ADI-FDTD method," *IEEE Trans. Antennas Propag.*, vol. 49, no. 11, pp. 1554–1559, 2001.

[33] M. E. Mortenson, *Mathematics for computer graphics applications*. Industrial Press Inc., 1999.

[34] J. Du and C. Roblin, "Stochastic surrogate models of deformable antennas based on vector spherical harmonics and polynomial chaos expansions: Application to textile antennas," *IEEE Trans. Antennas Propag.*, vol. 66, no. 7, pp. 3610–3622, 2018.

[35] G. Aversano, J. C. Parra-Alvarez, B. J. Isaac, S. T. Smith, A. Coussement, O. Gicquel, and A. Parente, "PCA and Kriging for the efficient exploration of consistency regions in Uncertainty Quantification," *Proceedings of the Combustion Institute*, vol. 37, no. 4, pp. 4461–4469, 2019.