Uncertainty analysis on FDTD computation with artificial neural network

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

***Abstract*—The artificial neural network (ANN) has appeared as a potential alternative for uncertainty quantification (UQ) in the finite difference time domain (FDTD) computation. It is applied to build a surrogate model for the compute-intensive FDTD simulation and to bypass the numerous simulations required for UQ. However, when the surrogate model utilizes the ANN, a considerable number of data is generally required for high accuracy and generating such large quantities of data becomes computationally prohibitive. To address this drawback, a number of adaptations for ANN are proposed which additionally improves the accuracy of the ANN in UQ for the FDTD computation while maintaining a low computational cost. The proposed algorithm is tested for application in bioelectromagnetics and considerable speed-up, as well as improved accuracy of UQ, is observed compared to traditional methods such as the non-intrusive polynomial chaos method.**

***Index Terms*—Artificial neural network (ANN), uncertainty quantification (UQ), finite difference time domain (FDTD), Debye media, biological tissues**

I. Introduction

The finite difference time domain (FDTD) method [1], [2] is a well-proven technique for transient and full-wave numerical simulation of the propagation of electromagnetic waves in inhomogeneous media. It is especially applied in bioelectromagnetics for numerical simulations. However, in the FDTD simulation of the human body, the input parameters, such as the complex permittivities of human tissues, are determined from measurements of dielectric properties with typical uncertainty of $\pm 10\%$. The ambiguity of these input parameters results in a degree of uncertainty in the system response, and subsequently the system response cannot be precisely determined. Therefore, in order to increase the reliability and accuracy of FDTD simulations, it is necessary to quantify the impact of uncertainties of input parameters on the system responses.

This paper addresses the uncertainty analysis of FDTD simulations of the human body, where a digital human phantom (DHP) [3] is utilized as an equivalent human model, and the one-pole Debye model [4] is implemented in the FDTD computation to characterize the complex frequency-dependent behaviour of the biological tissues. Each tissue in the DHP is associated with Debye parameters which are subject to some degree of uncertainty, leading to different frequency responses among the population. This problem is known as forward uncertainty quantification (UQ) or the uncertainty propagation.

Various UQ techniques have been proposed in the last decades. The traditional Monte Carlo Method (MCM) [5] proves to be inappropriate for UQ because a considerable number of FDTD simulations is required to reach a reasonable level of convergence. There exist some ideal alternatives to MCM, such as the non-intrusive polynomial chaos (NIPC) expansion method [6], [7], and the stochastic collocation based method [8]. However, both methods cannot efficiently handle the high-dimensional UQ problems, due to the *curse of dimensionality* [9] whereby the number of required simulations substantially grows when the number of random variables increases. Although various techniques, such as the hyperbolic scheme [10] and the analysis of variance (ANOVA) decomposition method [11], are applied to reduce the impact of the *curse of dimensionality*, the computational cost associated with the system simulations remains high.

Many advanced UQ techniques have been proposed over the last decade to alleviate the *curse of dimensionality* and improve the accuracy of UQ [12]–[14]. Ideas behind these techniques include sparse strategies [15], [16], surrogate modelling [17], model order reduction [18] and hierarchical approach [19], [20]. The surrogate modelling technique, which builds a simpler equivalent model for the original complex system, is an ideal candidate for the UQ of computationally-intensive systems. A surrogate model has the potential to accurately predict the system outputs and it allows to bypass thousands of otherwise necessary system simulations, thereby significantly improving the efficiency of traditional methods such as MCM and NIPC.

The artificial neural network (ANN) [21]–[23] has become one of the most promising surrogate modelling techniques, owing to its high flexibility and learning capability. This paper studies the ANN in UQ for the FDTD computation. The ANN model is trained using the data which include the input samples of Debye parameters and the desired outputs obtained from the FDTD simulations. In general, the accuracy of ANN can be improved by increasing the number of data. However, this way may lead to the computational inefficiency in UQ for the compute-intensive systems, such as the 3-dimensional (3D)

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 EXPOSE/PIEM/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/hj7mkc9263.1.

FDTD simulation. In order to improve the accuracy of ANN in UQ for the FDTD computation without increasing the number of system simulations, an activation function of ANN [24] is proposed.

The proposed activation function is inspired by the NIPC expansion and the work in [18]. The quadratic NIPC expansion, *i.e.* with the order of polynomials restricted to two, was demonstrated to provide sufficient accuracy [25] for the UQ for the FDTD computation. When multiple random variables are considered, it may be useful to treat each random variable as a single uncertain input while the remaining random variables are considered as constants [18]. The proposed activation function is therefore devised using the quadratic NIPC expansion for one random variable. This proposed activation function is also known as the polynomial activation function, which had been utilized on various applications of ANN. However, it has never been applied to quantify the uncertainty of the FDTD results.

The novelty of this paper lies in both the application and the method.

- In terms of the application, we analyse the uncertainty of the 3D FDTD simulations of the human body. The DHP used in this work is derived from the medical imaging of a healthy male subject. This representative model of the human body is used to produce the experimental results of UQ that are consistent with reality.
- In terms of the method, we utilize the ANN for UQ of the FDTD computation, where two techniques are proposed to improve the efficiency and accuracy of UQ via the ANN. Firstly, we propose an activation function specified for the FDTD computation. This enables the ANN to build an accurate surrogate model for the FDTD computation from limited data, and thus improves the efficiency and accuracy of UQ. Secondly, we introduce a series of termination criteria to the ANN to prevent the potential risk of overfitting [26] that usually occurs when the data is insufficient, thereby maintaining the accuracy of UQ.

This paper is organised as follows: Section II explains the principles of the ANN, and the proposed adaptations for the ANN. Section III details the numerical experiments for UQ by the MCM and the proposed method. Finally, the results obtained using these different techniques are presented and the merits of the proposed method evaluated.

II. Methodology

In the FDTD calculations, the relationship between electric field $\boldsymbol{E}$ and the electric flux density $\boldsymbol{D}$ in one-pole Debye model is written as [27] $\boldsymbol{D} = \epsilon_0 \left[\epsilon_\infty + \frac{\epsilon_S - \epsilon_\infty}{1 + \jmath\omega\tau} + \frac{\sigma_S}{\jmath\omega\epsilon_0}\right] \boldsymbol{E}$, where ϵ_0 is the permittivity of vacuum, ϵ_S is the static permittivity, ϵ_∞ is the optical perimittivity, ω is the angular frequency, $\jmath$ is the imaginary unit satisfying $\jmath = \sqrt{-1}$, σ_S is the static conductivity, and τ is the relaxation time. Each tissue in the DHP is associated with four Debye parameters $(\sigma_S, \epsilon_\infty, \epsilon_S, \tau)$.

We assume there are $\mathcal{K}$ independent Debye parameters of interest satisfying the normal distribution and $\mathcal{M}$ sets of input samples to an ANN. These $\mathcal{K}$ Debye parameters are considered as uncertain inputs and they are expressed in a vectorial form of $\boldsymbol{\xi} = [\xi_1, \xi_2, \cdots, \xi_\mathcal{K}]^T$. We define $\boldsymbol{\xi}^{(m)}$ as the m-th sample of $\boldsymbol{\xi}$ for $1 \leq m \leq \mathcal{M}$. Apart from these $\mathcal{K}$ Debye parameters, input parameters in the FDTD simulation, such as the location of the excitation, are treated as constant. Since the specific absorption rate (SAR) is usually of interest in bioelectromagnetics applications, the square of the electric field $|\boldsymbol{E}|^2$ is considered the output of the FDTD simulation calculated as in $|\boldsymbol{E}|^2 = \sum_{\hat{n}} \left(\left| E_x^{\hat{n}}(\hat{i}, \hat{j}, \hat{k}) \right|^2 + \left| E_y^{\hat{n}}(\hat{i}, \hat{j}, \hat{k}) \right|^2 + \left| E_z^{\hat{n}}(\hat{i}, \hat{j}, \hat{k}) \right|^2 \right)$, where $(\hat{i}, \hat{j}, \hat{k})$ is the point location of the observation and $\hat{n}$ is the FDTD time step. Fig. 1 is an example showing how the uncertainty of Debye parameters of ϵ_S and ϵ_∞ for muscle affects the FDTD response $|\boldsymbol{E}|^2$ in the 3-dimensional FDTD simulations, in which ϵ_S and ϵ_∞ are varied from -10% to 10% simultaneously by steps of 2%. Details of the experimental setup for Fig. 1 are presented in Section III. It is

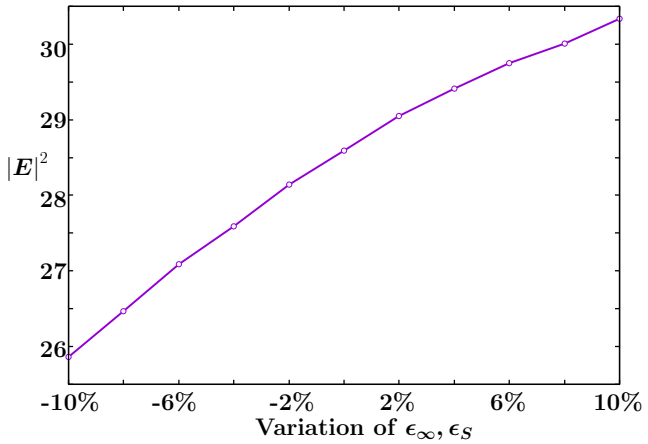

Fig. 1. Impact of uncertain Debye parameters of muscle on the system response

observed that the uncertainties of ϵ_S and ϵ_∞ induce a variation of up to 16% for the FDTD response $|\boldsymbol{E}|^2$. Furthermore, it is important to note that the numerical experiments of Fig. 1 only consider the uncertainties of Debye parameters from one tissue, *i.e.,* muscle. When multiple uncertain Debye parameters of tissues are considered, the FDTD response cannot be simply determined and it is therefore crucial to estimate the uncertainty of the FDTD response induced by these uncertain Debye parameters. This paper proposes an adaptive ANN to quantify the uncertainty of the FDTD simulation, in which a number of adaptations are introduced to the ANN aiming at improving the performance of the ANN in UQ. Details of the proposed method are presented as follows.

A. The architecture of ANN

The ANN is a machine learning algorithm utilized to model the underlying relationships between the input parameters and system output. An ANN typically has three types of layers of input, hidden and output layers. In this paper, the neurons in

the input layer represent the uncertain Debye parameters and the neuron in the output layer indicates the output of the ANN which is a single scalar of the prediction of $|\boldsymbol{E}|^2$.

Each neuron in the hidden layer receives its input $\mathcal{T}$ from the previous layer and converts $\mathcal{T}$ to the input of the successive layer by an activation function $f(\mathcal{T})$. We form a set of data pairs from the input $\boldsymbol{\xi}^{(m)}$ and the desired output $\left|\boldsymbol{E}^{(m)}\right|^2$ and train the ANN in order to optimise the weights of ANN which enables the ANN to make an accurate prediction of $\left|\boldsymbol{E}^{(m)}\right|^2$ for any input $\boldsymbol{\xi}^{(m)}$.

B. Polynomial activation function

This paper proposes a polynomial activation function to improve the accuracy of ANN in predicting the output of the FDTD computation. We define this activation function as

$$f_p(\mathcal{T}) = \mathcal{T} + \mathcal{T}^2. \tag{1}$$

This polynomial activation function is inspired by [18] and the NIPC expansion method. The work in [18] quantifies the uncertainty of the system $\mathcal{K}$ times when a system has $\mathcal{K}$ parameters. [18] considers only one parameter as an uncertain input at one time and the remaining $\mathcal{K}-1$ parameters are treated as constants. Thus, the number of polynomials is calculated based on the case of one parameter instead of $\mathcal{K}$ parameters thereby alleviating the *curse of dimensionality*.

The quadratic NIPC expansion method [25] achieves a high accuracy in estimating UQ for the FDTD computation. When a system has one random Debye parameter satisfying the normal distribution, $\psi_0(\xi_k) = 1, \psi_1(\xi_k) = \xi_k, \psi_2(\xi_k) = {\xi_k}^2 - 1$ are the quadratic Hermite polynomials. The NIPC expansion is formed by three polynomials as in $\sum_{\alpha=0}^{2} d_\alpha \psi_\alpha(\xi_k)$, where d_α is a coefficient of $\psi_\alpha(\xi_k)$. We set d_α for $0 \le \alpha \le 2$ to 1 for the sake of simplicity. Replacing ξ_k with $\mathcal{T}$, we design (1) as in $f_p(\mathcal{T}) = \sum_{\alpha=0}^{2} \psi_\alpha(\mathcal{T})$.

C. ANN with multiple hidden layers

The ANN with one hidden layer may underperform on the accuracy of the prediction of the system output when it is used to model a complex system. In such circumstances, multiple hidden layers are required. This paper defines an ANN with two hidden layers of the $1st$-hidden layer with G_1 neurons and the $2nd$-hidden layer with G_2 neurons.

Training an ANN with multiple hidden layers mainly involves three stages of forward propagation, backpropagation and the update of weights. Let ${\xi_k}^{(m)}$ be the m-th sample of ξ_k used for the FDTD computation and $\boldsymbol{\mathcal{X}}(\mathcal{M}) = \left\{{\tilde{\xi}_k}^{(m)}, m = 1 \sim \mathcal{M}, k = 1 \sim \mathcal{K}\right\}$ be a normalised sample set, where ${\tilde{\xi}_k}^{(m)}$ is the normalised ${\xi_k}^{(m)}$. In the stage of forward propagation, we calculate the output of ANN by

$$\widehat{\boldsymbol{E}} = f_p\big(f_p\big(\boldsymbol{\mathcal{X}}(\mathcal{M})\boldsymbol{\mathcal{W}}_1\big)\boldsymbol{\mathcal{W}}_2\big)\boldsymbol{\mathcal{W}}_3, \tag{2}$$

where the matrix $\boldsymbol{\mathcal{W}}_1 = \{\mathcal{W}_{1_{(kg_1)}}, k = 1 \sim \mathcal{K}, g_1 = 1 \sim G_1\}$ consists of the weights between the input layer and the $1st$-hidden layer. $\mathcal{W}_{1_{(kg_1)}}$ indicates the weight between k-th neuron in the input layer and g_1-th neuron in the $1st$-hidden layer. $\boldsymbol{\mathcal{W}}_2 = \{\mathcal{W}_{2_{(g_1 g_2)}}, g_1 = 1 \sim G_1, g_2 = 1 \sim G_2\}$ refers to the weights between the $1st$-hidden layer and the $2nd$-hidden layer, and $\boldsymbol{\mathcal{W}}_3 = \left[\mathcal{W}_{3_{(1)}}, \mathcal{W}_{3_{(2)}}, \ldots, \mathcal{W}_{3_{(G_2)}}\right]^T$ indicates the weights between the $2nd$-hidden layer and the output layer. Furthermore, when the input to f_p in (1) is a matrix or vector, the operation in f_p is element-wise.

The stages of backpropagation and update of weights are based on the gradient descent method. The ANN updates its weights in order to minimize the loss function of $\boldsymbol{L} = \frac{1}{2}(\widehat{\boldsymbol{E}} - \boldsymbol{\mathcal{E}})^2$, where $\boldsymbol{\mathcal{E}} = \left[\left|\boldsymbol{E}^{(1)}\right|^2, \cdots, \left|\boldsymbol{E}^{(\mathcal{M})}\right|^2\right]^T$ is a vector consisting of $\mathcal{M}$ $|\boldsymbol{E}|^2$ obtained from the $\mathcal{M}$ FDTD simulations. The procedure for forward propagation, backpropagation and update of weights is called an ANN iteration. Let ${\boldsymbol{\mathcal{W}}_i}^{(j)}$ for $i = 1 \sim 3$ be the weights of the ANN at j-th ANN iteration. The ANN updates its weights as in

$$\begin{aligned} {\boldsymbol{\mathcal{W}}_i}^{(j+1)} &= {\boldsymbol{\mathcal{W}}_i}^{(j)} - \eta {f_p(\boldsymbol{\mathcal{T}}_{i-1})}^{(j)^T} {\boldsymbol{\delta}_i}^{(j)} \text{ for } i = 2, 3 \\ {\boldsymbol{\mathcal{W}}_1}^{(j+1)} &= {\boldsymbol{\mathcal{W}}_1}^{(j)} - \eta {\boldsymbol{\mathcal{X}}(\mathcal{M})}^T {\boldsymbol{\delta}_1}^{(j)}, \end{aligned} \tag{3}$$

where $\boldsymbol{\mathcal{T}}_1 = \boldsymbol{\mathcal{X}}(\mathcal{M})\boldsymbol{\mathcal{W}}_1$ and $\boldsymbol{\mathcal{T}}_2 = f_p(\boldsymbol{\mathcal{T}}_1)\boldsymbol{\mathcal{W}}_2$ represent inputs to the $1st$-hidden layer and the $2nd$-hidden layer, respectively. η is the learning rate utilized to tune the update of weights. ${\boldsymbol{\delta}_i}^{(j)}$ for $i = 1 \sim 3$ are the error signals at the j-th ANN iteration used to measure how much $\boldsymbol{L}$ varies with the changes of $\boldsymbol{\mathcal{T}}_i$ for $i = 1 \sim 2$. ${\boldsymbol{\delta}_i}^{(j)}$ for $i = 1 \sim 3$ are written as

$$\begin{aligned} \boldsymbol{\delta}_3 &= \frac{\partial \boldsymbol{L}}{\partial \widehat{\boldsymbol{E}}} = \widehat{\boldsymbol{E}} - \boldsymbol{\mathcal{E}} \\ \boldsymbol{\delta}_i &= \frac{\partial f_p(\boldsymbol{\mathcal{T}}_i)}{\partial \boldsymbol{\mathcal{T}}_i} \odot \left(\boldsymbol{\delta}_{i+1} {\boldsymbol{\mathcal{W}}_{i+1}}^T\right), \text{ for } i = 1, 2 \end{aligned} \tag{4}$$

where $\odot$ denotes the element-wise multiplication of matrices. The ANN iteration is repeated for a certain number of times, each with updated weights and is terminated when the accuracy of the estimation of the system output via ANN reaches our expectation. The leave-one-out cross-validation (LOOCV) method [28] is utilized to quantify the accuracy.

D. Leave-one-out cross-validation

For a given dataset of $\mathcal{M}$ samples, the LOOCV method splits the dataset into the *training* data $\boldsymbol{\mathcal{X}}(\mathcal{M}_{tr})$, the *validation* data $\boldsymbol{\mathcal{X}}(\mathcal{M} - 1 - \mathcal{M}_{tr})$, and the *test* data, where $\mathcal{M}_{tr}$ is an integer which satisfies $0 < \mathcal{M}_{tr} < \mathcal{M} - 1$ and the *test* data contains one sample $\tilde{\boldsymbol{\xi}}^{(m)}$. ANN is trained in Section II-C using $\boldsymbol{\mathcal{X}}(\mathcal{M}_{tr})$ instead of $\boldsymbol{\mathcal{X}}(\mathcal{M})$. At each ANN iteration, ANN updates its weights in (3). These updated weights are utlised to calculate the output of ANN for $\boldsymbol{\mathcal{X}}(\mathcal{M} - 1 - \mathcal{M}_{tr})$ at the j-th ANN iteration as in ${\widehat{\boldsymbol{E}}_v}^{(j)} = f_p\left(f_p\left(\boldsymbol{\mathcal{X}}(\mathcal{M} - 1 - \mathcal{M}_{tr}){\boldsymbol{\mathcal{W}}_1}^{(j)}\right) \cdot {\boldsymbol{\mathcal{W}}_2}^{(j)}\right) \cdot {\boldsymbol{\mathcal{W}}_3}^{(j)}$, where ${\widehat{\boldsymbol{E}}_v}^{(j)} = \left[{\widehat{E}_v}^{(j)^{(1)}}, \cdots, {\widehat{E}_v}^{(j)^{(\mathcal{M}-1-\mathcal{M}_{tr})}}\right]$ is the output of

ANN of ${\widehat{E}_v}^{(j)(m)}$ using $\boldsymbol{\mathcal{X}}(\mathcal{M}-1-\mathcal{M}_{tr})$ at the j-th ANN iteration. Let ${\mathcal{L}_v}^{(j)}$ be the *validation* error $\mathcal{L}_v$ as in

$$ {\mathcal{L}_v}^{(j)} = \frac{\sum_{m=1}^{\mathcal{M}-1-\mathcal{M}_{tr}} \left({\widehat{E}_v}^{(j)(m)} - \left| \boldsymbol{E}^{(m)} \right|^2 \right)^2}{\mathcal{M}-1-\mathcal{M}_{tr}} \tag{5} $$

at the j-th ANN iteration.

The aim of using the *validation* data is prevention of the overfitting. $\mathcal{L}_v$ starts decreasing during the ANN iteration and later on increases when overfitting occurs. Therefore, we terminate the ANN iteration when $\dfrac{{\mathcal{L}_v}^{(j+1)}}{{\mathcal{L}_v}^{(j)}}$ is larger than a certain value ϖ in order to prevent the potential overfitting in ANN. This method is also known as early stopping.

Apart from the early stopping, other techniques, such as the *dropout* [29], can be utilized to avoid the overfitting as well. Since this paper deals with a two-hidden layer ANN with a linear activation function, the architecture of our ANN is relatively simple. Using the technique of early stopping solely is capable of detecting the overfitting.

The training of the ANN completes at the termination of the ANN iteration and we obtain ${\boldsymbol{\mathcal{W}}_i}^{(j)}$ for $i = 1 \sim 3$ as the *trained* ANN model. The *test* data is utilized to quantify the error of the *trained* ANN model. When $\tilde{\boldsymbol{\xi}}^{(m)}$ is used as the *test* data composed of one element, the *test* error ${\mathcal{L}_{ts}}^{(m)}$ is calculated as in

$$ \begin{aligned} {\mathcal{L}_{ts}}^{(m)} = \Big(& f_p \left(f_p \left(\tilde{\boldsymbol{\xi}}^{(m)} {\boldsymbol{\mathcal{W}}_1}^{(j)} \right) {\boldsymbol{\mathcal{W}}_2}^{(j)} \right) {\boldsymbol{\mathcal{W}}_3}^{(j)} \\ & - \left| \boldsymbol{E}^{(m)} \right|^2 \Big)^2 . \end{aligned} \tag{6} $$

We call the process of generation of the *trained* ANN model and calculation of ${\mathcal{L}_{ts}}^{(m)}$ a leave-one-out (LOO) iteration. Since we scan 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

$$ \mathcal{L}_l = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} {\mathcal{L}_{ts}}^{(m)} . \tag{7} $$

E. Uncertainty quantification of the FDTD computation

Section II-E introduces the proposed adaptations, including the polynomial activation function in Section II-B and the LOOCV method, into the ANN method described in Section II-C to quantify the uncertainty of the FDTD computation. The procedure of ANN for UQ for the FDTD computaion is as follows.

A) Calculation of $\mathcal{L}_l$
Given the dataset $\boldsymbol{\mathcal{X}}(\mathcal{M})$, we train the ANN using $\boldsymbol{\mathcal{X}}(\mathcal{M}_{tr})$ in Section II-C to obtain a *trained* ANN model. Varying m from 1 to $\mathcal{M}$, $\mathcal{L}_l$ in (7) is calculated when all $\tilde{\boldsymbol{\xi}}^{(m)}$ in $\boldsymbol{\mathcal{X}}(\mathcal{M})$ are used as the *test* data.

B) The final ANN model
We re-train the ANN in Section II-C using the entire $\mathcal{M}$ sets of input samples as the *training* data from scratch. After I ANN teration, the *training* error ${\mathcal{L}_{tr}}^{(j)}$ for the final ANN model is calculated as in

$$ {\mathcal{L}_{tr}}^{(j)} = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \left({\widehat{E}}^{(j)(m)} - \left| \boldsymbol{E}^{(m)} \right|^2 \right)^2 \tag{8} $$

at the end of each ANN iteration, where I is an integer and ${\widehat{E}}^{(j)(m)}$ indicates m-th element of $\widehat{\boldsymbol{E}}$ in (2) at j-th ANN iteration.
The ANN iteration is terminated when the conditions of
a) $j \geq I$
b) ${\mathcal{L}_{tr}}^{(j)} \leq \mathcal{L}_l$
c)

$$ \frac{\left| {\mathcal{L}_{tr}}^{(j)} - {\mathcal{L}_{tr}}^{(j-1)} \right|}{{\mathcal{L}_{tr}}^{(j-1)}} \leq b \tag{9} $$

and

$$ \frac{\left| {\mathcal{L}_{tr}}^{(j-1)} - {\mathcal{L}_{tr}}^{(j-2)} \right|}{{\mathcal{L}_{tr}}^{(j-2)}} \leq b, \tag{10} $$

where $0 < b < 1$
are all met.
These proposed conditions are inspired by the early stopping scheme. The Condition B)a aims at improving the computational efficiency. In the training process, the ANN needs some iterations to learn from the data and updates its weights accordingly. During the first I iterations, we do not calculate $\mathcal{L}_{tr}$. In general [22], $\mathcal{L}_{tr}$ should be less than $\mathcal{L}_l$ when the ANN model is well trained. $\mathcal{L}_l$ in Condition B)b is a constant throughout the ANN iteration. On the other hand, ${\mathcal{L}_{tr}}^{(j)}$ always decreases monotonously as the ANN iteration increases. $\mathcal{L}_{tr} > \mathcal{L}_l$ means that the ANN developed with the entire $\mathcal{M}$ sets of input samples is not fully trained and the accuracy of the system output via ANN can be further improved. The Condition B)b shall be met at a certain j when the model is fully trained. Condition B)c involves two inequalities of (9) and (10), which are utilized to detect the stable status of $\mathcal{L}_{tr}$. Satisfying (9) and (10) means that the accuracy of the estimation of the system output via ANN may not be improved with the further increase of the number of ANN iterations and that it induces a risk of overfitting if the ANN iteration continues. When the ANN iteration is terminated, we save ${\boldsymbol{\mathcal{W}}_i}^{(j)}$ for $i = 1 \sim 3$ as the final ANN model.

C) Uncertainty quantification using synthetic FDTD outputs
The final ANN model is utilized to make the predictions of $|\boldsymbol{E}|^2$. Let the matrix $\widetilde{\boldsymbol{\mathcal{X}}}(\mathcal{N}) = ({\tilde{\xi}_k}^{(n)}, n = 1 \sim \mathcal{N}, k = 1 \sim \mathcal{K})$ be a normalised sample set consisting of $\mathcal{N}$ sets of $\tilde{\boldsymbol{\xi}}^{(n)}$, where $\tilde{\boldsymbol{\xi}}^{(n)}$ indicates the n-th $\tilde{\boldsymbol{\xi}}$.
We use $\boldsymbol{\mathcal{X}}(\mathcal{M})$ to produce the final ANN model, then utilize some of $\widetilde{\boldsymbol{\mathcal{X}}}(\mathcal{N})$ as inputs to the final ANN model to predict $|\boldsymbol{E}|^2$. $\left| \boldsymbol{E}^{(m)} \right|^2$ corresponding to $\boldsymbol{\xi}^{(m)}$ is obtained from the FDTD simulation, whereas we do not run the FDTD simulation to acquire $\left| \boldsymbol{E}^{(n)} \right|^2$ corresponding to $\boldsymbol{\xi}^{(n)}$.

We calculate $\widehat{\boldsymbol{E}}$ in (2) using the final ANN model of ${\boldsymbol{\mathcal{W}}_i}^{(j)}$ for $i = 1 \sim 3$ obtained in StepB) of Section II-E, replacing $\boldsymbol{\mathcal{X}}(\mathcal{M})$ with $\widetilde{\boldsymbol{\mathcal{X}}}(\mathcal{N})$. $\widehat{\boldsymbol{E}}$ consists of $\mathcal{N}$ predictions $\widehat{E}^{(n)}$ of $|\boldsymbol{E}|^2$. These predictions are utilized to estimate the mean $\hat{\mu}(\mathcal{N})$ of $|\boldsymbol{E}|^2$ as in

$$\hat{\mu}(\mathcal{N}) = \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} \widehat{E}^{(n)} \tag{11}$$

and the standard deviation $\hat{\sigma}\ (\mathcal{N})$ of $|\boldsymbol{E}|^2$ as in

$$\hat{\sigma}\ (\mathcal{N})^2 = \frac{1}{\mathcal{N}-1} \sum_{n=1}^{\mathcal{N}} \left(\widehat{E}^{(n)} - \hat{\mu}(\mathcal{N}) \right)^2. \tag{12}$$

III. NUMERICAL EXPERIMENTS SETTING FOR UNCERTAINTY QUANTIFICATION

The MCM and the proposed method are utilized to quantify the uncertainty of the FDTD computation.

A. Radio environment setting

The simulation scenario is depicted in Fig. 2 where the FDTD space is $265 \times 490 \times 601$ voxels each with a resolution of 1 mm^3. The excitation is located 17 mm away from the human body and the observation location is in the middle of the prostate tissue. 10 layers of the complex frequency shifted-perfect matched layers (CFS-PML) [30], [31] are used to terminate the three-dimensional FDTD space. The DHP used in this work is provided by RIKEN (Saitama, Japan) [32] under the non-disclosure agreement between RIKEN and the University of Manchester. The usage was approved by RIKEN ethical committee.

A given human tissue is considered influential when the $|\boldsymbol{E}|^2$ at the observation location varies significantly due to variations of one or more of its four Debye parameters *i.e.* $\sigma_S, \epsilon_\infty, \epsilon_S, \tau$. The influential tissues for our scenario follows [33] wherein the influence of each of the four Debye parameters on $|\boldsymbol{E}|^2$ was investigated *i.e.* a sensitivity analysis was performed. The five influential tissues to be considered are: fat, skin, muscle, bone and prostate. Furthermore, the work in [33] indicates that the influences of σ_S and τ on $|\boldsymbol{E}|^2$ are negligible. As previously observed in [34], even though the conductivity variation has a negligible effect on $|\boldsymbol{E}|^2$, it has a clear impact on the SAR which is directly proportional to the product of the conductivity and $|\boldsymbol{E}|^2$. Our computation with the one-pole Debye model changes the conductivity by varying ϵ_∞. Thus, only two out of the four Debye parameters *i.e.* ϵ_∞ and $\epsilon_S - \epsilon_\infty \triangleq \Delta\epsilon$, are required for the simulations yielding a total of 10 (=2×5) uncertain input parameters. The procedures of generating samples for the 10 uncertain input parameters are as follows.

a) Generate 10^4 random ϵ_∞ and $\Delta\epsilon$ for each of 5 tissues yielding the normal distribution. Based on the work in [35], the variation of the relative permittivity and conductivity of each tissue should not exceed 10%. Thus, we consider the relative permittivity and the conductivity of each tissue vary within $\pm 10\%$. The probability descriptions of the 10 Debye parameters are presented in Table I.

ξ	Meaning of ξ	Average	Standard deviation
ξ_1	ϵ_∞ of bone	6.80	0.68
ξ_2	$\Delta\epsilon$ of bone	7.37	0.74
ξ_3	ϵ_∞ of skin	18.07	1.81
ξ_4	$\Delta\epsilon$ of skin	29.87	2.99
ξ_5	ϵ_∞ of muscle	28.93	2.88
ξ_6	$\Delta\epsilon$ of muscle	28.02	2.81
ξ_7	ϵ_∞ of fat	1.53	0.15
ξ_8	$\Delta\epsilon$ of fat	4.01	0.40
ξ_9	ϵ_∞ of prostate	32.82	3.28
ξ_{10}	$\Delta\epsilon$ of prostate	27.73	2.78

TABLE I
THE AVERAGE VALUES AND STANDARD DEVIATIONS OF THE 10 DEBYE PARAMETERS AND THEIR CORRESPONDING NOTATION

b) Randomly choose one sample out of the 10^4 samples for each uncertain Debye parameter and combine these chosen samples to produce 1 $\boldsymbol{\xi}$ composed of the 10 Debye parameters.

c) Repeat Stepb) 10^4 times to produce 10^4 $\boldsymbol{\xi}$. We do not choose the same samples as those chosen earlier.

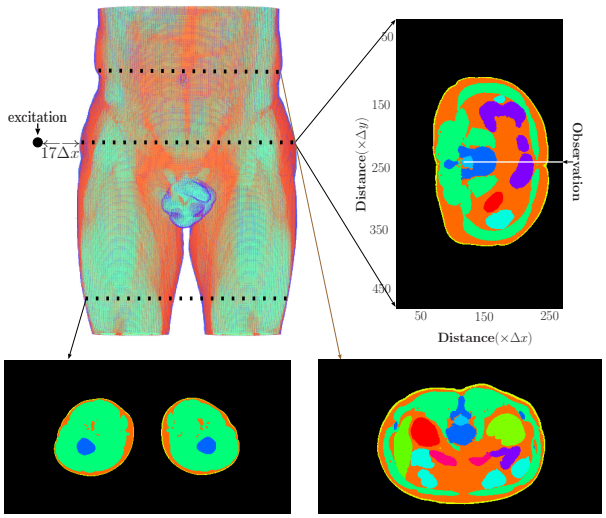

Fig. 2. Numerical simulation setup.

B. Numerical experiments for uncertainty quantification

a) The Monte Carlo method
The details of utilising the MCM for UQ can be found in [36], where the mean $\mu(\mathcal{M})$ and the standard deviation $\sigma(\mathcal{M})$ for $|\boldsymbol{E}|^2$ are calculated by

$$\mu(\mathcal{M}) = \frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \left| \boldsymbol{E}^{(m)} \right|^2 \tag{13}$$

and

$$\sigma(\mathcal{M})^2 = \frac{1}{\mathcal{M}-1} \sum_{m=1}^{\mathcal{M}} (\left| \boldsymbol{E}^{(m)} \right|^2 - \mu(\mathcal{M}))^2 \tag{14}$$

using the first $\mathcal{M}$ $|\boldsymbol{E}|^2$ among 10^4 $|\boldsymbol{E}|^2$ varying $\mathcal{M}$ from 2 to 10^4.

b) The ANN for UQ
Prior to the *training* of the ANN, we set its hyperparameters in order to maximize the accuracy of ANN, where the

number of layers of ANN is set to 2 and $G_1 = G_2$ are set to $5 = (\frac{\mathcal{K}}{2})$. Details regarding the hyperparameter tuning can be found in [37]. $\mathcal{M}_{tr}$ is set to 7, which makes the ratio of the size of the *training* set and the *validation* set close to the most commonly exercised 4:1 [38] in ANN. In our scenario, this ratio enables the ANN to have a stable performance and the potential overfitting problem can be properly detected.

We run the ANN iteration as described in Section II-C. When $j \geq I$, ${\mathcal{L}_v}^{(j)}$ is calculated in (5). There is no overfitting occurring during the first 200 ANN iterations. Thus, the ANN iteration is terminated when $\frac{{\mathcal{L}_v}^{(j+1)}}{{\mathcal{L}_v}^{(j)}} \geq \varpi$, where I and ϖ are set to 200 and 1.01, respectively. The setting of I is application-dependent and it can be influenced by many factors, such as the learning rate η and the size of dataset $\mathcal{M}$. A large value of η, such as $\eta > 0.01$, make the ANN learn from the data fast, resulting in the overfitting within a small number of ANN iterations, *i.e.* $j \leq 50$. Furthermore, the overfitting might occur when $\mathcal{L}_v$ stops decreasing. Setting ϖ to 1.01 allows us to detect the potential overfitting. When $j = 325$, $\frac{{\mathcal{L}_v}^{(j+1)}}{{\mathcal{L}_v}^{(j)}}$ becomes greater than 1.01. Therefore, we terminate the ANN iteration at the 325th ANN iteration and calculate ${\mathcal{L}_{ts}}^{(1)}$ in (6) using ${\boldsymbol{\mathcal{W}}_i}^{(325)}$ for $i = 1 \sim 3$ as the *trained* ANN model of the first LOO iteration.

$\mathcal{L}_l$ is calculated in (7) after 10 LOO iterations. The ANN is then re-trained using $\boldsymbol{\mathcal{X}}(10)$ as the *training* data. We terminate the ANN iteration when ${\mathcal{L}_{tr}}^{(j)} \leq \mathcal{L}_l$ and both (9) and (10) are met, where b is set to 0.01. The setting of b is application-dependent. 0.01 of b in our scenario is a reasonable value enabling us to detect the stable status of $\mathcal{L}_{tr}$. (9) and (10) are met when $j = 415$. Thus, we save ${\boldsymbol{\mathcal{W}}_i}^{(415)}$ for $i = 1 \sim 3$ as the final ANN model.

The final ANN model is utilized to make the predictions of $|\boldsymbol{E}|^2$. We use 10^4 sets of input samples in the MCM method. Therefore, we set $\mathcal{N}$ to 10^4 for comparison. $\widehat{\boldsymbol{E}}$ is calculated in (2) using ${\boldsymbol{\mathcal{W}}_i}^{(415)}$ for $i = 1 \sim 3$. $\hat{\mu}(\mathcal{N})$ and $\hat{\sigma}\ (\mathcal{N})$ are obtained in (11) and (12), respectively.

IV. Results and Discussions

We call the ANN with proposed adaptations as adaptive ANN. Section IV present $\mu(\mathcal{M})$, $\sigma(\mathcal{M})$, $\hat{\mu}(\mathcal{N})$, and $\hat{\sigma}\ (\mathcal{N})$ obtained from the MCM and the adaptive ANN. In order to evaluate the performance of the adaptive ANN, the input samples $\widetilde{\boldsymbol{\mathcal{X}}}(\mathcal{N})$, which are utilized to make predictions of $|\boldsymbol{E}|^2$ in the adaptive ANN, are the same as those used in the MCM of $\boldsymbol{\mathcal{X}}(\mathcal{M})$.

A. Results

1) Mean

Fig. 3 presents the $\mu(\mathcal{M})$ in (13) of the first $\mathcal{M}$ $|\boldsymbol{E}|^2$ obtained from the MCM and $\hat{\mu}(\mathcal{N})$ in (11) of the first $\mathcal{N}$ predictions of $|\boldsymbol{E}|^2$ obtained from the adaptive ANN varying $\mathcal{M}$ and $\mathcal{N}$ from 2 to 10^4. The accuracy of the estimation of the system output via the adaptive ANN can be calculated as $1 - \frac{\left|\hat{\mu}(10^4) - \mu(10^4)\right|}{\mu(10^4)} = 0.9963$.

Fig. 4 shows $\hat{\mu}(10^4)$ obtained from the adaptive ANN for 1000 experiments when $\mathcal{M} = 10$ and $\mathcal{K} = 10$. Each experiment produces a final ANN model using 10 sets of input samples to an ANN, which are chosen from 10^4 sets of input samples of the MCM based on the LHS method. The $\mu(10^4)$ from the MCM is presented as a straight line for comparison. The average and standard deviation of $\hat{\mu}(10^4)$ in Fig. 4 are 20.927 ± 0.098, while $\mu(10^4) = 20.925$.

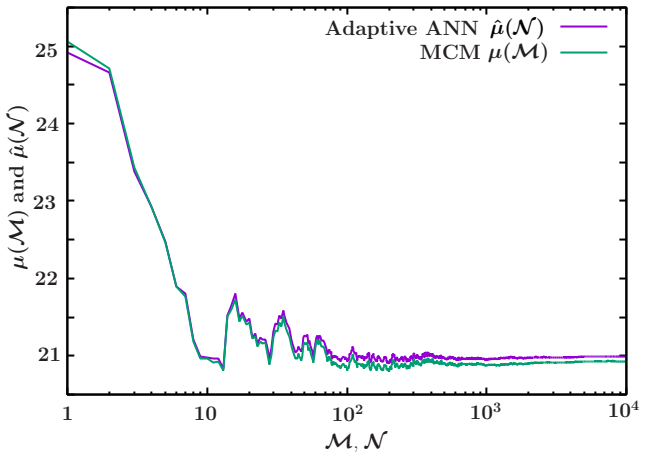

Fig. 3. $\mu(\mathcal{M})$ in (13) from the MCM and $\hat{\mu}(\mathcal{N})$ in (11) from the adaptive ANN.

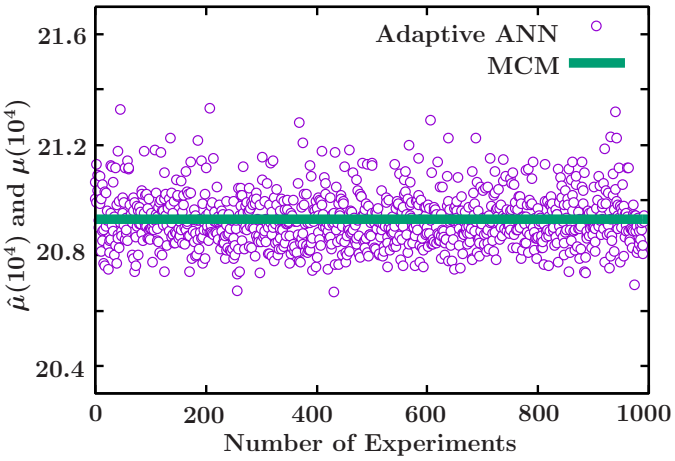

Fig. 4. $\hat{\mu}(10^4)$ in (11) from the adaptive ANN for 1000 experiments and $\mu(10^4)$ in (13) from the MCM.

2) Standard deviations

Fig. 5 presents $\sigma(\mathcal{M})$ in (14) of the first $\mathcal{M}$ $|\boldsymbol{E}|^2$ obtained from the MCM and $\hat{\sigma}\ (\mathcal{N})$ in (12) obtained from the adaptive ANN varying $\mathcal{M}$ and $\mathcal{N}$ from 2 to 10^4. The accuracy of the estimation of the system output via the adaptive ANN can be calculated as $1 - \frac{\left|\hat{\sigma}\ (10^4) - \sigma(10^4)\right|}{\sigma(10^4)} = 0.9883$.

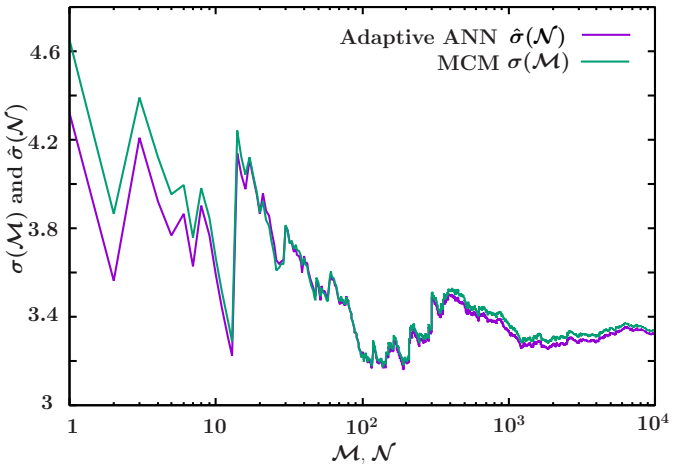

Fig. 5. $\sigma(\mathcal{M})$ in (14) from the MCM and $\hat{\sigma}$ $(\mathcal{N})$ in (12) from the adaptive ANN.

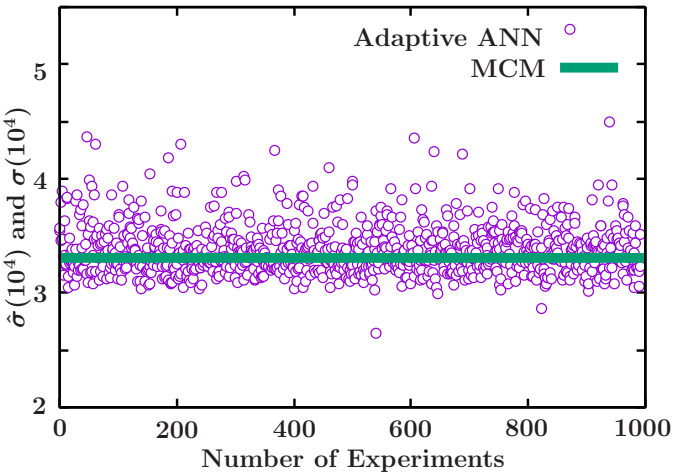

Fig. 6. $\hat{\sigma}$ (10^4) in (12) from the adaptive ANN for 1000 experiments and $\sigma(10^4)$ in (14) from the MCM.

Fig. 6 shows $\hat{\sigma}$ (10^4) obtained from the adaptive ANN for 1000 experiments when $\mathcal{M} = 10$ and $\mathcal{K} = 10$. $\sigma(10^4)$ from the MCM is presented as a straight line for comparison. The average and standard deviation of $\hat{\sigma}$ (10^4) in Fig. 6 are 3.375 ± 0.222, while $\sigma(10^4) = 3.348$.

B. Discussions

For the ANN based UQ techniques, the accuracy of UQ depends on the accuracy of the prediction of system output via ANN. The novelty of this paper is that we modify the activation function and the termination criteria of the ANN to maximize the accuracy in predicting the system output via ANN while maintaining a low computational cost. The following experiment is conducted to compare the proposed adaptive ANN with the traditional ANN from the viewpoint of the accuracy of the prediction of system output. We define the traditional ANN for regression analysis as the architecture which comprises of two hidden layers. The traditional ANN utilises the *linear* activation function of $f_l(\mathcal{T}) = \mathcal{T}$ and

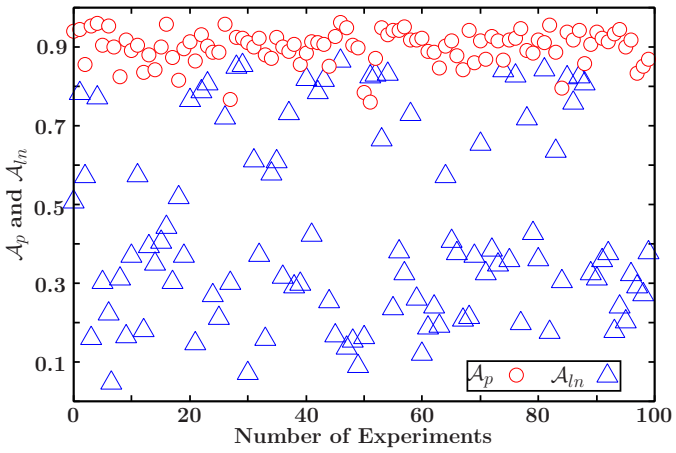

Fig. 7. 100 $\mathcal{A}_p$ from the adaptive ANN and 100 $\mathcal{A}_{ln}$ from the traditional ANN

terminates the ANN iteration when the training error becomes stable.

(a) Form the *test* data $\boldsymbol{\mathcal{X}}_{ts}(100)$ by randomly choosing 100 $\boldsymbol{\xi}$ out of the 10^4 $\boldsymbol{\xi}$ which are produced in Section III-A and obtain 100 $|\boldsymbol{E}|^2$ from the FDTD simulations using chosen $\boldsymbol{\xi}$.

(b) Randomly choose 10 $\boldsymbol{\xi}$ from the remaining 9900 $\boldsymbol{\xi}$. These chosen samples paired with their corresponding $|\boldsymbol{E}|^2$ from the FDTD simulations are used to build a surrogate model for the FDTD simulation using the adaptive ANN or the traditional ANN.

(c) Make predictions for $\boldsymbol{\mathcal{X}}_{ts}(100)$ from the two surrogate models obtained in (b). The accuracy of the adaptive ANN is calculated as in $\mathcal{A}_p = 1 - \frac{1}{100}\sum_{\widetilde{m}=1}^{100} \frac{\left|\widehat{E}_p^{\,(\widetilde{m})} - \left|\boldsymbol{E}^{(\widetilde{m})}\right|^2\right|}{\left|\boldsymbol{E}^{(\widetilde{m})}\right|^2}$, where $\widehat{E}_p^{\,(\widetilde{m})}$ is the prediction of $\left|\boldsymbol{E}^{(\widetilde{m})}\right|^2$ based on the adaptive ANN for $\widetilde{m} \in [1, 100]$. The accuracy of the traditional ANN is calculated as in $\mathcal{A}_{ln} = 1 - \frac{1}{100}\sum_{\widetilde{m}=1}^{100} \frac{\left|\widehat{E}_{ln}^{\,(\widetilde{m})} - \left|\boldsymbol{E}^{(\widetilde{m})}\right|^2\right|}{\left|\boldsymbol{E}^{(\widetilde{m})}\right|^2}$, where $\widehat{E}_{ln}^{\,(\widetilde{m})}$ is the prediction of $\left|\boldsymbol{E}^{(\widetilde{m})}\right|^2$ based on the traditional ANN.

We repeat the procedures of (b) and (c) 100 times to acquire 100 $\mathcal{A}_p$ and 100 $\mathcal{A}_{ln}$, each of which uses different 10 $\boldsymbol{\xi}$ to build the surrogate model based on the adaptive ANN or traditional ANN. 100 $\mathcal{A}_p$ and 100 $\mathcal{A}_{ln}$ obtained are presented in Fig. 7. The average and standard deviation of the 100 $\mathcal{A}_p$ and 100 $\mathcal{A}_{ln}$ are $89.94\% \pm 4.18\%$ and $44.21\% \pm 24.65\%$, respectively. This indicates that the proposed adaptations have the potential to significantly improve the accuracy of ANN in predicting the output of FDTD simulation while maintaining low computational cost.

Furthermore, we compare the proposed adaptive ANN with the state-of-the-art ANN-based UQ method proposed

in [39]. The work in [39] estimates the uncertainty of the SAR calculation through two neural networks, where the first neural network is designed based on the autoencoder neural network [40] for dimensionality reduction and the second neural network is a traditional ANN for UQ. We conduct the experiment to build surrogate models for the FDTD simulation using the proposed ANN and [39]. Fig. 8 shows the prediction accuracy of the surrogate model against the number of *training* data for these two methods, where the number of *training* data is varied from 10 to 1000 as in $\mathcal{M}_{tr} = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 500, 1000\}$. For each value of $\mathcal{M}_{tr}$, we construct a surrogate model using the proposed ANN and [39], respectively, and evaluate the prediction accuracy of the surrogate model on the *test* data $\boldsymbol{\mathcal{X}}_{ts}(100)$. From Fig. 8, it is observed that our proposed ANN outperforms [39], especially when the quantity of the *training* data is relatively small. It is important to note that generating one *training* data requires one system run. For the compute-intensive systems, it is impractical to utilize a large amount of *training* data for UQ. Accordingly, the proposed ANN demonstrates a high potential in efficiently handling the UQ problem for compute-intensive systems.

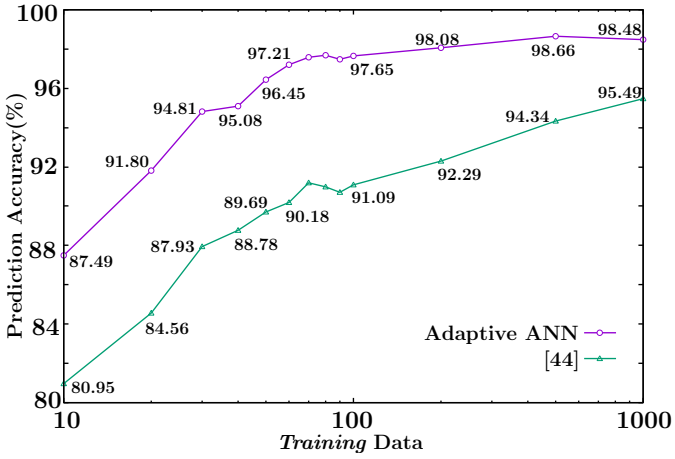

Fig. 8. Comparisons between the proposed ANN and [39] with the number of *training* data $\mathcal{M}_{tr} = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 500, 1000\}$

1) Comparisons with other UQ techniques:

The proposed adaptive ANN is utilized in UQ for the FDTD computation. We compare the proposed method with other state-of-the-art or classical UQ methods, which include the works in [39], [41] and [18], the NIPC expansion method and the traditional ANN, from the viewpoints of accuracy in UQ estimations and the computational efficiency. In the NIPC expansion method, we utilize the regression method to estimate the coefficients of polynomials, where the highest order of polynomials is set to 2. The work published in [41] builds a surrogate model using the sparse grid interpolation. We implement [41] using the sparse grid toolbox from the MATLAB [42], where the relative tolerance is set to 0.2 in our scenario. μ and σ obtained from these UQ techniques are presented in Table II, where the accuracy of μ and σ estimation is calculated by $1 - \frac{|\mu - \mu(10^4)|}{\mu(10^4)}$ and $1 - \frac{|\sigma - \sigma(10^4)|}{\sigma(10^4)}$, respectively.

The adaptive ANN outperforms other UQ techniques in terms of accuracy and computational efficiency which is measured by the number of FDTD simulations required. In our scenario, generating one set of input samples requires one run of the FDTD simulation with 5000 FDTD iterations, which takes about 1.4 hours to complete. 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 and the memory usage of one FDTD simulation was 9.0 GB .

The computational cost of an UQ technique is regarded as negligible compared with the one taken for FDTD simulation. The training process of ANN with 10 sets of input samples takes less than 10 seconds to complete. Therefore, when we evaluate the efficiency of UQ techniques, only the number of FDTD simulations required is taken into account. In Table II, the NIPC expansion method demands 66 FDTD simulations and achieves an accuracy of 78.11% for the σ estimation. Comparatively, our proposed method requires 10 FDTD simulations and achieves an accuracy of 98.83% for the σ estimation, which is about 6.6 times faster and 20% more accurate than the NIPC expansion method.

V. CONCLUSION

This paper proposes a number of adaptations for ANN to improve the accuracy of ANN in UQ for the FDTD computation while maintaining a low computational resources. We offer a versatile activation function to enable the ANN to effectively learn from the limited information. A series of termination criteria of ANN are proposed in order to maximize the accuracy of UQ. The main contribution of this paper lies in the significant improvement of the accuracy and the stability of UQ compared with the existing UQ techniques as alternatives to MCM.

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

[2] 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.

[3] M. Abalenkovs, F. Costen, J.-P. Bérenger, R. Himeno, H. Yokota, and M. Fujii, "Huygens subgridding for 3-D frequency-dependent finite-difference time-domain method," *IEEE Trans. Antenn. Propagat.*, vol. 60, no. 9, pp. 4336–4344, 2012.

[4] 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.

[5] A. Arduino, M. Chiampi, F. Pennecchi, L. Zilberti, and O. Bottauscio, "Monte carlo method for uncertainty propagation in magnetic resonance-based electric properties tomography," *IEEE Trans. Magn.*, vol. 53, no. 11, pp. 1–4, 2017.

UQ methods	FDTD simulations	μ	Accuracy of μ estimation	σ	Accuracy of σ estimation
NIPC method	66	20.942	99.92%	4.083	78.11%
Work in [41]	221	21.154	98.91%	3.267	97.52%
Work in [18]	30	20.720	99.02%	4.027	79.79%
Work in [39]	10	21.639	96.59%	3.922	82.93%
Our adaptive ANN	10	20.849	99.63%	3.311	98.83%
Traditional ANN	10	20.835	99.57%	2.983	89.06%

TABLE II
COMPARISONS OF THE PROPOSED METHOD WITH THE UQ METHODS OF NIPC METHOD, TRADITIONAL ANN, [39], [41] AND [18]

[6] 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.

[7] A. C. Austin and C. D. Sarris, "Efficient analysis of geometrical uncertainty in the fdtd method using polynomial chaos with application to microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 61, no. 12, pp. 4293–4301, 2013.

[8] D. Xiu and J. S. Hesthaven, "High-order collocation methods for differential equations with random inputs," *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 1118–1139, 2005.

[9] I. Jeong, B.-G. Gu, J. Kim, K. Nam, and Y. Kim, "Inductance estimation of electrically excited synchronous motor via polynomial approximations by least square method," *IEEE Trans. Ind. Appl.*, vol. 51, no. 2, pp. 1526–1537, 2015.

[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] I. M. Sobol, "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates," *Mathematics and computers in simulation*, vol. 55, no. 1-3, pp. 271–280, 2001.

[12] P. Rocca, N. Anselmi, A. Benoni, and A. Massa, "Probabilistic interval analysis for the analytic prediction of the pattern tolerance distribution in linear phased arrays with random excitation errors," *IEEE Trans. Antennas Propag.*, vol. 68, no. 12, pp. 7866–7878, 2020.

[13] L. Tenuti, N. Anselmi, P. Rocca, M. Salucci, and A. Massa, "Minkowski sum method for planar arrays sensitivity analysis with uncertain-but-bounded excitation tolerances," *IEEE Trans. Antennas Propag.*, vol. 65, no. 1, pp. 167–177, 2016.

[14] L. Poli, P. Rocca, N. Anselmi, and A. Massa, "Dealing with uncertainties on phase weighting of linear antenna arrays by means of interval-based tolerance analysis," *IEEE Trans. Antennas Propag.*, vol. 63, no. 7, pp. 3229–3234, 2015.

[15] M. Ahadi and S. Roy, "Sparse linear regression (spliner) approach for efficient multidimensional uncertainty quantification of high-speed circuits.," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 10, pp. 1640–1652, 2016.

[16] X. Yang, M. Choi, G. Lin, and G. E. Karniadakis, "Adaptive ANOVA decomposition of stochastic incompressible and compressible flows," *Journal of Computational Physics*, vol. 231, no. 4, pp. 1587–1614, 2012.

[17] C. Cui and Z. Zhang, "Stochastic collocation with non-gaussian correlated process variations: Theory, algorithms and applications," *IEEE Trans. Compon., Packag., Manuf. Technol.*, 2018.

[18] X. Cheng and V. Monebhurrun, "Application of different methods to quantify uncertainty in specific absorption rate calculation using a cad-based mobile phone model," *IEEE Trans. Electromagn. Compat.*, vol. 59, no. 1, pp. 14–23, 2017.

[19] A. Litvinenko, A. C. Yucel, H. Bagci, J. Oppelstrup, E. Michielssen, and R. Tempone, "Computation of electromagnetic fields scattered from objects with uncertain shapes using multilevel monte carlo method," *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, vol. 4, pp. 37–50, 2019.

[20] Z. Zhang, T.-W. Weng, and L. Daniel, "Big-data tensor recovery for high-dimensional uncertainty quantification of process variations," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 7, no. 5, pp. 687–697, 2017.

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

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

[23] 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.

[24] E. Soria-Olivas, J. D. Martn-Guerrero, G. Camps-Valls, A. J. Serrano-Lopez, J. Calpe-Maravilla, and L. Gomez-Chova, "A low-complexity fuzzy activation function for artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1576–1579, 2003.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] 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.

[31] 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.

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

[33] R. Hu, *Uncertainty Quantification for FD-FDTD Computations for the Human Body*. PhD thesis, University of Manchester, 2020.

[34] V. Monebhurrun, C. Dale, J.-C. Bolomey, and J. Wiart, "A numerical approach for the determination of the tissue equivalent liquid used during SAR assessments," *IEEE transactions on magnetics*, vol. 38, no. 2, pp. 745–748, 2002.

[35] A. M. Abduljabbar, M. E. Yavuz, F. Costen, R. Himeno, and H. Yokota, "Continuous wavelet transform-based frequency dispersion compensation method for electromagnetic time-reversal imaging," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 3, pp. 1321–1329, 2017.

[36] K. Binder, D. Heermann, L. Roelofs, A. J. Mallinckrodt, and S. McKay, "Monte carlo simulation in statistical physics," *Computers in Physics*, vol. 7, no. 2, pp. 156–157, 1993.

[37] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter, and A. Brenning, "Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data," *Ecological Modelling*, vol. 406, pp. 109–120, 2019.

[38] I. Guyon, "A scaling law for the validation-set training-set size ratio," *AT&T Bell Laboratories*, pp. 1–11, 1997.

[39] X. Cheng, C. Henry, F. P. Andriulli, C. Person, and J. Wiart, "A surrogate model based on artificial neural network for RF radiation modelling with high-dimensional data," *International journal of environmental research and public health*, vol. 17, no. 7, p. 2586, 2020.

[40] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[41] J. S. Ochoa and A. C. Cangellaris, "Random-space dimensionality reduction for expedient yield estimation of passive microwave structures," *IEEE Trans. Microw. Theory Tech.*, vol. 61, no. 12, pp. 4313–4321, 2013.

[42] Alexander, "Sparse Grid Interpolation," https://www.mathworks.com/matlabcentral/fileexchange/54289-sparse-grid-interpolation. Accessed: Sept. 3, 2019.