High-Order Deep Recurrent Neural Network With Hybrid Layers for Modeling Dynamic Behavior of Nonlinear High-Frequency Circuits

Fatemeh Charoosaei, Mostafa Noohi, Sayed Alireza Sadrossadat, *Member IEEE*, Ali Mirvakili, Weicong Na, *Member IEEE*, and Feng Feng, *Member IEEE*

***Abstract*— In this paper, a new technique for macromodeling of high-frequency circuits and components called high-order deep recurrent neural network (HODRNN) is proposed. This technique explores an alternative approach to learn RNN for time dependencies in more efficient way resulting in more accurate model. HODRNN uses more memory units to track previous hidden states, all of which are returned to the hidden layers as feedback through various weight paths. Moreover, new improved structure called Hybrid-HODRNN is proposed for further increasing the modeling accuracy of HODRNN. The proposed Hybrid-HODRNN uses hybrid layers with both single and high orders for taking advantage of HODRNN and also reducing the overfitting problem which finally leads to more accurate model. Additionally, the proposed method requires less training signals compared to the conventional shallow and deep RNNs in order to create a model with similar accuracy. Also, the obtained models from the proposed method are considerably faster than the transistor-level models while having similar accuracy. By modeling three high-frequency circuits in this paper, we conclude that the HODRNN and its Hybrid structure offer the ability to create a better macromodel of high-frequency nonlinear circuits than the conventional RNNs which verifies the superiority of the new macromodeling techniques.**

***Index Terms*—** Computer-aided design (CAD), deep learning, high-frequency circuits, high-order recurrent neural network, hybrid structure, macromodeling, nonlinear component, recurrent neural network.

I. Introduction

THE use of computer-aided methods to assist the design and modeling methodologies is an ever-growing trend for high-frequency circuits. This process has the potential to optimize and simplify the designer's workflow, increase productivity, and improve the quality of the design, to name a few. In addition, with the continuous increase in the speed and the frequency of the signals, it is important to verify the signal integrity (SI) while the data signals propagate through the system. In this manner, developing accurate as well as fast macromodels is crucial in the signal integrity-based design and analysis of such circuits and systems [1]-[3]. In this regard, artificial neural networks (ANN) have strongly contributed to the growth and development of computer-aided design (CAD) methods particularly for the development of high-frequency and microwave components and circuits. In addition, ANN-based CAD models have led to a significant improvement in the speed and efficiency of modeling, simulation, and optimization of such circuits and components [4], [5].

There is a particular type of ANN called recurrent neural network (RNN) that can be trained using input-output waveforms derived from time-domain measurements or simulations without the need to know the details of the primary circuit [6]-[9]. Also, the global approximation theorem for this structure states that RNN can model any nonlinear input-output relationship with any desired accuracy [10]. This structure, moreover, can capture dependency on consecutive data using a simple feedback mechanism. RNN modeling can also be suitable and powerful for macromodeling of nonlinear circuits [7], [11]. Most artificial neural network-based modeling methods have a simple feedforward structure that cannot accurately record the time-domain behavior of a nonlinear device. RNN which is considered as one of the most advanced methods in the field of nonlinear macromodeling [6]-[8], uses the recurrency mode as its feedback to more accurately model the dynamic behavior of nonlinear components. However, the shallow type of this neural network has difficulty capturing complex input-output relationships in nonlinear circuits. In order to reduce the abovementioned problem, a deep RNN (DRNN) structure is introduced [6]. This DRNN structure is composed of layers of accumulated recurrent neural networks that can embed different time intervals directly in its structure, and better model input-output complex relationships [12]. RNNs are very deep in time, and thus due to the problem of gradient disappearance in back-propagation, it turns out that learning process of RNN has difficulty in modeling the long-term dependence on consecutive data [13].

Two common and distinct approaches for nonlinear component modeling are behavioral and transistor-level models such as Spice models [9]. It is worth mentioning that neural networks are able to accurately receive nonlinearities while being as fast and accurate as aforementioned methods [14], [15].

This work was supported in part by the National Natural Science Foundation of China (Grant no. 62101382), the National Natural Science Foundation of China (Grant no. 61901010), and Scientific Research Project of Beijing Educational Committee (Grant no. KM202110005029).
(*Corresponding author: Weicong Na.*)

F. Charoosaie and S. A. Sadrossadat are with the Department of Computer Engineering, Yazd University, Yazd, Iran (email: charoosaei.fatemeh@gmail.com; alireza.sadr@yazd.ac.ir).

M. Noohi and A. Mirvakili are with the Department of Electrical Engineering, Yazd University, Yazd, Iran (email: m.noohi75@yahoo.com; ali.mirvakili@yazd.ac.ir).

W. Na is with the Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China (email: weicongna@bjut.edu.cn).

F. Feng is with the School of Microelectronics, Tianjin University, Tianjin, 300072, China (e-mail: ff@tju.edu.cn).

Regarding the modeling of nonlinear circuits and devices, following works have been reported: In [1], a state-space dynamic neural network (SSDNN) was introduced to model the transient behaviors of high-speed nonlinear circuits. In addition, stability analysis for neural modeling of nonlinear microwave circuits together with a new constrained training algorithm is introduced to satisfy the requirements of accuracy and stability of the resulting SSDNN models. In [2], using the internal space of an RNN, called the internal input-neuron space, a technique for developing the RNN structure in circuit modeling was proposed. Additionally, this technique is capable of reducing the amount of training data while maintaining the necessary modeling information. In [3], the adjoint SSDNN technique for modeling the behavior of nonlinear electronic and photonic components was introduced. This method is a newer version of SSDNN which adds the derivative information to the training patterns of nonlinear components making the training faster and more efficient while using less data without sacrificing model accuracy. In [6], a training algorithm based on real-time recurrent learning (RTRL) is introduced in order to train the deep RNN. In [7], a local feedback deep recurrent neural network (LFDRNNs) is presented to model the nonlinear behavior of a converter. This macromodeling method by using local feedbacks can be trained without knowing the internal details of the main circuit, and provides similar accuracy. In [8], the adjoint recurrent neural network (ARNN) technique is presented which adds derivative information to the training data set, yielding a more efficient and accurate training, while using fewer data. In [11], a macromodeling approach based on the RNN structure was proposed to learn the dynamic responses of nonlinear microwave circuits. In [16], it is stated that one of the most critical issues in a complex structure is reliability. Many factors such as package structure sizes, material properties, and process technologies are influential factors. In this reference, instead of using costly experimental tests such as thermal cycle testing, RNN is used which is a good choice for predicting electronic packaging reliability. The evolution of RNN structures to model the process of designing and analyzing electronic circuits and components has already been increased. In [17], RNN is used to simulate a high-speed channel in the time-domain that can manage the nonlinear behavior of I/O buffers. In [18], the RNN technique was utilized to obtain the behavioral modeling of a power amplifier with short-term and long-term memory effects. In [19], local-global feedback recurrent neural network (LGFRNN) is introduced which can be used for dynamic behavioral modeling of nonlinear circuits. This model is composed of time-delayed local and global feedbacks, and can alleviate the issue of slow convergence of the training phase.

This paper explores an alternative and more efficient approach for capturing dynamic behavior of nonlinear high-frequency circuits. In order to grasp this target, the conventional RNN is extended by using more memory units for tracking previous hidden states, all of which are fed back to the hidden layers as feedback through different weight paths. Indeed, the RNN standard structure is enriched to better model long-term dependencies in sequential data. This new structure is called high-order deep recurrent neural network (HODRNN). At each time stage, the proposed HODRNN directly combines several previous hidden states from different historical time stages by weighting different matrices to generate a feedback signal for each hidden layer. This novel macromodeling approach showed the reduction of testing error in less training time by using fewer layers and neurons per layer than the conventional RNN modeling method. Besides, the HODRNN method required less training data than the conventional RNN to generate models with similar accuracy, and provided higher speed compared to the transistor-level models. In this paper, moreover, an improved version of HODRNN called Hybrid-HODRNN, a combination of HODRNN layers with high and fewer orders, is proposed which not only reduces the testing errors due to overfitting suppression, but also requires lower training and testing time than the HODRNN modeling method due to its more efficient structure and training procedure. Additionally, the Hybrid-HODRNN method requires less training data than conventional RNN to generate similar accurate models. Also, it demonstrates higher speed-up compared to both the HODRNN-based and transistor-level models. With this proposed HODRNN and its hybrid structure, we are able to model complex high-frequency nonlinear and microwave circuits. This paper is organized as follows. Section II describes the background regarding the conventional RNN. Section III introduces the proposed HODRNN macromodeling approach. In this section, the structure of HODRNN together with its training procedure is discussed. Then, the novel macromodeling technique of Hybrid-HODRNN is introduced. Numerical results are provided in section IV. Eventually, section V concludes this paper.

II. BACKGROUND

A. Nonlinear circuit dynamics formulation

Assume N_I, N_O, and N_P to be the total numbers of input signals, output signals, and the parameters of the nonlinear circuit, respectively. Also, assume the vectors $\widehat{\boldsymbol{O}} = \left[\widehat{\boldsymbol{O}}_1 .. \widehat{\boldsymbol{O}}_{N_O}\right]$, $\boldsymbol{I} = \left[\boldsymbol{I}_1 ... \boldsymbol{I}_{N_I}\right]$ and $\boldsymbol{P} = [\boldsymbol{P}_1 ... \boldsymbol{P}_{N_P}]$ be the output signals, input signals and the circuit parameters of the nonlinear circuit, respectively.

The characteristics of the original nonlinear circuit can be described as a nonlinear state-space form as,

$$\begin{aligned} Z(t) &= G(Z(t), I(t), P, t) \\ \widehat{O}(t) &= E(Z(t), I(t), t) \end{aligned} \tag{1}$$

where $\boldsymbol{Z} = \left[\boldsymbol{Z}_1 ... \boldsymbol{Z}_{N_Z}\right]^T$ and N_z are the vectors of state variables and numbers of states, respectively. Also, G and E are nonlinear functions [11], [19]. For a nonlinear complex component, the original nonlinear equations described in (1) will be computationally extensive to solve. During design and optimization, repetitive evaluation of the circuit for different inputs is needed. As a result, a more convenient computational form that is easier to solve than the original complicated equations is required. This can be accomplished by converting the original complex equations to discrete-time one having a specific sampling rate such as recurrent neural network [6], [20]-[27].

B. Conventional Recurrent Neural Network

Conventional RNN is a type of neural network where each neuron's output from previous time step is fed as input to the current time step. In feedforward neural networks, there is no feedback or time-dependency on each neuron. However, in cases where the inputs/outputs are time-dependent and prediction of next time step is required, a recurrent structure such as RNN is required to be able to model this behavior [28]. RNN by use of a hidden state, remembers information of the past time steps. Each neuron's output equation in the conventional RNN is described as follows,

$$Z_i^l(t)=\sigma\left(\sum_{j=1}^{N_{l-1}} W_{i,j}^{l,F} Z_j^{l-1}(t)+\sum_{j=1}^{N_l} W_{i,j}^{l,R} Z_j^l(t-1)+b_i^l\right) \quad (2)$$

$$1\le i\le N_l, 1\le l\le L$$

where σ is the sigmoid activation function, $W_{i,j}^{l,F}$ is the feedforward weight connecting i^{th} neuron of layer l to j^{th} neuron of layer l-1 (both neurons at time step t), $W_{i,j}^{l,R}$ is the recurrent weight connecting i^{th} neuron of layer l at time step t to the j^{th} neuron of the same layer at time step t-1, $Z_j^{l-1}(t)$ is the output of j^{th} neuron of layer l-1 at time step t, $Z_j^l(t-1)$ is the output of j^{th} neuron of layer l at time step t-1, and b_i^l is the bias for i^{th} neuron of l^{th} layer. It is worth mentioning that backpropagation through time (BPTT) method should be used for calculating the gradient of RNN error function with respect to its weights [7], [29], [30]. The algorithm works its way backward through various layers and various times to find the partial derivative of the error with respect to the weights. Then, these gradients will be used to update the weights for the next iteration in order to decrease error in the training procedure.

III. The Proposed High-Order Deep Recurrent Neural Network Macromodeling Method

A. The drawback of the past modeling methods and the benefits of HODRNN

In the past few years, the ANNs have been used as a valuable computer aided design tool for modeling high frequency circuits by developing models based on input-output data.

The past macromodeling methods in the literature have some drawbacks. For instance, training the SSDNN requires solving nonlinear differential equations which makes it very time-consuming [1]. In addition, even though the adjoint SSDNN technique adds the derivative information to the training patterns of nonlinear components, and makes the training faster and more efficient without sacrificing model accuracy, this method relies on solving complex nonlinear differential equations and still suffers from time-consuming training and testing procedures. Moreover, the conventional RNN methods requires backpropagation through time procedure to train the network. Therefore, their gradients will vanish through time and they cannot capture the behavior of nonlinear circuits for long time dependencies. As a result, they provide less accurate models compared to HODRNN macromodeling method and their training trajectory will be slower due to inefficient gradient updates [13], [36].

One of the successful approaches to deal with vanishing gradient problem is long-short terms memory (LSTM) macromodeling approach [9]. LSTM relies on a fairly sophisticated structure made of gates to control the information flow to hidden units alleviating the problem of vanishing gradients. Despite its advantages, it has complicated structure with many trainable parameters and consequently slow to be trained which makes it hard to be scaled for larger tasks compared to HODRNN [13].

The proposed HODRNN macromodeling method in this paper takes the advantage of simplicity in the conventional RNN structure, but expands it to use more memory units. Therefore, the HODRNN has less trainable parameters compared to LSTM leading to faster training and less overfitting problem and consequently less testing error. Also, by using more memory units compared to conventional RNN, it can capture longer term dependencies and more complex input-output relationships. Therefore, the proposed HODRNN macromodeling method has both advantages of RNN and LSTM at the same time.

Additionally, in this paper the performance of the proposed HODRNN was improved by introducing the Hybrid-HODRNN structure which is a further advance over the proposed HODRNN method. Using the Hybrid method which will be explained later in details, not only the mentioned benefits of HODRNN will be kept intact but also in case of overfitting, it can help the HODRNN to reduce the overfitting impact on model accuracy.

B. Structure of High-Order Deep Recurrent Neural Network

High-order deep RNN (HODRNN) with order one is similar to that of conventional deep RNN on condition that only first order (dependence on only one previous time step) is utilized. However, the proposed HODRNN can consider multiple orders which means the dependency of current time to multiple previous time steps, and that is why it has a different structure compared to the conventional deep RNN. The proposed HODRNN can be described by the following equations, where $Z_1^0(t)\cdots Z_{N_l}^0(t)$ correspond to the network external inputs in the input layer.

$$Z_i^l(t)=\sigma\left(\begin{array}{l}\sum_{j=1}^{N_{l-1}} W_{i,j}^{l,F} Z_j^{l-1}(t)+\\ \sum_{m=1}^{M}\left(\sum_{j=1}^{N_l} W_{i,j}^{l,R_m} Z_j^l(t-m)\right)+b_i^l\end{array}\right) \quad (3)$$

$$1\le i\le N_l \quad , \quad 1\le l\le L$$

$$O(t)=\sum_{j=1}^{N_L} W_{1,j}^{L,F} Z_j^L(t) \quad (4)$$

where (3) and (4) are the output of hidden layer and output of the neural network, respectively (for simplicity only one-output model is considered).

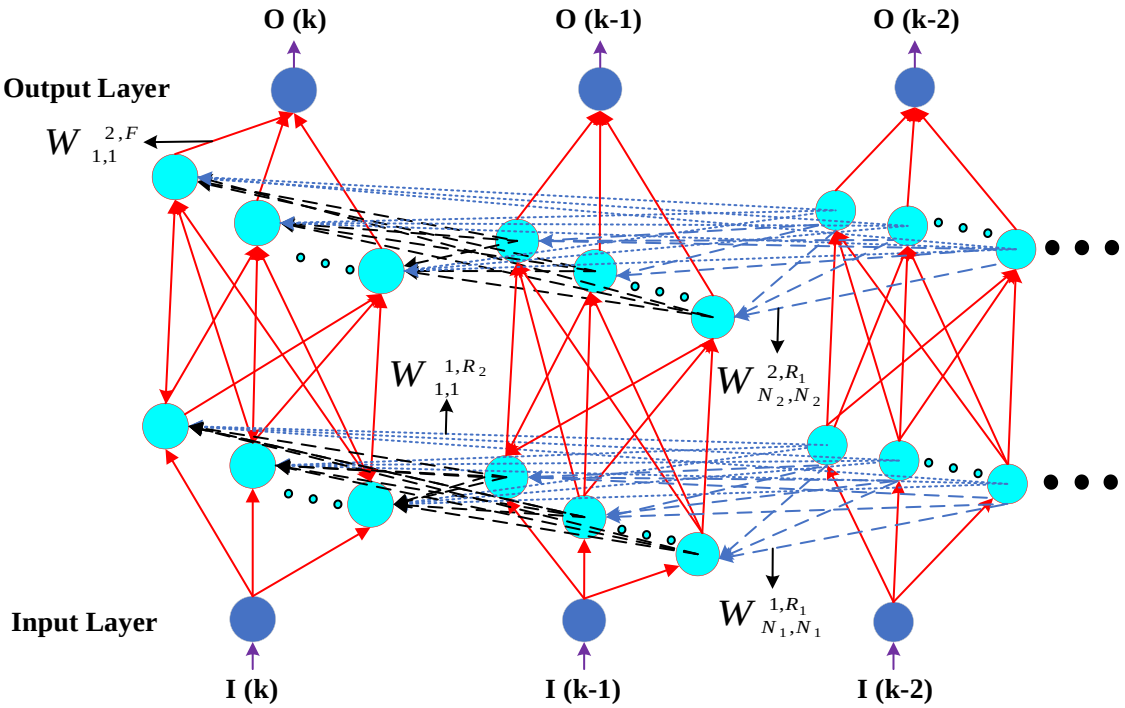


Fig. 1. High-order deep recurrent neural network structure with two hidden layers and order 2.

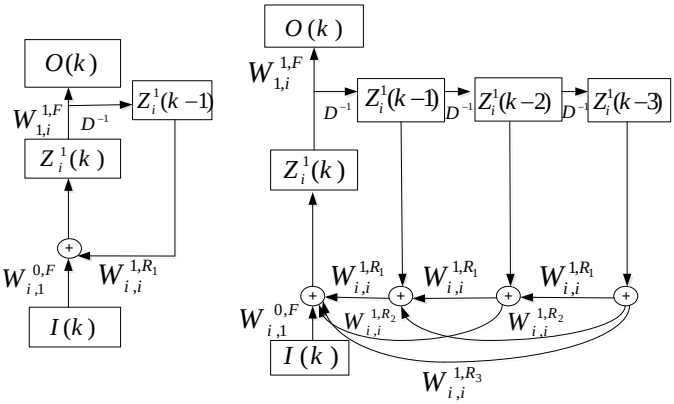


Fig. 2. Comparison of model structures between conventional RNN (first-order) and HODRNN of third-order. The D^{-1} denotes a time-delay unit.

In these equations L represents the number of hidden layers, M indicates the order of HODRNN, $W_{i,j}^{l,F}$ is the feedforward weight which connects i^{th} neuron of layer l to j^{th} neuron of layer l-1 (both neurons at time step t), and $W_{i,j}^{l,R_m}$ is the recurrent weight connecting i^{th} neuron of layer l at time step t to the j^{th} neuron of the same layer at time step t-m.

The structure of HODRNN with two hidden layers and order 2 is shown in Fig. 1. For simplicity, only the recurrent connections between time step (k) and (k-2) are shown in this figure. The comparison between the structures of conventional RNN (first-order) and HODRNN (third-order) is shown in Fig. 2. As it can be seen from this figure, a third-order HODRNN is unfolded in time which clearly indicates that the output of each hidden neuron at present time (current hidden neuron state) is explicitly decided by the input at current time, $I(t)$, and all three previous hidden neuron states.

C. Training of High-Order Deep Recurrent Neural Network

The HODRNN structure should be trained to make an appropriate model representing the dynamic behavior of a nonlinear circuit or component. There are two major steps in this training procedure: error and gradient calculations [19], [25]. As it can be understood from (7) and (8), the gradients of HODRNN error function with respect to its weights propagate backward through feedback paths to each hidden neuron state in the unfolded RNN [31]-[34]. Indeed, HODRNN can be learned using the same BPTT algorithm as conventional RNN, except that the error function at each time step need to be back-propagated to multiple feedback paths in the network rather than one path. The structure of the HODRNN must be trained by a set of input-output waveforms called training data obtained from either simulation or measurement [35]-[37].

Indeed, training is an optimization process which its objective function, called error function, is defined as follows:

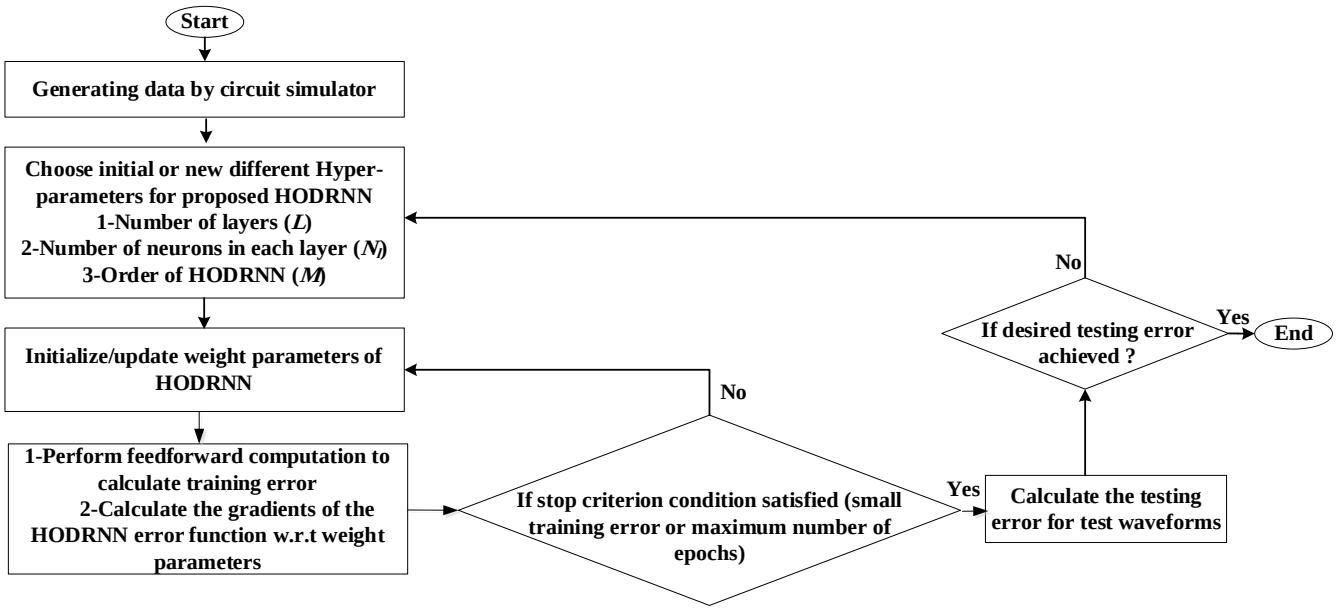


Fig. 3. Flowchart of the mcromodeling procedure by the proposed HODRNN method.

$$E_s = \frac{1}{2}\sum_{j=1}^{N_O}\sum_{k=1}^{N_t}(O_{j,s}(k) - \hat{O}_{j,s}(k))^2 \qquad (5)$$

where (5) is the error for the s***th*** training waveform, N_O *is* the number of outputs of the model and N_t denotes the number of time steps. Total error for all training waveforms is given as follows:

$$E = 1/2\sum_{s=1}^{N_s}\sum_{j=1}^{N_o}\sum_{k=1}^{N_t}(O_{j,s}(k) - \hat{O}_{j,s}(k))^2 \qquad (6)$$

where N_s is the number of training waveforms, $\hat{O}_{j,s}$ is the desired j^{th} output of original circuit for s^{th} training data, and $O_{j,s}$ is its corresponding predicted output waveform of HODRNN. Keep in mind that N_t is the total number of time steps sampled from the original training signals (obtained by simulation software) which could be hundreds of time steps (200, 300, …). However, *M* is the order of HODRNN (could be 3, 4, 5 …) which is the number of time steps that we should look back in time to calculate the state of the network at time step *t*.

The objective for training the model is to minimize *E* by adjusting feedforward and recurrent weights. HODRNN is trained using the gradient-based optimization technique. For employing efficient gradient-based optimization methods, derivatives of the error function with respect to each parameter of HODRNN should be obtained. Besides, due to the dependence of output at current time to the outputs of the multiple previous times, it is not applicable to utilize the same error back propagation approach used for conventional recurrent neural networks [7], [38]-[40].

For simplicity of calculating HODRNN gradients, assume we have only one input and one output. Accordingly, to calculate the derivative of the error *E* in terms of the interlayer feedforward weight $W_{i,j}^{l,F}$, the following recursive formula can be obtained.

$$\partial E_s(k)/\partial W_{i,j}^{l,F} = \frac{\partial E_s(k)}{\partial O}\times\sum_{p=1}^{N_L}\frac{\partial O}{\partial Z_p^L(k)}\times\frac{\partial Z_p^L(k)}{\partial W_{i,j}^{l,F}} \qquad (7)$$

where $Z_p^L(k)$ is the output of p^{th} neuron of layer *L* at time step *k*. To illustrate the recursive operation (chain rule), we extended the (7) into several steps as follows:

$$\frac{\partial O}{\partial Z_p^L(k)} = W_{1,p}^{L,F} \qquad (8)$$

According to (3),

$$\frac{\partial Z_p^L(k)}{\partial W_{i,j}^{l,F}} = \sigma'\times\left(\begin{array}{l}\sum_{q=1}^{N_{L-1}} W_{p,q}^{L-1,F}\times\frac{\partial Z_q^{L-1}(k)}{\partial W_{i,j}^{l,F}} + \\ \sum_{m=1}^{M}\left(\sum_{q=1}^{N_L} W_{p,q}^{L-1,R_m}\times\frac{\partial Z_q^L(k-m)}{\partial W_{i,j}^{l,F}}\right) \\ + Z_j^{L-1}(k)\ \ (only\ if\ \ l = L-1)\end{array}\right) \qquad (9)$$

where,

$$\frac{\partial Z_q^{L-1}(k)}{\partial W_{i,j}^{l,F}} = \sigma'\times\left(\begin{array}{l}\sum_{r=1}^{N_{L-2}} W_{q,r}^{L-2,F}\times\frac{\partial Z_r^{L-2}(k)}{\partial W_{i,j}^{l,F}} + \\ \sum_{m=1}^{M}\left(\sum_{r=1}^{N_{L-1}} W_{q,r}^{L-2,R_m}\times\frac{\partial Z_r^{L-1}(k-m)}{\partial W_{i,j}^{l,F}}\right) \\ + Z_j^{L-2}(k)\ \ (only\ if\ \ l = L-2)\end{array}\right) \qquad (10)$$

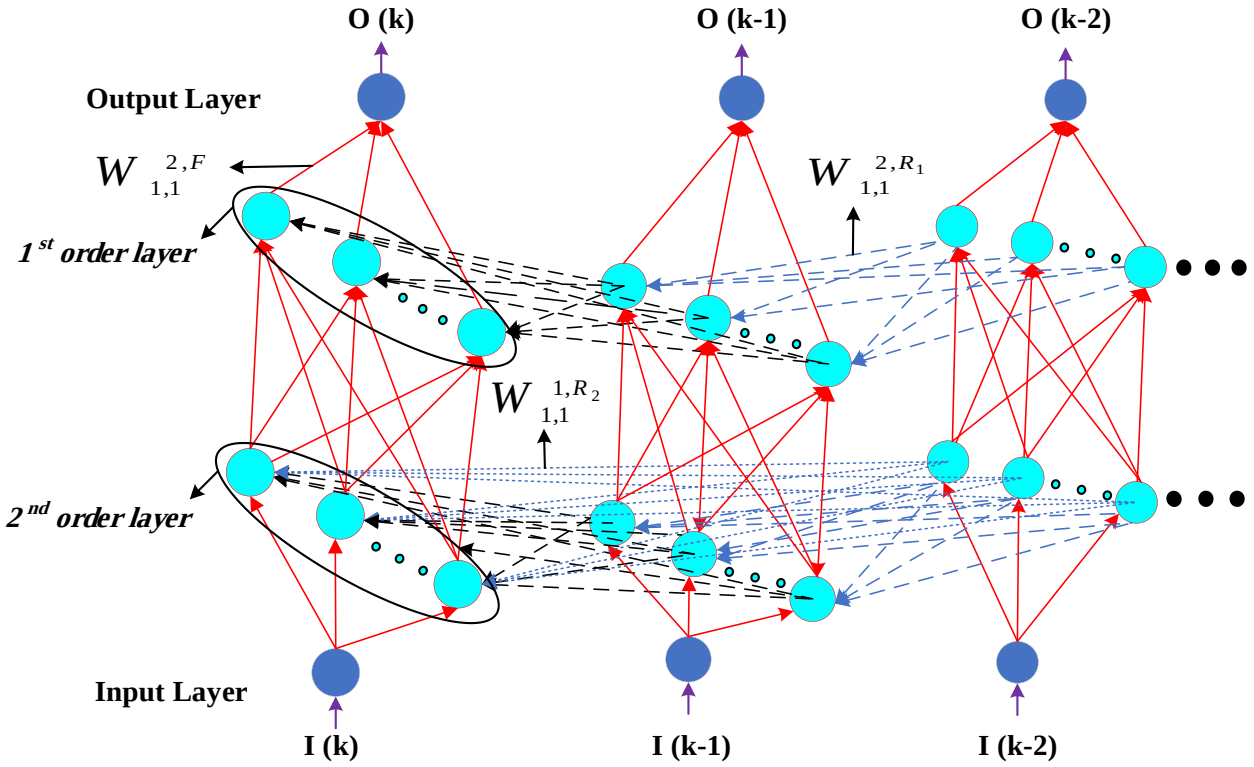


Fig. 4. Hybrid-HODRNN structure with two hidden layers: hidden layer one has order 2 and the hidden layer two has order 1.

and,

$$\frac{\partial Z_q^L(k-d)}{\partial W_{i,j}^{l,F}} = \sigma' \times \left(\begin{array}{l} \sum_{r=1}^{N_{L-1}} W_{q,r}^{L-1,F} \times \frac{\partial Z_r^{L-1}(k-d)}{\partial W_{i,j}^{l,F}} + \\ \sum_{m=1}^{M} \left(\sum_{r=1}^{N_L} W_{q,r}^{L-1,R_m} \times \frac{\partial Z_r^L(k-m-d)}{\partial W_{i,j}^{l,F}} \right) \\ + Z_j^{L-1}(k-d) \ \ (only\ if \ \ l = L-1) \end{array} \right) \quad (11)$$

$$1 \le d \le M$$

Similarly, we can calculate $\frac{\partial Z_p^{L-n}(k)}{\partial W_{i,j}^{l,F}}$ (where L-n is the hidden layer number, $2 < n < L$) and $\frac{\partial Z_r^L(k-m-d)}{\partial W_{i,j}^{l,F}}$ for other layers and times to get the exact gradients. Equation (12) will be calculated in this regard.

$$\frac{\partial E_s(k)}{\partial W_{i,j}^{l,F}} = \left(O(k) - \hat{O}(k)\right) \times \sum_{p=1}^{N_L} \left(W_{1,p}^{L,F} \times \sigma' \times \left(\begin{array}{l} \sum_{q=1}^{N_{L-1}} W_{p,q}^{L-1,F} \times \frac{\partial Z_q^{L-1}(k)}{\partial W_{i,j}^{l,F}} + \\ \sum_{m=1}^{M} \left(\sum_{q=1}^{N_L} W_{p,q}^{L-1,R_m} \times \frac{\partial Z_q^L(k-m)}{\partial W_{i,j}^{l,F}} \right) \\ + Z_j^{L-1}(k) \ \ (only\ if \ \ l = L-1) \end{array} \right) \right) \quad (12)$$

In (9), (10), (11), and (12) the last term will be added only when l=L-1, L-2, L-1, and L-1, respectively and σ' stands for the derivative of the activation function.

To calculate the derivative of the error E_s with respect to the recursive weights of each neuron (inter-time weights $W_{i,j}^{l,R_m}$), similar procedure should be performed.

Fig. 3 demonstrates the macromodeling procedure using the proposed HODRNN technique. Once a HODRNN macromodel is trained, it can then represent the parametric and dynamic input–output relationship of the original nonlinear circuit. So, it is evident that we can train HODRNN by using the BPTT just as conventional deep RNN but considering multiple time-dependencies and more complicated gradients [39, 32, 33, 41].

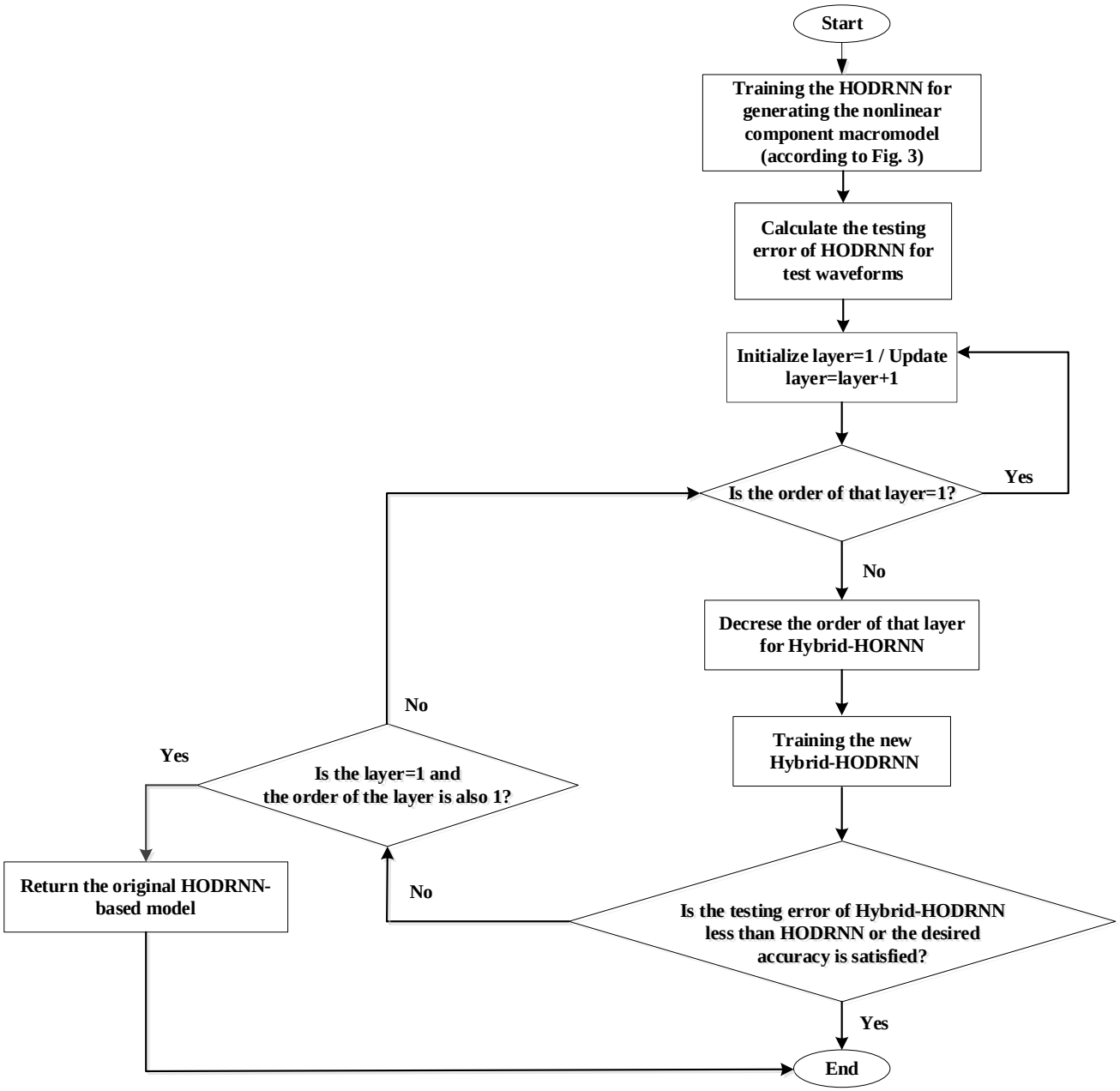


Fig. 5. Flowchart of the macromodeling procedure by the proposed Hybrid-HODRNN method.

D. Hybrid High-Order Deep Recurrent Neural Network (Hybrid-HODRNN)

To further improve the HODRNN performance, we have proposed the Hybrid-HODRNN structure combining conventional deep RNN with HODRNN techniques. In this way, the hidden layers of Hybrid-HODRNN can have different orders resulting in better model accuracy. As HODRNN structure contains many weights through different feedforward layers and multiple recurrent hidden states, by reducing the order of some of the hidden layers, number of parameters will be reduced while keeping the strength of higher orders for macromodeling of complex nonlinear circuits. Therefore, overfitting will be reduced resulting in less testing error and more accurate model. Thus, by combining high and low orders for different hidden layers, the proposed Hybrid-HODRNN method has the potential to significantly improve the conventional RNN macromodeling performance.

Fig. 4 shows the structure of Hybrid-HODRNN with two hidden layers where the first layer is of order 2 and the second layer has the simple 1st order structure. For simplicity, only the recurrent connections between time step (k) and (k-2) are shown in this figure. Moreover, Fig. 5 demonstrates the flowchart of nonlinear circuit macromodeling using the proposed Hybrid-HODRNN technique. The Hybrid-HODRNN can be described according to the following formula for a two hidden layers network.

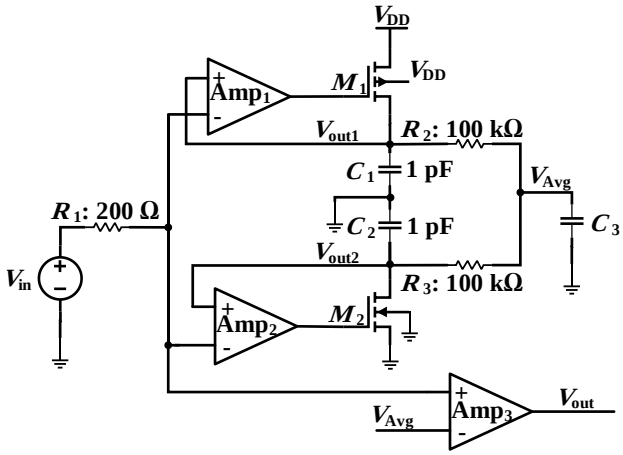


Fig. 6. A high-frequency DC generation circuit.

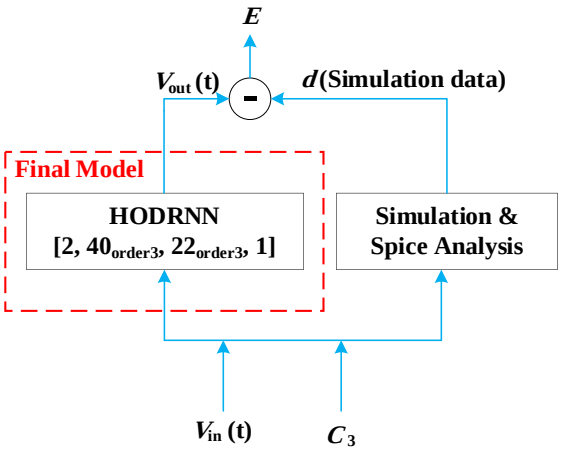


Fig. 7. Structure of the proposed HODRNN method for modeling the high-frequency DC generation circuit.

$$Z_i^1(t)=\sigma\left(\sum_{j=1}^{N_0}W_{i,j}^{1,F}I_j(t)+\sum_{m=1}^{2}\left(\sum_{j=1}^{N_1}W_{i,j}^{1,R_m}Z_j^1(t-m)\right)+b_i^1\right) \quad (13)$$

$$Z_i^2(t)=\sigma\left(\sum_{j=1}^{N_1}W_{i,j}^{2,F}Z_j^1(t)+\sum_{j=1}^{N_2}W_{i,j}^{2,R_1}Z_j^2(t-1)+b_i^2\right) \quad (14)$$

In above equation, N_0 is the number of model inputs. Moreover, in (13) layer is 1, and order is equal to 2. Likewise, in (14), layer is 2, and order is equal to 1. The training of Hybrid-HODRNN is similar to the HODRNN network keeping in mind that the hidden layer orders are different. As it can be seen from the hybrid structure, the number of weight parameters for layer 2 has been reduced from full HODRNN structure which will result in less overfitting problem and better model accuracy.

IV. Numerical Results

Five practical high-frequency nonlinear examples are used in this paper to demonstrate the validity of the proposed HODRNN as well as Hybrid-HODRNN macromodeling approaches.

A. Modeling a High-Frequency DC Generation Circuit

The schematic of a DC generation circuit is depicted in Fig. 6. For some applications such as high-frequency buffers, amplifiers, or comparators, it is needed to decide right at the average of incoming data signals. In addition, as the incoming data varies depending upon the input data, characteristics of the channel, and the system, it is not acceptable to specify the half of the signal's peak as its DC value. Indeed, in practical systems the amplitudes of the data will vary depending on the data and the channel frequency response. In these scenarios, the circuit of Fig. 6 can be used to accurately provide the average of the data signal. This circuit works based on determining the maximum and minimum of an input waveform and outputs the average of the two to slice the amplifier's input data in the center. The top part of Fig. 6 is a peak detector circuit, composed of amplifier Amp_1, MOSFET transistor M_1, and capacitor C_1. When the input, V_{in}, goes above the voltage stored on the capacitor C_1, V_{out1}, the output of the Amp_1 goes low, turning on the MOSFET and pulling the output, V_{out1}, towards V_{DD}. As the V_{out1} approaches V_{in}, the MOSFET starts to shut off. As a result, the output voltage across the capacitor, V_{out1}, corresponds to the peak voltage of the input signal. In the same manner, combination of the Amp_2, MOSFET transistor M_2, and capacitor C_2 makes the valley detector. This peak detector along with the valley detector, as is depicted in Fig. 6, can be used to generate a reference voltage that falls within the middle of the input data. The two resistors, R_1 and R_2, average the corresponding voltages of peak and valley detectors, and generate the DC average of the input signal, denoted as V_{Avg}. This DC average of the input is then fed to the bottom circuit where it is compared with the input data, and generates the output indicating the points where the input data is higher than its average. The Amp_1, Amp_2, and Amp_3 blocks in this figure are high-frequency amplifiers [42].

Fig. 7 shows a block diagram of the HODRNN structure used for modeling of this DC generation circuit. The output difference between the simulator and the HODRNN-based model indicated as E and is known as the error value. In this block diagram, C_3 as load capacitance is a static parameter, and V_{in} (t) and V_{out} (t) are input and output signals.

In order to build a macromodel for this circuit using HODRNN technique, square wave signals with the period of 1 ns were generated as training waveforms using circuit simulator. Also, rise/fall times of training signals were swept from 8 ps to 10 ps with steps of 0.2 ps. In this modeling, the static parameter which is the load capacitor, is varied from 2 fF to 5 fF. Another set of data, that were not used in training procedure, were generated as test signals.

The comparison of the absolute testing and training errors as well as training time among the conventional shallow RNN, deep RNN, proposed HODRNN-order2, order3, and order4 is demonstrated in Table I for the DC generation circuit. In addition, the number of parameters along with the modeling

TABLE I
COMPARISON AMONG THE CONVENTIONAL SHALLOW RNN, DEEP RNN, PROPOSED HODRNN-ORDER2 AND THE PROPOSED HODRNN-ORDER3 FOR MODELING A DC GENERATION CIRCUIT

No. of Parameters	Model	Structure	Training error	Testing error	Training time (s)
6,319	Shallow RNN	(2, 78, 1)	4.40×10^{-4}	9.06×10^{-4}	5,023
6,301	Deep RNN	(2, 35, 35, 35, 1)	3.54×10^{-4}	8.14×10^{-4}	4,773
6,997	Proposed HODRNN-Order2	(2, 38, 36, 1)	4.21×10^{-4}	6.58×10^{-4}	2,210
7,257	Proposed HODRNN-Order3	(2, 40, 22, 1)	4.46×10^{-4}	5.71×10^{-4}	2,185
7,311	Proposed HODRNN-Order4	(2,35, 20, 1)	4.62×10^{-4}	6.03×10^{-4}	1,688

TABLE II
COMPARISON OF CPU TIME AND SPEED-UP AMONG PROPOSED HODRNN-BASED AND TRANSISTOR-LEVEL MODELS FOR THE DC GENERATION CIRCUIT

Model	CPU time (ms)	Speed-up
Proposed HODRNN-Order4	26.402	3.56
Proposed HODRNN-Order3	25.164	3.77
Proposed HODRNN-Order2	24.319	3.90
Deep RNN	22.825	4.16
Shallow RNN	21.511	4.41
Transistor-Level	95	1

structure is provided in this table. Based on these results, it is evident that the proposed method provides a considerable reduction for training time compared to the conventional shallow RNN and deep RNN methods. Also, the HODRNN of orders 3 demonstrates superior performance than other orders for modeling this example. Additionally, training using the proposed methods provides much smaller testing error and makes this new macromodeling method a great candidate for large scale modeling problems such as complicated nonlinear circuits.

Table II provides the comparison of CPU time and speed-up among the proposed HODRNN-based and transistor-level models for the DC generation circuit. As it can be seen from the table, model generated using the proposed technique runs much faster than the transistor-level model, making this modeling approach suitable for high-frequency nonlinear applications.

The simulation results for the DC generation circuit are shown in Fig. 8. In this figure, test output waveforms generated by HODRNN-order3 is compared with deep RNN, and transistor-level models. As it is shown in Fig. 8, the model obtained from the proposed HODRNN method matches the original circuit outputs better than the model obtained from deep RNN method. Hence, as it is illustrated in this figure, it indicates the superiority of the proposed HODRNN method. In Fig. 9, test output waveforms generated by HODRNN-order3 is compared with deep RNN, and transistor-level models with more periods for DC generation circuit. By increasing the

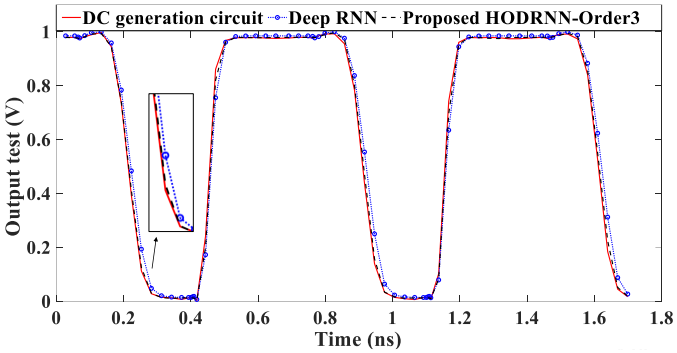


Fig. 8. Comparison of the outputs (test data waveforms) generated by HODRNN-based, deep RNN-based, and transistor-level models for DC generation circuit.

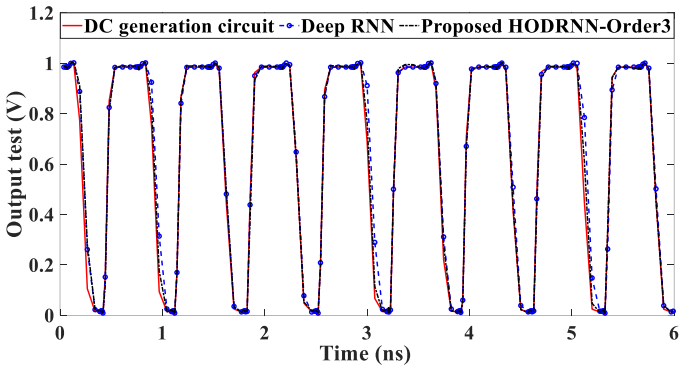


Fig. 9. Comparison of the outputs (test data waveforms) generated by HODRNN-based, deep RNN-based, and transistor-level models for DC generation circuit including many time steps.

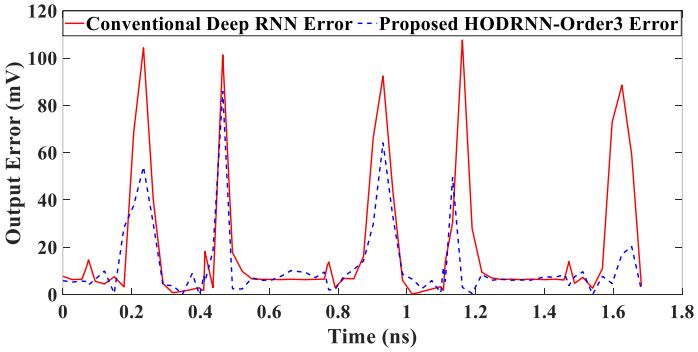


Fig. 10. Comparison of the absolute error between the original DC generation circuit and the conventional deep RNN-based model, and also between this circuit and the proposed HODRNN-Order3-based model.

number of periods, it can be seen that the model obtained from the proposed HODRNN method still matches the output of the main circuit with good accuracy. Fig. 10 is the error plot to compare the absolute error obtained from the conventional deep RNN and proposed HODRNN method. This error is the absolute value of subtracting the data of each method from the data of the transistor-level model. In this figure, it can be seen that the error of the proposed HODRNN is less than the conventional deep RNN method. In addition, the comparison of the required number of training data as well as testing and training errors between deep RNN and the model obtained from

TABLE III
COMPARISON OF THE NUMBER OF TRAINING DATA REQUIRED FOR GENERATING SIMILAR ACCURATE MODELS BETWEEN DEEP RNN AND PROPOSED HODRNN MACROMODELING METHODS FOR DC GENERATION CIRCUIT

Model	Number of Training data	Training error	Testing error
Deep RNN	40	3.54×10^{-4}	8.14×10^{-4}
Proposed HODRNN-Order3	20	5.83×10^{-4}	8.17×10^{-4}

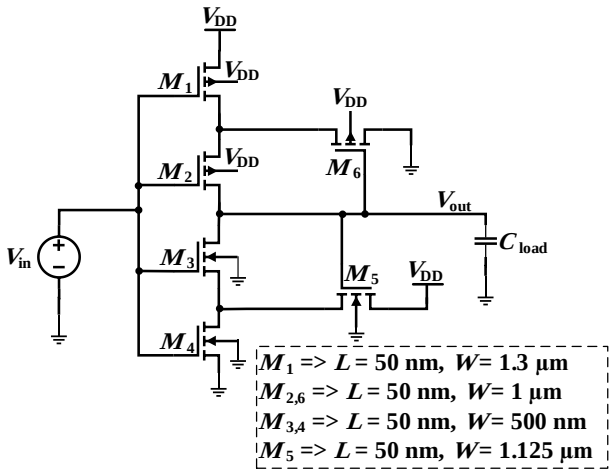


Fig. 11. A high-frequency Schmitt trigger circuit.

the proposed HODRNN method for this circuit is given in Table III. According to these results, a fewer number of training data is required by the proposed technique compared to the conventional method to obtain similar accurate models.

B. Modeling a High-Frequency Schmitt Trigger Circuit

The schematic of the Schmitt trigger is shown in Fig. 11 which is a good fit for high-frequency applications where a rapid decision should be made in the existence of a noisy environment. If the output is low, then M_6 is on and M_5 is off and M_3 and M_4 are on, providing a DC path to GND. On the other hand, if the output is high, $M5$ is on and $M6$ is off, and $M1$ and M_2 are on, providing a DC path to V_{DD} [42]. Fig. 10 shows a block diagram of the HODRNN structure which is used for modeling of Schmitt trigger circuit of Fig. 11. In Fig. 12, C_{load} which is the load of this circuit, is considered as the static parameter.

In order to build a macromodel for this circuit using HODRNN method, square wave signals with period of 1 ns were generated as training waveforms. Additionally, rise/fall times of training signals were swept from 48 ps to 50 ps with steps of 0.2 ps. The static parameter which is the load capacitor, is changed from 12 fF to 15 fF. Another set of data, which were not used in training procedure, were generated as test signals.

The comparison of the absolute testing and training errors as well as training time among the conventional shallow RNN, deep RNN, deep LSTM, proposed HODRNN-order2, order3, and order4 is given in Table IV for the Schmitt trigger circuit. Besides, the number of parameters together with the modeling structure is provided in this table. Based on the results of this

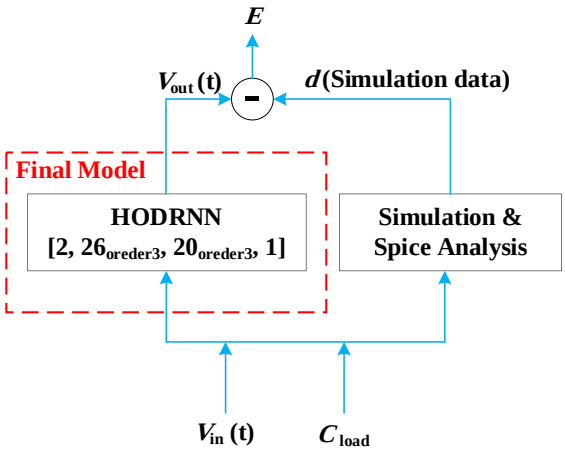


Fig. 12. Structure of the proposed HODRNN method for modeling the high-frequency Schmitt trigger circuit.

TABLE IV
COMPARISON AMONG THE CONVENTIONAL SHALLOW RNN, DEEP RNN, DEEP LSTM, PROPOSED HODRNN-ORDER2, ORDER3, AND ORDER4 FOR MODELING SCHMITT TRIGGER CIRCUIT

No. of Parameters	Model	Structure	Training error	Testing error	Training time (s)
3,079	Shallow RNN	(2, 54, 1)	4.18×10^{-4}	15.23×10^{-4}	3,160
3,001	Deep RNN	(2, 24, 24, 24, 1)	3.17×10^{-4}	9.99×10^{-4}	3,042
3,303	Deep LSTM	(2, 18, 14, 1)	5.18×10^{-4}	8.44×10^{-4}	5,175
3,386	Proposed HODRNN-Order2	(2, 25, 25, 5, 1)	5.16×10^{-4}	8.02×10^{-4}	2,344
3,841	Proposed HODRNN-Order3	(2, 26, 20, 1)	5.84×10^{-4}	6.21×10^{-4}	970
3,867	Proposed HODRNN-Order4	(2,21, 20, 1)	6.18×10^{-4}	7.93×10^{-4}	678

table, it is evident that the proposed method provides a remarkable reduction for training time compared to the conventional shallow RNN, deep RNN, and deep LSTM methods. Also, the HODRNN of orders 3 exhibits greater performance than other orders for macromodeling this Schmitt trigger circuit. Additionally, training using the proposed method provides much smaller testing error and makes this HODRNN-based macromodeling method a great choice for large scale modeling problems such as complex and high-frequency nonlinear circuits.

Table V provides the comparison of CPU time and speed-up among the proposed HODRNN-based and transistor-level models for the Schmitt trigger circuit. As it is reported in this table, model generated using the proposed technique runs much faster than the transistor-level model. Accordingly, this

TABLE V
COMPARISONS OF CPU TIME AND SPEED-UP AMONG PROPOSED HODRNN-BASED AND TRANSISTOR-LEVEL MODELS FOR THE SCHMITT TRIGGER CIRCUIT

Model	CPU Time (ms)	Speed-up
Proposed HODRNN-Order4	28.651	8.48
Proposed HODRNN-Order3	27.409	8.86
Proposed HODRNN-Order2	26.781	9.07
Deep LSTM	29.112	8.34
Deep RNN	24.983	9.72
Shallow RNN	24.324	9.99
Transistor-Level	243	1

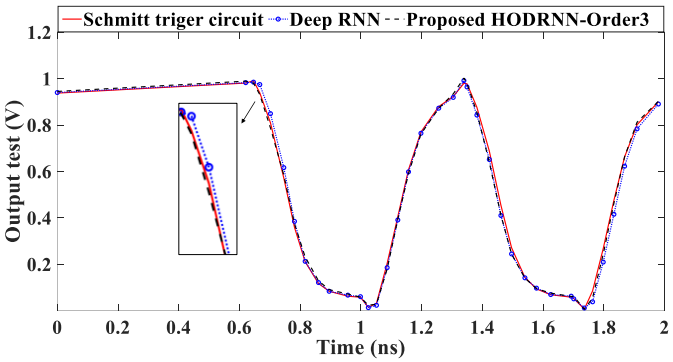


Fig. 13. Comparison of the outputs (test data waveforms) generated by HODRNN-based, deep RNN-based, and transistor-level models for Schmitt trigger circuit.

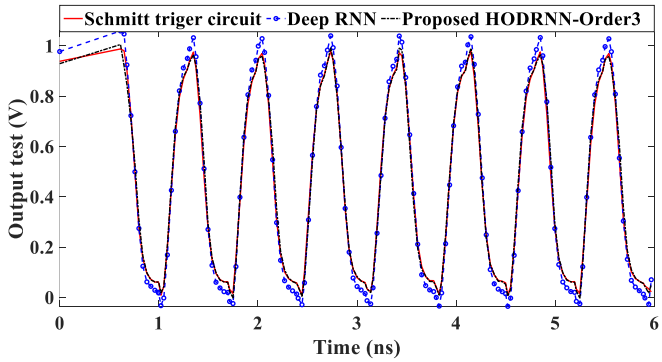


Fig. 14. Comparison of the outputs (test data waveforms) generated by HODRNN-based, deep RNN-based, and transistor-level models for Schmitt trigger circuit including many time steps.

modeling approach can be considered as a suitable solution for high-frequency nonlinear applications. The simulation results for the Schmitt trigger circuit are shown in Fig. 13. In this figure, test output waveforms generated by HODRNN-order3 is compared with deep RNN, and transistor-level models. As it is shown in this figure, the model attained from the proposed HODRNN method matches the original circuit outputs better than the model obtained from deep RNN method. Consequently, it indicates the superiority of the proposed HODRNN method as it is depicted in this figure.

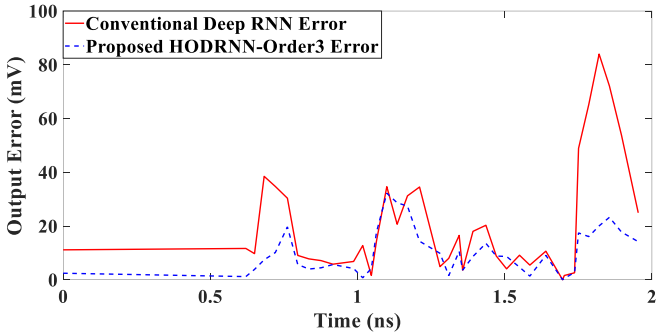


Fig. 15. Comparison of the absolute error between the original Schmitt trigger circuit and the conventional deep RNN-based model, and also between this circuit and the proposed HODRNN-Order3-based model.

TABLE VI
COMPARISON OF THE NUMBER OF TRAINING DATA REQUIRED FOR GENERATING SIMILAR ACCURATE MODELS BETWEEN DEEP RNN AND PROPOSED HODRNN MACROMODELING METHODS FOR SCHMITT TRIGGER CIRCUIT

Model	Number of Training data	Training error	Testing error
Deep RNN	40	3.17×10^{-4}	9.99×10^{-4}
Proposed HODRNN-Order3	23	6.39×10^{-4}	10.41×10^{-4}

In Fig. 14 test output waveforms generated by HODRNN-order3 is compared with deep RNN, and transistor-level models with more periods for Schmitt trigger circuit. By increasing the number of periods, it can be seen that the model obtained from the proposed HODRNN method still matches the output of the main circuit with good accuracy Fig. 15 is the error plot to compare the absolute error obtained from the conventional deep RNN and the proposed HODRNN methods which is the absolute value of subtracting the data of each method from the data of the transistor-level model. In this figure, it can be seen that the error of the proposed HODRNN is less than the conventional deep RNN method.

Additionally, the comparison of the required number of training data along with testing and training errors between deep RNN and the model obtained from the proposed HODRNN method for this circuit is reported in Table VI. According to the results of this table, the proposed technique demands a fewer number of training data compared to the conventional approach to obtain similar accurate models.

C. Modeling a High-Frequency Amplifier Circuit

The schematic of a high-frequency amplifier is shown in Fig. 16. This amplifier is composed of a folded cascode structure at the input stage, and a class AB output buffer at its output stage. Also, in order to increase the bandwidth for the high-frequency operation, the 1 kΩ feedback resistor is utilized [42].

Fig. 17 shows a block diagram of the HODRNN structure used for modeling of this high-frequency amplifier. In this block diagram, C_{load} as load capacitance is considered as one of the inputs in addition to the input voltage. Likewise, Fig. 18 shows the block diagram of the Hybrid-HODRNN structure for the same circuit.

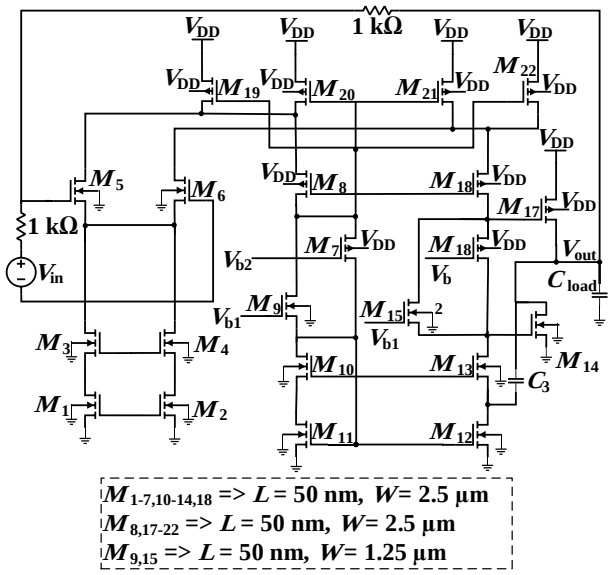


Fig. 16. A high-frequency amplifier with an input common-mode range.

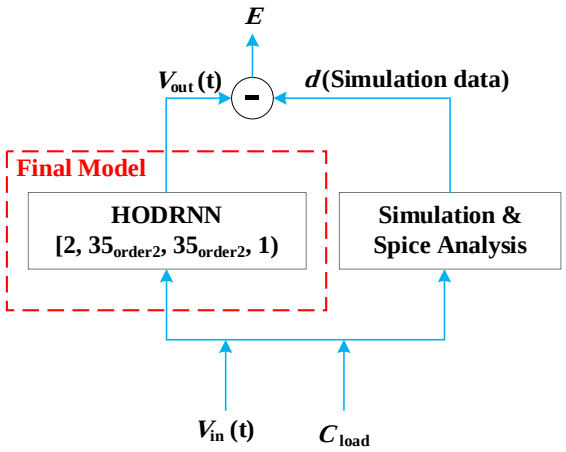


Fig. 17. Structure of the proposed HODRNN method for modeling the high-frequency amplifier circuit.

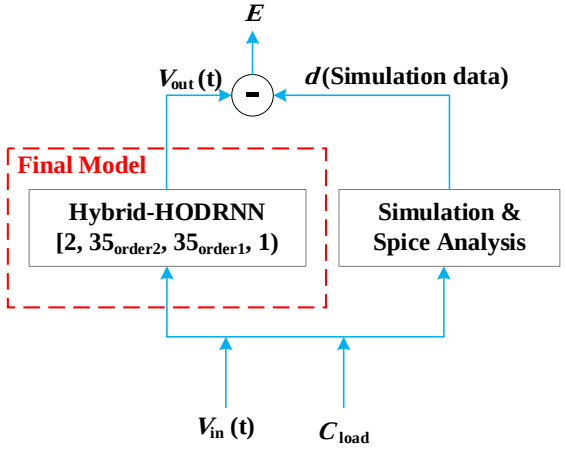


Fig. 18. Structure of the proposed Hybrid-HODRNN method for modeling the high-frequency amplifier circuit.

TABLE VII
COMPARISON AMONG DEEP RNN, PROPOSED HODRNN-ORDER2, ORDER3, AND PROPOSED HYBRID HIGH-ORDER RNN-ORDER2, 1 FOR MODELING A HIGH-FREQUENCY AMPLIFIER CIRCUIT

No. of Parameters	Model	Structure	Training error	Testing error	Training time (s)
4,651	Deep RNN	(2, 30, 30, 30, 1)	6.17 ×10^{-4}	10.52 ×10^{-4}	3,618
6,266	Proposed HODRNN-Order2	(2, 35, 35, 1)	6.30 ×10^{-4}	8.02 ×10^{-4}	2,851
6,421	Proposed HODRNN-Order3	(2, 30, 30, 1)	6.32 ×10^{-4}	10.22 ×10^{-4}	3,259
5,041	Proposed Hybrid HODRNN-Order2, 1	(2, 35, 35, 1)	6.69 ×10^{-4}	6.48 ×10^{-4}	2,051

TABLE VIII
COMPARISON OF CPU TIME AND SPEED-UP AMONG PROPOSED HODRNN-ORDER2, PROPOSED HYBRID-HODRNN, AND TRANSISTOR-LEVEL MODELS FOR MODELING A HIGH-FREQUENCY AMPLIFIER CIRCUIT

Model	CPU Time (ms)	Speed-up
Proposed HODRNN-Order3	25.071	11.61
Proposed HODRNN-Order2	23.851	12.20
Proposed Hybrid-HODRNN-Order2, 1	23.031	12.64
Deep RNN	22.864	12.73
Transistor-Level (Transient analysis)	291	1

In order to build a macromodel for this circuit using HODRNN method, square wave signals with period of 1 ns were generated as training waveforms. In addition, rise/fall times of training signals were swept from 8 ps to 10 ps with steps of 0.2 ps. In this example, the load capacitor is varied from 7 fF to 10 fF. Another set of data that were not used in training procedure were generated as test signals.

The comparison of the absolute testing and training errors as well as training time among the deep RNN, proposed HODRNN-order2, order3, and proposed Hybrid-HODRNN is demonstrated in Table VII for the high-frequency amplifier circuit. In this table, the number of parameters along with the modeling structure is also reported. Based on these results, it is apparent that the proposed Hybrid-HODRNN method offers a noticeable reduction for training time compared to the deep RNN method. Moreover, training using this proposed technique provides smaller testing error, and makes this new macromodeling method a great option for modeling complicated high-frequency nonlinear circuits.

Table VIII provides the comparison of CPU time and speed-up for the proposed Hybrid-HODRNN, as well as HODRNN-based, and transistor-level models for the high-frequency amplifier. According to the results provided in this table, model

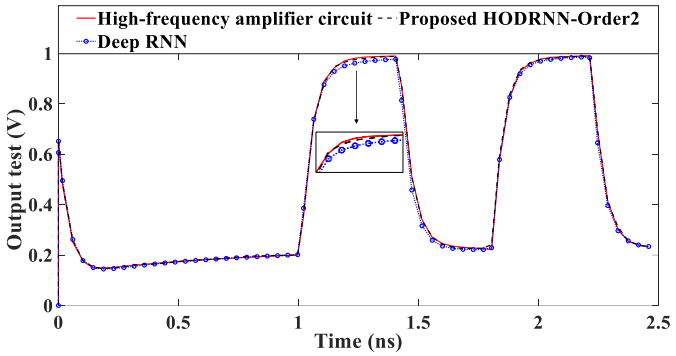


Fig. 19. Comparison of the outputs (test data waveforms) generated by HODRNN-based, deep RNN-based, and transistor-level models for the high-frequency amplifier.

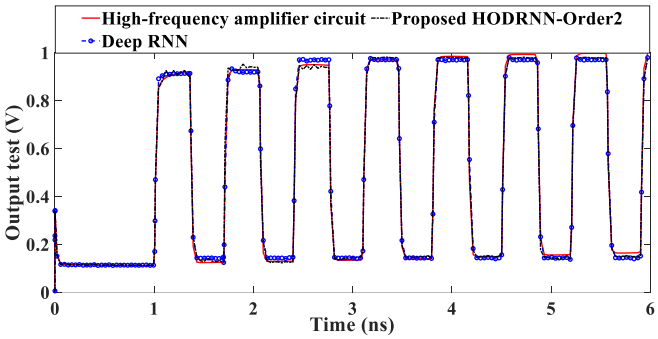


Fig. 20. Comparison of the outputs (test data waveforms) generated by HODRNN-based, deep RNN-based, and transistor-level models for the high-frequency amplifier including many time steps.

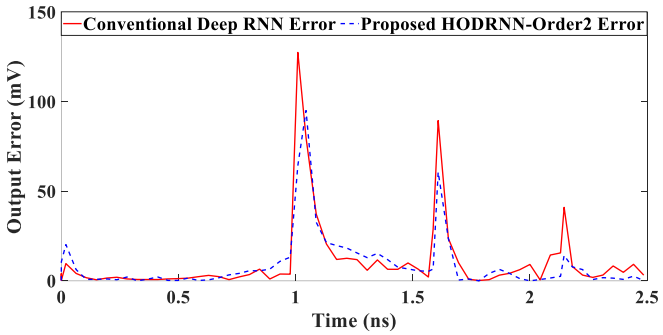


Fig. 21. Comparison of the absolute error between the original high-frequency amplifier circuit and the conventional deep RNN-based model, and also between this circuit and the proposed HODRNN-Order2-based model.

generated using the proposed techniques run much faster than the transistor-level model. Additionally, Hybrid-HODRNN performs a little better than that of HODRNN-based model in terms of speed-up. Accordingly, the Hybrid-HODRNN approach can be considered as an appropriate technique for modeling of high-frequency and nonlinear applications.

The simulation results for the high-frequency amplifier are illustrated in Fig. 19. In Fig. 19, test output waveforms generated by HODRNN-order2 is compared with deep RNN, and transistor-level models. As it is shown in this figure, the model achieved from the proposed HODRNN method matches

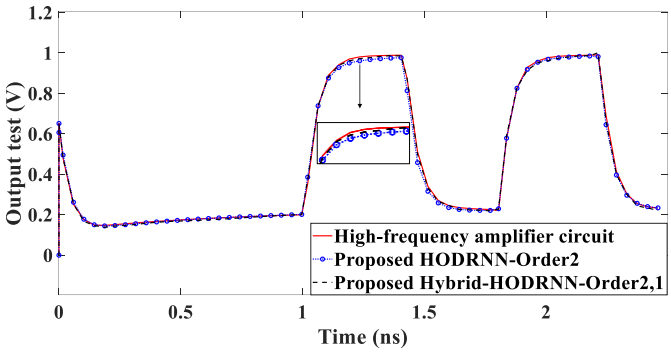


Fig. 22. Comparison of the outputs (test data waveforms) generated by proposed Hybrid-HODRNN, with that of HODRNN-based, and transistor-level models for the high-frequency amplifier.

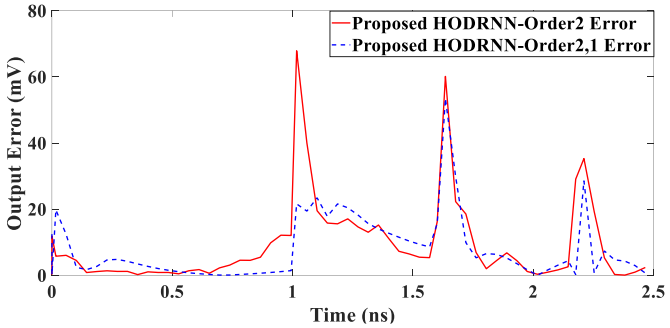


Fig. 23. Comparison of the absolute error between the original high-frequency amplifier circuit and the proposed HODRNN-Order2-based model, and also between this circuit and the Hybrid-HODRNN-Order2,1-based model.

the original circuit outputs better than the model obtained from deep RNN method. In Fig. 20 test output waveforms generated by HODRNN-order2 is compared to the deep RNN, and transistor-level models with more periods for high-frequency amplifier circuit. By increasing the number of periods, it can be seen that the model obtained from the proposed HODRNN method still matches the output of the main circuit with good accuracy. Fig. 21 is the error plot to compare the absolute error obtained from the conventional deep RNN and the proposed HODRNN methods. This error is the absolute value of the difference between the data extracted from each method and the data of the transistor-level model. In this figure, it can be seen that the error of the proposed HODRNN is less than the conventional deep RNN method. In Fig. 22, test output waveforms generated by HODRNN-order2 is compared to the proposed Hybrid-HODRNN, and transistor-level models. As it is shown in this figure, the model achieved from the proposed Hybrid-HODRNN method matches the original circuit outputs better than the model obtained from HODRNN method. As a result, it shows the superiority of the proposed Hybrid-HODRNN over the HODRNN method. Moreover, Fig. 23 is the error plot for comparing the absolute error obtained from the methods of proposed HODRNN-Order2 and proposed Hybrid-HODRNN-order2,1. In this figure, it can be seen that the error of the proposed Hybrid-HODRNN is less than the other method.

TABLE IX
COMPARISON OF THE NUMBER OF TRAINING DATA REQUIRED FOR GENERATING SIMILAR ACCURATE MODELS BETWEEN DEEP RNN AND PROPOSED HYBRID-HODRNN MACROMODELING METHODS FOR MODELING A HIGH-FREQUENCY AMPLIFIER CIRCUIT

Model	Number of Training data	Training error	Testing error
Deep RNN	40	6.17×10^{-4}	10.52×10^{-4}
Proposed Hybrid HODRNN-Order2, 1	15	8.48×10^{-4}	9.93×10^{-4}

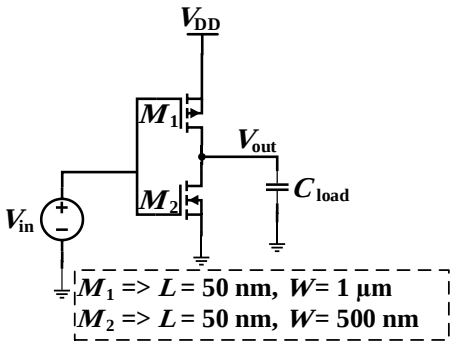


Fig. 24. A high-frequency CMOS inverter circuit.

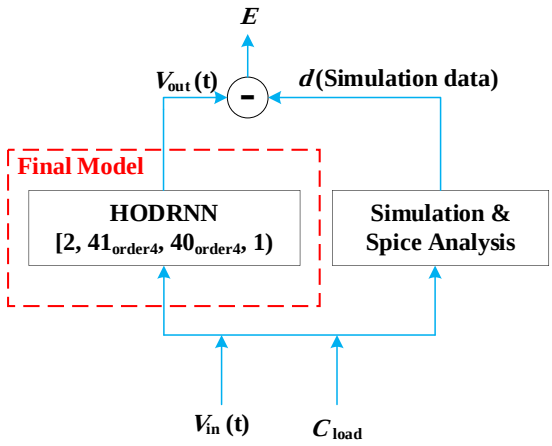


Fig. 25. Structure of the proposed HODRNN method for modeling the high-frequency CMOS inverter circuit.

Additionally, the comparison of the required number of training data along with testing and training errors between deep RNN and the model obtained from the proposed Hybrid-HODRNN method for this circuit is reported in Table IX. According to the results of this table, the proposed technique demands a fewer number of training data compared to the conventional approach to obtain similar accurate models.

D. Modeling a High-Frequency CMOS inverter Circuit

The schematic of the CMOS inverter is shown in Fig. 24 which is a good fit for high-frequency applications where a quick decision should be made in the existence of a noisy environment [9], [43]. CMOS inverter is indeed the combination of PMOS and NMOS transistors. Often times a chain of these CMOS inverters will be used to satisfy the requirements of the load. CMOS inverters can be used in a variety of applications such as driver and receiver buffers due to their negligible power dissipation, and high safety against noise. Additionally, for making buffers any even number of these inverters will be employed. If the input is low, M_2 is on and M_1 is off, and as a result, the output is low, and if the input is high, M_1 is on and M_2 is off, and as a result, the output is high. Other words, transistor M_1 generates the V_{DD} path to the output and M_2 creates the output path to GND.

TABLE X
COMPARISON AMONG THE CONVENTIONAL SHALLOW RNN, DEEP RNN, DEEP LSTM, PROPOSED HODRNN-ORDER2, ORDER3, ORDER4, AND ORDER5 FOR MODELING CMOS INVERTER CIRCUIT

No. of Parameters	Model	Structure	Training error	Testing error	Training time (s)
11,341	Shallow RNN	(2, 105, 1)	7.66×10^{-4}	27.14×10^{-4}	7,112
10,351	Deep RNN	(2, 45, 45, 45, 1)	6.54×10^{-4}	25.90×10^{-4}	7,260
11,053	Deep LSTM	(2, 26, 23, 20, 1)	7.72×10^{-4}	20.18×10^{-4}	8,315
12,400	Proposed HODRNN-Order2	(2, 45, 45, 23, 1)	8.14×10^{-4}	18.22×10^{-4}	6,036
14,879	Proposed HODRNN-Order3	(2, 45, 45, 8, 1)	9.56×10^{-4}	14.29×10^{-4}	5,880
14,927	Proposed HODRNN-Order4	(2, 41, 40, 1)	10.23×10^{-4}	11.08×10^{-4}	4,337
15,208	Proposed HODRNN-Order5	(2, 37, 37, 1)	10.86×10^{-4}	13.06×10^{-4}	3,562

Fig. 25 shows a block diagram of the HODRNN structure used for modeling of this CMOS inverter circuit. The output difference between the simulator and the HODRNN-based model indicated as E and is known as the error value. In this block diagram, C_{load} as load capacitance is a circuit parameter which is considered as one of the inputs in addition to the input voltage, and V_{in} (t) and V_{out} (t) are input and output signals respectively.

In order to build a macromodel for this circuit using HODRNN method, square wave signals with period of 1 ns were generated as training waveforms. Additionally, rise/fall times of training signals were swept from 20 ps to 50 ps with steps of 0.2 ps. The static parameter which is the load capacitor, and is changed from 5 pF to 10 pF. Another set of data, which were not used in training procedure, were generated as test signals.

The comparison of the absolute testing and training errors as well as training time between the conventional shallow RNN, deep RNN, deep LSTM, proposed HODRNN-order2, HODRNN-order3, HODRNN-order4, and the proposed HODRNN-order5 is given in Table X for the CMOS inverter circuit. Besides, the number of parameters together with the modeling structure is provided in this table. Based on the results of this table, it is evident that the proposed method provides a

TABLE XI
COMPARISONS OF CPU TIME AND SPEED-UP AMONG PROPOSED HODRNN-BASED AND TRANSISTOR-LEVEL MODELS FOR THE CMOS INVERTER CIRCUIT

Model	CPU Time (ms)	Speed-up
Proposed HODRNN-Order5	31.019	7.15
Proposed HODRNN-Order4	30.877	7.18
Proposed HODRNN-Order3	29.341	7.56
Proposed HODRNN-Order2	26.213	8.46
Deep LSTM	31.022	7.15
Deep RNN	23.772	9.33
Shallow RNN	16.115	13.77
Transistor-Level	222	1

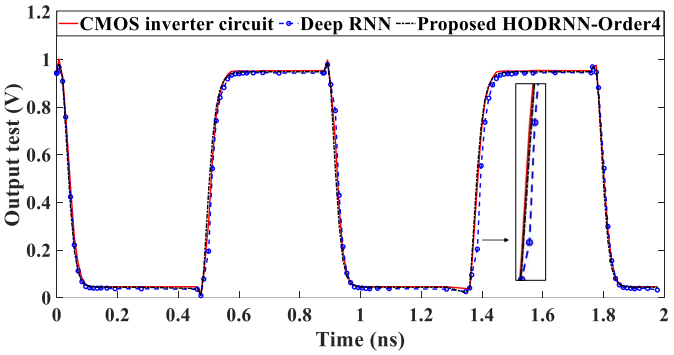


Fig. 26. Comparison of the outputs (test data waveforms) generated by HODRNN-based, deep RNN-based, and transistor-level models for CMOS inverter circuit.

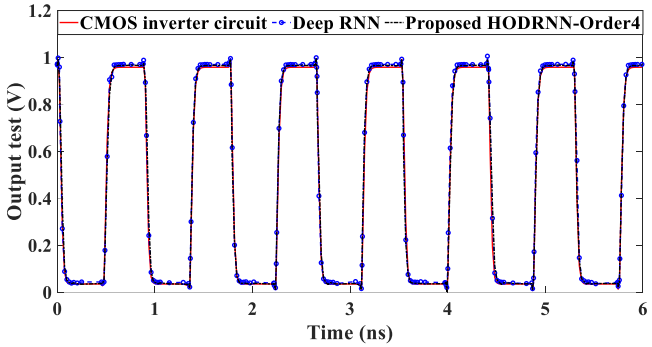


Fig. 27. Comparison of the outputs (test data waveforms) generated by HODRNN-based, deep RNN-based, and transistor-level models for CMOS inverter circuit including many time steps.

considerable reduction for training time compared to the conventional shallow RNN, deep RNN, and deep LSTM methods. Also, the HODRNN of orders 4 exhibits greater performance than other orders for macromodeling this CMOS inverter circuit. Additionally, training using the proposed method provides much smaller testing error and makes this HODRNN-based macromodeling method a great choice for high-frequency nonlinear circuits.

Table XI provides the comparison of CPU time and speed-up among the proposed HODRNN-based and transistor- models for the CMOS inverter circuit. As it can be seen from the table, model generated using the proposed technique runs much faster

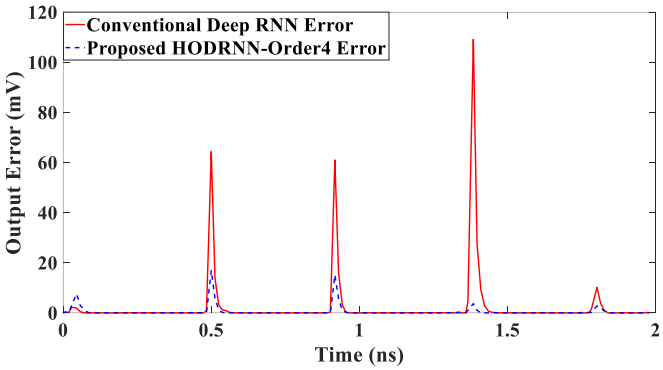


Fig. 28. Comparison of the absolute error between the original CMOS inverter circuit and the conventional deep RNN-based model, and also between this circuit and the proposed HODRNN-Order4-based model.

TABLE XII
COMPARISON OF THE NUMBER OF TRAINING DATA REQUIRED FOR GENERATING SIMILAR ACCURATE MODELS BETWEEN DEEP RNN AND PROPOSED HODRNN MACROMODELING METHODS FOR CMOS INVERTER CIRCUIT

Model	Number of Training data	Training error	Testing error
Deep RNN	40	6.54×10^{-4}	25.90×10^{-4}
Proposed HODRNN-Order4	17	8.23×10^{-4}	25.85×10^{-4}

than the transistor-level model, making this modeling approach suitable for nonlinear applications.

The simulation results for the CMOS inverter are illustrated in Fig. 26. In this figure, test output waveforms generated by HODRNN-order4 is compared with deep RNN, and transistor-level models. As it is shown in this figure, the model achieved from the proposed HODRNN method matches the original circuit outputs better than the model obtained from deep RNN method.

In Fig. 27 test output waveforms generated by HODRNN-order4 is compared with deep RNN, and transistor-level models with more periods for CMOS inverter circuit. By increasing the number of periods, it can be seen that the model obtained from the proposed HODRNN method still can match the output of the main circuit accurately. Fig. 28 is the error plot to compare the absolute error obtained from the conventional deep RNN and the proposed HODRNN methods which is the absolute value of subtracting the data of each method from the data of the transistor-level model. In this figure, it can be seen that the error of the proposed HODRNN is less than the conventional deep RNN method. In addition, the comparison of the required number of training data as well as testing and training errors between deep RNN and the model obtained from the proposed HODRNN method for this circuit is given in Table XII. According to these results, a fewer number of training data is required by the proposed technique compared to the conventional method to obtain similar accurate models.

E. Modeling a High-Frequency Multi-Stage Buffer Drived by Ring Oscillator

The schematic of the multi-stage buffer drived by ring oscillator circuit is shown in Fig. 29 [2], [3], [14]. A ring oscillator is a

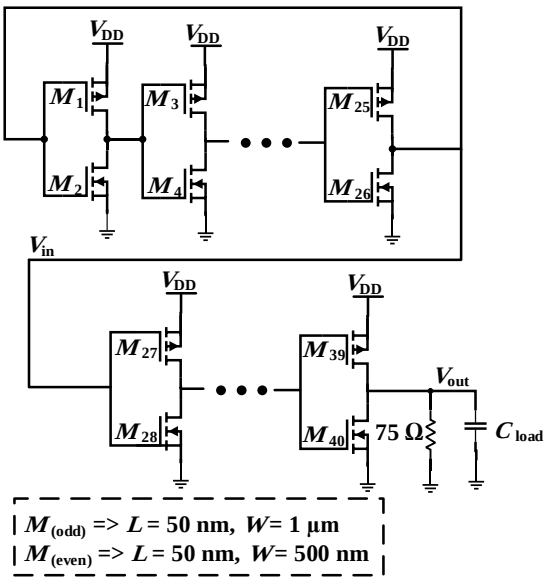


Fig. 29. A high-frequency multi-stage buffer drived by ring oscillator circuit.

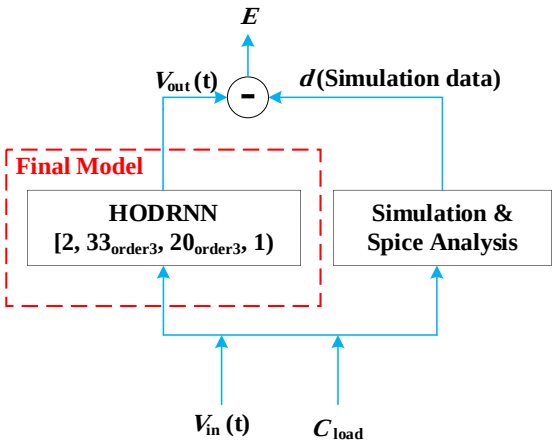


Fig. 30. Structure of the proposed HODRNN method for modeling the high-frequency multi-stage buffer drived by ring oscillator circuit.

circuit composed of an odd number of CMOS inverter circuits in a closed loop called a ring, whose output oscillates between two voltage levels. The inverters, are attached in a chain and the output of the last inverter is feedbacked into the first. The multi-stage buffer drived by ring oscillator circuit depicted in Fig. 29 consists of thirteen inverter circuits in the loop, and its output is connected to the circuits of the seven-level CMOS driver circuit. An oscillator is used to generate a signal which has a specific frequency that is beneficial for synchronizing the computation process in digital systems.

Fig. 30 shows a block diagram of the HODRNN structure used for modeling of this high-frequency multi-stage buffer drived by ring oscillator circuit. The output difference between the simulator and the HODRNN-based model indicated as E and is known as the error value. In this block diagram, C_{load} as load capacitance is considered as one of the inputs in addition to the input voltage, and V_{in} (t) and V_{out} (t) are the input and output signals respectively.

TABLE XIII
COMPARISON AMONG THE CONVENTIONAL SHALLOW RNN, DEEP RNN, DEEP LSTM, PROPOSED HODRNN-ORDER2, ORDER3, AND ORDER4 FOR MODELING MULTI-STAGE BUFFER DRIVED BY RING OSCILLATOR CIRCUIT

No. of Parameters	Model	Structure	Training error	Testing error	Training time (s)
4,555	Shallow RNN	(2, 66, 1)	1.88×10^{-4}	7.63×10^{-4}	1,811
4,521	Deep RNN	(2, 45, 25, 15, 1)	1.35×10^{-4}	5.52×10^{-4}	1,766
4,687	Deep LSTM	(2, 20, 16, 6, 1)	2.41×10^{-4}	4.89×10^{-4}	3,405
4,943	Proposed HODRNN-Order2	(2, 38, 23, 1)	2.57×10^{-4}	4.15×10^{-4}	943
5,234	Proposed HODRNN-Order3	(2, 33, 20, 1)	2.68×10^{-4}	2.95×10^{-4}	653
5,393	Proposed HODRNN-Order4	(2, 28, 20, 1)	2.74×10^{-4}	3.51×10^{-4}	397

In order to build a macromodel for this circuit using HODRNN method, the output of ring oscillator is used as input training waveforms meaning that V_{DD} of ring oscillator were swept from 0.9 V to 1.2 V with steps of 0.01 V and input training signals for multi-stage buffer were generated. Also, the load capacitance of the multi-stage buffer was swept from 1.3 pF to 1.7 pF in order to take into account the effect of buffer load on modeling of its output. Another set of data, which were not used in training procedure, were generated as test signals.

The comparison of the absolute testing and training errors as well as training time between the conventional shallow RNN, deep RNN, deep LSTM, proposed HODRNN-order2, order3, and order4 is given in Table XIII for the multi-stage buffer drived by ring oscillator circuit. Besides, the number of parameters together with the modeling structure is provided in this table. Based on the results of this table, it is evident that the proposed method provides a remarkable reduction for training time compared to the conventional shallow RNN, deep RNN, and deep LSTM methods.

Also, the HODRNN of order 3 exhibits greater performance than other orders for macromodeling this multi-stage buffer. Additionally, training using the proposed method provides much smaller testing error and makes this HODRNN-based macromodeling method a great choice for large scale modeling problems such as complex and high-frequency nonlinear circuits.

Table XIV provides the comparison of CPU time and speed-up among the proposed HODRNN-based and transistor-level models for the multi-stage buffer drived by ring oscillator circuit. As it can be seen from the table, model generated using the proposed technique runs much faster than the transistor-level model, making this modeling approach suitable for high-frequency nonlinear applications.

The simulation results for the multi-stage buffer drived by ring oscillator circuit are illustrated in Fig. 31. In this figure, test output waveforms generated by HODRNN-order3 is compared with deep RNN, and transistor-level models. As it is shown in this figure, the model achieved from the proposed HODRNN

TABLE XIV
COMPARISONS OF CPU TIME AND SPEED-UP AMONG PROPOSED HODRNN-BASED AND TRANSISTOR-LEVEL MODELS FOR THE MULTI-STAGE BUFFER DRIVED BY RING OSCILLATOR CIRCUIT

Model	CPU Time (ms)	Speed-up
Proposed HODRNN-Order4	27.449	6.88
Proposed HODRNN-Order3	25.114	7.52
Proposed HODRNN-Order2	22.741	8.31
Deep LSTM	27.668	6.83
Deep RNN	20.108	9.39
Shallow RNN	18.984	9.95
Transistor-Level	189	1

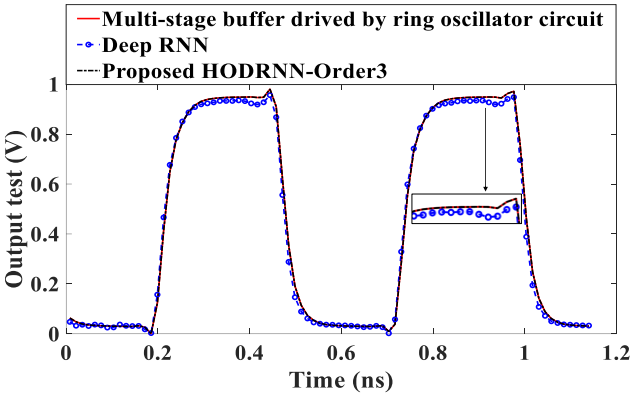


Fig. 31. Comparison of the outputs (test data waveforms) generated by HODRNN-based, deep RNN-based, and transistor-level models for multi-stage buffer drived by ring oscillator circuit.

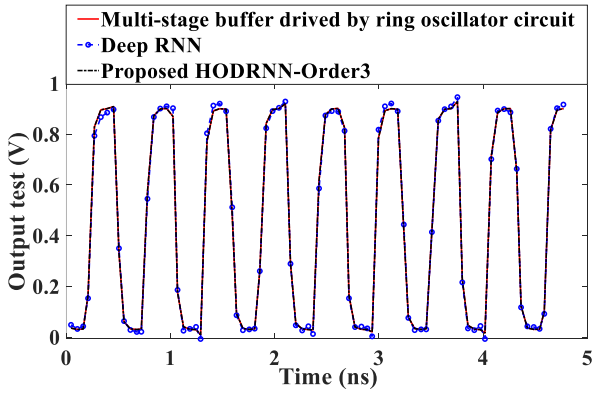


Fig. 32. Comparison of the outputs (test data waveforms) generated by HODRNN-based, deep RNN-based, and transistor-level models for multi-stage buffer drived by ring oscillator circuit including many time steps.

method matches the original circuit outputs better than the model obtained from deep RNN method. In Fig. 32 test output waveforms generated by HODRNN-order3 is compared with deep RNN, and transistor-level models with more periods for multi-stage buffer drived by ring oscillator circuit. By increasing the number of periods, it can be seen that the model obtained from the proposed HODRNN method still matches the outputs of the main circuit with good accuracy. By taking the absolute value of subtracting the data of each method from the

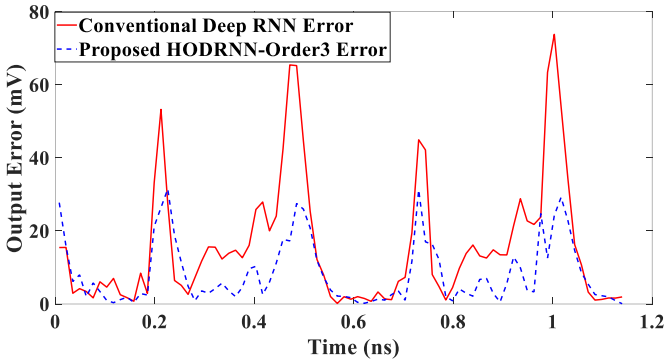


Fig. 33. Comparison of the absolute error between the original multi-stage buffer drived by ring oscillator circuit and the conventional deep RNN-based model, and also between this circuit and the proposed HODRNN-Order3-based model.

TABLE XV
COMPARISON OF THE NUMBER OF TRAINING DATA REQUIRED FOR GENERATING SIMILAR ACCURATE MODELS BETWEEN DEEP RNN AND PROPOSED HODRNN MACROMODELING METHODS FOR MULTI-STAGE BUFFER DRIVED BY RING OSCILLATOR CIRCUIT

Model	Number of Training data	Training error	Testing error
Deep RNN	41	1.35×10^{-4}	5.52×10^{-4}
Proposed HODRNN-Order3	18	2.56×10^{-4}	6.78×10^{-4}

data of the transistor-level model, the error plot is obtained as depicted in Fig. 33. In this figure, it can be seen that the error of the proposed HODRNN is less than the conventional deep RNN method.

Additionally, the comparison of the required number of training data along with testing and training errors between deep RNN and the model obtained from the proposed HODRNN method for this circuit is reported in Table XV. According to the results of this table, the proposed technique demands a fewer number of training data compared to the conventional approach to obtain similar accurate models.

V. Conclusion

In this paper, a novel method called HODRNN is proposed to model nonlinear high-frequency circuits which uses more memory units than the conventional shallow and deep RNNs to capture longer-term dependencies. The training time for the proposed HODRNN method not only is reduced considerably compared to the conventional RNN methods, but also the smaller testing error and more accurate macromodel has been achieved using the proposed modeling method. Also, the model generated from this proposed technique runs much faster than the transistor-level model. In addition, model generation using the proposed technique requires a fewer number of training data compared to the conventional RNN modeling methods to obtain similar accurate models. Moreover, an improved version of HODRNN called Hybrid-HODRNN which uses hybrid layers with both single and high orders is proposed reducing the overfitting problem leading to better model accuracy compared to the HODRNN. The training time for the proposed Hybrid-

HODRNN method also reduced noticeably compared to the conventional RNN methods due to its efficient structure and training procedure. Additionally, similar to the HODRNN, the Hybrid-HODRNN required less training data to obtain similar accurate model in comparison with the conventional RNN modeling methods. As a result, the aforementioned advantages show the superiority of both Hybrid-HODRNN and HODRNN methods over conventional RNN techniques. Consequently, the proposed macromodeling methods are great options for large scale modeling problems such as complicated nonlinear and high-frequency circuits. Three practical nonlinear high-frequency circuit examples were used in this paper to demonstrate the validity of the proposed macromodeling approaches.

REFERENCES

[1] Y. Cao, R. Ding, and Q.-J. Zhang, "State-space dynamic neural network technique for high-speed IC applications: modeling and stability analysis," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 6, pp. 2398-2409, Jun. 2006.

[2] Y. Cao and Q.-J. Zhang, "A new training approach for robust recurrent neural-network modeling of nonlinear circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 57, no. 6, pp. 1539-1553, Jun. 2009.

[3] S. A. Sadrossadat, P. Gunupudi, and Q.-J. Zhang, "Nonlinear electronic/photonic component modeling using adjoint state-space dynamic neural network technique," in *IEEE Trans. Compon., Packag. and Manuf. Tech.*, vol. 5, no. 11, pp. 1679-1693, Nov. 2015.

[4] H. Kabir, L. Zhang, M. Yu, P. H. Aaen, J. Wood, and Q.-J. Zhang, "Smart modeling of microwave devices," *IEEE Microw. Magazine*, vol. 11, no. 3, pp. 105-118, Apr. 2010.

[5] Q.-J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design-from theory to practice," *IEEE Trans. Microw. Theory Tech.*, vol. 51, no. 4, pp. 1339-1350, Apr. 2003.

[6] Z. Naghibi, S. A. Sadrossadat, and S. Safari, "Time-domain modeling of nonlinear circuits using deep recurrent neural network technique," *AEU-Int. J. of Electron. and Commun.*, vol. 100, pp. 66-74, Feb. 2019.

[7] M. Noohi, A. Mirvakili, and S. A. Sadrossadat, "Modeling and implementation of nonlinear boost converter using local feedback deep recurrent neural network for voltage balancing in energy harvesting applications," in *Int. J. Circuit Theory and Appl.*, vol. 49, no. 12, pp. 4231-4247, Sep. 2021.

[8] Z. Naghibi, S. A. Sadrossadat, and S. Safari, "Adjoint recurrent neural network technique for nonlinear electronic component modeling," *Int. J. Circuit Theory and Appl.*, vol. 50, no. 4, pp. 1119-1129, Apr. 2022.

[9] M. Moradi, S. A. Sadrossadat, and V. Derhami, "Long Short-Term Memory Neural Networks for Modeling Nonlinear Electronic Components," *IEEE Trans. Compon., Packag. Manuf. Tech.*, vol. 11, no. 5, pp. 840-847, May. 2021.

[10] A. M. Schäfer and H. G. Zimmermann, "Recurrent neural networks are universal approximators," *Int. Conf. Artificial Neural Networks*, Springer, Sep. 2006, pp. 632-640.

[11] Y. Fang, M. C. Yagoub, F. Wang, and Q.-J. Zhang, "A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks," *IEEE Trans. Microw. Theory Tech.*, vol. 48, no. 12, pp. 2335-2344, Dec. 2000.

[12] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," *Adv. Neural Inf. Process. syst. (NIPS)*, Jan. 2013, pp. 190-198.

[13] R. Soltani and H. Jiang, "Higher order recurrent neural networks," Apr. 2016, *arXiv:1605.00064*. [Online]. Available: https://arxiv.org/abs/1605.00064

[14] H. Yu, H. Chalamalasetty, and M. Swaminathan, "Behavioral modeling of steady-state oscillators with buffers using neural networks," *IEEE 27th Conf. on Electr. Perform. Electron. Packag. and Syst. (EPEPS)*, San Jose, CA, USA, Nov. 2018, pp. 307-309.

[15] B. Li, P. Franzen, Y. Choi, and C. Cheng, "Receiver behavior modeling based on system identification," *27th IEEE Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, San Jose, CA, USA, Oct. 2018, pp. 299-301.

[16] S. Fu, Y. Tseng, and K. Chiang, "Study on Data Effect of Using RNN Model to Predict Reliability Life of Wafer Level Packaging," *15th Int. Microsyst., Packag., Assembly and Circuits Tech. Conf. (IMPACT*), Taipei, Taiwan, Nov. 2020, pp. 200-203.

[17] T. Nguyen, T. Lu, J. Sun, Q. Le, K. We, and J. Schut-Aine, "Transient simulation for high-speed channels with recurrent neural network," *IEEE 27th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, San Jose, CA, USA, Nov. 2018, pp. 303-305.

[18] C. Zhang, S. Yan, Q.-J. Zhang, and J.-G. Ma, "Behavioral modeling of power amplifier with long term memory effects using recurrent neural networks," in *IEEE Int. Wireless Symp. (IWS)*, Beijing, China, Apr. 2013, pp. 1-4.

[19] Z. Naghibi, S. A. Sadrossadat, and S. Safari, "Dynamic behavioral modeling of nonlinear circuits using a novel recurrent neural network technique," *Int. J. Circuit Theory Appl.*, vol. 47, no. 7, pp. 1071-1085, Jul. 2019.

[20] J. Xu, M. C. Yagoub, R. Ding, and Q.-J. Zhang, "Neural-based dynamic modeling of nonlinear microwave circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 50, no. 12, pp. 2769-2780, Dec. 2002.

[21] N. Mirzaie and R. Rohrer, "A Macromodeling Approach for Analog Behavior of Digital Integrated Circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits and Syst.*, vol. 39, no. 12, pp. 5025-5031, Dec. 2020.

[22] K. Batselier, Z. Chen, H. Liu, and N. Wong, "A tensor-based volterra series black-box nonlinear system identification and simulation framework," *IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Austin, TX, USA, Jan. 2016, pp. 1-7.

[23] D. De Jonghe and G. Gielen, "Efficient analytical macromodeling of large analog circuits by transfer function trajectories," *IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, Nov. 2011, pp. 91-94.

[24] Y. Zhang, N. Fong, and N. Wong, "Piecewise-polynomial associated transform macromodeling algorithm for fast nonlinear circuit simulation," *18th Asia and South Pac. Des. Autom. Conf. (ASP-DAC)*, Yokohama, Japan, Jan. 2013, pp. 515-520.

[25] T. McConaghy and G. G. Gielen, "Template-free symbolic performance modeling of analog circuits via canonical-form functions and genetic programming," *IEEE Trans. Comput.-Aided Design Integr. Circuits and Syst.*, vol. 28, no. 8, pp. 1162-1175, Aug. 2009.

[26] D. De Jonghe, D. Deschrijver, T. Dhaene, and G. Gielen, "Extracting analytical nonlinear models from analog circuits by recursive vector fitting of transfer function trajectories," *Design, Autom. & Test in Europe Conf. & Exhibition (DATE)*, Grenoble, France, Mar. 2013, pp. 1448-1453.

[27] Y. Song, M. P. Sai, and H. Yu, "Zonotope-based nonlinear model order reduction for fast performance bound analysis of analog circuits with multiple-interval-valued parameter variations," *Design, Autom. & Test in Europe Conf. & Exhibition (DATE)*, Dresden, Germany, Mar. 2014, pp. 1-6.

[28] A. Tjandra, S. Sakti, and S. Nakamura, "Tensor decomposition for compressing recurrent neural network," *Int. Joint Conf. on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, Oct. 2018, pp. 1-8.

[29] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," *IEEE Int. Conf. Acoust., Speech Signal process.*, Vancouver, BC, Canada, Oct. 2013, pp. 8624-8628.

[30] Y. Bengio, P. Frasconi, and P. Simard, "The problem of learning long-term dependencies in recurrent networks," in *IEEE Int. Conf. Neural Networks*, San Francisco, CA, USA, Aug. 1993, pp. 1183-1188.

[31] T. Linzen, E. Dupoux, and Y. Goldberg, "Assessing the ability of LSTMs to learn syntax-sensitive dependencies," *Trans. of the Assoc. for Comput. Ling.*, Dec. 2016, pp. 521-535.

[32] S. Gu, S. Levine, I. Sutskever, and A. Mnih, "Muprop: Unbiased backpropagation for stochastic neural networks," Nov. 2015, *arXiv:1511.05176*. [Online]. Available: https://arxiv.org/abs/1511.05176

[33] A. X. M. Chang, B. Martini, and E. Culurciello, "Recurrent neural networks hardware implementation on FPGA," Nov. 2015, *arXiv:1511.05552*. [Online]. Available: https://arxiv.org/abs/1511.05552

[34] A. Graves, "Generating sequences with recurrent neural networks," Aug. 2013, *arXiv:1308.0850*. [Online]. Available: https://arxiv.org/abs/1308.0850

[35] C. Zhang and P. C. Woodland, "High order recurrent neural networks for acoustic modelling," in *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, Calgary, AB, Canada, Sep. 2018, pp. 5849-5853.

[36] H. Huang and B. Mak, "To Improve the Robustness of LSTM-RNN Acoustic Models Using Higher-Order Feedback from Multiple Histories," *Int. Speech Commun. Assoc. (INTERSPEECH)*, Stockholm, Sweden, Aug. 2017, pp. 3862-3866.

[37] R. Yu, S. Zheng, A. Anandkumar, and Y. Yue, "Long-term forecasting using higher order tensor RNNs," 2017, arXiv:1711.00073. [Online]. Available: http://arxiv.org/abs/1711.00073

[38] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157-166, Mar. 1994.

[39] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *Int. Conf. Mach. Learn.*, Aug. 2013, pp. 1310-1318.

[40] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," *29th Int. Conf. Mach. Learn. (ICML)*, Madison, WI, USA, Jun. 2012, pp. 1881-1888.

[41] Y. Bengio and P. Frasconi, "Credit assignment through time: Alternatives to backpropagation," *Adv. in Neural inf. Process. Syst (NIPS)*, Nov. 1993, pp. 75-82.

[42] R. Jacob (Jake) Baker, CMOS Circuit Design, Layout, and Simulation, Fourth ed., Piscataway, NJ, USA: John Wiley & Sons, 2019.

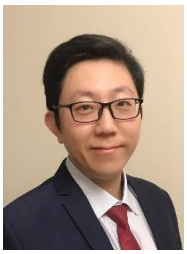
[43] B. Mutnury, M. Swaminathan, and J. P. Libous, "Macromodeling of nonlinear digital I/O drivers," *IEEE Trans. Adv. Packag.*, vol. 29, no. 1, pp. 102–113, Feb. 2006.

Fatemeh Charoosaei received the B.Eng. degree in Information Technology Engineering from Islamic Azad University of Najafabad, Isfahan, Iran, in 2017. She is currently pursuing master's degree in artificial intelligence at Yazd University, Computer Engineering Department, Yazd, Iran. Her research interests include computer-aided design, deep learning, and neural networks and their applications in modeling and simulation of circuits and components.

Mostafa Noohi received the B.Eng. degree in Electrical Engineering from Shahid Bahonar College, Shiraz, Iran, in 2018 and master's degree in Electrical Engineering from Yazd University, Yazd, Iran, in 2021. He is currently pursuing Ph.D. degree in Electrical Engineering at Sahand University of Technology, Tabriz, Iran. His research interests include computer-aided design, artificial intelligence, and hybrid energy harvesting.

Sayed Alireza Sadrossadat (S'12 - M'14) received B.Sc. degree from University of Tehran, Tehran, Iran, in 2007, master's degree from University of Waterloo, Waterloo, ON, Canada, in 2010, and Ph.D. degree from Carleton University, Ottawa, ON, Canada, in 2015. He is currently the chair of Artificial Intelligence Group, Computer Engineering Department, Yazd University, Yazd, Iran. His current research interests include optimization and neural network-based modeling of linear/nonlinear components and circuits, computer-aided design, deep learning, very large-scale integration design, probabilistic design, and yield maximization. Mr. Sadrossadat has been a Technical Reviewer for several IEEE/IET journals, such as IEEE Transactions on Neural Network and Learning Systems, IEEE Transactions on Microwave Theory and Technique, IEEE Transactions on Vary Large-Scale Integration Systems, IET Microwave and Propagation, IET Electronic Letters, IEEE Canadian Journal of Electrical and Computer Engineering, and Scientific Reports-Nature.

Ali Mirvakili received his B.Sc., master's, and Ph.D. degree in Electronics Engineering from Yazd University, K. N. Toosi University of Technology, and Tufts University in 2005, 2008, and 2014, respectively. He was at Advanced Integrated Circuits and Systems Lab., Tufts University, and student member of Smart Lighting Engineering Reseach Center, from 2010 to 2014, where he worked on LED Driver Architecture Design and Tesbed Integration for Visible Light Communication (VLC). He was Electronics and Programming Intern at Osram Sylvania, USA, from 2014 to 2015, and Electrical Design Engineer at Sensata Technologies, USA, from 2015 to 2016. He joined with the department of Electrical Engineering at Yazd University in 2016 as Assistant Professor. His research interests are power management for energy harvesting, VLSI, RF, and mixed-signal circuit design for biomedical applications, and DC-DC converter control.



Feng Feng (Member, IEEE) received the B.Eng. degree in Tianjin University, Tianjin, China, in 2012, and the Ph.D. degree in the School of Microelectronics at Tianjin University, Tianjin, China, and the Department of Electronics at Carleton University, Ottawa, ON, Canada, in 2017. From 2017 to 2020, he was a Postdoctoral Fellow in the Department of Electronics at Carleton University, Ottawa, ON, Canada. He is currently an Associate Professor with the School of Microelectronics at Tianjin University, Tianjin, China. His research interests include electromagnetic parametric modeling and design optimization algorithms, deep neural network modeling method, space mapping algorithm and surrogate model optimization, finite-element method in electromagnetic simulation and optimization, and multiphysics modeling and optimization.

WEICONG NA received the B.Eng. degree from Tianjin University, Tianjin, China, in 2012, and the Ph.D. degrees from the School of Microelectronics, Tianjin University, and the Department of Electronics, Carleton University, Ottawa, ON, Canada, in 2018. She is currently with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. Her research interests include microwave circuit modeling and design, automated neural network model generation algorithm, EM field knowledge-based modeling and optimization, and deep neural network modeling for microwave applications.